MODULE 8

AWS
Zero to Hero

A Complete DevOps-Oriented Guide to
Amazon Web Services

Platform
Amazon Web Services

Focus
DevOps Workflow

Module Overview

Complete AWS Training Roadmap

01 **Cloud Computing Fundamentals**

AWS overview, cloud service models, traditional IT vs cloud benefits

02 **AWS Core Services**

Global infrastructure, service categories overview

03 **Security & Identity**

IAM, VPC, access management, network isolation

04 **Compute & Deployment**

EC2, containers, serverless, auto scaling

05 **Storage & Databases**

S3, RDS, DynamoDB, data management solutions

06 **Networking & CDN**

CloudFront, Route53, load balancers

07 **DevOps & Monitoring**

CI/CD pipelines, CloudFormation, CloudWatch

08 **Best Practices & Exam Tips**

Real workflows, interview questions, cheat sheet

CHAPTER 01

Cloud Computing Fundamentals

Understanding the foundation of cloud technology and why AWS leads the market

Cloud Concepts

Service Models

Business Benefits

What is AWS & Cloud Computing?

Amazon Web Services (AWS)

A comprehensive cloud computing platform that provides **on-demand access** to a wide range of IT resources over the internet.

- Compute Servers
- Storage Systems
- Managed Databases
- Networking
- DevOps Tools
- Managed Services

Cloud Computing Model

Instead of buying physical hardware, you rent IT resources on-demand, paying only for what you consume, with instant global accessibility.

Pay-As-You-Go

No upfront costs

Global Access

Anytime, anywhere

Instant Scaling

Elastic resources

Cloud Service Models

IaaS Infrastructure

Rent virtual machines, storage, networking. You manage the OS and applications.

EC2 VPC

PaaS Platform

Focus on application code while AWS manages the underlying infrastructure.

Elastic Beanstalk Lambda

SaaS Software

Use complete applications managed by AWS, like the AWS Console itself.

AWS Console Gmail

Traditional IT Problems vs Cloud Solutions

How cloud computing transforms IT infrastructure challenges

Traditional IT Challenges

High Hardware Costs

Significant upfront capital investment in physical servers, networking equipment, and data center infrastructure. Hardware depreciation and refresh cycles create ongoing expenses.

Maintenance Issues

Continuous need for hardware maintenance, software patching, security updates, and troubleshooting. Requires dedicated IT staff and 24/7 monitoring.

Scaling Difficulty

Manual procurement and deployment processes make it difficult to scale resources quickly. Over-provisioning to handle peak loads results in low average utilization.

Downtime & Availability

Single points of failure, disaster recovery complexity, and limited geographic distribution lead to service interruptions and data loss risks.

Cloud Solutions

Cost Savings

Pay only for what you use with no upfront hardware investment. Convert capital expenditure to operational expenditure, reducing total cost of ownership.

High Availability

Built-in redundancy across multiple Availability Zones and regions. Managed services include automatic failover and disaster recovery capabilities.

Scalability

Auto scaling resources up or down based on demand. Handle traffic spikes instantly without manual intervention or over-provisioning.

Faster Deployment

Deploy applications in minutes instead of months. Global infrastructure enables worldwide access with low latency.

CHAPTER 02

AWS Core Services

Exploring the fundamental building blocks of AWS cloud infrastructure

- Global Infrastructure
- Service Categories
- Regions & AZ

AWS Global Infrastructure & Service Categories

Global Infrastructure

Regions

Physical locations worldwide (e.g., Mumbai, US-East, Frankfurt). Each region contains multiple isolated data centers called Availability Zones.

Availability Zones (AZ)

Isolated data centers within a region, connected by low-latency networks. Each AZ has independent power, cooling, and security.

Edge Locations

Global network of CDN points for caching content closer to users, reducing latency and improving performance.

30+	90+	200+
Regions	AZ	Edge

Major Service Categories

Compute

EC2 | ECS | Lambda | Beanstalk

Storage

S3 | EBS | EFS

Database

RDS | DynamoDB | Aurora | OpenSearch

Networking

VPC | Route53 | ELB | CloudFront

DevOps

CodeCommit | CodePipeline | CodeBuild | CodeDeploy

CHAPTER 03

Security & Identity

Mastering AWS security foundations and access management best practices

IAM VPC Access Control

IAM: Identity & Access Management

The foundation of AWS security – controlling who can access what

What is IAM?

IAM is the **security backbone** of AWS, enabling you to manage access to AWS services and resources securely.

Create Users
Individual accounts for each person

Manage Permissions
Granular control over resource access

Secure Resources
Protect AWS services from unauthorized access

Critical Best Practice

Never use the root account for daily work! Create individual IAM users and follow the principle of least privilege.

