AWS Cloud Institute: Cloud Fundamentals 2 Syllabus

Course overview

In this course, you will focus on the concept of serverless computing and utilizing AWS Lambda. The course covers how to create and manage AWS Lambda functions, construct a serverless workflow, and discuss AWS Step Function activities.

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 online learning content in greater depth, connect with AWS instructors and ACI learners, and prepare for the weekly final assessment. All ILT sessions, except for 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.

Course completion requirements

At the end of each 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 final assessments with a score of 80% or higher to receive credit for the course. ILT attendance or watching ILT recordings is strongly encouraged but is not required for course completion.

Week 1: Security 3 – Part 1

Goal

Gain a deeper understanding of implementing security measures, using temporary credentials, applying security protocols to AWS Lambda functions, and using access controls based on tags for efficient resource management.

Learning objectives

- Review security benefits and differences for temporary and long-term credentials
- Describe AWS Security Token Service (AWS STS) APIs and provide scenarios for AWS STS using AWS SDKs and AWS CLI
- Use Amazon CloudWatch Logs and AWS CloudTrail to monitor and audit AWS resources accessed with temporary credentials
- Explain how to use AWS Identity and Access Management (IAM) roles to grant permissions to applications running on Amazon Elastic Compute Cloud (Amazon EC2) instances
- Provide temporary security credentials to a third-party API using an SDK
- Describe how to implement least privilege with Lambda execution roles
- Identify common AWS managed policies for Lambda
- Describe how to control function invocation with AWS IAM
- Explain trusted code execution
- Configure trusted code signing using AWS Lambda

Outline

1. **Working with Temporary Credentials**
    a. **Working with Temporary Credentials**

i. Managing Long-Term and Short-Term User Credentials
ii. Lab: Implementing an OIDC IdP for Enhanced Security and Identity
iii. The Role of AWS Security Token Service for Temporary Credentials
iv. Using AWS CLI and AWS SDK with Temporary Credentials
v. Logging AWS STS API Calls with AWS CloudTrail
vi. Activity: Monitoring Temporary Credential Use
vii. Using IAM Roles for Amazon EC2 Instances
viii. Tag-based Temporary Credentials
ix. Knowledge Check

2. **Security with AWS Lambda**
    a. **Security with Lambda**
        i. AWS Lambda Security Best Practices
        ii. Common AWS Managed Policies for AWS Lambda
        iii. AWS Lambda Function Invocation with IAM
        iv. Activity: Identifying Restrictions or AWS Lambda Invocations
        v. Code Signing for AWS Lambda
        vi. Knowledge Check

Hands-on labs

- **Lab: Implementing an OIDC IdP for Enhanced Security and Identity (60 min)**
  This lab demonstrates how to secure a web application by using temporary credentials.

.

Week 2: Security 3 – Part 2

Goal

Learn how to establish and maintain a data perimeter and use encryption to protect data in transit and at rest, and how to efficiently respond to security incidents.

Learning objectives

- Identify data classification models and schemes
- Define the objectives for a data perimeter
- Describe establishing a data perimeter on AWS for three different objectives
- Create a data perimeter
- Review the importance of encrypting data in transit and at rest
- Review methods for encrypting data at rest in Amazon Simple Storage Service (Amazon S3)
- Implement S3 bucket policies to enforce encryption
- Review encryption options for Amazon Relational Database Service (Amazon RDS)
- Monitor the encryption status of Amazon RDS instances
- Review encryption options for Amazon DynamoDB
- Verify encryption settings for DynamoDB
- Discuss AWS Encryption SDK
- Define an indicator of compromise
- Identify the aspects of AWS incident response
- Identify differences between traditional incident response and incident response in the cloud
- Provide examples of incident response playbooks
- Explore the Compromised IAM Credentials Playbook

