AWS Certified Cloud Practitioner

Comprehensive Study Guide

Exam Code: CLF-C02

AWS Certification Preparation

December 20, 2025

Contents

Chapter 1

Introduction

1.1 About This Guide

This comprehensive study guide is designed to help you prepare for and pass the **AWS Certified Cloud Practitioner (CLF-C02)** exam. This certification validates your overall understanding of the AWS Cloud, independent of specific technical roles.

1.2 Certification Overview

- **Exam Code:** CLF-C02
- **Duration:** 90 minutes
- **Question Format:** 65 questions (multiple choice and multiple response)
- **Passing Score:** 700 out of 1000
- **Cost:** $100 USD
- **Validity:** 3 years
- **Delivery:** Pearson VUE testing center or online proctored

1.3 Exam Domain Breakdown

Domain	**Percentage**
Domain 1: Cloud Concepts	24%
Domain 2: Security and Compliance	30%
Domain 3: Cloud Technology and Services	34%
Domain 4: Billing, Pricing, and Support	12%

Table 1.1: AWS Cloud Practitioner Exam Domains

1.4 Target Audience

This certification is ideal for:

- Individuals new to AWS Cloud
- Sales and marketing professionals
- Business analysts and project managers
- IT professionals transitioning to cloud
- Students and recent graduates
- Anyone seeking foundational AWS knowledge

Exam Tip

No technical prerequisites are required, but 6 months of exposure to AWS Cloud is recommended for success.

Chapter 2

Domain 1: Cloud Concepts (24%)

2.1 What is Cloud Computing?

2.1.1 Definition

Cloud computing is the **on-demand delivery** of IT resources over the Internet with **pay-as-you-go pricing**. Instead of buying, owning, and maintaining physical data centers and servers, you access technology services on an as-needed basis.

2.1.2 Six Advantages of Cloud Computing

1. **Trade capital expense for variable expense**
    - Pay only for what you consume
    - No upfront infrastructure costs
    - Lower Total Cost of Ownership (TCO)
2. **Benefit from massive economies of scale**
    - AWS achieves higher economies of scale
    - Lower pay-as-you-go prices
    - Prices decrease over time
3. **Stop guessing capacity**
    - Scale up or down based on demand
    - No over or under provisioning
    - Elastic resources
4. **Increase speed and agility**
    - Resources available in minutes
    - Faster experimentation and innovation
    - Reduced time to market
5. **Stop spending money running and maintaining data centers**

  - Focus on business differentiators
  - AWS manages infrastructure
  - Reduce operational burden

6. **Go global in minutes**

  - Deploy applications globally
  - Low latency for users worldwide
  - Multiple AWS Regions available

2.2 Cloud Computing Models

2.2.1 Infrastructure as a Service (IaaS)

- Basic building blocks for cloud IT
- Highest level of flexibility and control
- Example: Amazon EC2, Amazon S3
- You manage: OS, applications, data
- AWS manages: Hardware, networking, facilities

2.2.2 Platform as a Service (PaaS)

- Removes need to manage underlying infrastructure
- Focus on deployment and management of applications
- Example: AWS Elastic Beanstalk, AWS Lambda
- You manage: Applications, data
- AWS manages: Runtime, middleware, OS, servers

2.2.3 Software as a Service (SaaS)

- Completed product run and managed by service provider
- End-user applications
- Example: Amazon WorkMail, Amazon Chime
- You manage: User access, data input
- AWS manages: Everything else

2.3 Cloud Deployment Models

2.3.1 Cloud (Public Cloud)

- Fully deployed in the cloud
- All parts of application run in cloud
- Applications built on cloud or migrated
- Can be built on low-level infrastructure or higher-level services

2.3.2 Hybrid

- Connects cloud resources to on-premises infrastructure
- Integrates cloud with existing infrastructure
- Useful for legacy applications
- Common deployment model for many enterprises
- Uses AWS Direct Connect, VPN

2.3.3 On-Premises (Private Cloud)

- Resources deployed using virtualization and resource management tools
- Sometimes called ”private cloud”
- Uses AWS Outposts for on-premises AWS infrastructure
- Increased resource utilization

2.4 AWS Well-Architected Framework

The AWS Well-Architected Framework describes key concepts, design principles, and architectural best practices for designing and running workloads in the cloud.

2.4.1 Six Pillars

1. Operational Excellence

- Run and monitor systems to deliver business value
- Continually improve supporting processes and procedures
- Key principles: Perform operations as code, annotate documentation, make frequent small reversible changes
- Services: AWS CloudFormation, AWS Config, AWS CloudTrail, Amazon CloudWatch

2. **Security**
   - Protect information, systems, and assets
   - Key principles: Implement strong identity foundation, enable traceability, apply security at all layers
   - Services: AWS IAM, AWS Organizations, AWS KMS, AWS Shield, Amazon GuardDuty

3. **Reliability**
   - Ensure workload performs its intended function correctly and consistently
   - Recover from failures and dynamically acquire computing resources
   - Key principles: Automatically recover from failure, test recovery procedures, scale horizontally
   - Services: Amazon RDS Multi-AZ, AWS Auto Scaling, Amazon CloudWatch

4. **Performance Efficiency**
   - Use computing resources efficiently to meet requirements
   - Maintain efficiency as demand changes and technologies evolve
   - Key principles: Democratize advanced technologies, go global in minutes, use serverless architectures
   - Services: AWS Lambda, Amazon EBS, Amazon RDS, AWS Auto Scaling

5. **Cost Optimization**
   - Run systems to deliver business value at lowest price point
   - Key principles: Implement cloud financial management, adopt consumption model, measure overall efficiency
   - Services: AWS Cost Explorer, AWS Budgets, Reserved Instances, Savings Plans

6. **Sustainability**
   - Minimize environmental impacts of running cloud workloads
   - Key principles: Understand your impact, establish sustainability goals, maximize utilization
   - Services: Amazon EC2 Auto Scaling, AWS Lambda, Amazon S3 Intelligent-Tiering

Key Point

The Well-Architected Framework is frequently tested on the exam. Understand each pillar's purpose and key services.

2.5 Cloud Economics

2.5.1 Total Cost of Ownership (TCO)

- Financial estimate to identify direct and indirect costs
- Compare on-premises vs. cloud costs
- Includes: Server costs, storage costs, network costs, IT labor costs
- AWS TCO Calculator helps estimate savings

2.5.2 Capital Expenditure (CapEx) vs. Operational Expenditure (OpEx)

CapEx (On-Premises):

- Upfront purchase of physical infrastructure
- Fixed, sunk cost
- Depreciates over time
- Requires capacity planning

OpEx (Cloud):

- Pay for what you use
- Variable cost based on consumption
- No upfront commitment
- Scales with business needs

2.5.3 Migration Strategies (6 R's)

1. **Rehosting (Lift and Shift)**
   - Move applications without changes
   - Fastest migration approach
   - Optimize after migration
2. **Replatforming (Lift, Tinker, and Shift)**
   - Make a few cloud optimizations
   - Don't change core architecture
   - Example: Migrate database to RDS
3. **Repurchasing**
   - Move to a different product

- Often SaaS platforms
- Example: CRM to Salesforce

4. **Refactoring/Re-architecting**

   - Reimagine how application is architected
   - Use cloud-native features
   - Most expensive but highest benefit

5. **Retire**

   - Identify IT assets that are no longer useful
   - Shut down and remove from portfolio

6. **Retain**

   - Keep applications on-premises
   - Not ready to migrate
   - Hybrid deployment

Chapter 3

Domain 2: Security and Compliance (30%)

3.1 AWS Shared Responsibility Model

Important

This is one of the most critical concepts for the exam. Understand what AWS manages versus what the customer manages.

3.1.1 AWS Responsibility: Security OF the Cloud

AWS is responsible for protecting the infrastructure that runs all services:

- Physical security of data centers
- Hardware and networking components
- Compute, storage, database, and networking infrastructure
- AWS global infrastructure (Regions, Availability Zones, Edge Locations)
- Managed services (RDS, DynamoDB, etc.)

3.1.2 Customer Responsibility: Security IN the Cloud

Customers are responsible for:

- Customer data
- Platform, applications, Identity and Access Management (IAM)
- Operating system, network, and firewall configuration
- Client-side data encryption and data integrity authentication
- Server-side encryption (file system and/or data)
- Network traffic protection (encryption, integrity, identity)

- Security group configuration
- User access management

3.1.3 Shared Controls

- Patch Management: AWS patches infrastructure; customers patch OS and applications
- Configuration Management: AWS configures infrastructure; customers configure databases and applications
- Awareness and Training: AWS trains employees; customers train their staff

Exam Tip

For any security question, ask: "Who is responsible?" AWS handles the infrastructure; you handle what you put in the cloud.

3.2 AWS Identity and Access Management (IAM)

3.2.1 Core Components

Users

- Individual people or services
- Permanent named operators
- Can have long-term credentials (password, access keys)
- Should represent a physical person or application

Groups

- Collection of users
- Groups cannot be nested
- Users can belong to multiple groups
- Apply policies to groups for easier management

Roles

- Temporary credentials for users, applications, or services
- No username/password or access keys
- Can be assumed by anyone who needs it
- Best practice for EC2 instances accessing AWS services
- Can be used for cross-account access

Policies

- JSON documents defining permissions
- Attached to users, groups, or roles
- Define what actions are allowed or denied
- Follow principle of least privilege

3.2.2 IAM Best Practices

1. **Root account**: Use only for initial setup, then lock it away
    - Enable MFA on root account
    - Do not create access keys for root
    - Create individual IAM users
2. **Principle of Least Privilege**: Grant only permissions required
3. **Use Groups**: Assign permissions to groups, not individual users
4. **Enable MFA**: Especially for privileged users
5. **Use Roles**: For applications running on EC2
6. **Rotate Credentials**: Regularly change passwords and access keys
7. **Remove Unnecessary Credentials**: Delete unused users, roles, and access keys
8. **Use Policy Conditions**: For extra security (IP restrictions, time-based, MFA)

3.2.3 Multi-Factor Authentication (MFA)

MFA adds extra layer of protection:

- Something you know (password)
- Something you have (MFA device)
- MFA device options:
    - Virtual MFA device (Google Authenticator, Authy)
    - Hardware MFA device (YubiKey)
    - SMS text message (not recommended for root)

3.3 Security Services

3.3.1 AWS Organizations

- Centrally manage multiple AWS accounts
- Consolidated billing across all accounts
- Hierarchical grouping of accounts (Organizational Units)
- Service Control Policies (SCPs) for governance
- Automate account creation
- Centralize security and compliance

Key Features:

- Consolidated billing: One bill for all accounts, volume discounts
- SCPs: Control maximum available permissions
- OU structure: Organize accounts by business unit, environment, etc.

3.3.2 AWS Key Management Service (KMS)

- Create and manage cryptographic keys
- Control use of keys across AWS services
- Integrated with most AWS services
- Customer Master Keys (CMKs): AWS managed or customer managed
- Automatic key rotation available
- Audit key usage via CloudTrail

3.3.3 AWS Shield

DDoS protection service

- **AWS Shield Standard**:
  - Automatic protection for all AWS customers
  - No additional cost
  - Protects against common Layer 3/4 attacks
- **AWS Shield Advanced**:
  - $3,000/month per organization
  - Enhanced protection for EC2, ELB, CloudFront, Route 53, Global Accelerator
  - 24/7 access to DDoS Response Team (DRT)
  - Cost protection against usage spikes
  - Real-time attack notifications

3.3.4 Amazon GuardDuty

- Intelligent threat detection service
- Uses machine learning
- Monitors VPC Flow Logs, CloudTrail logs, DNS logs
- Identifies unauthorized or malicious activity
- No software to deploy
- 30-day free trial
- Integrates with EventBridge for automated responses

3.3.5 Amazon Inspector

- Automated security assessment service
- Assesses applications for vulnerabilities
- Checks for exposure, vulnerabilities, deviations from best practices
- Generates detailed security findings
- Prioritized list of security findings
- Supports EC2 instances and container images

3.3.6 AWS WAF (Web Application Firewall)

- Protects web applications from common exploits
- Deployed on CloudFront, Application Load Balancer, API Gateway
- Create custom rules to block attack patterns
- Protects against SQL injection, cross-site scripting (XSS)
- IP-based filtering, geo-blocking
- Rate-based rules to prevent DDoS

3.3.7 Amazon Macie

- Data security and privacy service
- Uses machine learning to discover and protect sensitive data
- Identifies Personally Identifiable Information (PII)
- Monitors S3 buckets
- Provides dashboards and alerts
- Helps meet compliance requirements (GDPR, HIPAA)

3.3.8 AWS Artifact

- On-demand access to AWS compliance reports
- Self-service portal for audit artifacts
- Download AWS security and compliance documents
- Examples: ISO certifications, SOC reports, PCI reports
- No cost
- Support compliance and regulatory requirements

3.4 Compliance

3.4.1 AWS Compliance Programs

AWS complies with numerous compliance programs:

- **HIPAA**: Health Insurance Portability and Accountability Act
- **PCI DSS**: Payment Card Industry Data Security Standard
- **SOC 1, 2, 3**: Service Organization Controls
- **ISO 27001**: Information security management
- **FedRAMP**: Federal Risk and Authorization Management Program
- **GDPR**: General Data Protection Regulation

3.4.2 AWS Config

- Assess, audit, and evaluate configurations
- Continuous monitoring of resource configurations
- Track configuration changes over time
- Compliance auditing and security analysis
- Config Rules: Define desired configurations
- Automated remediation of non-compliant resources

Chapter 4

Domain 3: Cloud Technology and Services (34%)

This is the largest domain, covering core AWS services across compute, storage, networking, databases, and more.

