

# Exam Guide

AWS Certified Cloud  
Practitioner



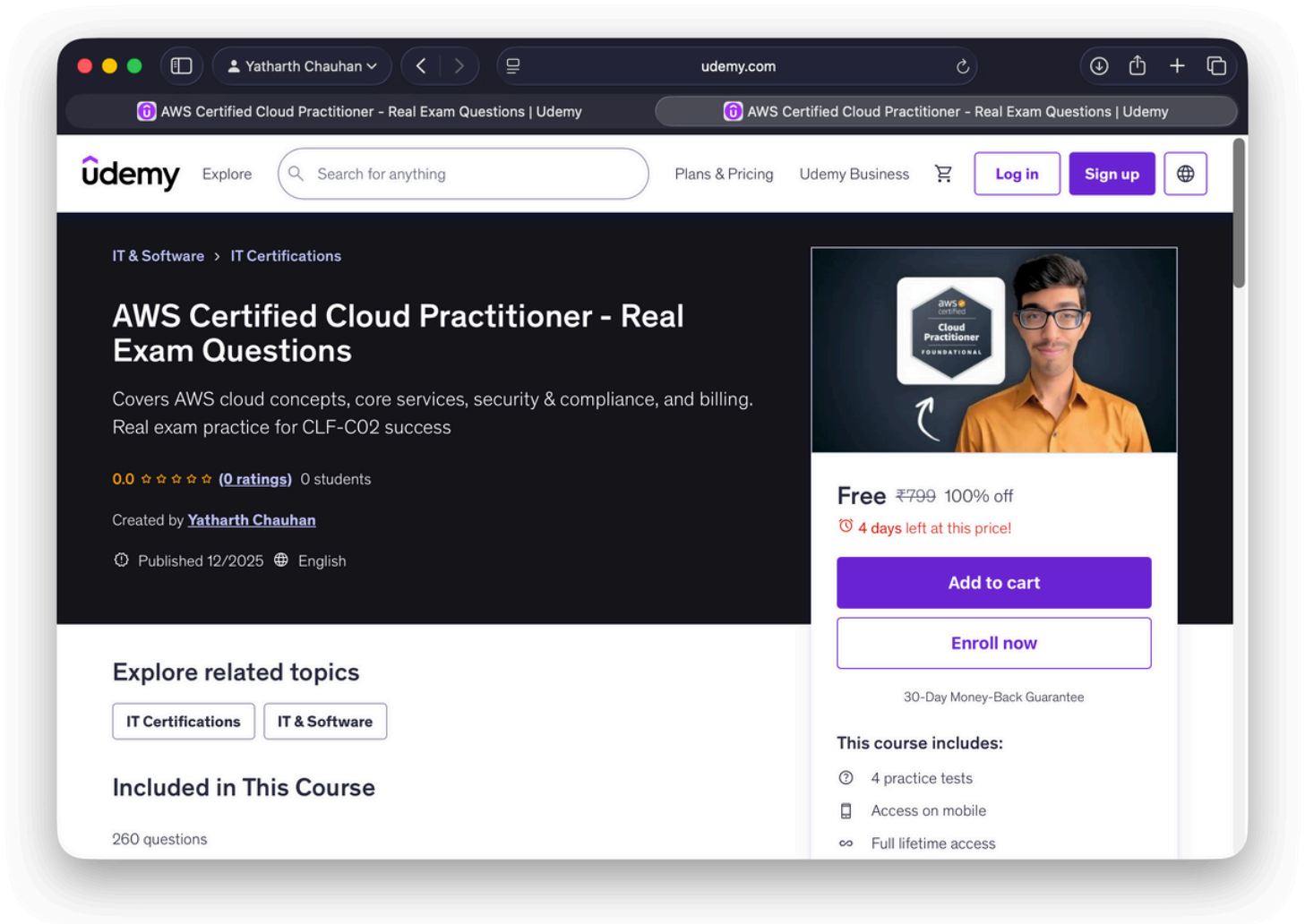
250+ FREE Exam Questions



**Yatharth Chauhan**

Follow for more content

# 250+ Real Exam Questions (100% OFF)



USE Coupon Code:  
**AWSYATRI046590702**

→ 4 days left to get **FREE**  
(Link is given in bio)

# AWS Cloud Practitioner Exam Guide - Yatharth Chauhan

## Part 1: Introduction and Exam Overview

### What is AWS Certified Cloud Practitioner?

The AWS Certified Cloud Practitioner (CLF-C02) certification is AWS's foundational level certification. This means it's the EASIEST and BEST STARTING POINT if you want to learn about AWS.

This certification proves that you:

- Understand what AWS Cloud is and why companies use it
- Know how to stay secure and follow rules when using AWS
- Can recognize and use the most common AWS services
- Understand how AWS pricing works and how to control costs
- Can make basic decisions about which AWS services to use

Think of this certification as proof that you have foundational knowledge of AWS, like having a high school diploma before going to college. It doesn't make you an expert, but it shows you understand the basics.

### Who Should Take This Exam?

- Complete beginners to AWS (even from non-IT backgrounds)
- Business professionals who work with AWS teams
- Students interested in cloud computing
- Anyone with 0-6 months of AWS exposure
- Career changers entering cloud computing field
- People who want to prove basic cloud knowledge

This is NOT for:

- Advanced AWS architects
- People with several years of AWS experience
- Those looking for advanced certifications (yet)

## What This Certification Does NOT Test

The exam specifically does NOT test your ability to:

- Write code or do programming
- Design complex cloud architectures
- Troubleshoot AWS problems
- Implement AWS solutions
- Perform load testing or performance testing
- Build and deploy applications
- Configure advanced networking

Focus only on UNDERSTANDING and RECOGNIZING, not on BUILDING or FIXING.

---

## Part 2: Exam Format and Scoring

### Question Types

The exam has TWO types of questions:

Type 1: Multiple Choice

- You see a question or statement
- You get FOUR possible answers
- ONLY ONE is correct
- You pick the one correct answer
- Example: "What is the primary benefit of using Amazon EC2?"
- A. Eliminates need for internet connection
- B. Provides scalable computing capacity (CORRECT)
- C. Removes all security concerns
- D. Automatically backs up all data

Type 2: Multiple Response

- You see a question or statement
- You get FIVE or MORE possible answers
- TWO or MORE are correct (you need to pick all correct ones)
- If you pick all correct AND no incorrect, you get full points
- If you miss even one or pick an incorrect one, you get zero points
- Example: "Which of the following are benefits of AWS?" (pick two)
- A. Pay only for what you use (CORRECT)

- B. Requires building your own data centers (INCORRECT)
- C. Provides global infrastructure (CORRECT)
- D. Eliminates internet usage (INCORRECT)

## Question Count

- Total questions shown: 65 (90 minutes)
- Scored questions: 50 (these affect your grade)
- Unscored questions: 15 (used by AWS to test new questions)
- **Important:** You don't know which questions are unscored, so answer all carefully

## Scoring System

- Minimum passing score: 700 out of 1000
- Scaled scoring means AWS adjusts difficulty levels
- No penalty for guessing (unanswered questions count as wrong)
- You need about 70% correct to pass
- Results show: Pass or Fail + scaled score + performance by domain

## Time Management

- Total time: 90 minutes
- 65 questions to answer
- Average time per question: About 1.5 minutes
- Don't spend more than 2-3 minutes on any one question
- Flag difficult questions and come back if time permits

---

## Part 3: Domain 1 - Cloud Concepts (24% of Exam)

Domain 1 covers the WHY of AWS - why companies use it and why it matters.

### Task 1.1: Define the Benefits of the AWS Cloud

## What Makes AWS Different from Traditional Data Centers?

Traditional Data Center (On-Premises):

- You buy servers and equipment upfront (big expense)
- You pay for the building, electricity, cooling, staff
- You pay whether servers are used or sitting idle
- You're stuck with whatever capacity you bought
- You manage and maintain everything
- Scaling up takes weeks or months
- Difficult to serve global customers

AWS Cloud:

- You rent servers by the hour (pay-as-you-go)
- AWS pays for building, electricity, cooling, staff
- You pay ONLY for what you actually use
- Scale up or down in seconds
- AWS maintains everything
- Already available worldwide
- Easy to serve customers globally

## Economics of Scale

AWS buys hardware in massive quantities. Think of it like:

- A farmer buying seeds one at a time = expensive
- A farmer buying 1 million seeds in bulk = much cheaper per seed

AWS passes these savings to customers through lower prices.

## Global Infrastructure Benefits

AWS has data centers (called Regions) all around the world:

- Americas: North America (US, Canada), South America (Brazil)
- Europe: Ireland, Germany, France, UK, Sweden, etc.
- Asia Pacific: Singapore, Tokyo, Sydney, Mumbai, etc.
- Middle East: Bahrain

Benefits:

- Deploy in seconds anywhere globally
- Serve customers from location near them (faster speed)

- Meet data residency laws (keep data in specific countries)
- Disaster recovery (if one region fails, use another)

## High Availability

High Availability means your application stays running even when problems happen.

Example:

- Traditional: One server fails = entire website down
- AWS: Servers in different locations = if one fails, others keep running

## Elasticity

Elasticity means the ability to grow and shrink automatically.

