

AWS Cloud Institute: Developer Fundamentals 3 Syllabus

Course Overview

In this course, you will continue to build your Python knowledge through web programming, starting with the basics of HTML, CSS, Javascript, and the Django framework for server-side development. You will learn how to deploy microservices for event-driven applications in a serverless fashion. In Cloud Capstone 1, you will design and develop a working application to demonstrate your ability to use AWS services and Python code to create a cloud-based solution to an employer-informed problem scenario.

Course Structure

Developer Fundamentals 3 is a combination of digital e-learning content (“modules”) that you can read, watch, and engage with on your own time (asynchronous content) and instructor-led training sessions (ILTs) that focus on the topics presented in the asynchronous content.

Developer Fundamentals 3 modules are covered over the course of 11 weeks and include approximately 79 hours of text, video, audio, hands-on learning activities, and knowledge checks (see Table 1). There are over a dozen ILT sessions offered each week that can be watched live or on-demand. Although ILT attendance is not required for course completion, each session is an opportunity to connect with and learn from your AWS instructor and other Developer Fundamentals 3 students.

As with all AWS Cloud Institute courses, the expectation is that you will complete your modules and associated assessments on a weekly basis. You also have the option to attend the ILTs live or watch the recorded version that will be posted online after each session. Taking advantage of this “flipped classroom” method is the best way to ensure that you develop the professional and technical skills required to be successful as a cloud application developer.

Course Sessions

Table 1. Module Overview by Content Type

	Content Type	Percentage of total course time (by hours)
1	e-learning (text, video, audio)	44% (43 hours)
2	Ungraded (formative) knowledge checks	9% (8.4 hours)
3	Hands-on learning activities (labs, activities and/or challenges)	15% (14.4 hours)
4	Graded (summative) assessments	13% (13 hours)
	Capstone Project	19% (19.2 hours)
5	Total Time	100% (98 hours)

Instructor-Led Training (ILT) Sessions

Mondays: Weekly Stand-Up, 12:00PM, 2:00PM & 4:00PM EST
Tuesdays: Weekly Spotlight, 12:00PM, 2:00PM & 4:00PM EST
Wednesdays: Weekly Focus Session, 12:00PM, 2:00PM & 4:00PM EST
Thursdays: Weekly Labology Session, 12:00PM, 2:00PM & 4:00PM EST

Fridays: Weekly Fun Friday Event, 4:00PM EST
M, T, W, Th: Office Hours, 9:00-11:30 AM, 05:30-8:00PM EST

All ILT sessions, with the exception of office hours, are recorded and will be available on-demand for you to watch when you have time. Multiple live ILT sessions are offered each day to accommodate a range of schedules. All ILT sessions, with the exception of Fun Friday events, are also offered on a PST schedule (i.e., Weekly Stand-Up 9:00AM, 11:00AM & 1:00PM PST).

Prerequisites

Developer Fundamentals 1 and Developer Fundamentals 2

Assessment and Grading Policy

Developer Fundamentals 3 is offered as a “pass/fail” course (no letter grades). You will be required to achieve a passing score of 85% or higher on all weekly summative assessments to successfully complete Developer Fundamentals 3. There are 11 summative assessments in this course. Each summative assessment will take the form of a 20-question multiple choice quiz presented at the end of your assigned modules each week. You will have unlimited opportunities to achieve a passing score on each summative assessment.

Attendance at ILT sessions is strongly encouraged, but will not count towards course completion. You should use the ILT sessions to explore concepts that were presented in your e-learning in greater depth, connect with your AWS instructors and fellow ACI learners, and prepare for the weekly summative assessment.

Course Completion Requirements

All learners must complete 11 weekly summative assessments with a score of 85% or higher to receive credit for Developer Fundamentals 3. ILT attendance or watching ILT recordings is strongly encouraged but is not required for course completion.

Python 3

Module Overview

This module is part of the Developer Fundamentals 3 course and covers how to develop a complete web application using the Django framework for Python. Users will build a web application and deploy for high availability using AWS Elastic Beanstalk.

Week 1: Web Technologies and HTML Overview

Goal:

Introduce the fundamental technologies used in a web application and learn basic HTML syntax.