4.1 AWS Global Infrastructure

4.1.1 Regions

- Geographic area with multiple Availability Zones
- Currently 33 Regions worldwide
- Each Region is completely isolated
- Choose Region based on:
  - Compliance requirements
  - Proximity to users (latency)
  - Available services (not all services in all Regions)
  - Pricing (varies by Region)

4.1.2 Availability Zones (AZs)

- One or more discrete data centers
- Each AZ has redundant power, networking, connectivity
- Physically separated within a Region
- Connected with high-bandwidth, low-latency networking
- Minimum of 3 AZs per Region (most have more)
- Deploy resources across multiple AZs for high availability

4.1.3 Edge Locations

- 400+ Edge Locations worldwide
- Used by CloudFront for content caching
- Lower latency for end users
- Also used by Route 53, AWS Shield, AWS WAF
- More locations than Regions

4.1.4 AWS Local Zones

- Extension of Region closer to users
- Single-digit millisecond latency
- For latency-sensitive applications
- Not available everywhere

4.1.5 AWS Wavelength

- Embeds AWS compute and storage in 5G networks
- Ultra-low latency applications
- For mobile edge computing

4.1.6 AWS Outposts

- Fully managed service extending AWS infrastructure to on-premises
- Same AWS APIs, tools, hardware on-premises
- Hybrid cloud deployment
- AWS manages and maintains the infrastructure

Exam Tip

Remember: Regions contain Availability Zones. Edge Locations are separate and used for content delivery.

4.2 Compute Services

4.2.1 Amazon EC2 (Elastic Compute Cloud)

Virtual servers in the cloud

Instance Types

- **General Purpose**: Balanced compute, memory, networking (T3, M5)
- **Compute Optimized**: High-performance processors (C5, C6)
- **Memory Optimized**: Large datasets in memory (R5, X1)
- **Storage Optimized**: High sequential read/write (I3, D2)
- **Accelerated Computing**: GPU, FPGA (P3, G4)

Pricing Models

1. **On-Demand**:
    - Pay per hour or per second
    - No upfront commitment
    - Highest per-hour cost
    - Good for unpredictable workloads
2. **Reserved Instances (RI)**:
    - 1 or 3 year commitment
    - Up to 75% discount vs. On-Demand
    - Standard RI: Most discount, can't change instance type
    - Convertible RI: Less discount, can change instance type
    - Good for steady-state workloads
3. **Savings Plans**:
    - Commit to consistent compute usage (measured in $/hour)
    - 1 or 3 year commitment
    - Up to 72% discount
    - More flexible than Reserved Instances
    - Applies to EC2, Lambda, Fargate
4. **Spot Instances**:
    - Bid on unused EC2 capacity
    - Up to 90% discount
    - Can be interrupted with 2-minute warning
    - Good for fault-tolerant, flexible workloads
    - Not suitable for critical workloads
5. **Dedicated Hosts**:
    - Physical server dedicated to your use

- Most expensive option
- For compliance requirements or licensing
- Socket/core visibility for licensing

6. **Dedicated Instances**:
    - Instances run on hardware dedicated to you
    - May share hardware with other instances in your account
    - Less expensive than Dedicated Hosts

Auto Scaling

- Automatically adjust capacity based on demand
- Scale out (add instances) or scale in (remove instances)
- Define scaling policies:
    - Target tracking: Maintain metric at target (e.g., 50% CPU)
    - Step scaling: Scale based on CloudWatch alarms
    - Scheduled scaling: Predictable load changes
- Improves availability and cost optimization
- Works with Elastic Load Balancing

4.2.2 AWS Lambda

Serverless compute service

- Run code without provisioning servers
- Pay only for compute time consumed
- Automatic scaling
- Supports multiple languages (Python, Node.js, Java, Go, C#, Ruby)
- Maximum execution time: 15 minutes
- Event-driven execution
- Integrated with many AWS services
- Use cases: Data processing, real-time file processing, backends

Benefits:

- No server management
- Continuous scaling
- Subsecond metering
- High availability built-in

4.2.3 Amazon Lightsail

- Easy-to-use cloud platform
- Virtual private servers (VPS)
- Predictable monthly pricing
- Includes compute, storage, networking
- Ideal for simple web applications, blogs, small businesses
- Pre-configured applications (WordPress, Magento)
- Good for beginners or simple workloads

4.2.4 AWS Elastic Beanstalk

- Platform as a Service (PaaS)
- Deploy and manage applications without infrastructure management
- Supports Java, .NET, PHP, Node.js, Python, Ruby, Go, Docker
- Automatically handles capacity provisioning, load balancing, auto-scaling
- You retain full control of underlying resources
- No additional charge (pay for resources used)
- Good for web applications

4.2.5 Amazon ECS (Elastic Container Service)

- Fully managed container orchestration service
- Run Docker containers
- Launch types:
  - EC2: You manage the underlying EC2 instances
  - Fargate: Serverless, AWS manages infrastructure
- Integrates with other AWS services
- Use cases: Microservices, batch processing

4.2.6 Amazon EKS (Elastic Kubernetes Service)

- Fully managed Kubernetes service
- Run Kubernetes without managing control plane
- Compatible with standard Kubernetes tooling
- Integrates with AWS services
- For teams already using Kubernetes

4.2.7 AWS Fargate

- Serverless compute engine for containers
- Works with ECS and EKS
- No need to provision or manage servers
- Pay for resources used by containers
- Focus on application, not infrastructure

Key Point

Remember the compute spectrum: EC2 (most control) → Containers (ECS/EKS) → Lambda (least control, most abstraction)

4.3 Storage Services

4.3.1 Amazon S3 (Simple Storage Service)

Object storage service

Key Concepts

- **Buckets**: Containers for objects, globally unique names
- **Objects**: Files stored in buckets (up to 5 TB)
- **Keys**: Unique identifier for objects
- **Durability**: 99.999999999% (11 nines)
- **Availability**: Varies by storage class

Storage Classes

1. **S3 Standard**:
   - Frequently accessed data
   - Low latency, high throughput
   - 99.99% availability
   - Most expensive
2. **S3 Intelligent-Tiering**:
   - Automatically moves objects between tiers
   - Optimizes costs for unknown access patterns
   - Small monthly monitoring fee
3. **S3 Standard-IA (Infrequent Access)**:

  - Less frequently accessed data
  - Lower storage cost, retrieval fee
  - 99.9% availability
  - Minimum 30-day storage

4. **S3 One Zone-IA**:
   - Single Availability Zone
   - 20% less cost than Standard-IA
   - 99.5% availability
   - For recreatable data

5. **S3 Glacier Instant Retrieval**:
   - Archive data with instant retrieval
   - Millisecond retrieval
   - Minimum 90-day storage

6. **S3 Glacier Flexible Retrieval**:
   - Archive data, retrieval in minutes to hours
   - Expedited (1-5 min), Standard (3-5 hours), Bulk (5-12 hours)
   - Minimum 90-day storage

7. **S3 Glacier Deep Archive**:
   - Lowest cost storage
   - Retrieval time: 12-48 hours
   - Minimum 180-day storage
   - For compliance archives

S3 Features

- **Versioning**: Keep multiple versions of objects
- **Lifecycle Policies**: Automatically transition or delete objects
- **Encryption**: Server-side and client-side encryption
- **Access Control**: Bucket policies, ACLs, IAM policies
- **Static Website Hosting**: Host static websites
- **Cross-Region Replication**: Replicate objects across Regions
- **Transfer Acceleration**: Fast transfer using CloudFront edge locations

4.3.2 Amazon EBS (Elastic Block Store)

Block storage for EC2 instances

- Persistent block storage
- Attached to single EC2 instance
- Automatically replicated within Availability Zone
- Snapshots stored in S3
- Can detach and reattach to different instances

Volume Types

- **General Purpose SSD (gp3, gp2)**: Balanced price/performance
- **Provisioned IOPS SSD (io2, io1)**: High-performance, mission-critical
- **Throughput Optimized HDD (st1)**: Low-cost, frequently accessed
- **Cold HDD (sc1)**: Lowest cost, infrequently accessed

4.3.3 Amazon EFS (Elastic File System)

- Managed NFS (Network File System)
- Shared storage for multiple EC2 instances
- Automatically scales
- Pay for storage used
- Regional service (spans multiple AZs)
- Use cases: Content management, web serving, data sharing

4.3.4 AWS Storage Gateway

- Hybrid cloud storage service
- Connects on-premises to AWS storage
- Three types:
  - File Gateway: NFS/SMB access to S3
  - Volume Gateway: iSCSI block storage
  - Tape Gateway: Virtual tape library
- For backup, disaster recovery, migration

4.3.5 AWS Snow Family

Physical devices for data migration

- **Snowcone**: 8 TB, smallest, portable
- **Snowball Edge**:
  - Storage Optimized: 80 TB
  - Compute Optimized: 42 TB + compute
- **Snowmobile**: 100 PB, shipping container
- Use cases: Large data migrations, edge computing, disaster recovery
- Ships to you, load data, ship back to AWS

Exam Tip

Choose storage based on use case: S3 for object storage, EBS for EC2 block storage, EFS for shared file system.

4.4 Database Services

4.4.1 Amazon RDS (Relational Database Service)

Managed relational database

- Supported engines: MySQL, PostgreSQL, MariaDB, Oracle, SQL Server, Amazon Aurora
- Automated backups, patching, scaling
- Multi-AZ deployment for high availability
- Read replicas for read-heavy workloads
- No access to underlying OS
- Pay for instance type and storage

Multi-AZ Deployments

- Synchronous replication to standby in different AZ
- Automatic failover
- High availability and disaster recovery
- No performance benefit (failover only)

Read Replicas

- Asynchronous replication
- Scale read-heavy workloads
- Can be in different Region
- Can be promoted to primary
- Up to 5 read replicas per primary

4.4.2 Amazon Aurora

- AWS proprietary database
- MySQL and PostgreSQL compatible
- 5x faster than MySQL, 3x faster than PostgreSQL
- Up to 15 read replicas
- Automatic scaling storage (10 GB to 128 TB)
- 6 copies across 3 AZs
- More expensive than RDS but better performance

4.4.3 Amazon DynamoDB

NoSQL database service

- Fully managed, serverless
- Key-value and document database
- Single-digit millisecond latency
- Automatic scaling
- Global tables for multi-region replication
- DynamoDB Accelerator (DAX): In-memory cache
- Pay per request or provisioned capacity
- Use cases: Mobile backends, gaming, IoT

4.4.4 Amazon ElastiCache

In-memory caching service

- Managed Redis or Memcached
- Improve application performance
- Reduce database load
- Microsecond latency
- Common use: Cache database queries, session stores

4.4.5 Amazon Redshift

Data warehouse service

- Petabyte-scale data warehouse
- Columnar storage
- Massively parallel processing
- SQL-based queries
- Integrate with BI tools
- Use cases: Analytics, business intelligence

4.4.6 Amazon Neptune

- Fully managed graph database
- Supports Property Graph and RDF
- Use cases: Social networks, recommendation engines, fraud detection

4.4.7 Amazon DocumentDB

- MongoDB-compatible document database
- Fully managed
- Scale storage and compute independently
- Use cases: Content management, catalogs

Key Point

Database selection: RDS for relational, DynamoDB for NoSQL key-value, Redshift for analytics, ElastiCache for caching.