Real-world example:

- Retail website gets 100 visitors per day normally
- Black Friday sale: Gets 10,000 visitors per day
- Without elasticity: Website crashes (too many visitors)
- With AWS elasticity: Automatically adds more servers for the day, then removes them after

You only pay for servers you actually use during peak times.

## Agility

Agility means ability to move and adapt quickly.

Traditional:

- Want to try new technology = order servers, takes months
- Mistake in planning = stuck with wrong equipment

AWS:

- Want to try new technology = spin up in hours
- Made a mistake = delete it immediately, minimal cost

## Task 1.2: Identify Design Principles of the AWS Cloud

AWS created the Well-Architected Framework to help you design good solutions. It has six pillars (think of pillars holding up a building).

## **Pillar 1: Operational Excellence**

This pillar is about RUNNING systems well.

Key ideas:

- Monitor and improve systems constantly
- Automate manual processes
- Document everything (so anyone can understand it)
- Prepare for failure and learn from incidents
- Use Infrastructure as Code (write code to create infrastructure)

Example: Instead of manually setting up servers, write a script that AWS can run to set them up identically every time.

## **Pillar 2: Security**

This pillar is about PROTECTING systems and data.

Key ideas:

- Apply security at every layer (don't just protect the outside)
- Enable traceability (log who did what and when)
- Implement least privilege (give minimum needed access)
- Automate security best practices
- Protect data in transit (while moving) and at rest (stored)
- Prepare for security incidents

## **Pillar 3: Reliability**

This pillar is about STAYING UP and WORKING PROPERLY.

Key ideas:

- Automatically recover from failures
- Test failure scenarios (disaster recovery drills)
- Scale horizontally (add more servers) not vertically (bigger servers)
- Stop guessing capacity (use auto scaling)
- Manage change through automation

Example: Have multiple servers so if one fails, others keep working.

## **Pillar 4: Performance Efficiency**

This pillar is about GETTING RESULTS QUICKLY.



Key ideas:

- Use appropriate technologies
- Monitor performance
- Make trade-offs (fast vs. cost vs. consistency)
- Use global services to reduce latency (delay)

Example: Use caching (storing frequently requested data) so you don't recalculate same result repeatedly.

## **Pillar 5: Cost Optimization**

This pillar is about SPENDING MONEY WISELY.

Key ideas:

- Use right-sized resources (not over-provisioning)
- Measure and monitor usage
- Stop spending on data center operations (AWS does it)
- Analyze and attribute expenditure (understand where money goes)
- Use managed services (let AWS manage for you)

Example: Stop paying for servers you're not using at night.

## **Pillar 6: Sustainability**

This pillar is about ENVIRONMENTAL RESPONSIBILITY.

Key ideas:

- Understand environmental impact
- Maximize utilization (use resources efficiently)
- Anticipate and plan for future needs
- Use managed services (shared infrastructure is more efficient)
- Reduce downstream and upstream impact

Example: AWS data centers use renewable energy, so your applications are greener than on-premises.

## **Task 1.3: Understand Benefits of and Strategies for Migration to AWS Cloud**

### **Why Migrate to AWS?**

Business benefits:

- Reduce capital expenses (no servers to buy)
- Increase operational agility (changes faster)
- Improve disaster recovery capability
- Enable innovation (focus on business, not infrastructure)
- Reduce environmental impact

Technical benefits:

- Access global infrastructure
- More security services available
- More database options
- Better scalability

## **AWS Cloud Adoption Framework (AWS CAF)**

AWS CAF is a GUIDE to help organizations migrate to AWS successfully. It divides responsibility into four perspectives:

Perspective 1: Business Perspective

- Role: Business leaders, managers
- Focus: Business outcomes and benefits
- Questions: How will AWS help our bottom line?

Perspective 2: People Perspective

- Role: HR, training, staff
- Focus: Organizational change and skills development
- Questions: Do our people have skills to use AWS?

Perspective 3: Governance Perspective

- Role: CIO, Risk, Compliance
- Focus: Policies, rules, and risk management
- Questions: Are we compliant and secure?

Perspective 4: Platform Perspective

- Role: IT leaders, architects
- Focus: Technical architecture and tooling
- Questions: What services do we need? How do we architect?

## **Migration Strategies**

There are six main strategies for moving to AWS (sometimes called the "Six Rs"):

### Strategy 1: Rehost (Lift and Shift)

- Move application to AWS exactly as is
- Don't change anything
- Fastest option
- Cost savings just from not running data center
- Example: Take your Windows server and run it on AWS EC2

### Strategy 2: Replatform (Lift, Tinker, and Shift)

- Move application to AWS with some changes
- Optimize for cloud but not redesign
- Get more benefits than pure rehost
- Still relatively fast
- Example: Migrate database to managed Amazon RDS instead of self-managed

### Strategy 3: Refactor/Re-architect

- Rebuild application using cloud-native services
- Takes more time and effort
- Gets most cloud benefits
- Example: Break monolithic application into microservices using Lambda

### Strategy 4: Repurchase

- Stop using current software
- Buy new cloud-based software (SaaS)
- Example: Stop using own email server, use Office 365 in cloud

### Strategy 5: Retire

- Turn off applications that aren't needed
- Stop paying for unused systems
- Example: Discover old system nobody uses, shut it down

### Strategy 6: Retain (Do Nothing)

- Keep some applications on-premises
- Valid choice for legacy systems not worth migrating
- Example: Keep 30-year-old system if migration cost exceeds benefit

## **AWS Database Migration Service (DMS)**

One of the hardest parts of migrating is moving data.

AWS DMS helps:

- Migrate databases from on-premises to AWS
- Works with different database types
- Minimize downtime during migration
- Can continue using old system while migrating

Example: Move your Oracle database to Amazon RDS without stopping your business.

## **AWS Snow Family**

For companies with HUGE amounts of data, internet transfer is too slow.

AWS Snow Family ships you hardware:

- AWS Snowball: A briefcase-sized device (50-80 TB)
- AWS Snowmobile: A truck-sized device (100 PB - petabytes!)

You:

1. Connect device to your network
2. Copy data onto it
3. Ship it to AWS
4. AWS uploads data to cloud
5. You get device back

Faster than uploading over internet for large amounts of data.

## **Task 1.4: Understand Concepts of Cloud Economics**

This task is about understanding how MONEY and COSTS work in AWS.

### **Fixed Costs vs. Variable Costs**

Fixed Costs (Traditional):

- Buy server for \$5,000 (one-time cost)
- Buy building space for \$2,000/month (whether you use it or not)
- Hire staff for \$4,000/month (whether busy or slow)
- These costs stay same regardless of business activity

Variable Costs (AWS Model):

- Pay \$0.10 per hour for EC2 server (only when running)
- Pay \$0.023 per GB for storage (only for storage you use)

- If you don't use it, you don't pay it
- Cost scales with business activity

AWS Advantage: You don't pay for unused capacity.

## **Cost Comparison: On-Premises vs. AWS**

On-Premises Costs:

- Servers: \$50,000 upfront
- Network equipment: \$10,000 upfront
- Building space: \$2,000/month
- Electricity: \$800/month
- Staff salaries: \$15,000/month
- Maintenance: \$1,000/month
- Total: \$18,800/month + initial \$60,000

AWS Costs (Example):

- Servers needed: \$1,000/month
- Storage: \$100/month
- Network: \$50/month
- No staff cost (AWS manages)
- No electricity cost (AWS pays)
- Total: \$1,150/month (no upfront)

After 6 months: On-prem = \$172,800; AWS = \$6,900

## **Rightsizing**

Rightsizing means using the RIGHT SIZE of resources, not too big or too small.

Problem: Over-provisioning

- Buy server that can handle 10,000 users
- Only 100 users use it
- Wasted money on unused capacity
- Common with traditional data centers

AWS Solution:

- Start with smaller server
- Monitor actual usage
- Scale up if needed

- Scale down if excess
- Only pay for what you need

## **Managed Services**

Managed Services are AWS services where AWS handles the complexity.

Self-Managed (You do the work):

- Setup and maintain EC2 servers
- Install database software
- Apply security patches
- Handle backups
- Requires staff

Managed Service Example (AWS does it):

- Amazon RDS (database)
- AWS handles installation, patches, backups, scaling
- You just use it
- Less staff needed
- Lower total cost

## **Licensing Models**

Different software has different licensing models:

Model 1: Traditional Purchase

- Buy license for \$500
- Pay annual maintenance \$50
- License locked to hardware
- If hardware breaks, software can't transfer

Model 2: BYOL (Bring Your Own License)

- You own your software license already
- Bring it to AWS
- Keep using same license
- AWS allows this for many products
- Can save money if you have existing licenses

Model 3: Included License (AWS)

- License included in AWS service price
- Amazon RDS includes database license

- No separate license fee
- Simpler pricing

## Automation Benefits

Automation means using software to do manual tasks.

Manual Process (Cost):

- Admin spends 2 hours manually setting up each server
- At \$50/hour = \$100 per server
- Setting up 100 servers = \$10,000 labor cost
- Takes 200 hours (spread over weeks)
- Mistakes happen

Automated Process:

- Write script once (2 hours)
- Run script to create 100 servers in 20 minutes
- Labor cost: \$100 (for writing script)
- Zero mistakes
- Massive time and cost savings

AWS CloudFormation enables this automation.

---

## Part 4: Domain 2 - Security and Compliance (30% of Exam)

Domain 2 covers how to keep AWS systems SECURE and COMPLIANT with rules.

