# AWS DevOps Course Content

## Week 1: Introduction to AWS and DevOps Principles

### Day 1: Introduction to AWS and Cloud Computing

**1. What is Cloud Computing?**

- Definition of Cloud Computing
- Types of Cloud Services (IaaS, PaaS, SaaS)
- Public, Private, and Hybrid Clouds
- Benefits of Cloud Computing

**2. Overview of AWS Global Infrastructure**

- AWS Regions and Availability Zones
- Edge Locations and Content Delivery Networks
- Key AWS Services Overview (Compute, Storage, Databases, Networking)

**3. Introduction to AWS Management Console**

- Navigating the AWS Management Console
- Overview of AWS Command Line Interface (CLI)
- Introduction to AWS SDKs

**4. Setting up AWS Free Tier Account**

- Creating an AWS Account
- Understanding Free Tier Limits
- Billing and Cost Management Dashboard

### Day 2: AWS Identity and Access Management (IAM)

**1. Introduction to IAM**

- Definition and Importance of IAM
- Core IAM Components (Users, Groups, Roles, Policies)
- IAM Best Practices

**2. Creating IAM Users, Groups, and Policies**

- Steps to Create IAM Users
- Creating and Assigning Groups
- Writing and Attaching Policies
- Assigning Policies to Users and Groups

**3. Role-based Access Control and MFA**

- Overview of Roles in IAM
- Creating and Using IAM Roles
- Configuring Multi-Factor Authentication (MFA)
- Best Practices for MFA

### 4. Security Best Practices

- Principle of Least Privilege
- Using AWS Organizations for Account Management
- Enabling IAM Access Analyzer
- Regular IAM Policy Audits

## Day 3: Introduction to DevOps

### 1. What is DevOps?

- Definition and History of DevOps
- DevOps vs. Traditional IT
- Key DevOps Goals and Metrics

### 2. DevOps Principles and Practices

- Continuous Integration and Continuous Deployment (CI/CD)
- Infrastructure as Code (IaC)
- Automation and Configuration Management
- Monitoring and Logging
- Collaboration and Communication

### 3. Benefits of DevOps

- Faster Time to Market
- Improved Deployment Frequency
- Lower Failure Rate of New Releases
- Shortened Lead Time for Changes
- Improved Mean Time to Recovery

### 4. Overview of DevOps Tools and Technologies

- Version Control Systems (Git, SVN)
- CI/CD Tools (Jenkins, CircleCI, AWS CodePipeline)
- Configuration Management Tools (Ansible, Chef, Puppet)
- Containerization (Docker, Kubernetes)
- Monitoring and Logging Tools (Prometheus, Grafana, ELK Stack)
- Cloud Platforms (AWS, Azure, Google Cloud Platform)

## Day 4: Version Control with Git and AWS CodeCommit

### 1. Introduction to Version Control Systems

- Definition and Importance of Version Control
- Centralized vs. Distributed Version Control Systems
- Key Concepts (Repositories, Commits, Branches, Merges)

### 2. Setting Up Git and GitHub

- Installing Git on Different Operating Systems
- Configuring Git (Username, Email, Aliases)
- Creating a GitHub Account
- Creating and Cloning Repositories on GitHub

### 3. Overview of AWS CodeCommit

- Introduction to AWS CodeCommit
- Benefits of Using CodeCommit
- Key Features of CodeCommit (High Availability, Security, Integration)

### 4. Creating and Managing Repositories in CodeCommit

- Creating a Repository in CodeCommit
- Cloning CodeCommit Repositories
- Adding, Committing, and Pushing Changes
- Managing Branches and Merges in CodeCommit

## Day 5: Continuous Integration and Continuous Delivery (CI/CD)

### 1. Introduction to CI/CD

- Definition of Continuous Integration (CI)
- Definition of Continuous Delivery (CD)
- Difference between Continuous Delivery and Continuous Deployment
- Key CI/CD Concepts and Practices

### 2. Benefits of CI/CD

- Faster Feedback Loops
- Improved Software Quality
- Reduced Risk of Deployment Failures
- Enhanced Collaboration and Communication

### 3. CI/CD Pipeline Overview

- Stages of a CI/CD Pipeline (Source, Build, Test, Deploy)
- Tools for Each Stage of the Pipeline
- Best Practices for Designing CI/CD Pipelines

