



# STUDY GUIDE

**JOY W. NG'ARU**

This study guide is free to access, reproduce, and share.  
It is not comprehensive, feel free to add more content and contribute to it on my GitHub repo:  
<https://github.com/JoyNgaru/aws-ccp-study-guide>

### **Acknowledgment**

I curated this study guide while studying for my AWS Certified Cloud Practitioner (CCP) exam. The goal was to break down major concepts into small, easy-to-understand guides. I hope that it will help you in your journey into AWS Cloud Computing.

Special thanks go to AWS/restart, **Emobilis, Ajira Kenya, and Mastercard** Foundation for giving me an opportunity to learn about AWS services.

I cannot forget to thank my Teachops at the program whose support was immensely felt: **Gloria Moraa, Anthony Nzuki, Kevin Kiruri, Naomi Jebet, Reuben Cheruiyot, Collins Mutai, and Benson Mutua.**

Memorable mentions to **Wesley Kibet and Emmanuel Odenyire** for their memorable support throughout the journey.

If you have any questions, kindly reach out to me at: [joywanjikum@gmail.com](mailto:joywanjikum@gmail.com) and I will be more than willing to assist. I wish you all the best!

**List of topics and subtopics for the AWS Cloud Practitioner exam**

Main Topic	Subtopics
<b>AWS Cloud Concepts</b>	- Global Infrastructure (Regions, Availability Zones, Edge Locations)
	- AWS Cloud Value Proposition (Scalability, Elasticity, Cost-Efficiency)
<b>Security and Compliance</b>	- Shared Responsibility Model (Responsibilities of AWS and the Customer)
	- Identity and Access Management (IAM) (Users, Groups, Roles, Policies)
	- AWS Web Application Firewall (WAF) (Protection from Web Exploits)
<b>Technology</b>	- Core Services (EC2, S3, RDS, Lambda, VPC)
	- Management and Governance (CloudWatch, CloudTrail, AWS Config)
	- Basic Architectural Principles (Fault Tolerance, High Availability, Load Balancing)
<b>Billing and Pricing</b>	- Pricing Models (On-Demand, Reserved, Spot Instances)
	- Billing and Account Management (Cost Explorer, Budgets)
	- Total Cost of Ownership (TCO) (Calculating and Optimizing TCO)
<b>AWS Cloud Adoption Framework</b>	- Overview (Aligning Technology and Business Strategies)
	- CAF Perspectives (Business, People, Governance, Platform, Security, Operations)
<b>AWS Well-Architected Framework</b>	- Pillars (Operational Excellence, Security, Reliability, Performance Efficiency, Cost Optimization)
<b>AWS Support Plans</b>	- Basic (Free Account and Billing Support, Documentation, Forums)
	- Developer (Business Hours Access, Guidance, <24h Case Response Time)
	- Business (24/7 Access, Faster Response Times, Contextual Guidance)
	- Enterprise (TAM, Concierge Support, Training for Complex Use Cases)

## AWS Global Infrastructure

The AWS Global Infrastructure is designed to provide **high availability, fault tolerance, and low latency for users around the world**. It consists of Regions, Availability Zones (AZs), Edge Locations, and data centers.

### 1. Regions

- **Description:** AWS Regions are geographically distinct locations that **consist of multiple, isolated, and physically separate Availability Zones**. Each Region is fully independent and designed to provide services and data replication within that Region.
- **Purpose:** Regions help AWS customers to deploy applications close to their end-users, ensuring **lower latency and better performance**. They also allow customers to comply with legal and regulatory requirements by choosing where to store and process data.
- **Example:** North America (N. Virginia), Europe (Ireland), Asia Pacific (Tokyo).

### 2. Availability Zones (AZs)

- **Description:** Availability Zones are isolated locations within an AWS Region. Each AZ **consists of one or more data centers** equipped with independent power, cooling, and networking.
- **Purpose:** AZs are designed to be independent from failures in other AZs, providing high availability and fault tolerance. Customers can architect their applications to use multiple AZs to protect against data center failures.
- **Example:** A Region like N. Virginia (us-east-1) might have multiple AZs, labeled as us-east-1a, us-east-1b, us-east-1c, etc.

### 3. Edge Locations

- **Description:** Edge Locations are **endpoints for AWS services used to cache copies of content closer to end-users**. They are part of AWS's content delivery network (CDN) called Amazon CloudFront.
- **Purpose:** They help **deliver content with lower latency and higher data transfer speeds**. Edge Locations are also used for services like AWS Global Accelerator and AWS Shield.
- **Example:** Edge Locations are globally distributed in major cities to serve content more efficiently to end-users.

### 4. Data Centers

- **Description:** Data centers are the physical facilities where AWS hosts its servers and networking equipment. Each data center is part of an Availability Zone, and multiple data centers can exist within a single AZ.
- **Purpose:** Data centers provide the foundational infrastructure for all AWS services, ensuring secure, scalable, and reliable operations.