4.5 Networking and Content Delivery

4.5.1 Amazon VPC (Virtual Private Cloud)

Isolated virtual network

- Your own private network in AWS
- Control IP address range (CIDR blocks)
- Create subnets in different Availability Zones
- Configure route tables, network gateways
- Isolated and secure

Components

- **Subnets**: Segments of VPC IP address range
  - Public subnet: Has route to Internet Gateway
  - Private subnet: No direct internet access
- **Internet Gateway (IGW)**: Allows communication with internet
- **NAT Gateway/Instance**: Allows private subnet instances to access internet
- **Route Tables**: Determine traffic routing
- **Security Groups**: Virtual firewall for instances (stateful)
  - Instance level
  - Allow rules only
  - Stateful: Return traffic automatically allowed
- **Network ACLs**: Subnet-level firewall (stateless)
  - Subnet level
  - Allow and deny rules
  - Stateless: Must explicitly allow return traffic
- **VPC Peering**: Connect two VPCs
- **VPN Gateway**: Connect on-premises to VPC via VPN
- **Direct Connect**: Dedicated network connection from on-premises

4.5.2 Amazon CloudFront

Content Delivery Network (CDN)

- Distribute content globally with low latency
- 400+ Edge Locations worldwide
- Cache content closer to users
- Integrates with S3, EC2, ELB, Route 53
- DDoS protection via AWS Shield
- Use cases: Static/dynamic content, video streaming, API acceleration

4.5.3 Amazon Route 53

DNS web service

- Domain name registration
- DNS routing
- Health checking
- Highly available and scalable
- Routing policies:
  - Simple: Single resource
  - Weighted: Distribute traffic across resources
  - Latency: Route to lowest latency
  - Failover: Active-passive failover
  - Geolocation: Route based on user location
  - Geoproximity: Route based on resource location

4.5.4 Elastic Load Balancing (ELB)

Distribute traffic across targets

Types

- **Application Load Balancer (ALB)**:
  - Layer 7 (HTTP/HTTPS)
  - Advanced routing (path, host-based)
  - Best for web applications
- **Network Load Balancer (NLB)**:

  - Layer 4 (TCP/UDP)
  - Extreme performance, low latency
  - Static IP addresses

- **Gateway Load Balancer**:
  - Layer 3 (IP)
  - For third-party virtual appliances

- **Classic Load Balancer**:
  - Previous generation
  - Layer 4 and 7
  - Being phased out

4.5.5 AWS Direct Connect

- Dedicated private connection from on-premises to AWS
- Bypass internet
- Consistent network performance
- Reduce bandwidth costs
- Access public and private services
- Takes weeks/months to provision

4.5.6 AWS VPN

- Site-to-Site VPN: Connect on-premises to AWS VPC over internet
- Client VPN: Connect individual users to AWS or on-premises
- Encrypted connection
- Quick to set up (minutes)
- Lower cost than Direct Connect

4.6 Management and Governance

4.6.1 AWS CloudFormation

- Infrastructure as Code (IaC)
- Define resources in templates (JSON or YAML)
- Automate infrastructure provisioning

- Version control infrastructure
- Consistent deployments
- No additional charge (pay for resources created)

4.6.2 AWS CloudTrail

- Governance, compliance, auditing
- Log all API calls in AWS account
- Who did what, when, and where
- Enabled by default (90 days)
- Store logs in S3 for long-term retention
- Integrate with CloudWatch Logs
- Essential for security and compliance

4.6.3 Amazon CloudWatch

Monitoring and observability

- Collect and track metrics
- Collect and monitor log files
- Set alarms
- Automatically react to changes
- Components:
  - CloudWatch Metrics: Numerical data points
  - CloudWatch Logs: Log collection and analysis
  - CloudWatch Alarms: Trigger actions based on metrics
  - CloudWatch Events/EventBridge: Event-driven automation
- Built-in metrics for most AWS services
- Custom metrics for applications

4.6.4 AWS Systems Manager

- Operational hub for AWS resources
- View operational data
- Automate tasks
- Patch management
- Run commands on multiple instances
- Session Manager: Secure shell access without SSH keys
- Parameter Store: Centralized secret and configuration management

4.6.5 AWS Trusted Advisor

Best practice recommendations

- Real-time guidance to provision resources
- Five categories:
  1. Cost Optimization: Reduce costs
  2. Performance: Improve performance
  3. Security: Close security gaps
  4. Fault Tolerance: Increase availability
  5. Service Limits: Check service quotas
- Free checks: 7 core checks
- Business/Enterprise Support: All checks + notifications
- Dashboard with action items

4.6.6 AWS Control Tower

- Set up and govern multi-account AWS environment
- Automated setup of landing zone
- Guardrails for governance
- Account Factory for account creation
- Built on AWS Organizations, CloudFormation, other services

4.6.7 AWS Service Catalog

- Create and manage catalogs of approved IT services
- Control which services users can deploy
- Standardize deployments
- Centralized management

4.7 Additional Services

4.7.1 Amazon SNS (Simple Notification Service)

- Pub/sub messaging service
- Send notifications to multiple subscribers
- Push notifications to mobile devices
- Email, SMS, HTTP endpoints
- Decoupled architecture
- Topics and subscriptions

4.7.2 Amazon SQS (Simple Queue Service)

- Fully managed message queue
- Decouple application components
- Standard queues: At-least-once delivery, best-effort ordering
- FIFO queues: Exactly-once processing, strict ordering
- Messages retained up to 14 days
- Unlimited throughput

4.7.3 AWS Step Functions

- Orchestrate workflows
- Visual workflow designer
- Coordinate multiple AWS services
- Serverless workflow service
- State machines for complex workflows

4.7.4 Amazon EventBridge

- Serverless event bus
- Connect applications using events
- Route events between AWS services, SaaS applications
- Previously called CloudWatch Events

4.7.5 AWS Batch

- Fully managed batch processing
- Run batch computing workloads
- Automatically provisions compute resources
- Dynamically scales based on volume
- Use cases: Data processing, rendering

4.7.6 Amazon Athena

- Interactive query service
- Analyze data in S3 using SQL
- Serverless
- Pay per query
- Use cases: Log analysis, business intelligence

4.7.7 AWS Glue

- Fully managed ETL (Extract, Transform, Load) service
- Prepare data for analytics
- Serverless
- Data catalog and metadata management

4.7.8 Amazon QuickSight

- Business intelligence service
- Create visualizations and dashboards
- Machine learning insights
- Serverless
- Pay per session

4.7.9 Amazon Kinesis

- Real-time data streaming
- Collect, process, analyze streaming data
- Services:
  - Kinesis Data Streams: Capture and store data streams
  - Kinesis Data Firehose: Load data into AWS data stores
  - Kinesis Data Analytics: Analyze streams with SQL
  - Kinesis Video Streams: Process video streams

4.7.10 Amazon SageMaker

- Build, train, deploy machine learning models
- Fully managed ML service
- Integrated Jupyter notebooks
- Pre-built algorithms
- One-click training and deployment

4.7.11 Amazon Rekognition

- Image and video analysis
- Facial analysis and recognition
- Object and scene detection
- Content moderation
- Machine learning-powered

4.7.12 Amazon Comprehend

- Natural language processing (NLP)
- Extract insights from text
- Sentiment analysis, entity recognition
- Topic modeling

4.7.13 Amazon Lex

- Build conversational interfaces (chatbots)
- Same technology as Amazon Alexa
- Automatic speech recognition (ASR)
- Natural language understanding (NLU)

4.7.14 AWS Migration Hub

- Track application migrations
- Single location to monitor migrations
- Integrate with migration tools
- Visualize migration progress

4.7.15 AWS Database Migration Service (DMS)

- Migrate databases to AWS
- Source database remains operational
- Supports homogeneous and heterogeneous migrations
- Continuous data replication

4.7.16 AWS Application Discovery Service

- Discover on-premises applications
- Plan migrations
- Collect server utilization and dependency data

Chapter 5

Domain 4: Billing, Pricing, and Support (12%)

5.1 AWS Pricing Fundamentals

5.1.1 Core Principles

- **Pay-as-you-go**: Pay only for what you use
- **Pay less when you reserve**: Reserved capacity discounts
- **Pay less with volume-based discounts**: Use more, pay less per unit
- **No upfront costs**: No capital expenditure
- **No termination fees**: Stop anytime

5.1.2 Free Tier

Three types:

1. **Always Free**: Never expire
   - DynamoDB: 25 GB storage
   - Lambda: 1 million requests/month
   - SNS: 1 million publishes
2. **12 Months Free**: Starting from account creation
   - EC2: 750 hours/month of t2.micro or t3.micro
   - S3: 5 GB standard storage
   - RDS: 750 hours/month of db.t2.micro
3. **Trials**: Short-term free trials
   - SageMaker: 2 months
   - Inspector: 90 days
   - Lightsail: 1 month

5.2 Pricing Models by Service Category

5.2.1 Compute Pricing

EC2:

- Pay for instance hours
- Varies by instance type, Region
- Additional charges: Data transfer, storage

Lambda:

- Pay for requests and compute time
- $0.20 per 1 million requests
- Compute time charged per GB-second
- 1 million free requests/month (always free)

5.2.2 Storage Pricing

S3:

- Pay for storage used (per GB/month)
- Request pricing (PUT, GET, etc.)
- Data transfer out
- Varies by storage class

EBS:

- Pay for provisioned storage (per GB/month)
- Snapshot storage (incremental)
- Varies by volume type

5.2.3 Database Pricing

RDS:

- Instance hours
- Storage (per GB/month)
- Backup storage
- Data transfer

DynamoDB:

- On-Demand: Pay per request
- Provisioned: Pay for read/write capacity units
- Storage (per GB/month)

5.2.4 Network Pricing

- Data transfer IN: Generally free
- Data transfer OUT to internet: Charged (tiered pricing)
- Data transfer between Regions: Charged
- Data transfer within same Region: Free or low cost
- CloudFront data transfer out: Lower than direct from services

5.3 Cost Management Tools

5.3.1 AWS Pricing Calculator

- Estimate monthly AWS costs
- Configure service specifications
- Create cost estimates for solutions
- Share estimates with stakeholders
- Free to use
- No account required

5.3.2 AWS Cost Explorer

- Visualize and analyze costs
- View up to 12 months of historical data
- Forecast future costs
- Filter by service, Region, tag, etc.
- Identify cost trends
- Default reports and custom reports
- Free, but API access costs extra

5.3.3 AWS Budgets

- Set custom cost and usage budgets
- Alert when exceeding thresholds
- Types:
  - Cost budgets
  - Usage budgets

  - Reservation budgets
  - Savings Plans budgets

- Send notifications via SNS
- First two budgets free, $0.02/day per additional budget

5.3.4 AWS Cost and Usage Report

- Most comprehensive cost data
- Detailed breakdown of usage and costs
- Delivered to S3 bucket
- Integrate with Athena, Redshift, QuickSight
- Hourly, daily, or monthly reports
- Free (pay for S3 storage)

5.3.5 AWS Cost Anomaly Detection

- Uses machine learning to detect unusual spending
- Automatically identifies anomalies
- Sends alerts
- Root cause analysis
- No additional cost

5.4 Consolidated Billing and AWS Organizations

5.4.1 Consolidated Billing

- One bill for all accounts in organization
- Combine usage across accounts for volume discounts
- Free feature of AWS Organizations
- Track charges by account
- No additional charge

Benefits:

- Volume pricing discounts
- Single payment method
- Easy tracking
- Free tier applies once per organization

5.5 AWS Support Plans

Feature	Basic	Developer	Business	Enterprise
Cost	Free	$29/month or 3% of monthly AWS usage	$100/month or 10% (3-10% tiered)	$15,000/month or 10% (3-10% tiered)
Use Case	All customers	Testing and development	Production workloads	Mission-critical workloads
Technical Support	None	Business hours via email	24/7 via email, chat, phone	24/7 via email, chat, phone
Response Time - General	N/A	< 24 hours	< 24 hours	< 24 hours
Response Time - System Impaired	N/A	< 12 hours	< 12 hours	< 12 hours
Response Time - Production Down	N/A	N/A	< 4 hours	< 4 hours
Response Time - Business Critical	N/A	N/A	< 1 hour	< 1 hour
Response Time - Mission Critical	N/A	N/A	N/A	< 15 minutes
Who Can Open Cases	N/A	1 primary contact	Unlimited contacts	Unlimited contacts
Trusted Advisor Checks	7 core checks	7 core checks	All checks	All checks
Third-Party Software Support	No	No	Yes	Yes
Architectural Guidance	No	General	Contextual to use case	Consultative
Technical Account Manager (TAM)	No	No	No	Yes

Feature	Basic	Developer	Business	Enterprise
Proactive Programs	No	No	No	Yes (Infrastructure Event Management, Well-Architected Reviews)
Concierge Support Team	No	No	No	Yes

Table 5.1: AWS Support Plan Comparison

Important

Memorize the support plans, especially response times and which plan includes TAM (Enterprise only).

5.5.1 Additional Support Resources

AWS Personal Health Dashboard

- Personalized view of service health
- Alerts for events impacting your resources
- Proactive notifications
- Detailed remediation guidance
- Available to all customers

AWS Health API

- Programmatic access to health information
- Requires Business or Enterprise Support
- Integrate with monitoring systems

AWS Managed Services (AMS)

- AWS operates infrastructure on your behalf
- 24/7 operations
- Incident management
- Patching, backup, monitoring
- Separate service with additional cost

AWS Professional Services

- Global team of experts
- Help design, architect, build, migrate
- Work alongside your team
- Consulting services (additional cost)

AWS Partner Network (APN)

- Global community of partners
- Consulting Partners: Professional services
- Technology Partners: Software solutions
- AWS Marketplace: Software and services

5.6 Cost Optimization Strategies

5.6.1 Right Sizing

- Match instance types to workload requirements
- Use CloudWatch metrics to identify underutilized resources
- Use AWS Compute Optimizer recommendations
- Downsize or change instance families

5.6.2 Reserved Capacity

- Reserved Instances for EC2, RDS, ElastiCache, Redshift
- Savings Plans for flexible commitment
- Up to 75% savings
- Analyze usage patterns before purchasing

5.6.3 Spot Instances

- Up to 90% discount
- For fault-tolerant, flexible workloads
- Batch jobs, big data, containerized workloads

5.6.4 Auto Scaling

- Scale resources based on demand
- Avoid over-provisioning
- Reduce costs during low-demand periods

5.6.5 Storage Optimization

- Use appropriate S3 storage classes
- Implement lifecycle policies
- Delete unused snapshots and volumes
- Use S3 Intelligent-Tiering for unknown patterns

5.6.6 Data Transfer Optimization

- Use CloudFront to reduce data transfer costs
- Keep data in same Region when possible
- Use VPC endpoints for S3 and DynamoDB
- Compress data before transfer

Chapter 6

Practice Questions and Exam Tips

6.1 Sample Questions by Domain

6.1.1 Domain 1: Cloud Concepts

Question 1: Which of the following are advantages of cloud computing? (Select TWO)

A. Trade capital expense for variable expense

B. Increase time to market

C. Benefit from massive economies of scale

D. Maintain infrastructure

Answer: A and C

Question 2: Which migration strategy involves moving applications to the cloud without making changes?

A. Repurchasing

B. Rehosting

C. Refactoring

D. Replatforming

Answer: B (Lift and Shift)

6.1.2 Domain 2: Security and Compliance

Question 3: According to the Shared Responsibility Model, which security aspect is AWS responsible for?