### 4. Tools and Best Practices

- Choosing the Right CI/CD Tools

- Integrating CI/CD Tools with Version Control Systems
- Automating Tests and Deployments
- Monitoring and Managing CI/CD Pipelines

# Week 2: AWS CI/CD Tools

## Day 1: AWS CodeBuild

### 1. Introduction to AWS CodeBuild

- What is AWS CodeBuild?
- Key Features of CodeBuild
- Use Cases for CodeBuild

### 2. Creating and Managing Build Projects

- Steps to Create a Build Project in CodeBuild
- Configuring Build Environments (Runtime, Compute Type)
- Specifying Source Repositories
- Defining Build Artifacts

### 3. Configuring Buildspec Files

- Introduction to buildspec.yml Files
- Structure of a buildspec.yml File
- Defining Phases (Install, Pre-build, Build, Post-build)
- Specifying Artifacts, Cache, and Environment Variables

### 4. Integrating CodeBuild with CodeCommit

- Connecting CodeBuild to CodeCommit Repositories
- Triggering Builds Automatically with CodeCommit Pushes
- Viewing Build Logs and Reports
- Troubleshooting Common Build Issues

## Day 2: AWS CodeDeploy

### 1. Introduction to AWS CodeDeploy

- What is AWS CodeDeploy?
- Key Features of CodeDeploy
- Use Cases for CodeDeploy

### 2. Creating Deployment Applications and Groups

- Steps to Create a Deployment Application
- Defining Deployment Groups
- Associating EC2 Instances and On-premises Servers

- Configuring Deployment Settings (Rollbacks, Notifications)

### 3. Deployment Strategies (In-Place and Blue/Green)

- Overview of In-Place Deployment Strategy
- Steps for Performing In-Place Deployments
- Overview of Blue/Green Deployment Strategy
- Steps for Performing Blue/Green Deployments

### 4. Integrating CodeDeploy with CodeCommit and CodeBuild

- Setting Up CodeDeploy Deployment Configurations
- Connecting CodeDeploy to CodeCommit and CodeBuild
- Automating End-to-End Deployments
- Monitoring and Troubleshooting Deployments

## Day 3: AWS CodePipeline

### 1. Introduction to AWS CodePipeline

- What is AWS CodePipeline?
- Key Features of CodePipeline
- Use Cases for CodePipeline

### 2. Creating and Managing Pipelines

- Steps to Create a Pipeline in CodePipeline
- Defining Pipeline Stages (Source, Build, Test, Deploy)
- Configuring Stage Actions and Transitions
- Managing Pipeline Revisions and Versions

### 3. Integrating CodeCommit, CodeBuild, and CodeDeploy

- Setting Up Source Stage with CodeCommit
- Configuring Build Stage with CodeBuild
- Defining Deploy Stage with CodeDeploy
- Automating the Entire CI/CD Pipeline

### 4. Monitoring and Managing Pipelines

- Viewing Pipeline Status and History
- Configuring Pipeline Notifications and Alerts
- Troubleshooting Pipeline Failures
- Best Practices for Pipeline Management

## Day 4: Hands-on Lab: Setting Up a CI/CD Pipeline

### 1. Step-by-step Guide to Create a CI/CD Pipeline

- Setting Up Source Repository with CodeCommit
- Creating Build Project with CodeBuild
- Defining Deployment Application with CodeDeploy
- Integrating Stages in CodePipeline

#### 2. Testing the Pipeline with Sample Applications

- Creating a Sample Application
- Committing Changes to Trigger Pipeline
- Monitoring Build and Deployment Processes
- Validating Application Deployment

#### 3. Debugging and Troubleshooting Common Issues

- Common CI/CD Pipeline Errors
- Analyzing Build and Deployment Logs
- Fixing Configuration and Code Issues
- Retriggering Failed Pipeline Stages

### Day 5: Project Review and Q&A

#### 1. Review of CI/CD Concepts

- Recap of Key CI/CD Principles and Practices
- Summary of AWS CI/CD Tools
- Importance of CI/CD in DevOps

#### 2. Project Walkthrough and Discussion

- Detailed Review of Hands-on Lab Project
- Discussion of Challenges and Solutions
- Sharing Best Practices and Tips

#### 3. Q&A Session and Troubleshooting

- Open Floor for Questions and Clarifications
- Addressing Specific Issues Faced by Participants
- Providing Additional Resources and Guidance