### Task 2.1: Understand the AWS Shared Responsibility Model

This is CRITICAL - understanding shared responsibility is essential for the exam and for using AWS securely.

### The Basic Concept

AWS and YOU both have security responsibilities. It's not AWS's responsibility to secure everything, and it's not your responsibility alone either.

Think of it like renting an apartment:

- Landlord (AWS) must: Fix roof, maintain walls, ensure building safety
- Tenant (You) must: Lock your door, not leave stove on, keep apartment clean

If thieves rob you because you left door unlocked, that's YOUR fault, not landlord's.

## **What AWS is Responsible For**

AWS handles security OF the cloud:

### **1. Physical Infrastructure**

- Data center buildings
- Locked doors and guards
- Earthquake-proof construction
- Fire suppression systems

### **2. Network Infrastructure**

- Cables and routers
- DDoS protection at AWS level
- Internet connectivity

### **3. Virtualization Infrastructure**

- Virtual machine hypervisors (software controlling VMs)
- Ensuring one customer's VM can't access another's

### **4. Foundation Services**

- EC2, RDS, S3 service itself
- API security
- Service availability

AWS must ensure:

- Hackers can't physically access data centers
- One customer can't access another customer's data
- Services are available and working
- AWS infrastructure is secure

## **What YOU are Responsible For**



You handle security IN the cloud (your data and applications):

1. Customer Data

- Protecting your actual data content
- Encrypting your sensitive data
- Managing who has access to your data

2. Applications

- Secure coding
- Application patching
- Application configuration

3. Operating Systems

- OS patches for EC2 (not needed for managed services)
- OS configuration
- OS security

4. Network and Firewall

- Choosing which ports to open
- Setting up security groups
- VPC configuration

5. Identity and Access

- Creating users
- Setting permissions
- Managing passwords
- Multi-factor authentication

6. Encryption Keys (sometimes)

- Managing encryption keys
- Key rotation
- Access control to keys

Example: You store password "secret123" in database

- AWS ensures database service runs securely
- AWS ensures only YOUR account can access database
- YOU must ensure password is encrypted and strong
- YOU must ensure people don't have unnecessary access

## How Responsibility Changes by Service

The shared responsibility model changes depending on the SERVICE. The more managed the service, the more AWS does.

Service Type 1: Unmanaged (You do most)

EC2 (Virtual Servers)

- AWS: Physical hardware, virtualization, network
- You: Operating system patches, application software, data, security groups, firewalls

Service Type 2: Partially Managed (AWS does some)

Amazon RDS (Managed Database)

- AWS: Physical hardware, virtualization, database software, patches, backups, high availability
- You: Encryption keys, database accounts and permissions, applications connecting to it

Service Type 3: Fully Managed (AWS does almost everything)

Amazon S3 (Object Storage)

- AWS: Physical hardware, virtualization, service, availability, patches
- You: Encryption, who has access, data content, encryption keys if you use them

Service Type 4: Complete SaaS (AWS does everything except your data)

Amazon SageMaker (Machine Learning)

- AWS: Everything except your data and algorithms
- You: Your input data, your machine learning models, your results

## Principle of Least Privilege

This is a security principle you will see on the exam multiple times.

Principle: Give users and applications **ONLY** the minimum permissions they need to do their job.

Example:

- Data analyst needs to read data from S3
- Bad: Give admin access to entire AWS account
- Good: Give permission to read only from specific S3 bucket

Why it matters:

- If account gets hacked, damage is limited
- Fewer mistakes (can't accidentally delete important things)
- Better security

## **Task 2.2: Understand AWS Cloud Security, Governance, and Compliance Concepts**

### **Encryption**

Encryption means scrambling data so only authorized people can read it.

Encryption in Transit

- Data moving from point A to point B
- Example: User sends data from laptop to AWS
- Use HTTPS (secure web), TLS, IPsec
- Protects against hackers intercepting data

Encryption at Rest

- Data stored somewhere
- Example: Data sitting in S3 storage
- Use AES-256, RSA, or other encryption
- Protects against someone stealing the hard drive

Why Both Matter:

- Encryption in transit: Protects during upload/download
- Encryption at rest: Protects when stored
- Need both for complete protection

### **Compliance and Governance**

Compliance means following rules and regulations.

Different industries have different requirements:

Healthcare:

- HIPAA: Protects patient medical information
- Must encrypt patient data
- Must log who accesses data
- Must track changes

Finance:

- PCI DSS: Protects credit card data
- SOX: Financial transparency

Government:

- FedRAMP: For US government contracts
- FISMA: For US government systems

Legal:

- GDPR: European data protection (right to be forgotten, data portability)
- CCPA: California privacy law

AWS Compliance Support:

AWS maintains certifications showing it meets standards:

- ISO 27001 (information security)
- SOC 2 (security and availability)
- PCI DSS (payment card security)
- HIPAA (healthcare)
- GDPR (Europe)
- Many others

This means if you use AWS properly, you can more easily comply with these standards.

## **Logging and Monitoring**

Logging: Recording what happens

Monitoring: Watching for problems

AWS CloudTrail:

- Logs all API calls (who did what, when, from where)
- Essential for compliance
- Essential for auditing

Amazon CloudWatch:

- Monitors application and system performance
- Collects metrics (CPU, memory, disk usage)
- Can trigger alerts

AWS Config:

- Tracks configuration changes
- "What changed and when did it change?"
- Good for compliance and troubleshooting

Audit Manager:

- Helps prepare for audits
- Collects evidence of compliance
- Simplifies compliance reporting

Why Logging Matters:

- Detect suspicious activity
- Investigate after incident
- Prove compliance to auditors
- Troubleshoot problems

## **Task 2.3: Identify AWS Access Management Capabilities**

### **AWS Identity and Access Management (IAM)**

IAM is how you manage who can do what in AWS.

IAM Components:

#### **1. Users**

- Individual people or applications
- Each has name and credentials
- Example: "john\_employee@company"

#### **2. Groups**

- Collection of users
- Easier to manage permissions for groups
- Example: "Finance\_Team" group contains all finance employees

### 3. Roles

- Set of permissions
- Can be assumed by users or services
- Temporary permissions
- Example: "Lambda\_Execution\_Role" allows Lambda to access databases

### 4. Policies

- Rules defining what actions are allowed
- Written in JSON (code-like format)
- Example: Policy saying "Allow read S3 bucket X" or "Allow modify EC2"

Example IAM Setup:

- Create user: "alice\_manager"
- Create role: "Manager\_Access\_Role"
- Add policy to role: "Allow manage EC2, view S3"
- Give role to alice\_manager
- Now alice can manage EC2 and view S3

## **AWS Root User Account**

Important: AWS root user is special.

What is root user?

- The FIRST account when you create AWS account
- Has complete access to everything
- Identified by email address (not username)

Why it's dangerous:

- Has too much power
- If hacked, entire account is compromised
- Hard to track who used it

Root User Protection:

Do NOT:

- Use root account for daily work

- Leave access keys unprotected
- Use for routine administration

DO:

- Create separate admin user with IAM
- Enable Multi-Factor Authentication (MFA) on root
- Store root credentials in safe place
- Use root account ONLY for essential tasks like:
  - Creating initial admin user
  - Changing email address
  - Modifying billing settings
  - Closing AWS account

## **Multi-Factor Authentication (MFA)**

MFA is security using MULTIPLE verification methods.

Example:

- Something you know: Password
- Something you have: Phone

With MFA:

1. Enter username and password
2. System sends code to your phone
3. You enter code from phone
4. Access granted only after correct code

Why it matters:

- Even if password is stolen, hacker can't get in without phone
- Dramatically improves security

MFA Types:

- Virtual authenticator app (Google Authenticator, Authy)
- Hardware security key
- SMS text message (least secure)

## **IAM Identity Center (AWS Single Sign-On)**

IAM Identity Center makes managing users easier, especially for organizations.

Without IAM Identity Center:

- Each application has separate login
- Each system has separate passwords
- Users forget which password goes where
- Chaos

With IAM Identity Center:

- Single login for all AWS applications
- One password to remember
- Can integrate with company directory (Active Directory)
- Easy onboarding and offboarding

## **Access Keys**

Access Keys are credentials for programmatic access (code accessing AWS).

What is access key?

- Access Key ID (like username)
- Secret Access Key (like password)

When to use:

- Developers accessing AWS from code
- Applications running on servers
- Automated scripts

Security rules:

- Don't share access keys (like passwords)
- Rotate regularly (create new, delete old)
- Don't put in code (use credential manager)
- Use least privilege

## **Credential Storage**

Never hardcode credentials in source code!