Learning Objectives:

- Describe the request-response flow of a web application.
- Explain the role of HTML, Cascading Style Sheets (CSS), and JavaScript in a web application.
- Identify basic HTML elements.
- Use basic HTML elements to build a simple webpage.
- Describe HTML development best practices.

Module Outline:

1. Web Technologies

- a. Web Development Practice Environment
- b. Introduction: Web Technologies
- c. Technologies Used in Web Applications
- d. Knowledge Check

2. HTML Overview

- a. Introduction: HTML Overview
- b. Getting Started with HTML
- c. HTML Basic Structure and Syntax
- d. Text Elements and Comments
- e. Images
- f. Links
- g. HTML Development Best Practices
- h. Lab: Creating an HTML Webpage
- i. Knowledge Check

Activities:

- **Lab: Creating an HTML Webpage (40 mins)**
 - This lab challenges you to create a basic HTML webpage that includes a variety of HTML elements. You will also learn how to host the page on a web server so that it is accessible from the internet.

Ungraded Assessments:

Ungraded assessments establish the bar of each learner's pre-existing knowledge of the content. The questions are designed to review basic information about technologies used in a web application, HTML features and syntax, and HTML best practices

Graded Assessment:

The graded assessment covers technologies used in web applications, HTML basic structure and syntax, HTML text elements, comments, images and links, and HTML development best practices.

Week 2: Organizing and Styling HTML Pages

Goal:

Learners observe how the structuring and layout of HTML tags impacts the readability and ease of consumption of web pages, identifying the use and purpose of Cascading Style Sheets (CSS), and differentiating between when to use HTML formatting tags and when to use CSS.

Learning Objectives:

- Identify HTML elements used for web content organization.
- Apply best practice for ordering and writing HTML elements.
- Build table and list elements for text placement.
- Organize and group text elements using HTML.

Module Outline:

1. Organizing HTML Elements

- a. Web Development Practice Environment
- b. Introduction: Organizing and Styling HTML Pages
- c. Benefits of Organizing HTML Elements
- d. Using Line Breaks and Horizontal Rule
- e. Creating Lists
- f. Spacing and Formatting Text
- g. Building with Tables
- h. Organizing with Divisions
- i. Lab: Organizing the Elements in a Webpage
- j. Knowledge Check

2. Cascading Style Sheets

- a. Introduction: Cascading Style Sheets
- b. Using Cascading Style Sheets
- c. Writing CSS Syntax
- d. Formatting Attributes
- e. Applying Spans
- f. Lab: Using CSS to Add Style to a Webpage
- g. Knowledge Check

Activities:

- **Lab: Organizing the Elements in a Webpage (60 min)**
 - This lab challenges you to continue from where you left off in the previous lab, **Python 3 – Lab 1: Creating an HTML Webpage**. In that lab, you created a basic HTML page for the AnyCompany bicycle parts business. In this lab, you make use of additional HTML elements, such as lists and tables, to organize the content on your webpage.
- **Lab: Using CSS to Add Style to a Webpage (60 min)**
 - This lab challenges you to continue where you left off in the previous lab. In the previous lab, you added tables, an unordered list, and other non-CSS stylings to an HTML page for the AnyCompany bicycle parts business. In this lab, you style the same webpage using CSS and HTML div elements.

Ungraded Assessments:

These ungraded assessments establish the bar of each learner's pre-existing knowledge on the content. These questions are designed to review basic information surrounding HTML and CSS, their use, application, and best practices for building a web page.

Graded Assessment:

The graded assessment covers the different types of HTML tags, the appropriate use case, and formatting. The assessment also tests the learner's knowledge of when to use CSS, how to use them appropriately, and the difference between HTML formatting and CSS use cases.

Week 3: Building an Interactive HTML Page

Goal:

Learn how to create a dynamic HTML page using an HTML form and JavaScript.

Learning Objectives:

- Describe the purpose of HTML form elements.
- Identify basic form elements and their main attributes.
- Use form elements to create an interactive HTML page.
- Describe the benefits of using client-side JavaScript.
- Explain basic JavaScript syntax.
- Use JavaScript to customize the behavior of a webpage.