A. Security group configuration

B. Physical security of data centers

C. Customer data encryption

D. IAM user management

Answer: B

Question 4: Which service provides DDoS protection at no additional cost?

A. AWS WAF

B. AWS Shield Advanced

C. AWS Shield Standard

D. Amazon GuardDuty

Answer: C

6.1.3 Domain 3: Technology and Services

Question 5: Which compute service allows you to run code without provisioning servers?

A. Amazon EC2

B. AWS Lambda

C. Amazon ECS

D. AWS Elastic Beanstalk

Answer: B

Question 6: Which storage class is most cost-effective for infrequently accessed data that needs millisecond retrieval?

A. S3 Standard

B. S3 Glacier Instant Retrieval

C. S3 Standard-IA

D. S3 Glacier Deep Archive

Answer: C

6.1.4 Domain 4: Billing and Pricing

Question 7: Which AWS Support plan provides a Technical Account Manager?

A. Basic

B. Developer

C. Business

D. Enterprise

Answer: D

Question 8: Which service helps you create cost estimates for AWS solutions?

A. AWS Cost Explorer

B. AWS Pricing Calculator

C. AWS Budgets

D. AWS Cost and Usage Report

Answer: B

6.2 Test-Taking Strategies

6.2.1 Before the Exam

1. **Review all exam objectives**: Ensure you've covered each domain
2. **Take practice exams**: Identify weak areas
3. **Get hands-on experience**: Create AWS account, experiment with services
4. **Review AWS documentation**: Especially FAQs for key services
5. **Get adequate rest**: Sleep well before exam day

6.2.2 During the Exam

1. **Read questions carefully**: Pay attention to keywords
   - ”MOST cost-effective”
   - ”BEST”
   - ”LEAST amount of effort”
   - ”NOT” or ”EXCEPT”
2. **Eliminate wrong answers**: Narrow down choices
3. **Watch for absolutes**: ”Always,” ”never,” ”all,” ”none” are often wrong
4. **Flag difficult questions**: Return to them later
5. **Manage time**: Don't spend too long on one question
6. **Trust your preparation**: Your first instinct is often correct
7. **Use process of elimination**: Remove obviously incorrect answers
8. **Review flagged questions**: Use remaining time to review

6.2.3 Common Question Patterns

- **Scenario-based**: Describe situation, ask for best solution
- **Select TWO/THREE**: Multiple correct answers
- **Best practice**: What's the recommended approach?
- **Cost optimization**: Which option is most cost-effective?
- **Security**: Which option is most secure?
- **High availability**: Which design ensures uptime?

6.3 Common Pitfalls to Avoid

1. **Confusing service names**: EC2 vs. ECS vs. EKS; S3 vs. EBS vs. EFS
2. **Not understanding Shared Responsibility**: Know what AWS vs. customer manages
3. **Mixing up support plans**: Especially response times and TAM
4. **Forgetting storage classes**: S3 storage tiers and use cases
5. **Confusing pricing models**: On-Demand vs. Reserved vs. Spot
6. **Not knowing AWS global infrastructure**: Regions, AZs, Edge Locations
7. **Overlooking "EXCEPT" questions**: Read carefully
8. **Assuming real-world complexity**: Choose AWS-recommended simple solution

6.4 Key Concepts to Memorize

Exam Tip

Focus on understanding concepts rather than memorization, but these facts appear frequently on the exam.

6.4.1 Numbers to Remember

- S3 durability: 99.999999999% (11 nines)
- Minimum AZs per Region: 3
- Edge Locations: 400+
- Lambda max execution: 15 minutes
- RDS read replicas: Up to 5
- Support plan response times
- Free tier: 12 months for EC2, S3, RDS

6.4.2 Service Comparisons

- S3 vs. EBS vs. EFS
- RDS vs. DynamoDB vs. Redshift
- EC2 vs. Lambda vs. Elastic Beanstalk
- CloudWatch vs. CloudTrail vs. Config
- SNS vs. SQS
- Security Groups vs. NACLs
- ALB vs. NLB

Chapter 7

Comprehensive Service Comparisons and Decision Trees

7.1 Storage Services Detailed Comparison

Feature	**Amazon S3**	**Amazon EBS**	**Amazon EFS**	**Instance Store**
Type	Object Storage	Block Storage	File Storage	Ephemeral Block
Use Case	Backups, archives, web content, data lakes	Boot volumes, databases, transactional data	Shared file systems, content management	Temporary data, caches, buffers
Access	HTTP/S API, SDK, CLI	Attached to EC2 instance	NFSv4 protocol, multiple EC2 instances	Direct attached to EC2
Durability	11 nines (99.999999999%)	Replicated within AZ	Replicated across AZs in Region	Lost when instance stops
Scalability	Unlimited	Up to 64 TB per volume	Petabyte scale	Fixed to instance type
Performance	Varies by class	Up to 256,000 IOPS	Scales with size	Highest IOPS for instance
Cost	Low per GB, varies by class	$0.08-0.125/GB-month	$0.30/GB-month	Included with instance
Availability	99.9-99.99% SLA	99.8-99.9%	99.99%	Dependent on instance
Backup	Versioning, lifecycle	Snapshots to S3	AWS Backup	Not persistent
Multi-AZ	Yes	No (single AZ)	Yes	No
Concurrent Access	Unlimited	Single EC2 instance	Thousands of instances	Single instance

Table 7.1: Storage Services Comparison

7.2 Database Services Detailed Comparison

Service	Type	Use Case	Scaling	Pricing Model	Managed
RDS	Relational (SQL)	Traditional apps, OLTP	Vertical, Read Replicas	Instance + storage	Fully managed
Aurora	Relational (MySQL/PostgreSQL compatible)	High-performance OLTP	Auto-scaling storage	Serverless or provisioned	Fully managed
DynamoDB	NoSQL Key-Value	Web/mobile apps, gaming, IoT	Auto horizontal	On-demand or provisioned	Fully managed
Redshift	Data Warehouse (OLAP)	Analytics, BI, big data	Add nodes	Nodes + storage	Fully managed
ElastiCache	In-memory cache	Session stores, leaderboards	Add nodes	Nodes (hourly)	Fully managed
Neptune	Graph database	Social networks, recommendations	Vertical	Instances	Fully managed
DocumentDB	Document (MongoDB compatible)	Content management, catalogs	Vertical, replicas	Instances	Fully managed
Timestream	Time series	IoT, DevOps metrics	Auto	Storage + queries	Fully managed

Table 7.2: Database Services Comparison

7.3 Compute Services Decision Tree

Choose Your Compute Service:

- **Need full control over OS and configuration?**
  - Yes → Use **EC2**
  - Want to save costs for predictable workloads? → Use **Reserved Instances**
  - Fault-tolerant batch workloads? → Use **Spot Instances**
- **Want to run code without managing servers?**
  - Event-driven, ¡ 15 min execution → Use **Lambda**
  - Long-running, stateless → Use **Fargate**
- **Need to deploy web applications quickly?**

  - Simple deployment, don't want infrastructure management → Use **Elastic Beanstalk**
  - Need predictable pricing for small projects → Use **Lightsail**

- **Using containers?**
  - Want AWS-native orchestration → Use **ECS**
  - Need Kubernetes compatibility → Use **EKS**
  - Don't want to manage servers → Use **Fargate** (with ECS or EKS)

- **Running batch processing jobs?**
  - Use **AWS Batch**

7.4 Networking Components Deep Dive

Component	**Purpose**	**Key Points**
Internet Gateway (IGW)	Connect VPC to internet	One per VPC; enables internet access for public subnets
NAT Gateway	Outbound internet from private subnets	Highly available; placed in public subnet; charged per hour + data
NAT Instance	Alternative to NAT Gateway	EC2 instance; you manage; lower cost but less reliable
VPC Peering	Connect two VPCs	Non-transitive; can be cross-account/region; no overlapping CIDRs
Transit Gateway	Hub for connecting VPCs	Simplifies complex network topologies; central management
VPN Gateway	VPN connection to on-premises	IPsec VPN; encrypted over internet; quick setup
Direct Connect	Dedicated connection to on-premises	Private, consistent bandwidth; expensive; takes weeks to provision
VPC Endpoints	Private connection to AWS services	No internet required; Interface or Gateway endpoints; reduce costs
PrivateLink	Private connectivity to services	Access services in other VPCs; doesn't require VPC peering

Table 7.3: VPC Components

7.5 Load Balancer Detailed Comparison

Feature	**ALB**	**NLB**	**Gateway LB**
OSI Layer	Layer 7 (Application)	Layer 4 (Transport)	Layer 3 (Network)
Protocol	HTTP, HTTPS, WebSocket	TCP, UDP, TLS	IP
Routing	Path-based, host-based, query string	IP address, port	N/A
Use Case	Web applications, microservices	High performance, low latency, static IP	Third-party appliances
Target Types	IP, instance, Lambda	IP, instance, ALB	IP, instance
Performance	Good	Extreme (millions req/sec)	High
Static IP	No (DNS only)	Yes (Elastic IP)	N/A
SSL Termination	Yes	Yes	No
WebSocket	Yes	Yes	No
Health Checks	Advanced	Basic	Advanced
Pricing	Per hour + LCU	Per hour + LCU	Per hour + LCU

Table 7.4: Load Balancer Types Comparison

7.6 Security Services Complete Matrix

Service	**What It Does**	**When to Use**
IAM	Identity and access management	Control who can access what in AWS
AWS Organizations	Multi-account management	Centralize billing, apply policies across accounts
AWS SSO	Single sign-on	Centrally manage access to multiple accounts and applications
Cognito	User authentication for apps	Add sign-up/sign-in to mobile and web apps
Directory Service	Managed Active Directory	Integrate AWS with existing Microsoft AD
Secrets Manager	Store and rotate secrets	Automatically rotate database credentials
KMS	Encryption key management	Create and control encryption keys
CloudHSM	Hardware security modules	Dedicated hardware for regulatory compliance
Certificate Manager	SSL/TLS certificates	Free certificates for ELB, CloudFront, API Gateway
WAF	Web application firewall	Protect against SQL injection, XSS attacks
Shield Standard	DDoS protection	Automatic protection (free)

Service	What It Does	When to Use
Shield Advanced	Enhanced DDoS protection	24/7 DDoS Response Team, cost protection ($3,000/month)
GuardDuty	Threat detection	Continuous monitoring for malicious activity
Inspector	Vulnerability assessment	Scan EC2 and container images for vulnerabilities
Macie	Data privacy and protection	Discover and protect sensitive data in S3
Detective	Security investigation	Analyze and investigate security issues
Security Hub	Security posture management	Centralized view of security alerts and compliance
Firewall Manager	Centralized firewall management	Manage WAF, Shield across accounts

Table 7.5: Security Services Matrix

7.7 Service Limits Quick Reference

Service	Default Limit	Notes
EC2 Instances (On-Demand)	20 per region	Can request increase
VPCs per Region	5	Can request increase
Internet Gateways per Region	5	One per VPC typically
S3 Buckets per Account	100	Soft limit, can increase
S3 Object Size	5 TB max	Use multipart upload for ¿ 100 MB
RDS DB Instances	40 per region	Can request increase
Lambda Concurrent Executions	1,000	Can request increase
Lambda Function Timeout	15 minutes max	Cannot be increased
CloudFormation Stacks	200 per region	Can request increase
IAM Users per Account	5,000	Use roles/federated identities instead
IAM Groups per Account	300	Plan group structure carefully

Table 7.6: Service Limits Quick Reference

Chapter 8

Common Exam Scenarios and Real-World Solutions

8.1 Scenario-Based Learning

8.1.1 Scenario 1: Cost Optimization for Predictable Workloads

Situation: A company runs a web application on EC2 instances that experiences predictable traffic Monday-Friday 9 AM-5 PM EST. Traffic is minimal on weekends and nights.

Current Setup:

- 10 m5.large instances running 24/7
- On-Demand pricing
- Monthly cost: $1,200

Question: What's the MOST cost-effective solution?
Analysis:

- Predictable schedule = opportunity for optimization
- Not running 24/7 = On-Demand might be wasteful
- Regular business hours = scheduled scaling
- Baseline capacity needed = Reserved Instances candidate

Recommended Solution:

1. Purchase **3-year Standard Reserved Instances** for 2-3 instances (baseline capacity)
    - Savings: Up to 75% on these instances
2. Configure **EC2 Auto Scaling with scheduled actions**:
    - Scale up Monday-Friday 8:30 AM EST (before traffic starts)
    - Scale down at 5:30 PM EST (after traffic ends)

  - Minimum capacity on weekends: 2-3 instances

3. Use **On-Demand for peak periods** during business hours
4. Store session data in **ElastiCache or DynamoDB** (not on instances)

Expected Savings: 40-60% reduction in monthly costs

8.1.2 Scenario 2: Designing for High Availability

Situation: An e-commerce company's application must remain available even if an entire Availability Zone fails. The application currently runs on a single EC2 instance with a MySQL database.

Question: How should you architect this for high availability?