Better ways:

### **1. AWS Secrets Manager**

- Store secrets securely in AWS
- Access from applications



- Automatically rotate credentials
  - Track who accessed secrets
2. AWS Systems Manager Parameter Store
    - Store configuration values
    - Reference from code
    - Can encrypt sensitive values
  3. Environment Variables
    - Set in deployment environment
    - Code reads from environment, not hardcoded

## **Authentication Methods**

Methods AWS supports:

1. Usernames and Passwords
  - Traditional login
  - Use strong passwords
2. Multi-Factor Authentication (MFA)
  - Second factor verification
  - Adds security
3. IAM Identity Center
  - Single Sign-On for organizations
  - Integrates with company directory
4. Federated Access
  - Login using other identity provider
  - Example: Login with company ID instead of AWS ID
  - Good for organizations
5. Cross-Account IAM Roles

- User in account A can assume role in account B
- Good for multi-account organizations
- Temporary access

## **Task 2.4: Identify Components and Resources for Security**

### **AWS Security Services**

There are many AWS services to help you stay secure:

#### **AWS GuardDuty**

- Threat detection service
- Monitors for suspicious activity
- Machine learning finds unusual patterns
- Alerts you to threats

#### **Amazon Inspector**

- Automated security assessments
- Scans EC2 instances for vulnerabilities
- Reports security issues found
- Helps you patch problems

#### **AWS Security Hub**

- Central security monitoring
- Brings security findings from multiple services
- Shows security posture across account
- Compliance checking

#### **AWS Shield**

- DDoS (Distributed Denial of Service) protection
- DDoS: Hackers overwhelm server with traffic to crash it
- AWS Shield Standard: Included automatically
- AWS Shield Advanced: More protection available

#### **AWS WAF (Web Application Firewall)**

- Protects web applications
- Filters requests before reaching application
- Blocks common attack types
- Protects against SQL injection, XSS, etc.

## AWS Network ACLs

- Act as network firewall
- Control traffic in and out of subnet
- Stateless (each direction controlled separately)
- Good for network-level security

## Security Groups

- Act as firewall for individual resources
- Control what traffic EC2 or database can receive
- Stateful (if incoming allowed, outgoing is too)
- Very commonly used

## AWS Trusted Advisor

Trusted Advisor is a tool that checks your AWS setup for best practices.

What it checks:

- Cost Optimization: Are you paying more than needed?
- Performance: Is everything running efficiently?
- Security: Do you have security issues?
- Fault Tolerance: Are you protected if things fail?
- Service Limits: Are you near AWS limits?

Example findings:

- "You have EC2 instance running 24/7 but data shows unused. Consider stopping it to save money."
- "Security group is open to all IPs. Maybe restrict access."
- "Load balancer has no targets. Can be deleted."

Trusted Advisor helps you optimize AWS usage.

## Where to Find Security Information

AWS Security Resources:

1. AWS Security Center
  - Official AWS security information
  - Best practices
  - Compliance information

## 2. AWS Security Blog

- Updates on security
- Best practices articles
- Threat alerts

## 3. AWS Knowledge Center

- FAQs about AWS security
- Troubleshooting guides
- Common questions answered

## 4. AWS Artifact

- Compliance reports
- Audit reports
- Certification documents
- Proves AWS meets various standards

## Third-Party Security Products

You're not limited to AWS security products.

AWS Marketplace offers:

- Antivirus software
- Firewalls
- Monitoring tools
- Compliance tools
- From third-party vendors

Common third-party security on AWS:

- Splunk (monitoring)
  - CrowdStrike (endpoint protection)
  - Palo Alto Networks (firewall)
  - DataDog (monitoring)
-

## Part 5: Domain 3 - Cloud Technology and Services (34% of Exam)

This is the BIGGEST domain. It covers the actual AWS services you need to know about.

### Task 3.1: Define Methods of Deploying and Operating in AWS Cloud

#### How to Access AWS

There are three main ways to access and control AWS:

Method 1: AWS Management Console

- Web browser-based
- Click buttons to manage resources
- Visual, easy to learn
- Best for: Learning, occasional tasks, visualization

Example: Login to [aws.amazon.com](https://aws.amazon.com), click to create EC2 instance

Method 2: AWS Command Line Interface (CLI)

- Type commands in terminal/command prompt
- More powerful than console
- Can automate with scripts
- Best for: Automation, repeated tasks, power users

Example: `aws ec2 run-instances --image-id ami-12345 --count 1`

Method 3: AWS SDKs and APIs

- Programming libraries and direct API calls
- Most powerful
- Applications can control AWS programmatically
- Best for: Applications, complex automation

Example: Python code calling AWS to create databases automatically

Infrastructure as Code (IaC):

- Write code that describes your infrastructure
- AWS CloudFormation turns code into actual resources
- Benefits: Repeatable, version controlled, documented

## Types of Cloud Deployment Models

Companies use cloud in different ways:

### Model 1: Cloud Deployment

- Everything runs in AWS cloud
- No on-premises infrastructure
- Maximum cloud benefits
- Example: New startup using only AWS

### Model 2: Hybrid Deployment

- Some resources on-premises, some in AWS
- Gradual migration path
- Maintain legacy systems while using cloud
- Requires connectivity between on-premises and AWS

Example: Company keeps database on-premises for security, uses AWS for web application

### Model 3: On-Premises (Private Cloud)

- Everything stays in company data center
- Not really "cloud" but sometimes companies set up cloud-like infrastructure
- No AWS involvement
- Maximum control but maximum cost

## Connectivity Options

How do you connect your on-premises systems to AWS?

### Option 1: Public Internet

- Use regular internet connection
- Cheapest option
- Less secure (traffic goes over public internet)
- Sufficient for many use cases

### Option 2: AWS VPN (Virtual Private Network)

- Encrypted connection over internet
- More secure than plain internet
- Still uses internet but encrypted
- Good for: Secure hybrid connections

### Option 3: AWS Direct Connect

- Dedicated network connection to AWS
- Private connection (not over internet)
- More expensive but more secure
- Consistent performance
- Good for: High-security, high-volume data transfer

## Task 3.2: Define the AWS Global Infrastructure

### AWS Regions

An AWS Region is a completely separate geographic location with its own data centers.

Current AWS Regions (simplified):

- North America: US East (Virginia), US West (California), Canada
- South America: Brazil
- Europe: Ireland, Germany, France, Spain, UK, Sweden
- Asia Pacific: Singapore, Tokyo, Sydney, Seoul, Mumbai, Malaysia, Indonesia
- Middle East: Bahrain
- Africa: South Africa

Why multiple regions?

#### 1. Disaster Recovery

- If entire region fails (earthquake, power outage)
- Use different region automatically
- Business continues

#### 2. Local Presence

- Deploy in region closest to users
- Faster performance (less delay)
- Better user experience

#### 3. Data Sovereignty

- Some countries require data stay in country

- Can choose which region to use
- Comply with local laws

#### 4. Cost

- Some regions cheaper than others
- Can choose based on budget

## Availability Zones

Inside each Region are Availability Zones (AZs).

What is an Availability Zone?

- A separate data center (or cluster of data centers)
- Within same region
- Physically separated from other AZs
- Connected by fast network

Why have multiple AZs?

If one AZ fails:

- Earthquake damages one data center
- Data center still has power in other AZs
- All services keep running

High Availability:

- Deploy application in multiple AZs
- If one AZ fails, others continue
- No downtime for users

Typical region has 3 AZs (some have more).

## Edge Locations and CloudFront

Problem: Users far from your data center experience slow speed.

Solution: Cache content closer to users.

CloudFront:

- AWS Content Delivery Network (CDN)
- Caches content at edge locations around world



- User gets content from nearby edge location (faster)
- Reduces load on main servers

How it works:

1. Video stored in AWS Region Virginia
2. User in London requests video
3. CloudFront edge location in London serves it
4. User gets fast speed (no waiting for cross-ocean transfer)

Use cases:

- Websites (static files served fast)
- Videos (faster streaming)
- Software distribution
- Any content served to global users

## **AWS Global Accelerator**

Similar to CloudFront but for dynamic content.

- Routes traffic through AWS network for faster speed
- Improves performance for applications
- Similar concept to CloudFront but different approach

## **Local Zones and Wavelength Zones**

Specialized infrastructure for specific needs:

Local Zones:

- Extensions of AWS Regions
- Deployed closer to users
- Very low latency (fast speed)
- For applications needing extreme speed

Wavelength Zones:

- AWS infrastructure embedded in 5G networks
- For mobile applications
- Extremely low latency
- Cutting-edge technology

## **Task 3.3: Identify AWS Compute Services**

Compute means the processing power and servers.

## **Amazon EC2 (Elastic Compute Cloud)**

EC2 is virtual servers in the cloud.

What is EC2?

- On-demand computing capacity
- Can create or delete servers instantly
- Pay by hour
- Complete control over operating system and software

EC2 Instance Types:

Different instance types for different needs:

General Purpose (T3, M5):

- Balanced compute, memory, network
- Good for web applications
- Good for development and testing
- Example: Web server handling moderate traffic

Compute Optimized (C5, C6):

- High CPU performance
- For computationally intensive tasks
- Example: Scientific simulations, batch processing

Memory Optimized (R5, R6):

- Lots of RAM memory
- For applications processing large datasets
- Example: In-memory databases, real-time analytics

Storage Optimized (I3, H1):

- High disk I/O performance
- For data-intensive applications
- Example: NoSQL databases, data warehousing

Accelerated Computing (P3, G4):

- GPU or FPGA processors
- For machine learning, graphics
- Example: Training AI models

## EC2 Purchasing Options:

### On-Demand:

- Pay by hour
- No commitment
- Most expensive
- Good for: Unpredictable workloads

### Reserved Instances:

- Commit for 1 or 3 years
- Gets 40-60% discount
- Good for: Predictable workloads (always running)

### Spot Instances:

- Bid for spare capacity
- Can be interrupted (AWS might need capacity)
- 70-90% discount
- Good for: Flexible workloads, batch jobs

### Savings Plans:

- Flexible pricing over time
- Can mix instance types
- Discount for commitment

### Dedicated Hosts:

- Entire physical server just for you
- For licensing compliance
- Most expensive option

### Capacity Reservations:

- Reserve capacity in specific AZ
- Guaranteed availability
- Pay whether you use it or not

## Load Balancers

Purpose: Distribute traffic across multiple servers.