Module Outline:

1. HTML Forms

- a. Web Development Practice Environment
- b. Introduction: HTML Forms
- c. HTML Form Elements
- d. Adding a Form to an HTML Document
- e. Input Form Element
- f. Select Form Element
- g. Button Element
- h. Lab: Adding a Form and Buttons to a Webpage
- i. Knowledge Check

2. Client-Side Scripting

- a. Introduction: Client-Side Scripting
- b. JavaScript in Frontend Web Development
- c. JavaScript Basic Syntax
- d. Lab: Adding JavaScript to a Webpage
- e. Knowledge Check

Activities:

- **Lab: Adding a Form and Buttons to a Webpage (45 min)**
 - This lab challenges you to continue from where you left off in the previous lab. In that lab, you added Cascading Style Sheets (CSS) to an HTML page for the AnyCompany bicycle parts business. In this lab, you add a form to the page to collect product order details and submit them. You also add buttons that can be used by the website user to change the look of the webpage.
- **Lab: Using JavaScript in a Webpage (45 min)**
 - In this lab, you add JavaScript logic to the page for event handling purposes. The JavaScript logic makes it possible for the end user to use the buttons to change the look of the page. You also add form validation to the product order form that you previously created. The JavaScript ensures that the user has selected at least one product before they can submit the form.

Ungraded Assessments:

Ungraded assessments establish the bar of each learner's pre-existing knowledge of the content. The questions are designed to review the HTML elements used to create a form and the use of JavaScript for client-side scripting.

Graded Assessment:

The graded assessment covers the topics of HTML forms, HTML form elements, client-side scripting, and JavaScript basic syntax.

Week 4: Django Overview

Goal:

Introduce the Django framework, its benefits, and how to use it in the Cloud9 IDE.

Learning Objectives:

- Describe the Django web framework and its benefits.
- Explain the Model View Template (MVT) design pattern.
- Set up a Django environment in AWS Cloud9.

Module Outline:

1. Django Overview

- a. Web Development Practice Environment
- b. Introduction: Django Overview
- c. What is Django?
- d. Introduction to the Model-View-Template (MVT) Design Pattern
- e. How Django Works
- f. Developing a Django Web Application in Cloud9
- g. Lab: Installing and Configuring Django
- h. Knowledge Check

Activities:

- **Lab: Create a Django Development Environment in AWS Cloud9 (45 min)**
 - In this first Django lab, you will install the Django web framework in a Python virtual environment and then create a basic Django project with a Hello World web app running in it.

Ungraded Assessments:

Ungraded assessments establish the bar of each learner's pre-existing knowledge of the content. The questions are designed to review the design pattern used by Django, the tools and features it provides, and how to use Django in the Cloud9 IDE.

Graded Assessment:

The graded assessment for this module is included in the graded assessment of the last module of week 4, Django Models. It covers the topics of the benefits of the Django framework, how it works, and how to set it up in Cloud9.

Week 4: Django Models

Goal:

Learn how to develop a model in a Django web application.

Learning Objectives:

- Describe the Model class.
- Discuss how to create, update, and migrate a model.
- Explain how to use a model to access database data.
- Create a model for a Django web application.

Module Outline:

1. Django Models

- a. Web Development Practice Environment
- b. Introduction: Django Models
- c. Model Definition
- d. Creating and Migrating a Model
- e. Data Operations through a Model
- f. Lab: Creating Models in Django
- g. Knowledge Check

Activities:

- **Lab: Creating Models in Django (45 min)**
 - In this lab, you create models in Django to support storing product information and order information in the database used by Django.

Ungraded Assessments:

Ungraded assessments establish the bar of each learner's pre-existing knowledge of the content. The questions are designed to review the role of a model in a Django web application and how to create, persist, and use a model.

Graded Assessment:

The graded assessment covers the topics of how a model works, and how to develop and use a model in a Django web application.

Week 5: Django Views

Goal:

Learn how to develop a view in a Django web application.

Learning Objectives:

- Describe how a Django view works.
- Explain how to use a view to process a web request.
- Create a view for a Django web application.

Module Outline:

1. Django Views

- a. Web Development Practice Environment
- b. Introduction: Django Views
- c. View Definition
- d. URL Mapping

- e. Processing a Request and Producing a Response
- f. Lab: Creating Views in Django
- g. Knowledge Check

Activities:

- **Lab: Creating Views in Django (90 min)**
 - In this lab, you create views for the bicycle supplies Django application to make it possible for users to place an order and view an order confirmation. The user will also be able to view order history and the line item details for any previous order.