IAM Core Components

Users
Individual identities representing a person or application that needs AWS access

Username | Credentials

Groups
Collections of users with shared permissions for easier management

DevTeam | OpsTeam

Roles
Temporary permissions for services (EC2, ECS) to access other AWS resources

EC2Role | LambdaRole

Policies
JSON documents defining precise permissions for actions on resources

AllowS3 | DenyEC2

IAM Workflow: Users assume roles → Groups organize users → Roles grant temporary access → Policies define permissions → Resources are protected

VPC: Virtual Private Cloud

Your isolated private network within AWS

What is VPC?

VPC is a **logically isolated section** of the AWS cloud where you can launch resources in a virtual network that you define.

Private Network

Isolated from other AWS customers

IP Control

Define IP address ranges and subnets

Resource Isolation

Secure resources from unauthorized access

Important Fact

Every EC2 instance runs inside a VPC. Even if you don't create one, AWS provides a default VPC in each region.

VPC Key Components

Subnets

Segments of VPC IP range. Public subnets have internet access, private subnets don't.

Public: 10.0.1.0/24 | Private: 10.0.2.0/24

Route Tables

Rules determining where network traffic is directed within VPC.

Local Route | Internet Route

Internet Gateway

Allows VPC resources to communicate with the internet.

IGW-12345

Security Groups

Virtual firewall controlling inbound/outbound traffic for instances.

Port 22 | Port 80 | Port 443

CHAPTER 04

Compute & Deployment

Virtual servers, containers, and serverless computing for modern applications

 EC2

 Containers

 Serverless

EC2: Elastic Compute Cloud Deep Dive

Virtual servers in the cloud - the foundation of AWS compute

What is EC2?

EC2 provides **resizable compute capacity** in the cloud. Virtual servers you can launch in minutes with complete control over the operating system and applications.

- ✓ Full OS control
- ✓ Flexible instance types
- ✓ Security groups
- ✓ Key pair access

Key Concepts

Instance Types

t2.micro (free tier), m5.large (general purpose), c5.xlarge (compute optimized)

AMI (Amazon Machine Image)

Pre-configured templates: Amazon Linux, Ubuntu, Windows Server

Key Pair

SSH key for secure remote access to Linux instances

Security Groups

Virtual firewall controlling inbound/outbound traffic

Elastic IP

Static public IP address that persists across instance stops

EC2 Deployment Steps

1. **Launch Instance**
   Choose region, select AMI, and begin instance creation process
2. **Choose Instance Type**
   Select based on CPU, memory, storage needs (e.g., t2.micro for free tier)
3. **Configure Network**
   Select VPC, subnets, assign public IP, configure security groups
4. **Add Storage**
   Configure EBS volumes (root and additional), set size and type
5. **Configure Security Group**
   Open ports: 22 (SSH), 80 (HTTP), 443 (HTTPS) as needed
6. **Connect via SSH**
   Use key pair: ssh -i key.pem user@ip

Deployment Workflow

SSH → Install Nginx/Docker → Configure firewall → Deploy application → Assign Elastic IP

Containers & Serverless Computing

Modern application deployment patterns

Container Ecosystem

ECR

Elastic Container Registry: Private Docker image repository for storing container images

Image Storage | Version Control

ECS

Elastic Container Service: Managed container orchestration. Run Docker containers without managing servers.

Fargate (Serverless) | EC2 Launch

EKS

Elastic Kubernetes Service: Managed Kubernetes control plane for advanced container orchestration.

Kubernetes | Advanced

Container Workflow

Build Docker image → Push to ECR → Create ECS task → Deploy container → Service discovery

Serverless: AWS Lambda

What is Lambda?

Run code without provisioning servers. Event-driven execution with automatic scaling.

0	Auto	100ms
Server Mgmt	Scaling	Billing

Use Cases

- ✓ API backends (API Gateway + Lambda)
- ✓ Data processing (S3 events, DynamoDB streams)
- ✓ Automation (CloudWatch events, scheduled tasks)

EC2 vs Lambda Comparison

EC2 — Server-Based

Continuous running | Full OS control | Hourly billing

Lambda — Serverless

Event-driven | No server mgmt | Per execution

Deployment Tools: Elastic Beanstalk & Auto Scaling

Elastic Beanstalk

Platform as a Service (PaaS)

Deploy applications easily without worrying about infrastructure. **AWS manages everything** - capacity provisioning, load balancing, auto scaling, monitoring.