Why needed?

- One server can only handle so much traffic

- One server failing takes down website
- Solution: Multiple servers, load balancer distributes traffic

How it works:

1. User requests reach load balancer
2. Load balancer sends each request to one server
3. If one server fails, load balancer routes to others
4. Users see no downtime

Types of load balancers:

- Application Load Balancer (ALB): For web applications
- Network Load Balancer (NLB): For extreme performance
- Classic Load Balancer: Legacy

## **Auto Scaling**

Auto Scaling automatically adds or removes servers based on demand.

How it works:

1. You set minimum and maximum servers (e.g., 2 to 10)
2. You set trigger conditions (e.g., if CPU above 80%)
3. If demand increases, more servers automatically added
4. If demand decreases, excess servers removed
5. You only pay for servers running

Example:

- Normal day: 2 servers running, costs \$200
- Black Friday surge: 15 servers running, costs \$1500
- Next day: Back to 2 servers, costs \$200 again

Benefits:

- Automatically scales for demand
- No manual intervention
- Saves money during low traffic
- Handles high traffic automatically

## **AWS Lambda**

Lambda is serverless computing.

What is serverless?

- You don't manage servers
- Just upload code
- AWS runs it when triggered
- Pay only for time code runs

How it works:

1. Write code (any supported language)
2. Upload to Lambda
3. Set trigger (e.g., file uploaded to S3, API call)
4. When triggered, Lambda runs your code
5. Results returned

Use cases:

- Image resizing (when image uploaded to S3)
- Processing form submissions
- Scheduled tasks
- Real-time data transformation

Advantages:

- No server management
- Automatic scaling
- Pay per execution (very cheap if not used much)
- Quick deployment

Limitations:

- Maximum 15 minutes execution time
- Can't customize server environment
- Performance less predictable than EC2

## **Amazon ECS and EKS (Containers)**

Containers are lightweight, portable application environments.

Think of container like:

- Traditional: Full computer in virtual machine
- Container: Just the application and its dependencies
- Much smaller and faster than full virtual machine

Amazon ECS (Elastic Container Service):

- Manage Docker containers
- AWS-built container orchestration
- Simpler than Kubernetes

Amazon EKS (Elastic Kubernetes Service):

- Run Kubernetes (industry standard)
- If you want Kubernetes features
- More complex but more powerful

When to use containers:

- Microservices (break big app into small independent pieces)
- Multiple versions of same service
- Consistent deployment environments
- Development and production match

## **AWS Fargate**

Fargate is "serverless containers."

- Run containers without managing servers
  - Like Lambda but for container applications
  - Containers start automatically
  - You pay only for resource used
- 

# **Part 6: AWS Storage and Database Services**

## **Storage Services**

### **Amazon S3 (Simple Storage Service)**

S3 is object storage (like a filing cabinet for files).

What can you store?

- Documents
- Images
- Videos
- Log files

- Backups
- Any file

Key features:

- Unlimited storage
- Highly available (stored in multiple AZs)
- Durable (won't lose data)
- Accessible from anywhere via internet

S3 Buckets:

- S3 storage organized into buckets (like folders)
- Bucket names globally unique
- Inside bucket are objects (files)

S3 Storage Classes (Pricing Options):

S3 Standard:

- Most expensive per GB
- Immediately available
- Most frequently accessed data

S3 Intelligent-Tiering:

- Automatically moves between classes
- For unpredictable access patterns
- Cost optimized

S3 Standard-IA (Infrequent Access):

- Cheaper storage but costs to retrieve
- For data accessed infrequently
- Keep backups here

S3 Glacier:

- Very cheap storage
- Retrieval takes minutes to hours
- For archives/backup
- Very low cost

S3 Glacier Deep Archive:

- Cheapest option
- Retrieval takes 12+ hours

- For long-term archives
- Minimal cost

Use cases:

- Website static files
- Backups
- Data archives
- Log files
- Application media

## **Amazon Elastic Block Store (EBS)**

EBS is block storage (like a hard drive for EC2).

How it works:

- Attach EBS volume to EC2 instance
- Use like regular hard drive
- Survives if EC2 shuts down

Use cases:

- EC2 operating system and application files
- Databases requiring fast storage
- Applications needing consistent performance

## **Amazon Elastic File System (EFS)**

EFS is file storage (shared file system).

Key feature: Multiple EC2 instances can access same files simultaneously.

Use cases:

- Shared file storage for multiple servers
- Development environments
- Content management systems

## **Amazon FSx**

FSx is managed file storage with specific operating system compatibility.

Variants:

- FSx for Windows (Windows-compatible)
- FSx for Lustre (Linux, high-performance computing)



## Database Services

### Amazon RDS (Relational Database Service)

RDS is managed database for structured data (like Excel spreadsheet with defined columns).

Supported databases:

- MySQL
- PostgreSQL
- MariaDB
- Oracle
- SQL Server

Why use RDS instead of self-managed database?

AWS handles:

- Installation
- Patching
- Backups
- Automatic failover
- Scaling
- High availability

You handle:

- Data content
- Database configuration
- Access control

### Amazon Aurora

Aurora is AWS's own database engine (based on MySQL/PostgreSQL).

Advantages:

- Very fast (5x faster than MySQL)
- Highly available (automatic replication)
- Scalable
- Reliable

Best for: High-performance, mission-critical applications.

## Amazon DynamoDB

DynamoDB is NoSQL database (unstructured data like documents).

Difference from RDS:

- RDS: Structured, defined columns, complex queries
- DynamoDB: Flexible structure, fast reads/writes, simpler queries

When to use DynamoDB:

- Real-time applications (like social media feeds)
- Mobile applications
- IoT data
- Web sessions
- Time-series data

Key benefits:

- Extremely fast
- Automatic scaling
- Pay for what you use (cheaper if traffic bursty)

## Other Database Services

Amazon Neptune:

- Graph database
- For relationships between data
- Example: Social networks, recommendation engines

Amazon MemoryDB for Redis:

- In-memory data store
- Extremely fast
- For caching and real-time leaderboards

Amazon Redshift:

- Data warehouse
- For analyzing huge amounts of data
- Different from operational databases

## Database Migration Service (DMS)

DMS helps migrate databases from on-premises or other clouds to AWS.

How it works:

- Connects to source database
- Copies data to target (AWS database)
- Can run continuously to sync changes
- Minimal downtime

When to use:

- Moving Oracle to RDS
- Moving on-premises database to AWS
- Consolidating multiple databases

## **Schema Conversion Tool (SCT)**

Helps convert database schemas between different database types.

Example:

- Converting Oracle database to PostgreSQL
  - SCT analyzes and converts schema
  - Saves development time
- 

## **Part 7: AWS Networking Services**

### **Amazon VPC (Virtual Private Cloud)**

VPC is your own private network within AWS cloud.

Key concepts:

Subnets:

- Division of VPC into smaller sections
- Can span only one Availability Zone
- Public subnet: Connected to internet
- Private subnet: Not directly connected to internet

Internet Gateway:

- Allows communication between VPC and internet
- Necessary for public subnets
- Allows incoming and outgoing internet traffic

NAT Gateway:

- For private subnets to access internet

- Translates private IP to public IP
- Outgoing only (internet can't initiate connections)

#### Security Groups:

- Firewall for individual resources
- Stateful (if outgoing allowed, incoming response allowed)
- Can specify which ports and IPs allowed

#### Network ACLs:

- Firewall for entire subnet
- Stateless (each direction controlled separately)
- More granular control than security groups

#### Route Tables:

- Rules for where traffic goes
- Determines how packets are routed
- Multiple routes can exist

#### Example VPC Setup:

- VPC: 10.0.0.0/16
- Public Subnet 1: 10.0.1.0/24 (has internet gateway)
- Private Subnet 1: 10.0.2.0/24 (no direct internet)
- Traffic between subnets: Controlled by route tables

## Amazon Route 53

Route 53 is DNS service (translates domain names to IP addresses).