Ungraded Assessments:

Ungraded assessments establish the bar of each learner's pre-existing knowledge of the content. The questions are designed to review the role of a view in a Django application, how a view works, and how to develop a view.

Graded Assessment:

The graded assessment for this module is included in the graded assessment of the last module of week 5, Django Templates. It covers the topics of configuring URL mappings for a view, creating a view function, view request processing, and view best practices.

Week 5: Django Templates

Goal:

Learn how to develop a template in a Django web application.

Learning Objectives:

- Describe the main elements of the Django template language (DTL).
- Explain how to use a template to display data from a view.
- Use the Model-View-Template (MVT) pattern in a webpage.
- Create and launch a complete Django web application in AWS Cloud9.

Module Outline:

1. **Django Templates**
 - a. Web Development Practice Environment
 - b. Introduction: Django Templates
 - c. The Django Template System
 - d. Organizing and Referencing Templates
 - e. Displaying Data from a View
 - f. Template Language Tags and Filters
 - g. MVT Example
 - h. Lab: Creating Templates in Django
 - i. Knowledge Check

Activities:

- **Lab: Creating Templates in Django (60 min)**
 - In this lab, you create Django templates to provide the end user interface for the AnyCompany bicycle parts web application. You also add order-handling features to the application.

Ungraded Assessments:

Ungraded assessments establish the bar of each learner's pre-existing knowledge of the content. The questions are designed to review the role of a template in a Django application, how a template works, and how to develop a template.

Graded Assessment:

The graded assessment covers the topics of creating a template, using the DTL, integrating a template with a view, and developing a complete MVT application.

Week 6: Django Administration

Goal:

Learn how to use the Django Administration website.

Learning Objectives:

- Describe how to activate and access the Django admin site.
- Create an admin user and navigate the Django admin site.
- Discuss how to manage models in the Django admin site.
- Explain how to maintain model data in the Django admin site.

Module Outline:

1. Django Administration

- a. Introduction: Django Administration
- b. Accessing the Django Admin Site
- c. Registering Models
- d. Managing Admin Users
- e. Maintaining Model Data
- f. Lab: Administering Django
- g. Knowledge Check

Activities:

- **Lab: Administering Django (40 min)**
 - In this lab, you activate the Django administrator UI and discover the benefits that it provides.

Ungraded Assessments:

Ungraded assessments establish the bar of each learner's pre-existing knowledge of the content. The questions are designed to review how to activate and access the Django Admin website, and how to use its features.

Graded Assessment:

The graded assessment for this module is included in the graded assessment of the last module of week 6, AWS Elastic Beanstalk Python Support. It covers the topics of enabling the Django Admin site, creating an administrative user, and managing models and model data.

Week 6: AWS Elastic Beanstalk Python Support

Goal:

Review the benefits of AWS Elastic Beanstalk and learn the tools it provides to support Python applications.

Learning Objectives:

- Describe the benefits and features of AWS Elastic Beanstalk.
- Describe the AWS Elastic Beanstalk Python platform and available tools.

Module Outline:

1. Review of Elastic Beanstalk

- a. Introduction: Elastic Beanstalk
- b. Benefits and Concepts of Elastic Beanstalk
- c. Elastic Beanstalk Web Server Environment Architecture
- d. Using Elastic Beanstalk with the Command Line Interface (CLI)
- e. Knowledge Check

2. Elastic Beanstalk Python Platform and Tools

- a. Introduction: Elastic Beanstalk Python Platform and Tools
- b. Using Elastic Beanstalk with the Python SDK (Boto)
- c. Using web application frameworks for Elastic Beanstalk
- d. Using Web Server Gateway Interface servers for Elastic Beanstalk
- e. Using proxy servers with Elastic Beanstalk
- f. Using the pipenv packaging tool for Elastic Beanstalk
- g. Knowledge Check

Ungraded Assessments:

Ungraded assessments establish the bar of each learner's pre-existing knowledge of the content. The questions are designed to review basic information about technologies used in deploying applications using AWS Elastic Beanstalk and the Python toolset available to be used with Beanstalk.

Graded Assessment:

The graded assessment for this module includes topics of enabling the Django Admin site, creating an administrative user, and managing models and model data. How to deploy applications using AWS Elastic Beanstalk and the available Python tools are also covered in this assessment.