## Key Features of AWS Global Infrastructure

1. **Scalability and Flexibility:** The infrastructure is designed to support a wide range of applications, from small startups to large enterprises, providing scalable resources as needed.
2. **Resiliency and Redundancy:** With multiple Regions, AZs, and data centers, AWS ensures high availability and data durability. Customers can replicate data across Regions for disaster recovery or across AZs for high availability.
3. **Security and Compliance:** AWS data centers are equipped with state-of-the-art security measures. The infrastructure is designed to comply with various global standards and regulations, providing a secure environment for sensitive data.

## How to Use This Infrastructure

- **Deploying Applications:** Choose a Region that best meets your needs for latency, legal requirements, and customer proximity. Utilize multiple AZs within a Region for high availability and fault tolerance.
- **Content Delivery:** Use Edge Locations via Amazon CloudFront to cache content closer to end-users, improving performance and reducing load on origin servers.
- **Disaster Recovery:** Implement cross-Region replication and backups to protect against regional failures and ensure business continuity.

**Basic Architectural Principles** - crucial for designing robust, scalable, and efficient systems.

Principle	Description	AWS Services & Features
Scalability	Design systems to handle varying workloads by scaling up (adding resources) or scaling out (adding instances).	- Amazon EC2 Auto Scaling - AWS Lambda - Amazon RDS Read Replicas
High Availability	Ensure the system is available and operational at all times by using multiple resources across different Availability Zones.	- Amazon Route 53 - Elastic Load Balancing (ELB) - Amazon RDS Multi-AZ Deployments
Fault Tolerance	Design systems to continue operating correctly even in the event of hardware or software failures.	- Amazon S3 Cross-Region Replication - AWS CloudWatch Alarms - AWS Elastic Beanstalk
Security	Protect data and resources from unauthorized access and vulnerabilities by implementing strong security measures.	- AWS Identity and Access Management (IAM) - AWS KMS - AWS Shield
Cost Optimization	Manage and optimize costs by using resources efficiently and implementing cost-saving strategies.	- AWS Cost Explorer - AWS Savings Plans - Amazon EC2 Spot Instances
Performance Efficiency	Design systems to provide high performance and optimize resource utilization to meet performance goals.	- Amazon CloudFront - AWS Global Accelerator - Amazon DynamoDB
Operational Excellence	Ensure systems are easy to manage, monitor, and maintain with proper operational processes and automation.	- AWS Systems Manager - Amazon CloudWatch - AWS Config

**Commented [JN1]: Summary**

- **Scalability:** Ability to handle increased loads by adding resources or instances.
- **High Availability:** System remains operational across multiple Availability Zones.
- **Fault Tolerance:** System continues to function despite failures.
- **Security:** Protection of data and resources from unauthorized access.
- **Cost Optimization:** Efficient use of resources to manage and reduce costs.
- **Performance Efficiency:** Optimization of resource use to meet performance requirements.
- **Operational Excellence:** Effective management, monitoring, and maintenance of systems.
- **Modularity:** Building systems from independent, manageable components.
- **Decoupling:** Reducing dependencies between system components for flexibility.
- **Automation:** Streamlining processes to minimize manual intervention and errors.

Principle	Description	AWS Services & Features
Modularity	Design systems using modular components that can be independently developed, tested, and scaled.	- AWS Lambda - Amazon ECS - AWS Step Functions
Decoupling	Separate components to reduce dependencies and improve system flexibility and fault tolerance.	- Amazon SQS - Amazon SNS - AWS EventBridge
Automation	Automate repetitive tasks and processes to reduce human error and increase efficiency.	- AWS CloudFormation - AWS CodePipeline - AWS OpsWorks



## AWS Cloud Adoption Framework (CAF)

The AWS Cloud Adoption Framework (CAF) provides a structured approach to help organizations transition to the cloud. It is organized into six perspectives, each focusing on specific aspects of cloud adoption.

Perspective	Description	Focus Areas
Business	Ensures that IT aligns with business needs and outcomes. Focuses on business capabilities, value realization, and organizational transformation.	Value Proposition, Business Case, Business Outcomes, KPI and Metrics
People	Addresses organizational culture and change management, emphasizing the people aspect of cloud adoption.	Organizational Change Management, Training, and Certification, Role and Skill Development
Governance	Focuses on aligning IT strategy with business strategy, ensuring compliance, risk management, and the establishment of effective governance.	Cloud Strategy, Cloud Governance, Risk Management, Financial Management, Compliance
Platform	Focuses on the design, implementation, and optimization of the cloud environment. Ensures the environment is scalable, resilient, and secure.	Cloud Architecture, Migration, Integration, Resource Provisioning, Optimization
Security	Ensures the organization has a robust security posture, addressing security management and compliance in the cloud environment.	Identity and Access Management, Data Protection, Infrastructure Security, Threat Detection and Response
Operations	Focuses on operational excellence in the cloud, including monitoring, incident management, and service management.	Operations Management, Monitoring and Reporting, Automation, Incident and Problem Management

**Commented [JN2]:** Developing a business case for cloud adoption, understanding the financial impact, and aligning cloud initiatives with business strategies.