Current Problems:

- Single point of failure (one EC2 instance)
- Database not redundant
- No automatic failover
- Session data tied to instance

Recommended Solution:

1. **Multi-AZ Application Tier**:
   - Deploy **Application Load Balancer** spanning multiple AZs
   - Create **Auto Scaling group** with minimum 2 instances across different AZs
   - Set desired capacity based on traffic patterns
   - Configure health checks on ALB and Auto Scaling
2. **Multi-AZ Database**:
   - Migrate MySQL to **Amazon RDS Multi-AZ**
   - Automatic failover to standby in different AZ
   - Synchronous replication
   - Minimal downtime during failover
3. **Stateless Application Design**:
   - Store session data in **ElastiCache** (Redis with Multi-AZ)
   - Or use **DynamoDB** for session storage
   - Enable sticky sessions on ALB if needed (but prefer stateless)
4. **Static Assets**:
   - Store in **S3** (automatically multi-AZ)
   - Use **CloudFront** for global distribution

5. **Monitoring**:
    - Set up **CloudWatch alarms** for health checks
    - Configure **SNS notifications** for failures

Architecture Benefits:

- Survives AZ failure
- Automatic scaling for traffic spikes
- Automatic failover for database
- No single point of failure

8.1.3 Scenario 3: Large Data Migration

Situation: A healthcare company needs to migrate 80 TB of medical imaging data from on-premises storage to S3. Compliance requires data to be encrypted and migration completed within 2 weeks.

Constraints:

- Internet connection: 100 Mbps
- Upload via internet would take: 74 days
- Deadline: 2 weeks
- Data must be encrypted
- HIPAA compliance required

Question: What's the best migration approach?

Analysis:

- Data volume too large for internet upload
- Time constraint eliminates internet-based solutions
- Security and compliance requirements
- Need physical device for transfer

Recommended Solution:

1. Use **AWS Snowball Edge Storage Optimized**:
    - 80 TB usable capacity per device
    - Order 1-2 devices (for redundancy)
    - 256-bit encryption built-in
    - HIPAA compliant
2. **Migration Process**:

   (a) Order Snowball device via AWS Console
   (b) AWS ships device (2-3 days)
   (c) Connect to network, unlock with credentials
   (d) Copy data using Snowball client (2-4 days for 80 TB)
   (e) Ship device back to AWS (2-3 days)
   (f) AWS uploads to S3 (1-2 days)

3. **S3 Configuration**:
   - Enable **S3 server-side encryption (SSE-S3 or SSE-KMS)**
   - Enable **versioning** for data protection
   - Configure **lifecycle policies** to transition older data to Glacier
   - Enable **S3 Object Lock** for compliance (WORM)

4. **Compliance**:
   - Use **AWS Artifact** to access HIPAA BAA
   - Sign Business Associate Addendum (BAA)
   - Enable **CloudTrail** for audit logging
   - Use **AWS Config** for compliance monitoring

Timeline: 7-12 days (meets 2-week deadline)
Alternative for ¿100 PB: Use **AWS Snowmobile**

8.1.4 Scenario 4: Serverless Application Architecture

Situation: A startup wants to build a mobile app backend with REST API. They have limited DevOps resources and want to minimize operational overhead while paying only for actual usage.

Requirements:

- REST API for mobile app
- User authentication
- Data storage
- Image storage
- Scalable to millions of users
- Minimal operational management
- Pay-per-use pricing

Question: What AWS services should they use?
Recommended Serverless Architecture:

1. **API Layer**:

  - **Amazon API Gateway**: Create and manage REST API
  - Features: Request throttling, API keys, caching, CORS
  - Pay per million API calls

2. **Compute Layer**:
   - **AWS Lambda**: Run business logic without servers
   - Languages: Node.js, Python, Java, Go, etc.
   - Auto-scaling built-in
   - Pay only for execution time

3. **Authentication**:
   - **Amazon Cognito**: User sign-up, sign-in, access control
   - User pools for authentication
   - Identity pools for AWS resource access
   - Social identity providers (Facebook, Google)
   - Free tier: 50,000 MAUs

4. **Data Storage**:
   - **Amazon DynamoDB**: NoSQL database
   - Single-digit millisecond latency
   - Automatic scaling
   - On-demand or provisioned capacity
   - Always-free tier: 25 GB storage

5. **Image Storage**:
   - **Amazon S3**: Store user-uploaded images
   - Lifecycle policies to move old images to Glacier
   - CloudFront for fast image delivery

6. **Optional Enhancements**:
   - **Amazon CloudFront**: CDN for API and static assets
   - **AWS AppSync**: GraphQL API (alternative to API Gateway + Lambda)
   - **Amazon SES**: Send transactional emails
   - **Amazon SNS**: Push notifications to mobile devices

Benefits:

- Zero server management
- Automatic scaling from 0 to millions of users
- Pay only for actual usage

- High availability built-in
- Focus on application code, not infrastructure
- Fast deployment and iteration

Cost Example:

- 1 million API requests: $3.50
- Lambda executions: $0.20
- DynamoDB: $1.25
- S3 storage (100 GB): $2.30
- Total: $7.25/month for 1M requests

8.1.5 Scenario 5: Compliance and Governance

Situation: A financial services company with 50 AWS accounts needs to ensure no S3 buckets are publicly accessible across the organization. They also need to track all changes and demonstrate compliance.

Requirements:

- Enforce no public S3 buckets
- Apply to all accounts
- Monitor compliance continuously
- Audit all changes
- Automated remediation preferred

Question: How can they enforce and monitor this policy?

Recommended Solution:

1. **AWS Organizations Setup**:
    - Group accounts using **Organizational Units (OUs)**
    - Example structure: Production OU, Development OU, Test OU
2. **Service Control Policies (SCPs)**:
    - Create SCP denying `s3:PutBucketPublicAccessBlock` with value False
    - Deny `s3:PutBucketPolicy` if it allows public access
    - Apply to root or specific OUs
    - SCPs define maximum permissions (even admins can't override)
3. **S3 Block Public Access**:
    - Enable **S3 Block Public Access** at organization level

  - Applies to all accounts in organization
  - Prevents accidental public exposure

4. **Continuous Monitoring**:
   - Enable **AWS Config** across all accounts
   - Deploy **s3-bucket-public-read-prohibited** rule
   - Deploy **s3-bucket-public-write-prohibited** rule
   - Automatic compliance reporting

5. **Automated Remediation**:
   - Configure **AWS Config auto-remediation**
   - Use AWS Systems Manager Automation documents
   - Automatically disable public access when detected

6. **Audit and Logging**:
   - Enable **CloudTrail** in all accounts
   - Centralize logs in dedicated security account
   - Track all S3 API calls
   - Set up **CloudWatch alarms** for policy violations

7. **Centralized Security**:
   - Use **AWS Security Hub** for centralized security view
   - Aggregates findings from Config, GuardDuty, Inspector
   - Compliance dashboards for standards (PCI DSS, CIS)

Additional Recommendations:

- Regular compliance reports using **AWS Artifact**
- Periodic access reviews
- Employee training on security best practices
- Implement least privilege IAM policies

8.1.6 Scenario 6: Disaster Recovery Strategy

Situation: An e-commerce company needs disaster recovery for their application. Their business requires:

- RPO (Recovery Point Objective): 1 hour
- RTO (Recovery Time Objective): 4 hours
- Currently running in us-east-1

Question: What DR strategy should they implement?
DR Strategy Options:

1. **Backup and Restore** (Lowest cost, highest RTO)
   - RPO: Hours to days
   - RTO: Hours to days
   - Good for: Non-critical workloads
   - Not suitable for this scenario
2. **Pilot Light** (Recommended for this scenario)
   - RPO: Minutes to hours
   - RTO: Hours
   - Keep minimal version running in DR region
   - Scale up during disaster
3. **Warm Standby**
   - RPO: Seconds to minutes
   - RTO: Minutes
   - Reduced version always running
   - Higher cost than Pilot Light
4. **Multi-Site Active/Active**
   - RPO: Near zero
   - RTO: Near zero
   - Highest cost
   - Overkill for 4-hour RTO requirement

Recommended Pilot Light Implementation:

1. **Data Replication**:
   - Use **RDS cross-region read replicas**
   - Replicate from us-east-1 to us-west-2
   - Meets 1-hour RPO requirement
2. **Application AMIs**:
   - Regularly copy AMIs to DR region
   - Keep AMIs up-to-date
   - Automate with Lambda
3. **Infrastructure as Code**:

  - Use **CloudFormation templates**
  - Pre-create VPC, subnets, security groups in DR region
  - Keep Auto Scaling groups in DR region with 0 capacity

4. **DNS Failover**:
   - Use **Route 53 health checks**
   - Configure failover routing policy
   - Automatic DNS failover to DR region

5. **Testing**:
   - Quarterly DR drills
   - Document runbooks
   - Measure actual RTO/RPO

Failover Process:

1. Detect primary region failure (Route 53 health check)
2. Promote RDS read replica to master
3. Update CloudFormation stack to scale up Auto Scaling
4. Route 53 automatically redirects traffic
5. Total time: 2-3 hours (meets 4-hour RTO)

8.1.7 Scenario 7: Hybrid Cloud Connectivity

Situation: A manufacturing company wants to extend their on-premises data center to AWS while maintaining consistent network performance for their ERP system.

Requirements:

- Consistent network latency
- Private connection (no internet)
- Bandwidth: 1 Gbps
- Access to multiple VPCs

Connection Options Analysis:
Recommended Solution: AWS Direct Connect

1. **Direct Connect Setup**:
   - Order 1 Gbps Direct Connect port
   - Work with AWS Direct Connect Partner
   - Provision takes 2-4 weeks
   - Set up cross-connect at colocation facility

Solution	**Pros**	**Cons**	**Best For**
Site-to-Site VPN	Quick setup (hours), low cost, encrypted	Variable latency, internet-based, limited bandwidth	Dev/test, temporary
AWS Direct Connect	Consistent performance, high bandwidth, private	Expensive, takes weeks, not encrypted by default	Production, high bandwidth
Direct Connect + VPN	Best of both worlds	Most expensive, complex	Regulated industries

2. **Multiple VPC Access**:

   - Use **Direct Connect Gateway**
   - Connect to multiple VPCs across regions
   - Single Direct Connect connection
   - Simplifies connectivity

3. **High Availability**:

   - Order second Direct Connect connection (different location)
   - Configure BGP for automatic failover
   - Or use VPN as backup connection

4. **Security**:

   - Layer VPN over Direct Connect for encryption
   - Or use **MACsec** encryption
   - Private VIF for VPC access
   - Public VIF for public AWS services

8.2 Common Troubleshooting Scenarios

8.2.1 Cannot Connect to EC2 Instance

Symptoms: SSH or RDP connection times out or refused
Troubleshooting Steps:

1. **Verify Instance Status**:

   - Check instance state is ”running”
   - Check status checks are passing
   - View system log for boot errors

2. **Check Security Group**:

   - Ensure inbound rule allows SSH (22) or RDP (3389)

- Verify source IP is allowed (0.0.0.0/0 or your IP)
- Check if security group changed recently

3. **Check Network ACL**:
   - Ensure NACL allows inbound traffic on port
   - Ensure NACL allows ephemeral outbound ports (1024-65535)
   - NACLs are stateless!
4. **Verify Network Configuration**:
   - Instance has public IP (if connecting from internet)
   - Instance in public subnet (has IGW route)
   - Or using bastion host for private subnet
5. **Check Key Pair**:
   - Using correct .pem/.ppk file
   - File permissions correct (chmod 400 for .pem)
   - Key pair matches instance
6. **Check Route Table**:
   - Subnet has route to IGW (0.0.0.0/0 → igw-xxx)
   - Or route to NAT Gateway for private subnet

8.2.2 S3 Access Denied Errors

Common Causes and Solutions:

1. **IAM Permissions**:
   - Verify IAM policy grants s3:GetObject, s3:PutObject
   - Check for explicit Deny statements
   - Verify resource ARN in policy matches bucket
2. **Bucket Policy**:
   - Check bucket policy doesn't deny access
   - Verify Principal in policy
   - Check for IP-based restrictions
3. **Block Public Access**:
   - If public access needed, disable Block Public Access
   - Check both bucket-level and account-level settings
4. **Encryption**:

  - If using SSE-KMS, verify KMS key policy
  - Ensure user has kms:Decrypt permission

5. **Cross-Account Access**:
   - Bucket policy must allow cross-account access
   - Assume role with correct permissions

8.2.3 Lambda Function Issues

Issue 1: Function Timing Out
Solutions:

- Increase timeout (default 3 sec, max 15 min)
- Optimize code performance
- Check VPC configuration (can add latency)
- Increase memory (also increases CPU)
- Investigate cold start delays

Issue 2: Insufficient Permissions
Solutions:

- Check Lambda execution role has required permissions
- Review CloudWatch Logs for permission errors
- Add necessary IAM policies to execution role
- For VPC: Ensure role has VPC execution permissions

Issue 3: Throttling
Solutions:

- Request concurrency limit increase
- Implement exponential backoff in calling application
- Use SQS to buffer requests
- Consider reserved concurrency for critical functions

Chapter 9

Additional AWS Services and Advanced Topics

9.1 Developer Tools and CI/CD

9.1.1 AWS CodeCommit

- Git-based source control repository
- Secure, highly available
- No size limits on repositories
- Integrates with existing Git tools
- Encrypted at rest and in transit
- Free tier: 5 active users per month

9.1.2 AWS CodeBuild

- Fully managed build service
- Compiles source code, runs tests, produces packages
- Scales automatically
- Pay only for build time
- Pre-configured environments or custom Docker images
- Integrates with CodeCommit, GitHub, Bitbucket

9.1.3 AWS CodeDeploy

- Automated deployment service
- Deploy to EC2, Lambda, on-premises servers
- Deployment strategies: In-place, blue/green

- Automatic rollback on failure
- Integration with existing CI/CD tools
- No additional charge (pay for resources)