Outline

1. **Introduction to Data Perimeters on AWS**
    a. **Data Perimeters on AWS**
        i. Data Classification
        ii. Establishing and Implementing Data Perimeters
        iii. Securing a Data Perimeter with a Zero Trust Architecture
        iv. Lab: Zero Trust Architecture for Service-to-Service Workloads
        v. Knowledge Check
2. **Data Encryption at Rest and In Transit**
    a. **Data Encryption at Rest and In Transit**
        i. Encrypting Data at Rest and Data in Transit
        ii. Activity: Choosing Encryption Tools
        iii. Encrypting Data at Rest in Amazon S3
        iv. Lab: Implementing Amazon S3 Bucket Policies to Enforce Encryption
        v. Encrypting Database Data at Rest
        vi. AWS Encryption SDK
        vii. Knowledge Check
3. **Incident Response**
    a. Incident Response
        i. AWS Incident Response
        ii. Incident Response: Traditional and Cloud
        iii. Incident Response Playbooks
        iv. Activity: Top Incident Response Tips
        v. Lab: Responding to Incidents in an AWS Environment
        vi. Knowledge Check

Hands-on labs

- **Lab: Zero Trust Architecture for Service-To-Service Workloads (75 min)**
  In this lab, you will apply Zero Trust principles to the service-to-service architectures used in many micro-architecture or distributed environments today.
- **Lab: Implementing Amazon S3 Bucket Policies to Enforce Encryption (60 min)**
  In this lab, you will explore the various ways to secure data stored in Amazon Simple Storage Service (Amazon S3).
- **Lab: Responding to Incidents in an AWS Environment (60 min)**
  In this lab, you will act as a member of the incident response team receiving an alert on a possible Amazon EC2 instance that might be compromised by a bad actor.

Great job! This is the best course of action because...

A request to GetSessionToken passes two parameters to the API: The SerialNumber and TokenCode of the user's MFA device.

*Choose **Continue** to move on...*

Continue

Week 3: Networking 3 – Part 1

Goal

Learn to design and implement secure hybrid network architectures on AWS, with a strong focus on core hybrid connectivity services like AWS Direct Connect, AWS Site-to-Site VPN, and Cloud WAN.

Learning objectives

- Define and describe use cases for hybrid network connectivity
- Identify AWS services you can use to connect on-premises and AWS
- Recall the definition, purpose, and use cases of AWS Direct Connect
- Explain how AWS Direct Connect works and describe its key components
- Identify the AWS Direct Connect connection types and describe the differences between them
- Differentiate between Direct Connect virtual interfaces and describe when to use them
- Define concept Virtual Private Network (VPN)
- Recall the definition, purpose, and use cases of AWS VPN
- Differentiate between the two types of AWS VPN: AWS Client VPN and AWS Site-to-Site VPN
- Describe how AWS Site-to-Site VPN works and define its key components
- Describe the different connection methods for a site-to-site connection to AWS and when to use each
- Describe how AWS Client VPN works and define its key components
- Describe AWS Client VPN authentication methods
- Describe how to troubleshoot AWS Client VPN
- Define wide area network (WAN)
- Explain the common challenges in wide area networks
- Recall the definition, purpose, and use cases of AWS Cloud WAN
- Describe how AWS Cloud WAN works and explain its core components and architecture
- Differentiate between AWS Cloud WAN and AWS Transit Gateway
- Describe the purpose and characteristics of an AWS Cloud WAN core network policy
- Demonstrate the ability to use core functionalities of AWS Client VPN service

Outline

1. **Hybrid Networking Overview**
    a. **Hybrid Networking Overview**
        i. Introducing Hybrid Networks
        ii. AWS Hybrid Connectivity Services Overview
        iii. Knowledge Check
2. **Using AWS Site-to-Site VPN to Connect Your Data Center to AWS**
    a. **Using AWS Site-to-Site VPN to Connect Your Data Center to AWS**
        i. Introducing AnyCompany Robots
        ii. VPN Overview
        iii. Key Components of AWS Site-to-Site VPN
        iv. How AWS Site-to-Site VPN Works
        v. Routing in AWS Site-to-Site VPN
        vi. Troubleshooting AWS Site-to-Site VPN
        vii. Knowledge Check
3. **Using AWS Client VPN to Connect Remote Users to the Cloud**
    a. **Using AWS Client VPN to Connect Remote Users to the Cloud**