## Week 3: Infrastructure as Code (IaC) with AWS CloudFormation

### Day 1: Introduction to Infrastructure as Code (IaC)

#### 1. What is IaC?

- Definition and Importance of IaC
- Comparison with Traditional Infrastructure Management

- Key IaC Principles and Practices

### 2. Benefits of IaC

- Consistency and Repeatability
- Improved Efficiency and Agility
- Enhanced Collaboration and Version Control
- Reduced Risk of Configuration Drift

### 3. Overview of IaC Tools (CloudFormation, etc.)

- Introduction to AWS CloudFormation
- Comparison with Other IaC

Tools (Terraform, Ansible)

- Use Cases for Different IaC Tools

### 4. Introduction to AWS CloudFormation

- Key Features of CloudFormation
- Basic Concepts (Templates, Stacks)
- CloudFormation vs. Manual Provisioning
- Supported AWS Resources

## Day 2: AWS CloudFormation Basics

### 1. CloudFormation Templates and Stacks

- Understanding CloudFormation Templates
- Structure of a CloudFormation Template
- Creating and Managing CloudFormation Stacks
- Viewing Stack Events and Resources

### 2. Writing Basic CloudFormation Templates

- Defining AWS Resources in YAML/JSON
- Using Parameters, Mappings, and Outputs
- Writing Simple Templates for EC2, S3, VPC
- Validating Templates with CloudFormation Designer

### 3. Deploying Stacks with CloudFormation

- Steps to Deploy a Stack
- Monitoring Stack Creation and Update
- Handling Stack Creation Failures
- Rolling Back Stack Changes

### 4. Managing Stack Updates and Rollbacks

- Updating Existing Stacks
- Using Change Sets for Safe Updates
- Handling Stack Update Failures
- Rolling Back to Previous Versions

## Day 3: Advanced CloudFormation Concepts

### 1. Parameters, Outputs, and Conditions

- Using Parameters for Template Customization
- Defining Outputs for Stack Information
- Implementing Conditions for Conditional Resources
- Best Practices for Parameters and Outputs

### 2. Using Intrinsic Functions

- Overview of CloudFormation Intrinsic Functions
- Common Intrinsic Functions (Ref, GetAtt, Fn::Join)
- Combining Functions for Complex Logic
- Practical Examples of Intrinsic Functions

### 3. Nested Stacks and Stack Sets

- Introduction to Nested Stacks
- Creating and Managing Nested Stacks
- Benefits and Use Cases for Nested Stacks
- Introduction to Stack Sets for Multi-Account Deployment

### 4. CloudFormation Drift Detection

- What is Drift Detection?
- Running Drift Detection on Stacks
- Handling Drift Detection Results
- Best Practices for Preventing Drift

## Day 4: Hands-on Lab: Creating Infrastructure with CloudFormation

### 1. Writing Templates for Common AWS Resources

- Writing Templates for EC2 Instances and Security Groups
- Creating S3 Buckets and Configuring Policies
- Defining VPCs, Subnets, and Route Tables
- Writing Templates for RDS Databases

### 2. Deploying Multi-Tier Applications

- Designing Multi-Tier Architecture
- Writing Templates for Multi-Tier Deployments

- Deploying Front-end and Back-end Resources
- Integrating Components in a Single Stack

### 3. Best Practices for Template Design

- Modularizing Templates for Reusability
- Using Nested Stacks for Complex Deployments
- Writing Readable and Maintainable Templates
- Validating and Testing Templates

### 4. Troubleshooting CloudFormation Issues

- Common CloudFormation Errors
- Debugging Template Syntax and Logic
- Analyzing Stack Events for Failures
- Resolving Resource Dependency Issues

## Day 5: AWS CloudFormation Best Practices and Security

### 1. IaC Best Practices

- Version Control for CloudFormation Templates
- Automating Template Deployment with CI/CD
- Using Parameter Store and Secrets Manager
- Implementing Template Testing and Validation

### 2. Security Considerations for CloudFormation

- Managing IAM Permissions for CloudFormation
- Encrypting Sensitive Data in Templates
- Using AWS Config to Monitor Resource Compliance
- Implementing Security Best Practices

### 3. Integrating CloudFormation with CI/CD Pipelines

- Setting Up CloudFormation in CodePipeline
- Automating Stack Creation and Updates
- Integrating with CodeBuild and CodeDeploy
- Monitoring and Managing Automated Deployments