#### How DNS works:

1. User types example.com in browser
2. Route 53 looks up IP address for example.com
3. Returns 192.0.2.1 (for example)
4. Browser connects to 192.0.2.1

#### Features:

- Register domain names
- DNS resolution (translate names to IPs)
- Health checking (test if servers are up)
- Routing policies (direct traffic based on rules)

#### Routing policies:

Simple routing:

- Route all traffic to one location

Weighted routing:

- Direct percentage of traffic to different locations
- Example: 80% to US, 20% to Europe

Latency-based routing:

- Route users to location with lowest latency
- Automatic performance optimization

Geo-location routing:

- Route based on geographic location
- Example: Serve different content to different countries

## **CloudFront and Global Accelerator**

Already covered earlier but important for networking:

CloudFront:

- Caches content globally
- Reduces latency for users
- Good for static content

Global Accelerator:

- Routes traffic through AWS network
- Improves performance for dynamic content
- Good for non-HTTP protocols

## **AWS VPN and Direct Connect**

Already covered but summary:

AWS VPN:

- Encrypted connection over internet
- For secure hybrid connectivity

AWS Direct Connect:

- Dedicated network connection
  - Private, not over internet
  - More secure and consistent
-

## Part 8: Domain 4 - Billing, Pricing, and Support (12% of Exam)

### Task 4.1: Compare AWS Pricing Models

#### Compute Purchasing Options

##### On-Demand Instances:

- Pay by hour or second
- No commitment
- Most expensive
- Maximum flexibility
- Best for: New projects, unpredictable load

Example: \$0.10 per hour, run 1,000 hours = \$100

##### Reserved Instances:

- Commit to 1 or 3 years
- Significant discount (30-60%)
- Payment upfront or monthly
- Good for steady workloads
- Best for: Production baseline

Example: 1-year commitment = \$50/month (vs. \$73/month on-demand)

##### Spot Instances:

- Bid for spare capacity
- 70-90% discount vs on-demand
- Can be interrupted (AWS might need capacity)
- No SLA (service level agreement)
- Best for: Flexible workloads, batch jobs, development

Example: \$0.02 per hour (vs. \$0.10 on-demand)

##### Savings Plans:

- Flexible pricing for 1 or 3 years
- Can change instance type
- Discount 10-50%
- More flexible than reserved instances
- Best for: Multi-service commitments

#### Dedicated Hosts:

- Entire physical server for you
- Most expensive option
- For license compliance
- Example: Some software licenses tied to physical hardware

#### Dedicated Instances:

- Instance runs on dedicated hardware
- Other AWS customers not on same hardware
- Slightly more expensive than on-demand
- For compliance/security

#### Capacity Reservations:

- Reserve capacity in specific AZ
- Guarantees you can launch instance when needed
- Pay whether you use it or not
- For mission-critical applications

## Data Transfer Charges

AWS charges for data going across regions or out of AWS.

#### Free Data Transfer:

- Within AWS region (between AZs)
- Into AWS from internet

#### Chargeable Data Transfer:

- Out of AWS to internet (\$0.12 per GB typical)
- Between regions (\$0.02 per GB typical)

#### Impact:

- Design applications to minimize data transfer
- Keep related services in same region
- Use CloudFront to avoid repeated downloads

## Storage Pricing

Different storage types have different costs:

S3 Standard: \$0.023 per GB

- Most expensive

- Immediately available

S3 IA: \$0.0125 per GB

- Cheaper storage
- Retrieval fees apply

Glacier: \$0.004 per GB

- Very cheap storage
- Slow retrieval

EBS: \$0.10 per GB per month

- For EC2 volumes
- Fast performance

## **Task 4.2: Understand Resources for Billing, Budget, and Cost Management**

### **AWS Budgets**

AWS Budgets is tool to set spending limits.

Features:

- Set budget for services
- Get alerts when approaching limit
- Forecast spending based on current usage
- Can automate actions (shut down resources when limit reached)

Example:

- Set budget: \$500 per month
- Alert at \$400 (80%)
- Action at \$500: Stop expensive resources

### **AWS Cost Explorer**

AWS Cost Explorer analyzes your AWS spending.

Features:

- See spending by service
- See spending over time
- Forecast future spending
- Identify cost optimization opportunities



Example findings:

- "EC2 spending increased 30% this month"
- "Unused S3 storage increasing costs"
- "Reserved instances saving \$500/month"

## **AWS Billing Conductor**

Billing Conductor helps manage costs in large organizations.

Features:

- Create custom billing groups
- Allocate costs to departments
- Show back costs to business units
- Enterprise billing management

## **AWS Pricing Calculator**

Free tool to estimate AWS costs before deploying.

How to use:

1. Select services you want to use
2. Input expected usage
3. Get estimated monthly cost
4. Export estimate

Helps:

- Budget for new projects
- Compare different configurations
- Understand cost drivers

## **AWS Organizations**

For managing multiple AWS accounts:

Features:

- Consolidated billing (combine all accounts)
- Shared Reserved Instances across accounts
- Cost allocation tags
- Centralized management

Example:

- Company has 5 AWS accounts
- Using AWS Organizations to combine into one bill
- All Reserved Instances shared across accounts
- Better costs than separate accounts

## **Cost Allocation Tags**

Tags are labels for AWS resources.

Example tags:

- Project: ProjectX
- Environment: Production
- Department: Finance
- CostCenter: 12345

Benefits:

- Track costs by tag
- Allocate costs to departments
- Understand spending by project
- Automatic cost reporting

## **Task 4.3: Identify AWS Technical Resources and AWS Support Options**

### **AWS Support Plans**

AWS offers different support levels:

Basic Support (Free):

- Included with all AWS accounts
- Email support
- Community forum access
- AWS Trusted Advisor (limited checks)
- No SLA

Developer Support (\$29/month minimum):

- Business hours email support
- <24 hour response for general questions

- AWS Trusted Advisor (full checks)
- Good for: Development and testing

Business Support (\$100/month minimum):

- 24/7 phone, chat, email support
- 1 hour response for urgent issues
- Technical guidance
- Good for: Production environments

Enterprise On-Ramp Support (\$5,500/month):

- 15 minute response for critical issues
- Proactive support
- Guidance for optimization
- Good for: Large organizations

Enterprise Support (\$15,000/month):

- 15 minute response for critical issues
- Dedicated Technical Account Manager
- Quarterly business reviews
- Good for: Mission-critical workloads

What increases cost:

- Faster response times
- 24/7 availability
- Dedicated support person
- Proactive optimization

## **AWS Trusted Advisor**

Trusted Advisor checks your AWS account for best practices.

Free checks:

- Service limits
- Security groups

Paid checks (require Business or Enterprise support):

- Cost optimization
- Performance
- Security (advanced)
- Fault tolerance

## **AWS Health Dashboard**

Shows AWS service health and personal account health.

Service Health:

- Shows if AWS services are having issues
- Helps you understand if problems are AWS or your account

Account Health:

- Shows issues specific to your account
- Alerts about maintenance
- Upcoming changes affecting you

## **Trusted Advisor and Health Dashboard Together**

Use both for complete picture:

- Trusted Advisor: Best practice recommendations
- Health Dashboard: Current status and issues

## **AWS Knowledge Center**

Repository of common questions and answers.

Good for:

- Troubleshooting
- Learning
- Finding solutions

## **AWS re:Post**

Community Q&A platform.

Good for:

- Ask questions
- Get community answers
- Share knowledge

## **AWS Professional Services**

For organizations needing expert help.

Services:

- Architecture review

- Implementation assistance
- Training
- Best practices guidance

## **AWS Solutions Architects**

AWS provides solutions architects to help design solutions.

Available through:

- Business/Enterprise support
- AWS Professional Services
- Enterprise account programs

## **AWS Marketplace**

Marketplace offers solutions from AWS partners.

What's available:

- Pre-built applications
- Cost management tools
- Governance tools
- Security tools
- Data products

Benefits:

- Pre-built vs. building from scratch
- Integration with AWS billing
- One-click deployment

## **AWS Partners**

AWS Partner Network includes:

Independent Software Vendors (ISVs):

- Companies building software on AWS
- Available in marketplace

System Integrators:

- Companies helping implement solutions
- Consulting services

Resellers:

- Companies selling AWS services

Benefits of AWS Partners:

- Partner training and certification
  - Co-marketing opportunities
  - Volume discounts
  - Support
- 

## **Part 9: AI/ML and Analytics Services**

These are important for the exam and growing in AWS.

### **Amazon SageMaker**

SageMaker is for machine learning (AI training).

What it does:

- Build, train, deploy machine learning models
- Pre-built algorithms
- Jupyter notebooks for development
- Automated machine learning (AutoML)

Use cases:

- Predictive analytics
- Image recognition
- Natural language processing
- Recommendation engines

### **Amazon Lex**

Lex is for building chatbots.

What it does:

- Conversational interface
- Understands natural language
- Can integrate with other AWS services

Use cases:

- Customer service chatbot

- Virtual assistant
- FAQ bot

## **Amazon Kendra**

Kendra is enterprise search service.

What it does:

- Search through business documents
- Returns relevant answers
- Uses AI to understand questions

Use cases:

- Search company documentation
- Find information across multiple sources
- Improve document discoverability

## **Amazon Rekognition**

Rekognition is image and video recognition.

What it does:

- Detect objects in images
- Recognize faces
- Read text in images
- Analyze videos

Use cases:

- Security camera analysis
- Photo organization
- Content moderation
- Document processing

## **Amazon Polly**

Polly is text-to-speech service.

What it does:

- Convert text to spoken audio
- Natural sounding voices
- Multiple languages

Use cases:

- Accessibility
- Audio content creation
- Virtual assistant voice

## **Amazon Transcribe**

Transcribe is speech-to-text service.

What it does:

- Convert audio to text
- Supports multiple languages
- Medical and legal vocabulary

Use cases:

- Meeting transcription
- Video captioning
- Voicemail transcription

## **Amazon Translate**

Translate is machine translation service.

What it does:

- Translate between languages
- Maintains context

Use cases:

- Website translation
- Document translation
- Customer support in multiple languages

## **Amazon Comprehend**

Comprehend is natural language processing service.

What it does:

- Detect language
- Extract entities (people, places, etc.)
- Sentiment analysis (happy, sad, neutral)



Use cases:

- Customer feedback analysis
- Content categorization
- Entity extraction from documents

## **Analytics Services**

### **Amazon Athena**

Query data in S3 using SQL.

