AWS CLOUD INSTITUTE

AWS Cloud Institute: Developer Intermediate 1 Syllabus

Course Overview

In this course, you will learn to deploy microservices for event-driven applications on containers or in a serverless fashion. You will design and develop a working application that demonstrates your skills and your ability to use AWS services and Python code to create cloud-based solutions to employer-informed problem scenarios (Capstone Project 1).

Instructor-Led Training (ILT) Sessions

Attendance at ILT sessions is strongly encouraged, but attendance does not count toward course completion. You can use the ILT sessions to explore concepts presented in the e-learning content in greater depth, connect with AWS instructors and ACI learners, and prepare for the weekly final assessment. 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.

Prerequisites

Developer Fundamentals

Course Completion Requirements

At the end of each week's assigned module, you will complete a multiple-choice final assessment. You have unlimited opportunities to achieve a passing score on each final assessment. You must complete the weekly final assessments with a score of 85% or higher to receive credit for the course. ILT attendance or watching ILT recordings is strongly encouraged, but is not required for course completion.

Capstone Project: Building a Customer Onboarding Application

Goal:

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.

Hands-on Labs:

- **Lab: Building a Customer Onboarding App - Lab 01 (90 mins)**
  - In this lab, you create an Amazon Simple Storage Service (Amazon S3) bucket to store the customers' documents, configure the bucket policy, and create an AWS Identity and Access Management (IAM) role with specific permissions to access the S3 bucket.
- **Lab: Building a Customer Onboarding App - Lab 02 (90 mins)**
  - In this lab, you create an Amazon DynamoDB table to store customer data, create an Amazon Simple Notification Service (Amazon SNS) topic to send application notifications, and add DynamoDB and Amazon SNS permissions to the AWS Lambda function AWS Identity and Access Management (IAM) role.
- **Lab: Building a Customer Onboarding App - Lab 03 (120 mins)**

  - In this lab, you add new permissions to the AWS Lambda function role, get started with the AWS supported integrated development environment (IDE) to develop the application code, create a Lambda function and configure its settings, and configure an Amazon Simple Storage Service (Amazon S3) event notification to invoke the Lambda function.
- **Lab: Building a Customer Onboarding App - Lab 04 (120 mins)**
  - In this lab, you will learn how to deploy resources using AWS Serverless Application Model (AWS SAM) from the AWS SAM command line interface (CLI). These are the same resources you deployed in previous labs using the AWS Management Console. You will also learn how to parse the customer details from a comma-separated values (.csv) file and write it to an Amazon DynamoDB table.
- **Lab: Building a Customer Onboarding App - Lab 05 (120 mins)**
  - In this lab, you will add new permissions to the AWS Lambda function role for Amazon Rekognition. Then, you will update the Lambda function code to perform the CompareFaces action with the Amazon Rekognition service, publish the Amazon Simple Notification Service (Amazon SNS) topic notifications, and update an Amazon DynamoDB table based on the result of comparing the images with Amazon Rekognition.
- **Lab: Building a Customer Onboarding App - Lab 06 (120 mins)**
  - In this lab, you will add new permissions to the Lambda function role for Amazon Textract. Then, you will update the Lambda function code to perform ID document analysis using the Amazon Textract service, publish notifications to an Amazon Simple Notification Service (Amazon SNS) topic, and update an Amazon DynamoDB table based on the result of analysis performed by Amazon Textract.
- **Lab: Building a Customer Onboarding App - Lab 07 (120 mins)**
  - In this lab, you will create a new AWS Lambda function that mimics a third-party validation of the customer's driver's license. You will also create an HTTP API using Amazon API Gateway and integrate it with the Lambda function.
- **Lab: Building a Customer Onboarding App - Lab 08 (120 mins)**
  - In this lab, you will create an Amazon Simple Queue Service (Amazon SQS) queue to submit the license for third-party integration, create the Lambda function to submit the license, and develop the Lambda function code.
- **Lab: Building a Customer Onboarding App - Lab 09 (120 mins)**
  - In this lab, you will prepare to refactor your document Lambda function to perform its tasks in an asynchronous flow. This involves breaking the function's code into smaller functions that can run asynchronously.
- **Lab: Building a Customer Onboarding App - Lab 10 (120 mins)**
  - In this lab, you will build an AWS Step Functions state machine workflow to orchestrate the running of the AWS Lambda functions that you created in the previous lab. You will also enable AWS X-Ray to help you debug your solution.

Week 1: Python 3 – Web Technologies and HTML Overview

Goal:

Understand 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

Hands-on Labs:

- **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.

Week 2: Python 3 – Organizing and Styling HTML Pages

Goal:

Learn how the structuring and layout of HTML tags impacts the readability and ease of consumption of web pages, identify the use and purpose of Cascading Style Sheets (CSS), and differentiate 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

Hands-on Labs:

- **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 use 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.

Week 3: Python 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

Hands-on Labs:

- **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.

Week 4: Python 3 – Django Overview

Goal:

Learn about the Django framework, its benefits, and how to use it in an integrated development environment (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 an IDE.

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
g. Lab: Installing and Configuring Django
h. Knowledge Check

Hands-on Labs:

- **Lab: Create a Django project in an IDE (45 min)**
  - In this lab, you 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.

Week 4: Python 3 – 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

Hands-on Labs:

- **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.

Week 5: Python 3 – 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

Hands-on Labs:

- **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.

Week 5: Python 3 – 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 an IDE.

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

Hands-on Labs:

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

Week 6: Python 3 – 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

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

Week 6: Python 3 – AWS Elastic Beanstalk Python Support

Goal:
Review the benefits of AWS Elastic Beanstalk and learn about 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

Hands-on Labs:

None

Week 7: Python 3 – 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

Hands-on Labs:

- **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.

Week 8: Serverless Applications – 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

Hands-on Labs:

None

Week 8: Serverless Applications – 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 AWS 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

Hands-on Labs:

None

Week 8: Serverless Applications – 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, including 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

Hands-on Labs:

- **Lab: Working with DynamoDB (60 min)**
  - In this lab, you gain hands-on experience working with Amazon 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, store it in variables, and create new resources.
- **Lab: Working with Lambda (45 min)**
  - In this lab, you continue from where you left off in the previous lab. There is an Amazon DynamoDB table named LabCustomers already in place that contains 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.

Week 9: Serverless Applications – 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 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

Hands-on Labs:

- **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. Then, you update the webpage to invoke the endpoint.

Week 10: Serverless Applications – Microservice Integration Patterns – 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 can 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!

Hands-on Labs:

- **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.
- Understand 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

Hands-on Labs:

None

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

Hands-on Labs:

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