### 4. Automating Infrastructure Deployment

- Using CloudFormation StackSets for Multi-Region Deployment
- Implementing Self-Healing Infrastructure
- Leveraging AWS Config Rules for Compliance
- Advanced Automation with AWS Step Functions

# Week 4: AWS Core Services for DevOps

# Day 1: AWS Elastic Compute Cloud (EC2)

### 1. Introduction to EC2

- Overview of EC2 Service
- EC2 Use Cases and Benefits
- Understanding EC2 Instance Types

### 2. Launching and Managing EC2 Instances

- Steps to Launch an EC2 Instance
- Configuring Instance Settings
- Connecting to EC2 Instances
- Stopping, Starting, and Terminating Instances

### 3. EC2 Instance Types and Pricing Models

- Overview of EC2 Instance Families
- Understanding On-Demand, Reserved, and Spot Instances
- Selecting the Right Instance Type
- Estimating Costs and Managing Budgets

### 4. Security Groups and Key Pairs

- Creating and Configuring Security Groups
- Understanding Inbound and Outbound Rules
- Generating and Using Key Pairs
- Best Practices for Securing EC2 Instances

# Day 2: AWS Simple Storage Service (S3)

### 1. Introduction to S3

- Overview of S3 Service
- S3 Use Cases and Benefits
- Key Concepts (Buckets, Objects, Keys)

### 2. Creating and Managing S3 Buckets

- Steps to Create an S3 Bucket
- Configuring Bucket Settings and Policies
- Uploading, Downloading, and Deleting Objects
- Managing Bucket Versions and Tags

### 3. S3 Storage Classes and Lifecycle Policies

- Understanding S3 Storage Classes (Standard, IA, Glacier)
- Choosing the Right Storage Class
- Implementing Lifecycle Policies for Data Management

- Transitioning and Expiring Objects

### 4. Securing S3 Buckets and Data

- Implementing Bucket Policies and ACLs
- Configuring IAM Policies for S3 Access
- Using S3 Encryption (SSE-S3, SSE-KMS)
- Monitoring S3 Access with CloudTrail and CloudWatch

## Day 3: AWS Virtual Private Cloud (VPC)

### 1. Introduction to VPC

- Overview of VPC Service
- Key VPC Concepts (Subnets, Route Tables, Gateways)
- Benefits of Using VPC

### 2. Creating and Configuring VPCs

- Steps to Create a VPC
- Defining Subnets and Availability Zones
- Configuring Route Tables and Internet Gateways
- Setting Up NAT Gateways and Bastion Hosts

### 3. Security Groups and Network ACLs

- Creating and Configuring Security Groups
- Understanding Network ACLs
- Implementing Network Security Best Practices
- Monitoring VPC Traffic with Flow Logs

### 4. Advanced VPC Configurations

- Setting Up VPC Peering Connections
- Implementing PrivateLink and Interface Endpoints
- Configuring VPC Endpoints for S3 and DynamoDB
- Best Practices for VPC Design

## Day 4: AWS Relational Database Service (RDS)

### 1. Introduction to RDS

- Overview of RDS Service
- Benefits of Using Managed Databases
- Supported Database Engines (MySQL, PostgreSQL, etc.)

### 2. Creating and Managing RDS Instances

- Steps to Launch an RDS Instance

- Configuring Database Settings
- Connecting to RDS Databases
- Managing Database Backups and Snapshots

### 3. RDS Backup and Restore

- Understanding RDS Backup Mechanisms
- Configuring Automated Backups
- Performing Manual Backups and Restores
- Implementing Point-in-Time Recovery

### 4. Security Best Practices for RDS

- Securing RDS Instances with IAM Roles
- Implementing Encryption at Rest and in Transit
- Configuring Security Groups and Network ACLs
- Monitoring and Auditing RDS Instances

## Day 5: Hands-on Lab: Setting Up a Secure and Scalable Web Application

### 1. Designing a Secure VPC Architecture

- Planning VPC Layout and Subnet Allocation
- Configuring Route Tables and Gateways
- Setting Up Security Groups and Network ACLs
- Ensuring High Availability and Redundancy

### 2. Launching EC2 Instances and RDS within VPC

- Creating EC2 Instances in Private and Public Subnets
- Setting Up RDS Instances with Security Best Practices
- Configuring EC2 to RDS Connectivity
- Implementing Auto Scaling and Load Balancing