9.1.4 AWS CodePipeline

- Continuous delivery service
- Automates release pipeline
- Integrates with CodeCommit, CodeBuild, CodeDeploy
- Third-party integrations (GitHub, Jenkins)
- Visual workflow builder
- $1 per active pipeline per month

9.1.5 AWS CodeStar

- Unified user interface for development activities
- Quickly develop, build, and deploy applications
- Project templates for various languages and platforms
- Integrated dashboard for monitoring
- Team collaboration features
- No additional charge

9.1.6 AWS Cloud9

- Cloud-based IDE (Integrated Development Environment)
- Write, run, and debug code in browser
- Supports 40+ programming languages
- Built-in terminal with AWS CLI
- Collaborative coding features
- Pay only for underlying EC2 instance

9.2 Application Integration Services

9.2.1 Amazon EventBridge

- Serverless event bus service
- Connect applications using events
- Formerly CloudWatch Events
- Event sources: AWS services, custom applications, SaaS apps
- Event patterns for filtering
- Multiple targets per rule
- Pay per event

9.2.2 Amazon MQ

- Managed message broker service
- Supports Apache ActiveMQ and RabbitMQ
- For migrating existing applications using message brokers
- Alternative to SNS/SQS for specific protocols
- Industry-standard APIs and protocols (MQTT, AMQP, STOMP)
- Single-instance or active/standby deployment

9.2.3 AWS App Mesh

- Service mesh for microservices
- Monitor and control communications
- Works with ECS, EKS, EC2
- Provides observability, traffic management
- Based on Envoy proxy
- No additional charge (pay for resources)

9.3 End User Computing

9.3.1 Amazon WorkSpaces

- Managed Desktop-as-a-Service (DaaS)
- Virtual Windows or Linux desktops
- Access from any device
- Persistent desktop storage
- Integrated with Active Directory
- Pricing: Monthly or hourly
- Use cases: Remote work, contractors, BYOD

9.3.2 Amazon AppStream 2.0

- Application streaming service
- Stream desktop applications to browser
- No need to install applications locally
- Scales automatically
- Pay as you go
- Use cases: Training, POC, software trials

9.3.3 Amazon WorkDocs

- Secure document storage and collaboration
- Similar to Dropbox or Google Drive
- 1 TB storage per user
- File comments and feedback
- Active Directory integration
- Mobile and desktop apps

9.3.4 Amazon WorkLink

- Secure mobile access to internal websites
- No VPN required
- Renders content in browser on AWS
- Only pixels sent to device
- Protects corporate network
- Per user per month pricing

9.4 IoT Services

9.4.1 AWS IoT Core

- Connect IoT devices to cloud
- Supports billions of devices
- MQTT, HTTPS, WebSockets protocols
- Device shadow for state management
- Rules engine for data processing
- Integration with other AWS services

9.4.2 AWS IoT Greengrass

- Extend AWS to edge devices
- Local compute, messaging, and data caching
- Run Lambda functions at the edge
- Operate offline
- Secure communication with cloud
- ML inference at edge

9.4.3 AWS IoT Analytics

- Analytics for IoT data
- Process and analyze IoT data
- Pre-built analytics templates
- Integration with QuickSight
- SQL queries on time-series data
- Machine learning integration

9.5 Media Services

9.5.1 Amazon Elastic Transcoder

- Convert media files between formats
- Scalable media transcoding
- Pre-configured presets
- Pay per minute of transcoding
- Integration with S3, CloudFront

9.5.2 AWS Elemental MediaConvert

- File-based video transcoding
- Broadcast-grade features
- Supports various formats and codecs
- On-demand or reserved pricing
- More advanced than Elastic Transcoder

9.5.3 Amazon Kinesis Video Streams

- Capture, process, and store video streams
- Millions of devices
- Playback, analytics, ML integration
- Use cases: Security cameras, live streaming
- Pay for data ingested and consumed

9.6 Additional AI/ML Services

9.6.1 Amazon Polly

- Text-to-speech service
- Natural sounding speech
- Multiple languages and voices
- Neural TTS for more natural sound
- Pay per character
- Use cases: E-learning, accessibility

9.6.2 Amazon Transcribe

- Automatic speech recognition (ASR)
- Convert speech to text
- Real-time and batch processing
- Speaker identification
- Custom vocabulary
- Pay per second of audio

9.6.3 Amazon Translate

- Neural machine translation
- Translate text between languages
- 75+ languages supported
- Custom terminology
- Pay per character
- Use cases: Localization, content creation

9.6.4 Amazon Forecast

- Time-series forecasting service
- Uses machine learning
- No ML expertise required
- More accurate than traditional methods
- Use cases: Demand forecasting, resource planning

9.6.5 Amazon Kendra

- Intelligent search service
- ML-powered enterprise search
- Natural language queries
- Learns from user interactions
- Connects to various data sources

9.6.6 Amazon Personalize

- Real-time recommendations
- Same technology as Amazon.com
- No ML expertise required
- Real-time and batch recommendations
- Use cases: Product recommendations, personalized content

9.6.7 Amazon Textract

- Extract text and data from documents
- OCR and form recognition
- Preserves structure and relationships
- Works with PDFs, images
- Pay per page

9.7 Business Applications

9.7.1 Amazon Connect

- Cloud-based contact center
- Omnichannel customer service
- AI-powered chatbots
- Pay as you go
- Integration with other AWS services
- Use cases: Customer support, helpdesk

9.7.2 Amazon Simple Email Service (SES)

- Email sending and receiving
- Transactional and marketing emails
- High deliverability
- Pay per email sent
- Free tier: 62,000 emails/month (from EC2)
- Email validation and filtering

9.7.3 Amazon Pinpoint

- Marketing communication service
- Email, SMS, push notifications
- Customer segmentation
- Campaign management
- Analytics and engagement metrics
- Pay for messages sent

9.7.4 AWS WorkMail

- Managed email and calendar service
- Alternative to Exchange/Gmail
- Web-based access
- Mobile and desktop clients
- Active Directory integration
- $4 per user per month

9.7.5 Amazon Chime

- Video conferencing and communication
- Meetings, chat, business calling
- Screen sharing
- Per user per day pricing
- Alternative to Zoom, Teams

9.8 Management and Governance (Advanced)

9.8.1 AWS License Manager

- Manage software licenses
- Track license usage
- Set usage limits
- Prevent license violations
- Works with Microsoft, Oracle, SAP, etc.

9.8.2 AWS Service Catalog

- Create and manage catalogs of IT services
- Standardized products
- Control which services users can deploy
- Version control for products
- Governance and compliance
- Self-service portal for end users

9.8.3 AWS Well-Architected Tool

- Review workload architecture
- Compare against best practices
- Six pillars assessment
- Get improvement plan
- Free service
- Periodic reviews recommended

9.8.4 AWS Personal Health Dashboard

- Personalized view of AWS service health
- Alerts for service events affecting you
- Proactive notifications
- Remediation guidance
- Available to all customers
- Different from Service Health Dashboard (global status)

9.8.5 AWS Compute Optimizer

- Recommends optimal AWS resources
- ML-based recommendations
- Analyzes historical utilization
- Suggests EC2 instance types, EBS volumes, Lambda memory
- Cost savings opportunities
- No additional charge

9.9 Migration and Transfer Services

9.9.1 AWS Application Discovery Service

- Discover on-premises applications
- Plan migrations
- Collect server utilization and dependency data
- Agentless or agent-based discovery
- Export data for analysis
- Integration with Migration Hub

9.9.2 AWS Migration Hub

- Track application migrations
- Single location to monitor migrations
- Works with migration tools
- Visualize migration progress
- No additional cost

9.9.3 AWS Server Migration Service (SMS)

- Migrate on-premises servers to AWS
- Incremental replication
- Minimal downtime
- Supports VMware, Hyper-V, Azure
- No additional charge
- Being replaced by Application Migration Service

9.9.4 AWS DataSync

- Transfer data between on-premises and AWS
- Up to 10x faster than open-source tools
- Automated data transfer
- Supports NFS, SMB protocols
- Transfers to S3, EFS, FSx
- Pay per GB transferred

9.9.5 AWS Transfer Family

- Fully managed SFTP, FTPS, FTP
- Transfer files to/from S3 or EFS
- No infrastructure to manage
- Integration with existing auth systems
- Pay per protocol enabled + data transfer

9.10 Networking (Advanced)

9.10.1 AWS Global Accelerator

- Improve availability and performance
- Uses AWS global network
- Static anycast IP addresses
- Automatic failover
- Health checks
- Use cases: Gaming, IoT, VoIP
- Different from CloudFront (not caching)

9.10.2 AWS App Mesh

- Service mesh for microservices
- Monitor and control communications
- Works with ECS, EKS, EC2
- Based on Envoy proxy
- Traffic management, observability

9.10.3 AWS Cloud Map

- Service discovery
- Register application resources
- Discover services via API or DNS
- Health checking
- Integration with ECS, EKS

9.11 Additional Storage Services

9.11.1 Amazon FSx

- Fully managed file systems
- **FSx for Windows File Server**:
  - Windows native file system
  - SMB protocol
  - Active Directory integration
  - For Windows applications
- **FSx for Lustre**:
  - High-performance file system
  - ML, HPC, video processing
  - Integration with S3
  - Sub-millisecond latencies

9.11.2 AWS Backup

- Centralized backup service
- Automate and manage backups
- Backup across AWS services
- Backup policies and retention
- Compliance reporting
- Pay for storage used

9.12 Blockchain and Quantum

9.12.1 Amazon Managed Blockchain

- Create and manage blockchain networks
- Supports Hyperledger Fabric and Ethereum
- Fully managed
- Scales automatically
- Use cases: Supply chain, finance

9.12.2 Amazon Braket

- Quantum computing service
- Develop quantum algorithms
- Test on quantum simulators
- Run on quantum hardware
- Pay for simulation and quantum tasks

9.13 Exam Tips for Service Selection

9.13.1 Database Selection Decision Tree

Choose Database Based on Requirements:

- **Need SQL and ACID transactions?**
  - Traditional app, existing database → **RDS**
  - Need extreme performance → **Aurora**
  - Specific needs: Oracle/SQL Server → **RDS** with that engine
- **Need NoSQL?**
  - Key-value, scale to millions of requests/sec → **DynamoDB**
  - Document database, MongoDB compatible → **DocumentDB**
  - Graph relationships → **Neptune**
- **Need caching?**
  - In-memory cache → **ElastiCache**
  - Redis features needed → ElastiCache for Redis
  - Simple caching → ElastiCache for Memcached
- **Need data warehouse/analytics?**
  - OLAP, BI, big data → **Redshift**
  - Query data in S3 → **Athena**
  - Real-time analytics → **Kinesis Data Analytics**
- **Time-series data?**
  - IoT, metrics, logs → **Timestream**

Requirement	**Best Choice**	**Why**
Full OS control	EC2	Complete flexibility
Serverless, event-driven	Lambda	No servers, pay per use
Deploy app quickly	Elastic Beanstalk	PaaS, managed
Containers, Kubernetes	EKS	Kubernetes compatible
Containers, AWS native	ECS	Simpler than EKS
Containers, no servers	Fargate	Serverless containers
Batch processing	AWS Batch	Optimized for batch
Simple website, blog	Lightsail	Predictable pricing
Table 9.1: Compute Service Selection		

9.13.2 Compute Selection Flowchart

9.13.3 Storage Selection Guide

Use Case	**Service**	**Reason**
Backups, archives	S3 Glacier	Lowest cost
Website content	S3 + CloudFront	Scalable, global
EC2 boot volume	EBS	Block storage
Shared file system	EFS	Multi-attach
Windows file shares	FSx for Windows	SMB, AD integration
HPC workloads	FSx for Lustre	High performance
Temporary data	Instance Store	Highest IOPS
Offline transfer	Snow Family	Large data volumes

Table 9.2: Storage Selection Guide

Chapter 10

Study Plan and Resources

10.1 Recommended Study Timeline

10.1.1 4-Week Study Plan

Week 1: Cloud Concepts and Foundations

- Study Domain 1: Cloud Concepts
- Understand cloud computing fundamentals
- Learn AWS Well-Architected Framework
- Complete AWS Cloud Practitioner Essentials course
- Practice: Create AWS account, explore console

Week 2: Security and Compliance

- Study Domain 2: Security and Compliance
- Master Shared Responsibility Model
- Learn IAM thoroughly
- Review security services
- Practice: Set up IAM users, groups, roles, MFA

Week 3: Technology and Services

- Study Domain 3: Cloud Technology and Services
- Focus on core services: EC2, S3, RDS, VPC
- Learn other services at high level
- Practice: Launch EC2, create S3 bucket, set up VPC
- Take mid-point practice exam

Week 4: Billing and Review

- Study Domain 4: Billing, Pricing, and Support
- Memorize support plans
- Review all domains
- Take multiple practice exams
- Review weak areas
- Final review day before exam

10.1.2 2-Week Intensive Plan

For those with IT background or time constraints:

Week 1:

- Days 1-2: Cloud Concepts, Security
- Days 3-5: Core services (compute, storage, database, networking)
- Days 6-7: Remaining services, practice exam

Week 2:

- Days 8-9: Billing, Support plans, cost optimization
- Days 10-12: Practice exams, review weak areas
- Days 13-14: Final review, rest before exam

10.2 Official AWS Resources

10.2.1 Free Resources

1. **AWS Cloud Practitioner Essentials**
   - Free digital course on AWS Skill Builder
   - Covers all exam domains
   - Approximately 6 hours
   - `https://aws.amazon.com/training/digital/`
2. **AWS Exam Guide**
   - Official exam content outline
   - Lists all topics tested
   - Download from AWS Training and Certification
3. **AWS Whitepapers**

- Overview of AWS
- AWS Well-Architected Framework
- AWS Pricing
- Available at https://aws.amazon.com/whitepapers/

4. **AWS Documentation**
   - Comprehensive service documentation
   - FAQs for each service
   - Best practices guides
5. **AWS Free Tier**
   - Hands-on practice at no cost
   - 12 months free for many services
   - Always-free tier for others

10.2.2 Paid Resources

1. **AWS Official Practice Exam**
   - $20 USD
   - 20 questions
   - Same format as real exam
   - Purchase through AWS Training
2. **AWS Classroom Training**
   - Instructor-led courses
   - Virtual or in-person
   - Varies by location and provider

10.3 Third-Party Resources

10.3.1 Online Courses

- **A Cloud Guru / Pluralsight**
- **Udemy**: AWS Certified Cloud Practitioner courses
- **Linux Academy / A Cloud Guru**
- **Coursera**: AWS Cloud Practitioner specializations

10.3.2 Practice Exams

- Tutorials Dojo practice exams
- Whizlabs practice tests
- ExamTopics (free community questions)

10.3.3 Books

- AWS Certified Cloud Practitioner Study Guide (Sybex)
- AWS Certified Cloud Practitioner Exam Guide

10.3.4 YouTube Channels

- AWS Online Tech Talks
- FreeCodeCamp AWS courses
- Exam Pro

10.4 Hands-On Practice

Important

Hands-on experience is crucial for exam success. These labs will help you understand AWS services beyond theory and are likely to stay within Free Tier limits if done carefully.