Upload Code | AWS Deploys | Managed Infra

Supported Platforms

- Node.js
- Python
- Java
- PHP
- .NET
- Docker

Deployment Process

1. Create application
2. Upload code package
3. AWS provisions resources
4. Application deployed

Load Balancer & Auto Scaling

ELB

Elastic Load Balancer distributes incoming traffic across multiple EC2 instances to ensure no single server is overwhelmed.

ALB (HTTP) | NLB (TCP) | GWLB

Health checks ensure traffic only routes to healthy instances

Auto Scaling

Automatically increases/decreases the number of EC2 instances based on demand, optimizing cost and performance.

Scaling Triggers

CPU > 70% | Memory > 80% | Request Count

Benefits

Cost Optimization | High Availability | Fault Tolerance

CHAPTER 05

Storage & Databases

Leveraging AWS storage solutions and managed database services

S3 RDS DynamoDB

S3: Simple Storage Service

Scalable object storage for the cloud

What is S3?

S3 provides **object storage** through a web service interface. Store and retrieve any amount of data from anywhere on the web.

- ✓ 99.999999999% durability
- ✓ Unlimited scalability
- ✓ Pay-as-you-go
- ✓ Static website hosting

Core Concepts

Buckets

Containers for objects. Globally unique name. Regional resource.

Objects

Files + metadata. Key (name), Value (data), Version ID.

Versioning

Keep multiple versions. Protect against deletions.

Storage Classes & Features

Storage Classes

S3 Standard	Frequent access
S3 IA	Infrequent access
S3 Glacier	Archive
S3 Deep Archive	Long-term backup

Lifecycle Policies

Automate transitions between storage classes based on age.

Day 0: Standard | 30d: IA | 90d: Glacier

Static Website Hosting

Host static websites directly from S3 with HTTP endpoints.

- ✓ Index & error documents
- ✓ Custom domain support
- ✓ CloudFront integration

AWS Database Services

Managed databases for every application need

RDS

Relational Database Service: Managed relational databases with automated backups, patching, and scaling.

Supported Engines

MySQL | PostgreSQL | Oracle | SQL Server | MariaDB

Features

- ✓ Multi-AZ deployment
- ✓ Read replicas
- ✓ Automated backups
- ✓ Point-in-time recovery

Use For

Traditional applications, complex queries, ACID transactions

DynamoDB

Managed NoSQL Database: Serverless, fast, and flexible database for applications requiring consistent performance.

Key Features

- ✓ Serverless (no infrastructure)
- ✓ Single-digit millisecond latency
- ✓ Automatic scaling
- ✓ Global tables (multi-region)

Data Model

Hash Key | Range Key

Key-value and document data model with JSON support

Use For

Web/mobile apps, gaming, IoT, real-time analytics

OpenSearch

Search & Analytics Engine: Fork of Elasticsearch for real-time search, analytics, and visualization.

Capabilities

- ✓ Full-text search
- ✓ Log analytics
- ✓ Real-time dashboards
- ✓ Security analytics