        a. Key Components of AWS Client VPN
        b. How AWS Client VPN Works
        c. Lab: Configuring AWS Client VPN
        d. Knowledge Check
4. **Using AWS Direct Connect to Create a Dedicated Network Connection to AWS**
    a. **Using AWS Direct Connect to Create a Dedicated Network Connection to AWS**
        a. Key Components of AWS Direct Connect
        b. How Direct Connect Works
        c. Comparing Direct Connect and AWS VPN
        d. Direct Connect Resiliency Architecting
        e. Knowledge Check
5. **Using AWS Cloud WAN to Create a Global Network**
    a. **Using AWS Cloud WAN to Create a Global Network**
        a. Introducing WANs
        b. Introduction to AWS Cloud WAN
        c. Key Components and Architecture of AWS Cloud WAN
        d. Knowledge Check

Hands-on labs

- **Lab: Creating AWS Client VPN (60 min)**
  In this lab, you will use AWS Client VPN to generate the necessary certificates and keys required for mutual authentication needed for the Client VPN endpoint, configure the Client VPN endpoint and install OpenVPN on an Amazon EC2 instance, and test the connectivity.

Week 4: Networking 3 – Part 2

Goal

Understand the foundations and configurations of multi-layered security solutions using AWS services like AWS Network Firewall, AWS Shield, AWS WAF, and Amazon Route 53 Resolver DNS Firewall to protect the network and applications from various threats.

Learning objectives

- Understand the purpose of the security pillar of the AWS Well-Architected Framework
- Describe an example of an architecture with multi-layered security and identify the AWS services that can be used for multi-layered network security
- Recall the purpose of the DNS protocol
- Describe common DNS-level attacks, with a focus on DNS exfiltration
- Recall the definition of Route 53
- Define Route 53 Resolver DNS Firewall and describe its key components
- Explain how to use Route 53 resolver rule groups and rules
- Describe the high-level steps for using Route 53 Resolver DNS Firewall
- Explain the different actions that can be taken for filtered DNS queries such as allow, block, and alert
- Demonstrate how to create DNS firewall rules to block DNS queries containing specific domain names
- Define and describe the most common types of Distributed denial of service (DDoS) attacks
- Explain the core features and benefits of AWS Shield to protect applications from DDoS attacks
- Explain why it’s beneficial to go beyond security groups and NACLs for network security

- Recall the definition and purpose of a network firewall
- Recall the definition and purpose of AWS Network Firewall and describe its key features, core components, and use cases
- Differentiate between AWS Network Firewall centralized, distributed, and combined deployment types
- Demonstrate how to create a basic firewall policy with rules to filter inbound and outbound traffic
- Describe the integration of Network Firewall with other AWS services like Amazon VPC, ALB, CloudWatch
- Describe common L7 attacks and their impact
- Define AWS WAF and describe its key features
- Identify AWS resources that AWS WAF can protect and describe the security automations available for AWS WAF
- Describe the purpose of AWS WAF Web ACLs, rules, and rule groups
- Describe AWS WAF web ACL components
- Describe how web ACLs handle requests
- Describe the AWS WAF rule types and explain when to use them