Week 7: Deploying a Django Web Application to AWS Elastic Beanstalk

Goal:

Learn how to prepare a Django web application for production and deploy it to AWS Elastic Beanstalk.

Learning Objectives:

- Identify common AWS services used with a Django web application.
- Describe how to configure a Django web application with an Amazon Simple Storage Service (Amazon S3) bucket, Amazon ElastiCache cache, and an Amazon Relational Database Service (Amazon RDS) database.
- Outline steps for deploying a Django web application to AWS Elastic Beanstalk.

Module Outline:

1. Deploying a Django Web Application

- a. Introduction: Deploying a Django Web Application
- b. Moving from Development to Production
- c. Using Amazon S3 for Media Files
- d. Using Amazon ElastiCache for Session Storage
- e. Switching to an Amazon RDS Database
- f. Deployment to AWS Elastic Beanstalk
- g. Lab: Deploy a Django Web Application on AWS Elastic Beanstalk
- h. Knowledge Check

Activities:

- **Lab: Deploy a Django Web Application on AWS Elastic Beanstalk (90 min)**
 - In this lab, you update the web application so that the architecture is more scalable and resilient.

Ungraded Assessments:

Ungraded assessments establish the bar of each learner's pre-existing knowledge of the content. The questions are designed to review information surrounding deploying a Django web application to AWS Elastic Beanstalk, changing the backend database, and best practices surrounding planning and deployment.

Graded Assessment:

The graded assessment covers topics discussed in the course, such as preparing an application for deployment to production, using AWS managed services for scalability, security and performance, and deploying to AWS Elastic Beanstalk.

Modern Application Architecture and Developing Serverless Applications

Module Overview

This module is part of the Developer Fundamentals 3 course and teaches you how to use microservices in a modern application architecture. You will learn how to design applications using microservices and deploy them in a serverless fashion. You will also learn how to use frameworks to facilitate the deployment of serverless applications.

Week 8: Modern Application Architecture Overview

Goal:

Describe the three characteristics of modern application architectures and introduce the microservice architecture.

Learning Objectives:

- Describe the characteristics of a modern application.
- Identify the benefits of building modern applications on AWS.
- Describe microservice design patterns.
- Describe how to decompose an application into microservices.

Module Outline:

1. Introduction to Modern Application Architecture

- a. Introduction: Modern Application Architecture
- b. Modern Application Architecture
- c. Overview of AWS Services and Tools for Building Modern Applications
- d. Knowledge Check

2. Introduction to Microservices

- a. Introduction: Microservices
- b. Monolithic Versus Microservice Architecture
- c. The Twelve-Factor App Methodology
- d. Microservices Communication Protocols
- e. How to Design an Application Using Microservices
- f. Knowledge Check

Videos (Total Time 10:04):

- Coming Up (1:23)
- How to Build This with Jillian Forde (8:41)

Ungraded Assessments:

Ungraded assessments establish the bar of each learner's pre-existing knowledge of the content. The questions are designed to review basic information about technologies used in developing microservices in a modern application architecture and how to design applications using microservices and deploy them in a serverless fashion.

Graded Assessment:

The graded assessment for this module is included in the graded assessment of the last module of week 1, Microservices Deployment Patterns. It covers the topics of the benefits of the how to design application using microservices and the modern application architecture.

Week 8: Microservice Data Storage Patterns

Goal:

Identify the need for data persistence within serverless applications, and the AWS services that are commonly used for storing data in serverless application architectures.

Learning Objectives:

- Describe the data storage characteristics of a microservice.
- Identify microservice data persistence patterns.
- Choose the right Amazon Web Services data storage service.

Module Outline:

1. Microservice Data Storage

- a. Introduction: Microservice Data Storage
- b. Understanding Microservice Data Storage
- c. Managing Microservice Data Persistence
- d. Choosing the Right Microservice Storage Service
- e. Using DynamoDB for Microservice Storage

f. Knowledge Check

Ungraded Assessments:

These ungraded assessments establish the bar of each learner's pre-existing knowledge on the content. These questions are designed to review information surrounding the need for persisting data in serverless application architectures, and the AWS storage services commonly used for this purpose, with a focus on the role of Amazon DynamoDB.