What it does:

- Run SQL queries against data in S3
- Pay per query executed
- No servers to manage

Use cases:

- Analyze log files in S3
- Ad-hoc analysis
- Data lake queries

### **Amazon Kinesis**

Real-time data streaming.

What it does:

- Collect real-time data
- Process streaming data
- Build real-time applications

Use cases:

- Real-time analytics
- IoT data processing
- Live leaderboards

### **AWS Glue**

Data integration service for ETL (Extract, Transform, Load).

What it does:

- Extract data from sources

- Transform/clean data
- Load into data warehouse

Use cases:

- Data pipeline creation
- Data catalog
- Data preparation for analytics

## **Amazon QuickSight**

Business intelligence and visualization.

What it does:

- Create dashboards
- Visualize data
- Ad-hoc analysis

Use cases:

- Executive dashboards
  - Business metrics visualization
  - Data-driven decisions
- 

# **Part 10: Application Integration and Other Services**

## **Amazon SNS (Simple Notification Service)**

SNS is message publishing service.

What it does:

- Publish messages
- Subscribe to topics
- Send notifications

Use cases:

- Alert notifications
- Pub/sub messaging
- Application alerts

## **Amazon SQS (Simple Queue Service)**

SQS is message queue service.

What it does:

- Queue messages
- Decouple applications
- Process messages asynchronously

Use cases:

- Application decoupling
- Asynchronous processing
- Batch processing

## **Amazon EventBridge**

EventBridge routes events between services.

What it does:

- Receives events
- Routes based on rules
- Triggers actions

Use cases:

- Serverless application integration
- Event-driven architectures
- Cross-service communication

## **AWS Lambda (Additional Details)**

Already covered but additional uses:

Common triggers:

- S3 file upload
- DynamoDB changes
- API calls
- Scheduled events
- SNS messages
- SQS messages

Cost model:

- Pay per execution
- Free tier: 1 million requests/month
- Very cheap for low-volume applications

## **Amazon Connect**

Connect is contact center service.

What it does:

- Cloud-based call center
- IVR (Interactive Voice Response)
- Chat and screen sharing

Use cases:

- Customer support center
- Sales call center
- Technical support

## **Amazon SES (Simple Email Service)**

SES is email sending service.

What it does:

- Send emails programmatically
- Bulk email sending
- Email templates

Use cases:

- Transactional emails
  - Marketing emails
  - Application notifications
- 

# **Part 11: Management and Governance Services**

## **AWS CloudFormation**

CloudFormation is Infrastructure as Code.

What it does:

- Define infrastructure in code (JSON or YAML)
- Create entire environment with one command
- Version control your infrastructure
- Repeatable deployments

Example:

- Write CloudFormation template describing 10 EC2 instances, load balancer, RDS database
- Run template
- All resources created automatically
- Change template, run again, updates automatically

Benefits:

- No manual clicking
- Repeatable and consistent
- Version controlled
- Easy to tear down (delete entire stack)

## **AWS CloudTrail**

Already mentioned but important details:

CloudTrail logs every API call made in AWS account.

Logged information:

- Who made the call (user, role)
- What service was called
- What action was performed
- When (timestamp)
- From where (IP address)
- Success or failure

Uses:

- Security investigation (who changed what?)
- Compliance proof
- Troubleshooting
- Audit trail

## **Amazon CloudWatch**

CloudWatch monitors AWS resources.

What it tracks:

- CPU utilization
- Memory usage
- Disk I/O

- Network traffic
- Application-specific metrics

Features:

- Create dashboards
- Set alarms (alert when threshold exceeded)
- View logs
- Create metrics

## **AWS Config**

Config tracks configuration changes.

What it does:

- Record resource configurations
- Track changes over time
- Evaluate compliance
- Alert when non-compliant

Example:

- Security group changed from 0.0.0.0/0 to 10.0.0.0/8
- Config logs this change
- Alerts about security change
- Shows what changed and when

## **AWS Auto Scaling**

Already covered but importance for management:

Auto Scaling automatically adjusts resources based on demand.

What it monitors:

- CPU utilization
- Request count
- Custom metrics

Typical setup:

- Minimum instances: 2
- Maximum instances: 10
- Scale-up rule: If CPU > 70%, add instance
- Scale-down rule: If CPU < 30%, remove instance

## AWS Systems Manager

Systems Manager provides operational capabilities.

Features:

- Run commands on EC2 instances
- Patch management
- Session management (connect to EC2 without SSH)
- Parameter store (store configuration)

Use cases:

- Bulk operations on many instances
  - Automated patching
  - Remote administration
- 

## Part 12: Exam Preparation and Test-Taking Strategies

### Study Plan

Week 1: Cloud Concepts

- Read Domain 1 section multiple times
- Watch AWS overview videos
- Understand AWS value proposition
- Learn AWS Well-Architected Framework

Week 2: Security and Compliance

- Study shared responsibility model (critical!)
- Learn IAM basics thoroughly
- Understand security services
- Practice security concepts

Week 3: Technology Services

- Study compute services (EC2, Lambda, containers)
- Learn storage services (S3, EBS, EFS)
- Study databases (RDS, DynamoDB)
- Understand networking basics

Week 4: Billing and Review

- Study pricing models
- Learn cost management tools

- Review AWS support options
- Full review of all domains

## Practice Questions Strategy

Sample question format:

"A company wants to reduce costs while maintaining same capacity. They run EC2 instances 24/7. Which option provides best savings?"

- A. Use Spot instances (WRONG - can be interrupted)
- B. Use On-Demand instances (WRONG - most expensive)
- C. Use Reserved instances (CORRECT - significant discount)
- D. Use Dedicated Hosts (WRONG - most expensive)

How to answer:

1. Read entire question carefully
2. Identify what's being asked (cost optimization with same capacity)
3. Consider each option
4. Eliminate clearly wrong answers
5. Choose best remaining option

Multiple response example:

"Which of the following are responsibilities of the customer when using Amazon RDS?" (Select TWO)

- A. Patching the database software (WRONG - AWS does this)
- B. Setting database backups schedule (CORRECT)
- C. Securing physical data centers (WRONG - AWS does this)
- D. Managing encryption of data (CORRECT)
- E. Choosing RDS instance type (CORRECT - but only select 2)

Answer: D and B (or D and E)

## Exam Day Tips

Before Exam:

- Get good sleep night before
- Eat healthy breakfast
- Arrive early to testing center



- Use restroom before starting
- Minimize stress

During Exam:

- Read questions completely before answering
- Don't rush (you have time)
- Mark difficult questions to review
- Answer all questions (no penalty for guessing)
- For multiple response questions, read all options before selecting

Time Management:

- Read question: 30 seconds
- Answer question: 60 seconds
- Move to next: If uncertain, mark and move on
- If time permits at end, review marked questions

Mindset:

- Trust your study
- Don't second-guess correct answers
- Remember: 70% correct = pass
- You don't need to know everything
- Focus on understanding concepts

## **Common Mistakes to Avoid**

Mistake 1: Not understanding shared responsibility model

- This is tested heavily
- You need to know what AWS does vs. what you do
- Different for each service

Mistake 2: Confusing similar services

- EC2 vs. Lambda (virtual servers vs. serverless)
- RDS vs. DynamoDB (relational vs. NoSQL)
- S3 vs. EBS (object storage vs. block storage)

Mistake 3: Not reading questions carefully

- Missing "NOT," "EXCEPT," "MOST LIKELY"
- Changing meaning of question

Mistake 4: Overthinking simple questions

- Some questions are straightforward
- Don't add complexity that isn't there

Mistake 5: Not managing time

- Spending 10 minutes on one question
  - Running out of time at end
- 

## **Part 13: Final Review and Key Concepts to Remember**

### **Domain 1: Cloud Concepts (MUST KNOW)**

Key concepts:

- AWS costs less because of economies of scale
- Global infrastructure enables local deployment
- Elasticity and high availability key advantages
- Well-Architected Framework has 6 pillars
- AWS CAF helps with migration planning
- Fixed costs vs. variable costs (AWS is variable)
- Managed services reduce operational burden

### **Domain 2: Security and Compliance (MOST IMPORTANT)**

Key concepts:

- Shared responsibility model: AWS security OF cloud, you security IN cloud
- Principle of least privilege: Give minimum needed access
- IAM is how you manage access
- Protect AWS root user account
- Multi-factor authentication improves security
- Encryption in transit and at rest both important
- CloudTrail logs everything for compliance

### **Domain 3: Technology Services (BIGGEST DOMAIN)**

Key concepts:

- EC2: Virtual servers, multiple instance types, multiple purchasing options
- Lambda: Serverless, short-running code
- RDS: Managed relational databases

- DynamoDB: NoSQL database
- S3: Object storage, multiple storage classes
- VPC: Your private network
- Load balancers distribute traffic
- Auto scaling adjusts resources automatically
- CloudFront caches content globally

## **Domain 4: Billing (DON'T OVERLOOK)**

Key concepts:

- On-Demand: Most expensive, no commitment
- Reserved Instances: 1-3 year commitment, 30-60% discount
- Spot Instances: 70-90% discount but can be interrupted
- Savings Plans: Flexible commitment discount
- AWS Cost Explorer analyzes spending
- AWS Budgets set spending limits
- Data transfer between regions costs money
- Reserved instances can be shared across accounts