Outline

1. **Exploring Security Fundamental**
    a. **Exploring Security Fundamentals**
        i. Understanding Multi-layered Security
        ii. Exploring Common Threats
        iii. Knowledge Check
2. **Filtering Network Traffic with AWS Network Firewall**
    a. **Filtering Network Traffic with AWS Network Firewall**
        i. AWS Network Firewall Overview
        ii. How AWS Network Firewall Works
        iii. AWS Network Firewall Deployment Models
        iv. Lab: Using AWS Network Firewall for Inbound/Outbound Traffic
        v. Knowledge Check
3. **Protecting Your Network Against DDoS Attacks with AWS Shield**
    a. **Protecting Your Network Against DDoS Attacks with AWS Shield**
        i. AWS Shield Overview
        ii. How AWS Shield Advanced Works
        iii. Knowledge Check
4. **Protecting Web Applications from Common Exploits with AWS WAF**
    a. **Protecting Web Applications from Common Exploits with AWS WAF**
        i. AWS WAF Overview
        ii. How AWS WAF Works
        iii. AWS WAF Intelligent Threat Mitigation
        iv. Knowledge Check
5. **Filtering DNS Traffic Using the Amazon Route 53 Resolver DNS Firewall**
    a. **Filtering DNS Traffic Using the Amazon Route 53 Resolver DNS Firewall**
        i. Amazon Route 53 Review
        ii. Amazon Route 53 Resolver Firewall Overview
        iii. Knowledge Check

Hands-on labs

- **Lab: Using AWS Network Firewall for Inbound/Outbound Traffic (60 min)**
  This lab demonstrates the use of AWS Network Firewall to filter outbound web traffic using resources that are provisioned as part of the lab.

Week 5: Databases and Caching: Part 1

Goal

Explore characteristics of purpose-built databases and describe the main components and common use cases for DynamoDB. Practice creating, manipulating, and deleting DynamoDB tables in labs and activities.

Learning objectives

- Describe trends in data generation and consumption
- Describe the characteristics and use cases of different purpose-built databases
- Compare NoSQL and SQL database characteristics and use cases
- Define serverless computing
- Describe the purpose, main features, and common use cases of DynamoDB
- Describe the purposes of the core components of DynamoDB
- Describe the primary keys and indexes supported by DynamoDB
- Describe the data types supported by DynamoDB
- Create, manipulate, and delete a DynamoDB table
- Add query data within a DynamoDB table

Outline

1. **Developing Modern Applications Using DynamoDB**
    a. **Getting Started**
        i. Meet AnyCompany Video Games
    b. **Understanding Purpose-Built Databases**
        i. Trends in Data Generation and Consumption
        ii. Characteristics and Use Cases of Purpose-Built Databases
        iii. Comparing NoSQL and SQL Databases
        iv. DynamoDB
        v. Knowledge Check
    c. **Understanding DynamoDB Components**
        i. DynamoDB Tables
        ii. DynamoDB Keys and Table Design
        iii. Interacting with DynamoDB
        iv. DynamoDB Secondary Indexes
        v. Lab: Working with Amazon DynamoDB Tables and Indexes
        vi. Knowledge Check

Hands-on labs

- **Lab: Working with Amazon DynamoDB Tables and Indexes (45 min)**
  AnyCompany Video Games is looking to enhance its in-game experience by implementing a robust player database to support a friend service feature. You are tasked with creating the appropriate DynamoDB table and indexes tailored to meet the specific needs of this gaming environment. In this lab, you learn how to create

a DynamoDB table optimized for storing player data. You also create secondary indexes to provide fast and efficient access to player data, so your gaming platform can deliver seamless experiences for players connecting with friends in real time.

Week 6: Databases and Caching: Part 2

Goal

Explore DynamoDB concepts that can help optimize performance and secure application data. Learn about DynamoDB features, like table capacity modes, read and write capacity units, auto scaling, Time to Live (TTL), DynamoDB Streams, and global tables. Practice adjusting table capacity in DynamoDB and processing DynamoDB Streams using AWS Lambda.

Learning objectives

- Compare provisioned and on-demand capacity modes
- Compare and describe read and write capacity units
- Explain and implement DynamoDB auto scaling
- Explain what DynamoDB Time to Live (TTL) is and how it can be used
- Process Amazon DynamoDB Streams using AWS Lambda
- Describe the security and resiliency mechanisms offered by DynamoDB
- Describe DynamoDB global and replica tables
- Describe DynamoDB global tables use cases
- Create a DynamoDB global table
- Describe how to use AWS IAM with DynamoDB
- Explain how to use Amazon Virtual Private Cloud (Amazon VPC) endpoints for DynamoDB access control