Graded Assessment:

The graded assessment covers topics discussed in the course, including why persistent storage is needed in serverless applications, how to select the appropriate storage service based on performance, manageability, reliability, and security requirements, and Amazon DynamoDB's specific suitability for persisting data in serverless applications.

Week 8: Microservice Deployment Patterns

Goal:

Describe the serverless and container-based deployment patterns for a microservice and identify the AWS services that support them.

Learning Objectives:

- Identify the types of patterns used to deploy microservices, serverless or container based.
- Describe the AWS services used to deploy microservices in a serverless fashion.
- Describe the AWS services used to deploy microservices using container technology.
- Demonstrate how to write and troubleshoot Lambda functions, including Python functions.

Module Outline:

- 1. Types of Microservice Deployment Patterns**
 - a. Introduction: Microservice Deployment Patterns
 - b. Serverless and Container-Based Deployments
 - c. Serverless Microservices
 - d. Knowledge Check
- 2. Building Serverless Microservices with AWS Lambda**
 - a. Introduction: Serverless Microservices with AWS Lambda
 - b. Developing a Python Lambda Function
 - c. Writing Good Lambda Functions
 - d. Knowledge Check

Activities:

- **Lab: Working with DynamoDB (60 min)**
 - In this lab, you start building the solution. Specifically, you gain hands-on experience working with DynamoDB. You start by establishing the hosting infrastructure for the web frontend of your web application on Amazon S3. You then work to create the DynamoDB table that stores application data. You use various AWS Command Line Interface (AWS CLI) commands to

retrieve resource information and store the information as a variable for use later using variables and create resources.

- **Lab: Working with Lambda (45 min)**
 - In this lab, you start with what you achieved in the last lab. There is an Amazon DynamoDB table named LabCustomers already in place and it contains a few items of sample data. There is also an Amazon Simple Storage Service (Amazon S3) bucket configured to host a website, and it contains the web frontend for your application in the form of a single index.html page.

Ungraded Assessments:

These ungraded assessments establish the bar of each learner's pre-existing knowledge on the content. These questions are designed to review information surrounding AWS serverless services such as AWS Lambda, Amazon ECS, Amazon EKS, and AWS Fargate. Deployment patterns for microservices will also be assessed, including what AWS services can be used to deploy microservices in a serverless and container fashion.

Graded Assessment:

The graded assessment covers topics discussed in the course, including deployment patterns for deploying microservices in a serverless fashion to AWS services such as AWS Lambda, Amazon ECS, Amazon EKS, and AWS Fargate.

Week 9: Microservice Integration Patterns – Part 1

Goal:

Gain a deeper knowledge of microservice integration patterns through key concepts such as synchronous and asynchronous communication patterns, integrating Amazon API Gateway to benefit process flow, and the introduction of AWS Step Functions for effective process flow.

Learning Objectives:

- Describe microservice integration pattern types
- Identify Amazon Web Services (AWS) services used for synchronous and asynchronous integration
- Use API Gateway for microservice integration

Module Outline:

- 1. Microservice Integration Patterns**
 - a. Identifying Communication Patterns
 - b. API Gateway Integration Pattern
 - c. Decoupled Messaging Integration
 - d. Publish-Subscribe Integration
 - e. Event-Based Integration
 - f. Knowledge Check
- 2. Orchestrating Microservices with AWS Step Functions**
 - a. Orchestrating Microservices with AWS Step Functions
 - b. Working With Step Functions
 - c. Identifying Step Function use Cases
 - d. API Operations
 - e. Knowledge Check

Activities:

- **Lab: Working with API Gateway (60 min)**
 - In this lab, you use API Gateway to define a REST API endpoint that can invoke the Lambda functions. You then update the webpage to invoke the endpoint.

Ungraded Assessments:

These ungraded assessments establish the bar of each learner's pre-existing knowledge on the content. These questions are designed to review information surrounding deploying integration and communication patterns, best practices for integrating microservices into your architecture and the use of AWS Step Functions for process integration.

Graded Assessment:

The graded assessment covers topics discussed in the course, such as the types of communication patterns and use cases for each one, the AWS services used to integrate microservices synchronously and asynchronously, the benefits and use cases for using Amazon API Gateway for microservices, and AWS Step Function best practices.