## **Critical Mindset**

Remember why each service exists:

- Every AWS service solves a specific problem
- Exam questions present problems
- You choose which service solves it

Example:

- Problem: Need fast database (CORRECT: DynamoDB)
  - Problem: Need relational database (CORRECT: RDS)
  - Problem: Need cheap archival storage (CORRECT: Glacier)
  - Problem: Need to distribute web content globally (CORRECT: CloudFront)
- 

## **Part 14: Quick Reference Glossary**

API: Application Programming Interface (way to interact with services)

AZ: Availability Zone (single data center or cluster)

CloudFront: Content Delivery Network caching service

CloudTrail: Audit logging service

CloudWatch: Monitoring service

DynamoDB: NoSQL database service

EBS: Elastic Block Storage (hard drive for EC2)

EC2: Elastic Compute Cloud (virtual servers)

ECS: Elastic Container Service (container orchestration)

EFS: Elastic File System (shared file storage)

EKS: Elastic Kubernetes Service (Kubernetes)

FSx: File storage with specific OS compatibility

IAM: Identity and Access Management

IaC: Infrastructure as Code (CloudFormation)

Lambda: Serverless compute service

NAT Gateway: Network Address Translation for private subnets

RDS: Relational Database Service (managed database)

Route 53: DNS service

S3: Simple Storage Service (object storage)

SageMaker: Machine learning service

VPC: Virtual Private Cloud (private network)

WAF: Web Application Firewall

---

## Part 15: Practice Scenarios

### Scenario 1: Retail Company During Holiday Season

Company situation:

- Normal traffic: 100 requests/second
- Black Friday traffic: 5,000 requests/second
- Spike lasts 3 days
- After holiday, traffic returns to normal

Which services to use?

- Amazon EC2 with Auto Scaling (for traffic spike)
- Elastic Load Balancer (distribute traffic across servers)
- Amazon S3 (serve static content fast)
- CloudFront (cache content globally)
- On-Demand or Spot instances (don't commit to 3-year for temporary need)

Cost savings:

- Auto Scaling: Only pay for servers during peak (3 days) vs. 365 days
- CloudFront: Reduces load on main servers
- Spot instances: Save 70-90% on instance cost

## **Scenario 2: Healthcare Company Compliance**

Company needs:

- HIPAA compliance (patient data protection)
- Audit trail (know who accessed what)
- Data encryption
- Backup capability

Which services to use?

- AWS IAM (control access, principle of least privilege)
- CloudTrail (audit who accessed what)
- AWS KMS (encryption key management)
- CloudWatch (monitor access)
- RDS with encryption (managed database with protection)
- AWS Backup (automated backups)
- AWS Artifact (get HIPAA compliance reports)

Why these services:

- HIPAA requires proof of security controls
- AWS provides these controls
- Audit trails prove compliance
- Encryption protects patient data

## **Scenario 3: Startup With Unpredictable Load**

Startup situation:

- Small team, limited budget

- Traffic unpredictable (goes viral sometimes)
- Need to scale quickly
- Don't want to manage infrastructure

Which services to use?

- AWS Lambda (serverless, pays per execution)
- Amazon DynamoDB (NoSQL, automatic scaling)
- Amazon API Gateway (serverless API)
- CloudFront (cache to reduce costs)
- AWS S3 (static content storage)

Cost model:

- Free tier covers first 1 million requests
- Pay only if exceeds free tier
- Scales automatically
- Minimal operational overhead

## **Scenario 4: Enterprise Multi-Region Deployment**

Company needs:

- Disaster recovery (if one region fails)
- Low latency for global users
- Compliance with local data laws
- Cost optimization

Architecture:

- Primary region: US
- Secondary region: Europe (for compliance, disaster recovery)
- Tertiary region: Asia-Pacific (for local performance)

Services:

- Route 53 (failover DNS if region fails)
- RDS with Multi-Region read replicas (data replicated)
- CloudFront (edge locations cache globally)
- Auto Scaling (each region scales independently)
- AWS DMS (migrate data between regions)

---

## **Part 16: Final Checklist Before Taking Exam**

### Study Readiness:

- Have you read all 4 domains at least twice?
- Can you explain shared responsibility model?
- Do you know 5 compute service options?
- Can you compare EC2 purchasing options?
- Understand IAM basics?
- Know difference between services (EC2 vs. Lambda, RDS vs. DynamoDB)?

### Practice:

- Taken at least 1 practice exam?
- Scored 70%+?
- Reviewed wrong answers?
- Understood why correct answers are correct?

### Confidence:

- Do you feel ready?
- If not, where are weak spots?
- Study weak areas for 1-2 more days

### Logistics:

- Know exam date and time?
- Know testing center location?
- Know how to get there?
- Planned for traffic/delays?
- Have valid ID?

### Health:

- Getting 8 hours sleep before exam?
- Not stressed?
- Can take test in calm mind?

---

## Conclusion

You now have a comprehensive guide covering all aspects of the AWS Certified Cloud Practitioner exam.

### Key Takeaways:

1. AWS Certified Cloud Practitioner is foundational level
2. 4 domains tested: Cloud Concepts (24%), Security (30%), Technology (34%),

### Billing (12%)

3. Need 700 out of 1000 score to pass (approximately 70%)
4. Focus on understanding concepts, not memorizing details
5. Shared responsibility model is critical for security questions
6. Know when to use which service
7. Understand pricing models and cost optimization
8. Practice with sample questions
9. Get good sleep before exam
10. Trust your study and don't panic

Remember: This certification is meant to be achievable. It's entry-level. If you've read this guide thoroughly and practiced questions, you're ready.

Now go pass this exam!

Best of luck on your AWS Certified Cloud Practitioner certification!



# All AWS Certifications (Total:13)

	A	B	C	D	E	F	G	H	I	J	K	L
	Certification Name	Level	Experience Required	Prerequisites	Exam Cost	Retake Cost	Number of Questions	Duration (Minutes)	Passing Score	Required Cei	Maintenance	Official Credential Link
1	AWS Certified Cloud Practitioner	Foundational	0-6 months	None	\$100	N/A	65	90	70% (700/1000)	None	Every 3 years	<a href="https://aws.amazon.com/cer">https://aws.amazon.com/cer</a>
2	AWS Certified AI Practitioner	Foundational	0-6 months	None	\$100	N/A	65	90	70% (700/1000)	None	Every 3 years	<a href="https://aws.amazon.com/cer">https://aws.amazon.com/cer</a>
3	AWS Certified CloudOps Engineer - Associate	Associate	1+ year operations	None	\$150	\$100	65	130	72% (720/1000)	None	Every 3 years	<a href="https://aws.amazon.com/cer">https://aws.amazon.com/cer</a>
4	AWS Certified Solutions Architect - Associate	Associate	1+ year with AWS	None	\$150	\$100	65	130	72% (720/1000)	None	Every 3 years	<a href="https://aws.amazon.com/cer">https://aws.amazon.com/cer</a>
5	AWS Certified Developer - Associate	Associate	1+ year development	None	\$150	\$100	65	130	72% (720/1000)	None	Every 3 years	<a href="https://aws.amazon.com/cer">https://aws.amazon.com/cer</a>
6	AWS Certified Data Engineer - Associate	Associate	1+ year data role	None	\$150	\$100	65	130	72% (720/1000)	None	Every 3 years	<a href="https://aws.amazon.com/cer">https://aws.amazon.com/cer</a>
7	AWS Certified Machine Learning Engineer - Ass	Associate	1+ year ML services	None	\$150	\$100	65	130	72% (720/1000)	None	Every 3 years	<a href="https://aws.amazon.com/cer">https://aws.amazon.com/cer</a>
8	AWS Certified Solutions Architect - Professional	Professional	2+ years with AWS	Solutions Archt	\$300	\$200	75	180	75% (750/1000)	Solutions Arc	Every 3 years	<a href="https://aws.amazon.com/cer">https://aws.amazon.com/cer</a>
9	AWS Certified DevOps Engineer - Professional	Professional	2+ years DevOps	Developer or !	\$300	\$200	75	180	75% (750/1000)	Developer or	Every 3 years	<a href="https://aws.amazon.com/cer">https://aws.amazon.com/cer</a>
10	AWS Certified Generative AI Developer - Profes	Professional	2+ years AWS	None	\$300	\$200	75	180	75% (750/1000)	None	Every 3 years	<a href="https://aws.amazon.com/cer">https://aws.amazon.com/cer</a>
11	AWS Certified Advanced Networking - Specialty	Specialty	5+ years networking	None	\$300	\$200	65	170	75% (750/1000)	None	Every 3 years	<a href="https://aws.amazon.com/cer">https://aws.amazon.com/cer</a>
12	AWS Certified Security - Specialty	Specialty	5+ years security	None	\$300	\$200	65	170	75% (750/1000)	None	Every 3 years	<a href="https://aws.amazon.com/cer">https://aws.amazon.com/cer</a>
13	AWS Certified Machine Learning - Specialty	Specialty	2+ years ML workload	None	\$300	\$200	65	180	75% (750/1000)	None	Every 3 years	<a href="https://aws.amazon.com/cer">https://aws.amazon.com/cer</a>
14												

Comment **#awsyatri** to get sheet in your DM



Connect is Required to  
Enable DM



# Follow me on LinkedIn

→ to stay updated

**Follow Now**