Outline

1. **Exploring DynamoDB Advanced Concepts**
    a. **Optimizing DynamoDB**
        i. Capacity Modes
        ii. Read and Write Capacity Units
        iii. DynamoDB Auto Scaling
        iv. Lab: Adjusting Table Capacity in DynamoDB
        v. Capturing and Tracking Changes to DynamoDB Tables
        vi. Lab: Process DynamoDB Streams Using AWS Lambda
        vii. Knowledge Check
    b. **Securing DynamoDB**
        i. DynamoDB Security and Resiliency
        ii. DynamoDB Global Tables
        iii. Tutorial: Creating a Global Table
        iv. DynamoDB Access Control
        v. VPC Endpoints for DynamoDB
        vi. Knowledge Check

Hands-on labs

- **Lab: Adjusting Table Capacity in DynamoDB (60 min)**
  In this lab, you use Amazon DynamoDB provisioned read/write capacity mode to manage the throughput capacity. You use CloudWatch to monitor different DynamoDB metrics, affecting the performance of your table. You then use Amazon DynamoDB auto scaling feature to dynamically adjust provisioned throughput capacity on the table, to handle sudden increases in traffic without throttling.
- **Lab: Processing Amazon DynamoDB Streams Using AWS Lambda (60 min)**
  In this lab, you enable a DynamoDB stream on a table and connect an AWS Lambda function to the stream as a listener to process real-time data changes in the table. You then enable Time To Live (TTL) for automatic item expiration, set time requirements for item deletion, and verify findings via lambda logs.
- **Tutorial: Creating a Global Table**
  Learn how to create an Amazon DynamoDB global table in the DynamoDB console.

Week 7: Databases and Caching: Part 3

Goal

Learn how to describe use cases, strategies, and best practices for caching. Explore cloud services, including Amazon DynamoDB Accelerator (DAX), Amazon ElastiCache, and Amazon CloudFront to implement caching and analyze caching behavior.

Learning objectives

- Define caching
- Provide a high-level overview of the AWS services that support caching and their use cases
- Describe the challenges database caching solves
- Explain how DAX works as an in-memory cache for DynamoDB to improve response times for read operations
- Identify use cases where using DAX can improve application performance by reducing read latency
- Describe best practices for DAX cluster sizing, node types, encryption, and availability based on application requirements
- Compare and contrast caching reads using DAX versus other strategies like pre-fetching data
- Configure a DAX cluster and integrate it with an existing DynamoDB table for caching
- Describe Amazon ElastiCache, its benefits, and use cases
- Explain the benefits of caching services like Redis and Memcached and compare their features
- Explain how Amazon ElastiCache works with Redis or Memcached
- Describe Amazon ElastiCache caching strategies
- Implement Amazon ElastiCache caching logic
- Describe Amazon CloudFront
- Analyze caching behavior using Amazon CloudFront access logs and metrics to identify optimization opportunities
- Describe troubleshooting techniques for common Amazon CloudFront caching issues related to object invalidation, restricted content access, and origin overload scenarios
- Add an Amazon CloudFront distribution to a WordPress instance

Outline

1. **Caching on AWS**
    a. **Understanding Caching**
        i. Caching
        ii. Database Caching
        iii. Knowledge Check
    b. **Caching DynamoDB with DAX**
        i. DAX Benefits and Functionality
        ii. DAX Use Cases
        iii. DAX Best Practices
        iv. Comparing DAX to Other Caching Strategies
        v. Lab: Configuring a DAX Cluster for Caching with an Existing Amazon DynamoDB Table
        vi. Knowledge Check
    c. **Caching Your Application Data with Amazon ElastiCache**
        i. ElastiCache
        ii. Comparing ElastiCache to Alternatives
        iii. ElastiCache for Memcached
        iv. ElastiCache for Redis
        v. Lab: Using Amazon ElastiCache to Implement Caching
        vi. ElastiiCache Advanced Concepts
        vii. Demo: Database Caching with ElastiCache for Redis
        viii. Knowledge Check
    d. **Using Amazon CloudFront to Implement Global Caching**
        i. CloudFront
        ii. CloudFront Cache and Origin Functionality
        iii. Optimizing CloudFront
        iv. Integrating CloudFront with Other AWS Services
        v. Lab: Adding an Amazon CloudFront Distribution for Dynamic Content Acceleration
        vi. Troubleshooting CloudFront
        vii. Knowledge Check