Week 10: Microservice Integration Patterns: AWS Step Functions Lab – Part 2

Goal: In this lab, you learn how AWS Step Functions can be used to coordinate the invocation of Lambda functions to create a workflow. In this specific implementation, the Step Functions state machine that you create coordinate the actions necessary to run a Trivia game.

Learning Objectives:

- Define what a step function is within AWS.
- Identify the purpose of step functions.
- Build AWS Step Functions to assist with integrating microservices.

Module Outline:

1. **Lab Orchestrating Microservices with AWS Step Functions**
 - a. Welcome to the Microservice Integration Patterns Lab
 - b. Lab: Using AWS Step Functions with AWS Lambda
 - c. Lab Complete!

Activities

- **Lab: Working with Step Functions (60 min)**
 - This lab provides additional hands-on learning opportunities to work with AWS Step Functions. This lab follows the Week 2 course titled, Microservice Integration Patterns and reinforces the topics included in the section titled Orchestrating Microservices with AWS Step Functions.

Week 10: Serverless Application Deployment Frameworks – Part 1

Goal:

Discover how AWS CloudFormation can be used in conjunction with the AWS Cloud Development Kit (AWS CDK) and the AWS Serverless Application Model (AWS SAM) to deploy serverless applications in an efficient, reliable, and secure manner.

Learning Objectives:

- Identify AWS deployment frameworks for serverless applications.
- List best practices for serverless deployments.

Module Outline:

1. **Serverless Application Deployment Frameworks**
 - a. Overview of Deployment Frameworks for Serverless Applications
 - b. Knowledge Check
2. **Working with AWS CloudFormation**
 - a. Understanding AWS CloudFormation
 - b. Creating and Managing CloudFormation Templates
 - c. Working with CloudFormation Templates
 - d. Knowledge Check
3. **Working with AWS CDK**
 - a. Understanding AWS Cloud Development Kit
 - b. Creating and Managing Apps using AWS CDK Toolkit
 - c. Working with AWS CDK Apps using AWS CDK
 - d. AWS CDK Best Practices
 - e. Knowledge Check

Ungraded Assessments:

These ungraded assessments establish the bar of each learner's pre-existing knowledge on the content. These questions are designed to review information surrounding AWS CloudFormation and how CloudFormation can be used to deploy infrastructure using the AWS CDK.

Graded Assessment:

The graded assessment covers topics discussed in Step Functions and Serverless Deployment Frameworks course, such the characteristics and benefits of step functions in a serverless architecture, State Machine concepts and terminology, CloudFormation concepts, terminology, repository, and development options, and integrations with the AWS CDK.

Week 11: Serverless Application Deployment Frameworks – Part 2

Goal:

Discover how AWS CloudFormation can be used in conjunction with the AWS Cloud Development Kit (AWS CDK) and the AWS Serverless Application Model (AWS SAM) to deploy serverless applications in an efficient, reliable, and secure manner.

Learning Objectives:

- Describe the features of AWS Serverless Application Model (AWS SAM).
- Build and deploy an application using AWS SAM.
- Create resources in an AWS account using a SAM template.

Module Outline:

1. **Working with AWS Serverless Application Model**
 - a. Introduction: Working with AWS Serverless Application Model
 - b. AWS Serverless Application Model Templates

- c. Using AWS SAM Command Line Interface
- d. Lab: Working with AWS Serverless Application Model (AWS SAM)
- e. AWS SAM Best Practices
- f. Knowledge Check

Activities

- **Lab: Working with AWS Serverless Application Model (AWS SAM) (60 min)**
 - In this lab, you will see how using the AWS SAM service makes it more convenient to build application services like those you created manually in a previous lab. These services included a DynamoDB table, two Lambda functions, and an HTTP API that has both GET and POST methods using API Gateway.

Ungraded Assessments:

These ungraded assessments establish the bar of each learner's pre-existing knowledge on the content. These questions are designed to review information surrounding AWS SAM and how it can be used to build and deploy serverless applications.

Graded Assessment:

The graded assessment covers topics discussed in the course, such as the capabilities, benefits, and components of an AWS SAM development environment, AWS SAM syntax, keywords and formatting, AWS SAM best practices, and how to use the AWS SAM CLI to deploy infrastructure.

Capstone Project: Building a Customer Onboarding App

In this capstone project, you will design and develop a working application to demonstrate your ability to use AWS services and Python code to create a cloud-based solution to an employer-informed problem scenario.