### 3. Configuring S3 for Static Content

- Creating S3 Buckets for Static Content
- Setting Up Bucket Policies and Access Controls
- Implementing Static Website Hosting on S3
- Using CloudFront for Content Delivery

### 4. Integrating Components into a Single Application

- Connecting Front-end and Back-end Components
- Ensuring Secure Communication Between Services
- Monitoring Application Performance and Health
- Finalizing and Testing the Application Deployment

# Week 5: Monitoring, Logging, and Security in AWS

## Day 1: AWS CloudWatch

### 1. Introduction to Monitoring and Logging

- Importance of Monitoring and Logging
- Key Concepts and Terminology
- Overview of AWS Monitoring Services

### 2. Setting Up CloudWatch Metrics and Alarms

- Creating and Viewing CloudWatch Metrics
- Configuring Alarms for Resource Monitoring
- Setting Up Notifications and Alerts
- Best Practices for CloudWatch Alarms

# 3. Using CloudWatch Logs and Log Insights

Enabling CloudWatch Logs for AWS Services

- Creating Log Groups and Streams
- Querying Logs with CloudWatch Log Insights
- Analyzing Log Data for Troubleshooting

### 4. Custom Metrics and Dashboards

- Publishing Custom Metrics to CloudWatch
- Creating CloudWatch Dashboards
- Visualizing Metrics and Logs in Dashboards
- Sharing and Managing Dashboards

## Day 2: AWS CloudTrail and Config

### 1. Introduction to AWS CloudTrail

- What is CloudTrail?
- Key Features and Benefits of CloudTrail
- Configuring CloudTrail for AWS Accounts

### 2. Managing CloudTrail Logs and Insights

- Viewing and Querying CloudTrail Logs
- Setting Up CloudTrail Insights
- Analyzing CloudTrail Logs for Security Audits

- Best Practices for CloudTrail Management

### 3. Introduction to AWS Config

- Overview of AWS Config Service
- Key Features and Benefits of AWS Config
- Setting Up AWS Config Rules

### 4. Monitoring Resource Compliance with AWS Config

- Defining Compliance Rules and Policies
- Viewing and Managing Resource Compliance
- Using Config Rules for Security and Governance
- Automating Remediation with AWS Config

## Day 3: AWS Identity and Access Management (IAM) Best Practices

### 1. Advanced IAM Policies and Permissions

- Writing Advanced IAM Policies
- Using Policy Variables and Conditions
- Managing Cross-Account Access
- Implementing Policy Versioning and Rollback

### 2. IAM Roles and Resource-Based Policies

- Creating and Managing IAM Roles
- Using Resource-Based Policies
- Delegating Access Across AWS Accounts
- Best Practices for Role Management

### 3. IAM Policy Simulator and Access Analyzer

- Using IAM Policy Simulator for Testing
- Setting Up IAM Access Analyzer
- Analyzing Access Control Policies
- Identifying and Resolving Policy Issues

### 4. Securing AWS Accounts with IAM Best Practices

- Implementing Multi-Factor Authentication (MFA)
- Using AWS Organizations for Account Management
- Regular IAM Policy Audits
- Enforcing Security Best Practices

## Day 4: Security and Compliance in AWS

### 1. AWS Shared Responsibility Model

- Understanding AWS Shared Responsibility Model
- Customer and AWS Responsibilities
- Implementing Security Best Practices

### 2. Data Encryption and Key Management

- Using AWS Key Management Service (KMS)
- Implementing Encryption at Rest and in Transit
- Managing Encryption Keys and Policies
- Best Practices for Data Encryption

### 3. AWS Security Services Overview

- Introduction to AWS Security Hub
- Using Amazon GuardDuty for Threat Detection
- Implementing Amazon Macie for Data Protection
- Overview of AWS WAF and Shield for Web Security

### 4. Compliance and Governance in AWS

- Understanding Compliance Frameworks (PCI, HIPAA, GDPR)
- Using AWS Artifact for Compliance Documentation
- Implementing AWS Config Rules for Compliance
- Automating Compliance Monitoring and Reporting

## Day 5: Review

### 1. Course Review and Q&A

- Recap of Key Concepts and Topics
- Addressing Remaining Questions and Concerns
- Providing Additional Resources and Guidance
- Discussing Next Steps and Advanced Topics

### 2. Final Assessment and Feedback

- Conducting Final Assessment
- Collecting Feedback from Participants
- Providing Course Completion Certificates
- Discussing Future Learning Paths and Certifications