Hands-on labs

- **Lab: Configuring a DAX Cluster for Caching with an Existing Amazon DynamoDB Table (60 min)**
  In this lab, you set up a DAX cluster to cache data from an existing DynamoDB table. By implementing caching with DAX, applications that use DynamoDB can achieve significant performance improvements and reduce demand on the underlying tables. This can help improve latency for end users and optimize costs.
- **Lab: Using Amazon ElastiCache to Implement Caching (60 min)**
  In this lab, you create an ElastiCache for Redis cluster, authorize access to your ElastiCache cluster, and connect to your cluster to run commands. Additionally, you also clean up your resources by deleting an ElastiCache cluster.
- **Lab: Adding an Amazon CloudFront Distribution for Dynamic Content Acceleration (60 min)**
  In this lab, you create an Amazon CloudFront distribution in the AWS Management Console, customize your Amazon CloudFront distribution, and log in to your Dynamic Application.

Week 8: Compute 3: Part 1

Goal

Review how to use Amazon EC2 with applications, with a particular focus on Python dependencies. Practice running a Python script on an Amazon EC2 instance and build on existing Lambda knowledge by using key Lambda functions for applications.

Learning objectives

- Describe Amazon EC2 benefits and use cases for applications
- Describe remote administration and package management on Amazon EC2
- Create AWS Lambda functions.
- Describe the benefits and uses of AWS Lambda
- Describe the packaging of Python code for Lambda
- Describe Lambda function logging
- Describe Lambda function handler and event

Outline

1. **Using Amazon EC2 and Lambda for Applications**
    a. **Using Amazon EC2 for Applications**
        i. Building your Amazon EC2 Expertise
        ii. Working with Amazon EC2 Instance Profiles
        iii. Remote Administration on Amazon EC2
        iv. Package Management on Amazon EC2
        v. Python Dependencies in Amazon EC2
        vi. Lab: Apply Key Amazon EC2 Functions Used for Applications
        vii. Knowledge Check
    b. **Using Lambda for Applications**
        i. Serverless
        ii. AWS Lambda
        iii. Authoring Lambda Functions
        iv. Managing Packages in Lambda
        v. Monitoring Lambda Functions
        vi. Lambda Example
        vii. Lab: Applying Key Lambda Functions to Applications
        viii. Knowledge Check

Hands-on labs

- **Lab: Applying Key Amazon EC2 Functions Used for Applications (30 Mins)**
  In this lab, you will learn how to apply key Amazon EC2 functions used for running applications. You will launch an Amazon EC2 instance, connect to it using AWS Session Manager, install Python and necessary dependencies, and run a simple Python script that acts as a "URL checker."
- **Lab: Applying Key Lambda Functions to Applications (60 Mins)**
  In this lab, you will deploy the URL checker application on AWS Lambda, allowing you to run code without provisioning or managing servers. This is a convenient and cost-effective option for running applications. By the end of this lab, you will have successfully created and invoked a Lambda function that checks the availability of various URLs.

Week 9: Compute 3: Part 2

Goal

Review how container computing works and gain a better understanding of container computing services, including Amazon Elastic Kubernetes Service (Amazon EKS), Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Container Registry (Amazon ECR). Further compare container computing services with Amazon EC2 and Amazon Lambda to identify key differentiators for selecting a compute service.