10.4.1 Prerequisites

Create Your AWS Account:

1. Visit `https://aws.amazon.com`
2. Click "Create an AWS Account"
3. Provide email, password, and AWS account name
4. Enter contact information
5. Provide payment method (credit/debit card required)
6. Verify identity via phone
7. Select Support Plan (choose Basic - Free)
8. Wait for account activation (can take up to 24 hours)

Exam Tip

Set up billing alerts immediately to avoid unexpected charges. AWS Free Tier is generous, but mistakes can incur costs.

10.4.2 Lab 1: Set Up Billing Alerts and Budget

Duration: 15 minutes

Objective: Protect yourself from unexpected charges

Steps:

1. Sign in to AWS Management Console
2. Click your account name (top right) → "Account"
3. Scroll to "IAM User and Role Access to Billing Information"
4. Click "Edit" and enable "Activate IAM Access"
5. Navigate to CloudWatch service
6. Select your region (N. Virginia - us-east-1)
7. Go to "Alarms" → "Billing" → "Create alarm"
8. Click "Select metric" → "Billing" → "Total Estimated Charge"
9. Select USD and click "Select metric"
10. Set threshold: Greater than $5 (or your preferred amount)
11. Click "Next"
12. Create new SNS topic: "Billing-Alerts"
13. Enter your email address
14. Click "Create topic"
15. Confirm subscription via email
16. Complete alarm creation
17. Navigate to AWS Budgets
18. Click "Create budget"
19. Select "Cost budget" → "Next"
20. Set budget amount: $10/month
21. Configure alerts at 80% and 100%
22. Create budget

10.4.3 Lab 2: IAM Users, Groups, Roles, and MFA

Duration: 30 minutes

Objective: Understand IAM security best practices

Steps:

Secure Root Account

1. Navigate to IAM service in AWS Console
2. Click "Dashboard" - review security recommendations
3. Click "Add MFA" for root account
4. Select "Virtual MFA device"
5. Install Google Authenticator or Authy on your phone
6. Scan QR code with authenticator app
7. Enter two consecutive MFA codes
8. Click "Assign MFA"
9. Store root account credentials securely

Create IAM Admin User

1. In IAM, click "Users" → "Add users"
2. Username: "admin-user"
3. Select "Provide user access to AWS Management Console"
4. Choose "I want to create an IAM user"
5. Set custom password or autogenerate
6. Uncheck "Users must create a new password at next sign-in"
7. Click "Next"
8. Click "Attach policies directly"
9. Search and select "AdministratorAccess"
10. Click "Next" → "Create user"
11. Download credentials (save securely)
12. Copy console sign-in URL

Create IAM Groups

1. Click "User groups" → "Create group"
2. Group name: "Developers"
3. Attach policies:
   - AmazonEC2ReadOnlyAccess
   - AmazonS3FullAccess

4. Click ”Create group”
5. Create another group: ”Administrators”
6. Attach policy: AdministratorAccess
7. Create group

Create Additional IAM Users

1. Create user ”developer-1”
2. Enable console access
3. Set password
4. Click ”Next”
5. Add user to ”Developers” group
6. Create user
7. Create user ”developer-2”
8. Repeat process, add to ”Developers” group

Create IAM Role for EC2

1. Click ”Roles” → ”Create role”
2. Select ”AWS service”
3. Use case: ”EC2”
4. Click ”Next”
5. Attach policy: ”AmazonS3ReadOnlyAccess”
6. Click ”Next”
7. Role name: ”EC2-S3-ReadOnly-Role”
8. Description: ”Allows EC2 instances to read from S3”
9. Click ”Create role”

Test IAM Policies

1. Sign out from root account
2. Sign in as ”developer-1”
3. Try to access S3 (should work)
4. Try to access IAM (should be denied)
5. Sign out

10.4.4 Lab 3: Launch and Configure EC2 Instance

Duration: 45 minutes

Objective: Launch a web server on EC2

Steps:

1. Navigate to EC2 service
2. Select region (e.g., us-east-1)
3. Click "Launch Instance"
4. Name: "MyWebServer"
5. Select "Amazon Linux 2023 AMI" (Free tier eligible)
6. Instance type: "t2.micro" (Free tier eligible)
7. Create new key pair:
   - Key pair name: "my-key-pair"
   - Type: RSA
   - Format: .pem (Linux/Mac) or .ppk (Windows/PuTTY)
   - Click "Create key pair"
   - Save file securely
8. Network settings:
   - Create security group: "web-server-sg"
   - Allow SSH (port 22) from "My IP"
   - Allow HTTP (port 80) from "Anywhere" (0.0.0.0/0)
9. Configure storage: 8 GB gp3 (default, Free tier)
10. Expand "Advanced details"
11. Scroll to "User data" and paste:

    ```
    #!/bin/bash
    yum update -y
    yum install -y httpd
    systemctl start httpd
    systemctl enable httpd
    echo "<h1>Hello from AWS EC2</h1>" >
      /var/www/html/index.html
    ```

12. Click "Launch instance"
13. Wait for instance state: "Running"
14. Select instance, copy "Public IPv4 address"
15. Open browser, navigate to http://[public-ip]
16. You should see "Hello from AWS EC2"

Connect to EC2 via SSH

1. Open terminal (Linux/Mac) or PuTTY (Windows)
2. Navigate to directory with key pair
3. Set permissions (Linux/Mac): `chmod 400 my-key-pair.pem`
4. Connect: `ssh -i my-key-pair.pem ec2-user@[public-ip]`
5. Accept fingerprint (type ”yes”)
6. You’re now connected to your EC2 instance
7. Verify web server: `sudo systemctl status httpd`

Create AMI (Amazon Machine Image)

1. In EC2 Console, select your instance
2. Click ”Actions” → ”Image and templates” → ”Create image”
3. Image name: ”MyWebServer-AMI”
4. Image description: ”Web server with Apache”
5. Click ”Create image”
6. Go to ”AMIs” in left menu
7. Wait for status: ”Available”
8. You can now launch new instances from this AMI

Create Snapshot

1. Go to ”Volumes” in EC2 left menu
2. Select volume attached to your instance
3. Click ”Actions” → ”Create snapshot”
4. Description: ”WebServer backup”
5. Add tag: Key=”Name”, Value=”WebServer-Snapshot”
6. Click ”Create snapshot”
7. Go to ”Snapshots” to view

Cleanup (Important!)

1. Terminate instance: Select instance → ”Instance state” → ”Terminate”
2. Delete snapshot: Select snapshot → ”Actions” → ”Delete”
3. Deregister AMI: Select AMI → ”Actions” → ”Deregister”
4. Delete associated snapshot created by AMI

10.4.5 Lab 4: Amazon S3 Storage and Website Hosting

Duration: 40 minutes

Objective: Master S3 storage features

Create S3 Bucket

1. Navigate to S3 service
2. Click ”Create bucket”
3. Bucket name: ”my-website-[your-name]-[random-numbers]” (must be globally unique)
4. Region: Select your preferred region
5. Uncheck ”Block all public access” (for website hosting)
6. Acknowledge warning
7. Enable ”Bucket Versioning”
8. Add tag: Key=”Project”, Value=”Learning”
9. Click ”Create bucket”

Upload Objects

1. Click on your bucket name
2. Click ”Upload”
3. Create a simple HTML file (index.html):

   ```
   <!DOCTYPE html>
   <html>
   <head><title>My AWS Website</title></head>
   <body>
     <h1>Welcome to My S3 Website</h1>
     <p>This is hosted on Amazon S3</p>
   </body>
   </html>
   ```

4. Upload index.html
5. Create error.html:

   ```
   <!DOCTYPE html>
   <html>
   <head><title>Error</title></head>
   <body><h1>404 - Page Not Found</h1></body>
   </html>
   ```

6. Upload error.html
7. Upload a test image or text file
8. Click ”Upload”

Enable Static Website Hosting

1. In your bucket, go to ”Properties” tab
2. Scroll to ”Static website hosting”
3. Click ”Edit”
4. Enable ”Static website hosting”
5. Hosting type: ”Host a static website”
6. Index document: index.html
7. Error document: error.html
8. Click ”Save changes”
9. Copy the bucket website endpoint URL

Configure Bucket Policy

1. Go to ”Permissions” tab
2. Scroll to ”Bucket policy”
3. Click ”Edit”
4. Paste the following policy (replace BUCKET-NAME):

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::BUCKET-NAME/*"
    }
  ]
}
```

5. Click ”Save changes”
6. Open website endpoint URL in browser
7. Your website should be live!

Configure Versioning

1. Edit index.html locally, change content
2. Upload new version to S3 (same filename)
3. In bucket, select index.html
4. Click "Versions" tab
5. You'll see multiple versions
6. Click on older version to view/restore

Create Lifecycle Policy

1. Go to "Management" tab
2. Click "Create lifecycle rule"
3. Rule name: "Archive-Old-Files"
4. Choose rule scope: Apply to all objects
5. Lifecycle rule actions:
   - Transition current versions: 30 days → S3 Standard-IA
   - Transition current versions: 90 days → S3 Glacier Flexible Retrieval
   - Expire current versions: 365 days
6. Click "Create rule"

Enable Server-Side Encryption

1. Go to "Properties" tab
2. Scroll to "Default encryption"
3. Click "Edit"
4. Select "Server-side encryption with Amazon S3 managed keys (SSE-S3)"
5. Enable "Bucket Key"
6. Click "Save changes"

Cleanup

1. Empty bucket: Click "Empty" button
2. Confirm by typing "permanently delete"
3. Delete bucket: Click "Delete"
4. Confirm by entering bucket name

10.4.6 Lab 5: VPC, Subnets, and Network Configuration

Duration: 60 minutes
Objective: Build custom VPC with public and private subnets

Create VPC

1. Navigate to VPC service
2. Click "Create VPC"
3. Select "VPC and more" (creates VPC with subnets)
4. Name tag: "MyVPC"
5. IPv4 CIDR: 10.0.0.0/16
6. Number of AZs: 2
7. Number of public subnets: 2
8. Number of private subnets: 2
9. NAT gateways: None (to stay in free tier)
10. VPC endpoints: None
11. Click "Create VPC"
12. Wait for creation to complete

Review VPC Components

1. Go to "Your VPCs" - verify VPC created (10.0.0.0/16)
2. Go to "Subnets" - you should see:
   - Public subnet 1: 10.0.0.0/20 (AZ a)
   - Public subnet 2: 10.0.16.0/20 (AZ b)
   - Private subnet 1: 10.0.128.0/20 (AZ a)
   - Private subnet 2: 10.0.144.0/20 (AZ b)
3. Go to "Internet Gateways" - verify IGW attached to VPC
4. Go to "Route Tables" - review routes:
   - Public route table: has route to IGW (0.0.0.0/0 → igw-xxx)
   - Private route table: local routes only

Create Security Groups

1. Go to ”Security Groups”
2. Click ”Create security group”
3. Name: ”WebServer-SG”
4. Description: ”Allow HTTP and SSH”
5. VPC: Select your VPC
6. Inbound rules:
    - Type: SSH, Source: My IP
    - Type: HTTP, Source: 0.0.0.0/0
7. Outbound rules: Leave default (all traffic allowed)
8. Click ”Create security group”
9. Create another security group: ”Database-SG”
10. Description: ”Allow MySQL from WebServer”
11. Inbound rule:
    - Type: MySQL/Aurora (port 3306)
    - Source: Custom, select ”WebServer-SG”
12. Click ”Create security group”

Launch EC2 in Custom VPC

1. Go to EC2 service
2. Launch instance
3. Name: ”VPC-Test-Instance”
4. AMI: Amazon Linux 2023
5. Instance type: t2.micro
6. Network settings:
    - VPC: Select MyVPC
    - Subnet: Select public subnet
    - Auto-assign public IP: Enable
    - Security group: Select WebServer-SG
7. Launch instance
8. Verify you can access instance via public IP

Create Network ACL

1. Go to ”Network ACLs” in VPC
2. Click ”Create network ACL”
3. Name: ”Custom-NACL”
4. VPC: Select MyVPC
5. Click ”Create”
6. Select your NACL
7. Go to ”Inbound rules” tab
8. Click ”Edit inbound rules”
9. Add rules:
   - Rule 100: HTTP (80), Source: 0.0.0.0/0, Allow
   - Rule 110: SSH (22), Source: 0.0.0.0/0, Allow
   - Rule 120: Custom TCP (1024-65535), Source: 0.0.0.0/0, Allow
   - Rule *: All traffic, Deny (default)
10. Go to ”Outbound rules” tab
11. Add similar rules for outbound
12. Associate with subnet if desired

Cleanup

1. Terminate EC2 instance
2. Delete custom security groups
3. Delete custom NACLs
4. Delete VPC (this will delete associated subnets, route tables, IGW)

10.4.7 Lab 6: Amazon RDS Database

Duration: 30 minutes
Objective: Launch managed MySQL database
Steps:

1. Navigate to RDS service
2. Click ”Create database”
3. Choose: ”Standard create”
4. Engine: MySQL

5. Version: Latest (default)
6. Templates: "Free tier"
7. DB instance identifier: "mydatabase"
8. Master username: admin
9. Master password: Create secure password (save it!)
10. DB instance class: db.t3.micro or db.t4g.micro (Free tier)
11. Storage: 20 GB General Purpose SSD (gp3)
12. Uncheck "Enable storage autoscaling"
13. Connectivity:
    - Don't connect to EC2 instance
    - VPC: Default VPC (or your custom VPC)
    - Public access: No (recommended)
    - VPC security group: Create new
    - New VPC security group name: "rds-mysql-sg"
14. Additional configuration:
    - Initial database name: "mydb"
    - Disable automated backups (to stay free tier)
    - Disable encryption (optional, for simplicity)
15. Click "Create database"
16. Wait 5-10 minutes for creation
17. Once available, click on database name
18. Note the endpoint URL

Connect to RDS (requires EC2 in same VPC)

1. Launch EC2 instance in same VPC
2. SSH into EC2 instance
3. Install MySQL client:

   ```
   sudo yum install -y mariadb105
   ```

4. Modify RDS security group to allow port 3306 from EC2 security group
5. Connect to RDS:

```
mysql -h [rds-endpoint] -u admin -p
```

6. Enter password
7. Run commands:

   ```
   SHOW DATABASES;
   USE mydb;
   CREATE TABLE users (id INT, name VARCHAR(50));
   INSERT INTO users VALUES (1, 'John');
   SELECT * FROM users;
   ```

8. Exit: `exit;`

Cleanup

1. Go to RDS console
2. Select your database
3. Click ”Actions” → ”Delete”
4. Uncheck ”Create final snapshot”
5. Check ”I acknowledge...”
6. Type ”delete me”
7. Click ”Delete”

10.4.8 Lab 7: CloudWatch Monitoring and Alarms

Duration: 25 minutes
Objective: Monitor EC2 and create alarms
Steps:

1. Launch a t2.micro EC2 instance (if not already running)
2. Navigate to CloudWatch service
3. Click ”All metrics”
4. Click ”EC2” → ”Per-Instance Metrics”
5. Search for your instance ID
6. Select ”CPUUtilization” metric
7. View graph (will be low initially)
8. Click ”Actions” → ”Create alarm”

9. Set threshold: CPUUtilization > 70%
10. Click ”Next”
11. Create new SNS topic: ”EC2-Alerts”
12. Enter email address
13. Click ”Create topic”
14. Confirm email subscription
15. Click ”Next”
16. Alarm name: ”High-CPU-Alert”
17. Description: ”Alert when CPU exceeds 70%”
18. Click ”Next” → ”Create alarm”

Test Alarm (Optional)

1. SSH into EC2 instance
2. Install stress tool:

   ```
   sudo yum install -y stress
   ```

3. Run CPU stress:

   ```
   stress --cpu 2 --timeout 300
   ```

4. Wait 5-10 minutes
5. Check CloudWatch alarm - should transition to ”ALARM” state
6. You’ll receive email notification

View CloudWatch Logs

1. In CloudWatch, go to ”Logs” → ”Log groups”
2. Click ”Create log group”
3. Name: ”/aws/my-application”
4. Click ”Create”
5. Explore other log groups (if any exist from other services)

Cleanup

1. Delete CloudWatch alarm
2. Delete SNS topic
3. Delete log group
4. Terminate EC2 instance

10.4.9 Lab 8: AWS Cost Management Tools

Duration: 30 minutes

Objective: Explore billing and cost tools

AWS Pricing Calculator

1. Visit `https://calculator.aws`
2. Click "Create estimate"
3. Search "EC2"
4. Click "Configure"
5. Region: Select your region
6. Quick estimate: 10 t3.medium instances
7. Pricing model: On-Demand
8. Click "Add to my estimate"
9. Search "S3"
10. Standard storage: 1000 GB
11. Add to estimate
12. Search "RDS"
13. Add MySQL database, db.t3.medium
14. Add to estimate
15. Review total estimated monthly cost
16. Compare different pricing models (Reserved, Savings Plans)
17. Export estimate
18. Share estimate (get shareable link)

AWS Cost Explorer

1. Go to Billing and Cost Management console
2. Click "Cost Explorer"
3. Click "Launch Cost Explorer" (if first time)
4. Wait for data to populate (takes 24 hours for new accounts)
5. View "Monthly costs by service"
6. Filter by service (EC2, S3, etc.)
7. Group by service, region, or tag
8. View forecast
9. Create custom report
10. Save report for future use

Review Bills

1. Go to "Bills" in Billing console
2. View current month charges
3. Expand services to see detailed breakdown
4. Check Free Tier usage
5. Download bill as CSV

AWS Budgets

1. Go to "Budgets" in Billing console
2. Review budget created in Lab 1
3. Create additional budget:
   - Type: Usage budget
   - Service: EC2
   - Usage type: Running hours
   - Amount: 750 hours/month (Free tier limit)
   - Set alert at 80%
4. Create budget

Trusted Advisor

1. Navigate to Trusted Advisor
2. Review 7 core checks (available on Basic support):
   - S3 Bucket Permissions
   - Security Groups - Specific Ports Unrestricted
   - IAM Use
   - MFA on Root Account
   - EBS Public Snapshots
   - RDS Public Snapshots
   - Service Limits
3. Click on each check to view details
4. Take action on any recommendations
5. Refresh checks

10.4.10 Lab 9: Lambda Serverless Function

Duration: 25 minutes
Objective: Create and test Lambda function
Steps:

1. Navigate to Lambda service
2. Click "Create function"
3. Choose "Author from scratch"
4. Function name: "HelloWorldFunction"
5. Runtime: Python 3.12
6. Architecture: x86_64
7. Click "Create function"
8. In code editor, replace default code with:

```
import json

def lambda_handler(event, context):
    name = event.get('name', 'World')
    return {
        'statusCode': 200,
        'body': json.dumps(f'Hello, {name}!')
    }
```

9. Click ”Deploy”
10. Click ”Test”
11. Event name: ”TestEvent”
12. Event JSON:

```
{
  "name": "AWS Student"
}
```

13. Click ”Save”
14. Click ”Test”
15. View execution results
16. Check ”Execution result” shows: ”Hello, AWS Student!”
17. View CloudWatch Logs (click ”Monitor” → ”View logs in CloudWatch”)

Configure Lambda Trigger

1. In function overview, click ”Add trigger”
2. Select ”API Gateway”
3. Create new API
4. API type: HTTP API
5. Security: Open
6. Click ”Add”
7. Copy API endpoint URL
8. Open in browser: [URL]?name=YourName
9. You should see: ”Hello, YourName!”

Cleanup

1. Delete Lambda function
2. Delete API Gateway (if created)

10.4.11 Lab 10: CloudFormation Infrastructure as Code

Duration: 20 minutes

Objective: Deploy infrastructure using template

Steps:

1. Navigate to CloudFormation service
2. Click "Create stack"
3. Choose "Template is ready"
4. Select "Upload a template file"
5. Create file `simple-stack.yaml`:

```
AWSTemplateFormatVersion: '2010-09-09'
Description: Simple S3 bucket stack
Resources:
  MyS3Bucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: !Sub 'cf-bucket-${AWS::AccountId}'
      VersioningConfiguration:
        Status: Enabled
Outputs:
  BucketName:
    Description: Name of S3 bucket
    Value: !Ref MyS3Bucket
```

6. Upload file
7. Click "Next"
8. Stack name: "MyFirstStack"
9. Click "Next"
10. Add tags (optional)
11. Click "Next"
12. Review and click "Submit"
13. Watch stack creation progress
14. Go to "Outputs" tab to see bucket name
15. Verify bucket created in S3 console

Update Stack

1. Edit template to add public access block
2. Click "Update" on stack
3. Upload new template
4. Follow update process
5. Observe change set

Cleanup

1. Select stack
2. Click "Delete"
3. Confirm deletion
4. Wait for deletion to complete
5. Verify S3 bucket is deleted

10.4.12 Additional Practice Recommendations

1. **Explore AWS Free Tier Dashboard** - Monitor usage regularly
2. **Try DynamoDB** - Create table, add items, query
3. **Set up CloudTrail** - View API call history
4. **Configure AWS Config** - Track resource configurations
5. **Experiment with Auto Scaling** - Create launch template and auto scaling group
6. **Use AWS CLI** - Install and configure, run basic commands
7. **Explore different regions** - Note service availability differences
8. **Review service quotas** - Understand limits
9. **Practice stopping/starting resources** - Understand state management
10. **Document your learnings** - Take notes, create diagrams

Important

Always clean up resources after labs to avoid charges. Set billing alerts and check your billing dashboard daily during hands-on practice.

10.5 Final Preparation Checklist

10.5.1 One Week Before

Complete all study materials

Take at least 3 practice exams

Score consistently above 80%

Review all flagged topics

Revisit Shared Responsibility Model

Memorize support plans

10.5.2 One Day Before

Light review only

Don't study new material

Prepare exam center logistics or test computer

Get good sleep

Relax and stay confident

10.5.3 Exam Day

Arrive early (testing center) or test connection (online)

Bring two forms of ID (testing center)

Read questions carefully

Manage time effectively

Review answers before submitting

10.6 After the Exam

10.6.1 Immediate Next Steps

- Receive preliminary pass/fail immediately
- Official score report within 5 business days
- Digital badge available via Credly
- Certificate downloadable from AWS Certification Account
- Certification valid for 3 years

10.6.2 Maintaining Certification

- Recertify before expiration
- Stay updated with AWS changes
- Pursue associate-level certifications
- Join AWS certification community

Chapter 11

Quick Reference

11.1 Service Cheat Sheet

11.1.1 Compute

- **EC2**: Virtual servers
- **Lambda**: Serverless functions
- **Elastic Beanstalk**: PaaS for web apps
- **Lightsail**: Simple VPS
- **ECS/EKS**: Container orchestration
- **Fargate**: Serverless containers

11.1.2 Storage

- **S3**: Object storage
- **EBS**: Block storage for EC2
- **EFS**: Network file system
- **Storage Gateway**: Hybrid cloud storage
- **Snow Family**: Physical data migration

11.1.3 Database

- **RDS**: Managed relational database
- **Aurora**: High-performance MySQL/PostgreSQL
- **DynamoDB**: NoSQL database
- **ElastiCache**: In-memory cache
- **Redshift**: Data warehouse

11.1.4 Networking

- **VPC**: Virtual private network
- **CloudFront**: CDN
- **Route 53**: DNS
- **ELB**: Load balancing
- **Direct Connect**: Dedicated network connection

11.1.5 Security

- **IAM**: Identity and access management
- **Organizations**: Multi-account management
- **KMS**: Key management
- **Shield**: DDoS protection
- **WAF**: Web application firewall
- **GuardDuty**: Threat detection
- **Macie**: Data privacy

11.1.6 Management

- **CloudWatch**: Monitoring
- **CloudTrail**: API logging
- **Config**: Resource configuration tracking
- **CloudFormation**: Infrastructure as Code
- **Trusted Advisor**: Best practices
- **Systems Manager**: Operations hub

11.2 Acronym Guide

- **AZ**: Availability Zone
- **IAM**: Identity and Access Management
- **VPC**: Virtual Private Cloud
- **EC2**: Elastic Compute Cloud
- **S3**: Simple Storage Service
- **RDS**: Relational Database Service

- **EBS**: Elastic Block Store
- **EFS**: Elastic File System
- **ELB**: Elastic Load Balancing
- **ALB**: Application Load Balancer
- **NLB**: Network Load Balancer
- **CDN**: Content Delivery Network
- **DNS**: Domain Name System
- **NAT**: Network Address Translation
- **ACL**: Access Control List
- **CIDR**: Classless Inter-Domain Routing
- **SLA**: Service Level Agreement
- **RPO**: Recovery Point Objective
- **RTO**: Recovery Time Objective
- **HA**: High Availability
- **DR**: Disaster Recovery

11.3 Exam Day Reminders

Exam Tip

- 90 minutes, 65 questions
- Passing score: 700/1000
- Flag difficult questions for review
- Eliminate obviously wrong answers
- Trust your preparation
- Stay calm and focused

Good luck on your AWS Cloud Practitioner exam!