Integrations

Kinesis | CloudWatch | S3

Use For

Application search, log analytics, observability

Database Selection Guide

RDS
Complex queries, joins, transactions

DynamoDB
High throughput, simple queries, serverless

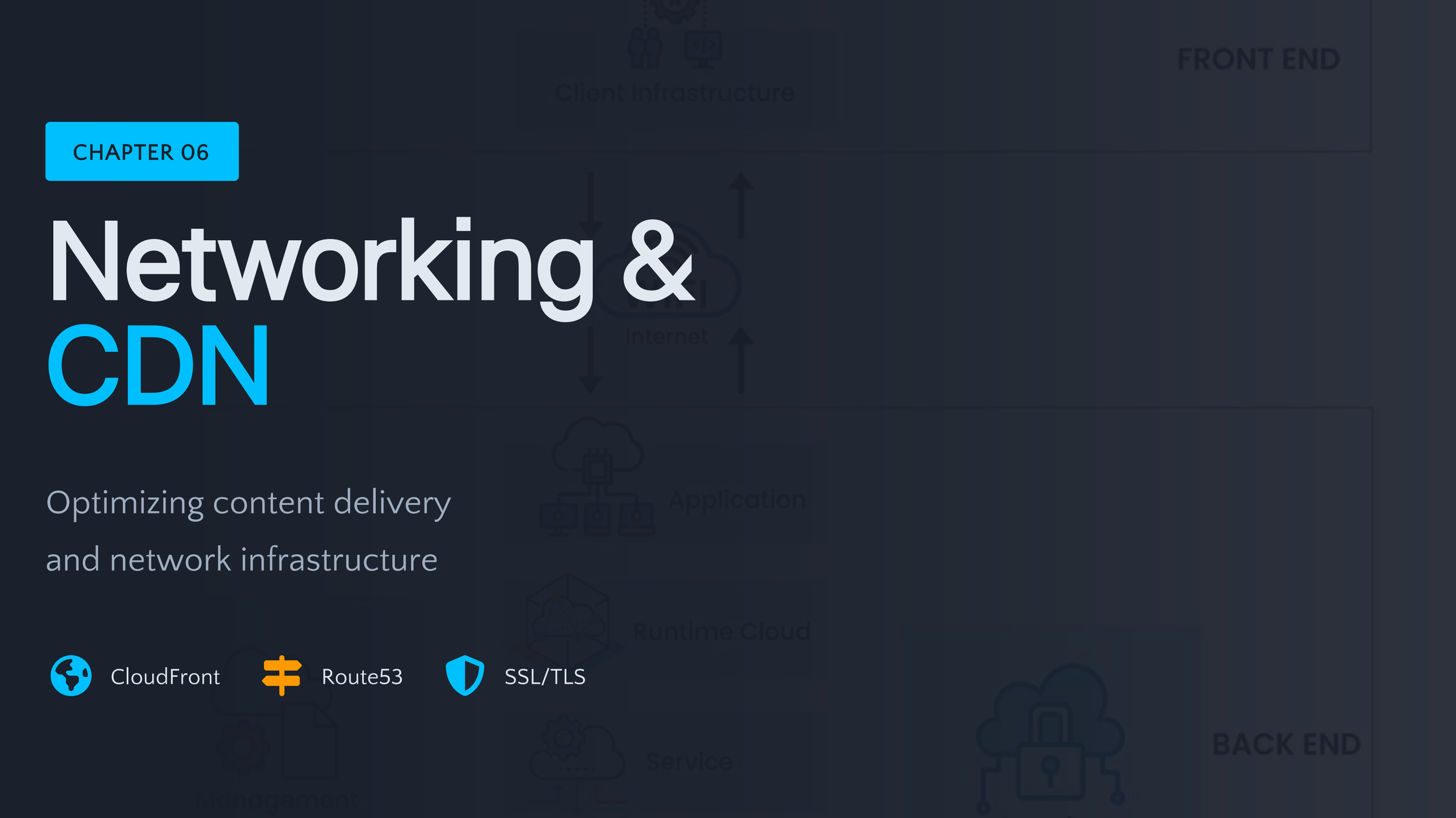
OpenSearch
Full-text search, log analytics

CHAPTER 06

Networking & CDN

Optimizing content delivery and network infrastructure

CloudFront Route53 SSL/TLS



CloudFront CDN & Advanced Networking

CloudFront CDN

Content Delivery Network that caches content at edge locations worldwide for faster delivery to users.

How It Works

Content cached at 200+ edge locations. Users get content from nearest edge.

Origins

S3 | EC2 | ELB | Custom

Benefits

- ✓ Faster content delivery
- ✓ Reduced latency
- ✓ Lower origin load
- ✓ DDoS protection

Route53

Scalable DNS and domain registration. Routes end users to Internet applications.

Domain Registration | DNS Routing | Health Checks

Advanced Networking Concepts

SSL/TLS

Encrypt data in transit. SSL/TLS certificates for HTTPS.

Certificate Manager | Free SSL

SSH

Secure Shell for remote server access. Uses key pairs.

Port 22 | Key Pair

Nginx Reverse Proxy

Sits between clients and backend servers. Load balancing, SSL termination.

Load Balance | SSL Terminate

IPv4 / IPv6

IP addressing protocols. IPv6 provides larger address space.

IPv4: 32-bit | IPv6: 128-bit

VPC Routing

Route tables control traffic flow. Security groups act as virtual firewalls.

Route Tables | Security Groups | NACL

CHAPTER 07

DevOps & Monitoring

Implementing CI/CD pipelines
and infrastructure automation

CodePipeline CloudFormation CloudWatch

CI/CD Services & CloudFormation

AWS Code Services

CodeCommit

Git repository service. Secure, scalable, managed Git repositories in the cloud.

Git Compatible | Encryption

CodePipeline

CI/CD orchestration. Automate build, test, and deploy phases.

Visual Workflow | Integration

CodeBuild

Build & test service. Compile source code, run tests, produce artifacts.

Pay per build | Docker builds

CodeDeploy

Automated deployment. Deploy to EC2, Lambda, ECS with minimal downtime.