Learning objectives

- Describe how Docker is used with AWS container services
- Explain how Amazon EKS manages the Kubernetes control plane and can also manage elements of the data plane
- Describe how pods communicate with each other and interact with hosts in Amazon EKS clusters
- Describe where Amazon EKS and Amazon ECS each integrate with other AWS services.
- Explain how upgrades are handled in Amazon EKS
- Outline the Amazon EKS and Amazon ECS pricing structure and manage cost
- Explain how to create an Amazon ECS cluster
- Use Amazon ECR with Docker
- Use an existing Docker container with an Amazon EKS cluster
- Use an existing Docker container with Amazon ECS
- Compare application logging, package management, and runtime management for Amazon EC2, Lambda, and containerized applications

Outline

1. **Practice with Container Computing**
    a. **Container Computing Overview**
        i. Containers Concepts Overview
        ii. Getting Started with Docker
        iii. Using Image Repositories
        iv. Lab: Using Dockerfile with Amazon ECR
        v. Deploying Containers at Scale
        vi. Knowledge Check
    b. **Working with Amazon EKS**
        i. Amazon EKS Overview
        ii. Configuring Amazon EKS
        iii. Integrating Amazon EKS with Other Services
        iv. Maintaining your Amazon EKS Cluster
        v. Managing Amazon EKS Costs
        vi. Lab: Deploying Containerized Applications to Amazon EKS
        vii. Knowledge Check
    c. **Working with Amazon ECS**
        i. Amazon ECS Overview
        ii. Defining Tasks and Services
        iii. Configuring Amazon ECS
        iv. Using Amazon ECS with External Services
        v. Managing Amazon ECS Costs
        vi. Lab: Deploying Containers on AWS Fargate Using Amazon ECS and Amazon ECR
        vii. Knowledge Check

   d. **Selecting a Compute Service**
      i. Comparing AWS Compute Services
      ii. Addressing Python Deployment Considerations
      iii. Knowledge Check

Hands-on labs

- **Lab: Using Dockerfile with Amazon ECR (30 min)**
  In this lab, you will learn how to build Docker images, test them locally, and then push them to ECR repositories for storage and deployment.
- **Lab: Deploying Containerized Applications to Amazon EKS (60 min)**
  In this hands-on lab, you deploy and run a pre-built containerized application on an Amazon Elastic Kubernetes Service (Amazon EKS) cluster using an AWS Cloud9 development environment. You will gain experience with the workflow of managing Kubernetes on AWS using multiple tools and approaches.
- **Lab: Deploying Containers on AWS Fargate Using Amazon ECS and Amazon ECR (60 min)**
  In this hands-on lab, you will learn how to build, deploy, and run containerized applications on AWS using Amazon Elastic Container Service (Amazon ECS), Elastic Container Registry (Amazon ECR), and Fargate.

Week 10: Architecting 2: Part 1

Goal

Learn the key principles and benefits of the AWS Well-Architected Framework and explore foundational architectural concepts of modern cloud applications, including the shift from monolithic to microservice designs. Examine the characteristics and use cases of event-driven architectures, and gain hands-on experience building decoupled architectures with services like Amazon EventBridge, Amazon SQS, Amazon SNS, and DynamoDB.

Learning objectives

- Describe the benefits of the six pillars of the Well-Architected Framework
- Discuss customer use cases for the Sustainability pillar of the Well-Architected Framework
- Compare monolithic and microservice applications
- Discuss the shifts that occur when changing from monolithic to microservice applications
- Describe client-service communications in modern applications
- Describe the benefits of Event Driven Architectures (EDAs), including their key characteristics and features, and common use cases
- Explain why EDAs promote the use of microservices and sustain eventual consistency
- Describe how EDAs support strategic reactions to infrastructure changes and autoscaling based on custom data
- Describe ideal use cases for using EventBridge and DynamoDB to decouple architectures
- Describe use cases for using DynamoDB and Amazon RDS to achieve eventual consistency
- Describe how to use Amazon SQS, Amazon SNS, and DynamoDB in event-driven architectures
- Build a serverless architecture with Amazon SQS and Amazon SNS
- Describe how to use DynamoDB Streams to decouple code
- Build decoupled architectures with EventBridge

Outline

1. **Building Modern, Event-Driven Applications**
   a. **Understanding Modern Applications**
      i. The Well-Architected Framework
      ii. Modern Applications

iii. Microservices
iv. Converting to Microservices
v. Knowledge Check

b. Designing Modern Event-Driven Applications

i. Event-Driven Architectures
ii. Eventual Consistency in Event-Driven Architectures
iii. Knowledge Check

c. Implementing Event-Driven Architectures With AWS Services

i. Using Amazon SQS in Event-Driven Architectures
ii. Using Amazon SNS in Event-Driven Architectures
iii. Lab: Using Amazon SNS and SQS in Event-Driven Architectures
iv. Using DynamoDB in Event-Driven Architectures
v. Using EventBridge in Event-Driven Architectures
vi. Lab: Building Decoupled Architectures with Amazon EventBridge
vii. Knowledge Check

Hands-on labs

- **Lab: Using Amazon SNS and SQS in Event-Driven Architectures (60 min)**
  In this lab, you configure an Amazon Simple Storage Service (Amazon S3) bucket to invoke an Amazon Simple Notification Service (Amazon SNS) notification whenever an object is added to an Amazon S3 bucket. You learn how to create and interact with Amazon Simple Queue Service (Amazon SQS) queues, and learn how to invoke an AWS Lambda function using Amazon SQS.
- **Lab: Building Decoupled Architectures with Amazon EventBridge (90 min)**
  In this lab, you will learn how to use a HTTP API to drive event driven architecture and invoke AWS Lambda functions by triggering EventBridge rules directly from Amazon API Gateway.

Week 11: Architecting 2: Part 2

Goal

Understand the difference between batch and real-time approaches to processing application data, with a focus on the AWS services used to support each processing type. Practice using Amazon Kinesis Data Steaming to ingest and analyze real-time data.

Learning objectives

- Describe batch processing
- Describe best practices for batch processing based on the Well-Architected Framework
- Describe batch processing using AWS Step Functions, AWS Lambda, Amazon Elastic Map Reduce (Amazon EMR), and AWS Batch
- Describe how to use Amazon EMR
- Describe real-time data processing, streaming applications, and the AWS services that support them
- Describe the stream processing paradigm
- Describe Amazon Kinesis Data Streaming, its common use cases, and how it is optimized for real-time data streaming
- Describe how to use Amazon Kinesis, DynamoDB, AWS OpenSearch, GraphQL, Amazon MemoryDB for Redis, and Amazon ElastiCache to support real-time application architectures and their ideal use cases

Outline

1. **Batch and Real-Time Processing Architectures**
    a. **Designing a Batch Processing Application Architecture**
        i. Batch Processing Architectures
        ii. Implementing Batch Processing with AWS
        iii. How Step Functions Supports Batch Processing
        iv. Use Cases: Step Functions for Batch Processing
        v. Knowledge Check
    b. **Designing Real-Time Applications Architecture**
        i. Lab Environment: Navigating Through Kinesis
        ii. Real-Time Processing Architectures
        ii. Implementing Real-Time Processing with AWS
        iii. Use Cases: Using AWS Services for Real-Time Processing
        iv. Kinesis Data Streams and OpenSearch Service
        v. Lab: Navigating Through Kinesis
        vi. Real-Time Data Storage Options
        vii. Evaluating Storage Options
        viii. Activity: Selecting the Appropriate Storage Option
        ix. Knowledge Check

Hands-on labs

- **Lab Environment: Navigating Through Kinesis (60 min)**
  This contains a lab, "Navigating Through Kinesis" that requires 30-60 minutes to provision.
- **Lab: Navigating Through Kinesis (60 min)**
  This is a two-part lab. First, you will create a Lambda function from a blueprint, create an Amazon Kinesis Stream, and then trigger the function with data from your stream and monitor the process with Amazon CloudWatch. Next, you will learn the basics of event-driven programming using Amazon DynamoDB, its Streams feature, and AWS Lambda. You will walk through the process of building a real-world application using AWS Triggers, which combines DynamoDB Streams and Lambda.