Blue/Green | Rolling

CI/CD Workflow

CodeCommit → CodePipeline triggers → CodeBuild → Test → CodeDeploy → Production

CloudFormation (IaC)

Infrastructure as Code

Define AWS resources in **YAML/JSON templates.** Create, update, and delete stacks predictably.

YAML/JSON | Version Control

Benefits

Automation
No manual resource creation

Reproducibility
Same infra across environments

Version Control
Track infrastructure changes

Sample Template Structure

```
Resources:
  MyEC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-123
      InstanceType: t2.micro
```

Monitoring, Logging & Messaging

CloudWatch

Metrics

Collect and track metrics from AWS resources and applications.

CPU | Memory | Disk | Network

Logs

Centralized log management. Aggregate, monitor, and analyze log files.

Log Groups | Log Streams

Alarms

Send notifications or take actions when metrics breach thresholds.

SNS | Auto Scaling | Lambda

CloudTrail

API activity tracking. Log and monitor AWS API calls for auditing and compliance.

- ✓ Who made the call
- ✓ What action was taken
- ✓ When it occurred

Messaging & Analytics

SNS

Pub/Sub messaging. Send notifications to multiple subscribers.

Email | SMS | Lambda

SQS

Queue-based messaging. Decouple and scale microservices.

Standard | FIFO

SES

Email service. Send marketing, transactional, and notification emails.

SMTP | API

Analytics & Data Pipeline

Athena	SQL queries on S3
Glue	ETL service
Redshift	Data warehouse
Kinesis	Stream processing

CHAPTER 08

DevOps Workflow & Best Practices

Real-world deployment patterns and exam preparation strategies

Real Workflow Exam Tips Best Practices

Real AWS DevOps Deployment Workflow

Complete CI/CD pipeline from code to production

Production-Ready CI/CD Pipeline

1. ### Developer Pushes Code
   Code changes pushed to GitHub or CodeCommit repository trigger the pipeline automatically
2. ### CodePipeline Orchestrates
   Pipeline triggers build process, runs tests, and coordinates deployment stages
3. ### CodeBuild Creates Docker Image
   Builds application, runs tests, creates optimized Docker image, and pushes to ECR
4. ### ECS Deploys Container
   Pulls image from ECR, deploys container to Fargate or EC2 cluster with auto scaling
5. ### Load Balancer Exposes App
   Application Load Balancer routes traffic to containers. Health checks ensure availability
6. ### CloudWatch Monitors & Logs
   Collects metrics, aggregates logs, triggers alarms for anomalies. Enables debugging and optimization

Flask + MongoDB Mini Project

Deploy a Python Flask application with MongoDB backend using Docker and AWS services.

Tech Stack

Flask | MongoDB | Docker | EC2/ECS

Deployment Steps

- ✓ Build Docker image locally
- ✓ Push to ECR repository
- ✓ Create ECS task definition

Key Integration Points

- **Source Control**
  GitHub/CodeCommit
- **Build & Test**
  CodeBuild
- **Container Registry**
  ECR
- **Container Runtime**
  ECS/Fargate

AWS Cheat Sheet & Exam Tips

AWS Service Cheat Sheet

Service	Purpose
EC2	Virtual server in the cloud
S3	Object storage service
IAM	Identity & access management
VPC	Private network in AWS
ECR	Docker image repository
ECS	Run containers (managed)
Lambda	Serverless code execution
RDS	Managed relational database
CloudWatch	Monitoring & observability
CloudFormation	Infrastructure as Code

Exam Focus Points

🔑 IAM Policies

Understand JSON structure, least privilege principle, and the difference between users, groups, roles, and policies.

🛡️ EC2 Security Groups

Know how security groups work as virtual firewalls, stateful vs stateless rules, and common port configurations.

🖧 VPC Networking

Understand subnets, route tables, internet gateways, and how resources communicate within VPC.

🔄 CI/CD Flow

Know the order: CodeCommit → CodeBuild → ECR → ECS/EC2 deployment with load balancer.

🔄 Docker + AWS Integration

ECR stores images, ECS runs containers, Fargate is serverless option vs EC2 launch type.

ZERO → HERO COMPLETE

AWS Mastery Achieved

You now have comprehensive AWS knowledge covering cloud fundamentals, security, compute, storage, networking, DevOps, and real-world deployment workflows.

29
Topics Covered
From basics to advanced

90+
AWS Services
Explained with examples

100%
DevOps Ready
Production workflows

Remember for Exams & Interviews

- **Always explain WHY cloud**
  Cost, scalability, availability benefits
- **Relate AWS with DevOps tools**
  CodePipeline, Docker, ECS
- **Emphasize security & scalability**
  IAM, VPC, auto scaling
- **Use real service names**
  Be specific, not generic