**Commented [JN3]:** Identifying key roles required for cloud operations, training staff on cloud technologies, and fostering a culture of innovation and agility.

**Commented [JN4]:** Establishing cloud governance policies, setting up security and compliance controls, and managing risks associated with cloud adoption.

**Commented [JN5]:** Designing the foundational infrastructure, selecting cloud services, and implementing security measures for applications and data

**Commented [JN6]:** Implementing security best practices, managing identity and access controls, and ensuring data protection and compliance.

**Commented [JN7]:** Setting up monitoring and logging, managing incidents and problems, and optimizing operations for efficiency and cost-effectiveness.

**AWS Cloud Value Proposition** - benefits and advantages of using AWS Cloud services.

Aspect	Description
Cost Efficiency	<b>Pay-as-You-Go Pricing:</b> Only pay for what you use, with no upfront costs or long-term commitments. <b>Cost Savings:</b> Discounts available through Reserved Instances, Savings Plans, and Spot Instances.
Scalability	<b>Auto-Scaling:</b> Automatically adjusts resources based on demand. <b>Elasticity:</b> Easily scale up or down to match workload requirements.
Global Reach	<b>Global Network:</b> AWS has a vast network of data centers worldwide, enabling low-latency access and high availability. <b>Regional Services:</b> Deploy applications in multiple regions to enhance global presence.
Flexibility and Innovation	<b>Wide Range of Services:</b> Offers over 200 fully featured services, allowing you to choose the right tools for your needs. <b>Rapid Deployment:</b> Quickly deploy applications and services with flexibility and agility.
Security and Compliance	<b>Robust Security:</b> Built-in security features, including encryption, IAM, and compliance certifications. <b>Shared Responsibility Model:</b> Clear division of security responsibilities between AWS and customers.
Reliability and High Availability	<b>Resilience:</b> Designed for fault tolerance and high availability with multiple Availability Zones and automated failover. <b>Backup and Recovery:</b> Comprehensive tools for data backup, disaster recovery, and business continuity.
Performance	<b>Optimized Performance:</b> Services and infrastructure are optimized for high performance and low latency. <b>Advanced Technologies:</b> Leverage cutting-edge technologies like machine learning, artificial intelligence, and serverless computing.
Managed Services	<b>Ease of Management:</b> Managed services reduce the operational overhead of maintaining infrastructure. <b>Focus on Core Business:</b> Allows you to focus on building applications and services rather than managing hardware.
Support and Ecosystem	<b>AWS Support Plans:</b> Access to various levels of support, including 24/7 help from AWS experts. <b>Extensive Ecosystem:</b> Rich ecosystem of partners, third-party tools, and integrations.
Agility and Speed	<b>Faster Time to Market:</b> Accelerate development and deployment cycles. <b>Experimentation:</b> Quickly test and iterate on new ideas with scalable resources.

**Commented [JN8]: Summary**

- Cost Efficiency:** Flexible pricing models and cost-saving options.
- Scalability:** Automatically adjust resources and scale applications easily.
- Global Reach:** Deploy applications globally with a broad network of data centers.
- Flexibility and Innovation:** Wide range of services and rapid deployment capabilities.
- Security and Compliance:** Comprehensive security features and compliance with industry standards.
- Reliability and High Availability:** Built for high availability and fault tolerance.
- Performance:** Optimized for high performance with advanced technologies.
- Managed Services:** Reduced operational overhead with managed services.
- Support and Ecosystem:** Access to support plans and a rich ecosystem of partners and tools.
- Agility and Speed:** Accelerate development and deployment, and experiment with new ideas.

### AWS Well-Architected Framework (WAF)

The AWS Well-Architected Framework (WAF) provides best practices and architectural guidance to help design and operate reliable, secure, efficient, and cost-effective systems in the cloud. It is based on five pillars.

WAF Pillar	Principles
Operational Excellence	<ol style="list-style-type: none"><li>1. Perform operations as code</li><li>2. Annotate documentation</li><li>3. Make frequent, small, reversible changes</li><li>4. Refine operations procedures frequently</li><li>5. Anticipate failure</li><li>6. Learn from operational failures</li></ol>
Security	<ol style="list-style-type: none"><li>1. Implement a strong identity foundation</li><li>2. Enable traceability</li><li>3. Apply security at all layers</li><li>4. Automate security best practices</li><li>5. Protect data in transit and at rest</li><li>6. Prepare for security events</li></ol>
Reliability	<ol style="list-style-type: none"><li>1. Automatically recover from failure</li><li>2. Test recovery procedures</li><li>3. Scale horizontally to increase aggregate system availability</li><li>4. Stop guessing capacity</li><li>5. Manage change in automation</li></ol>
Performance Efficiency	<ol style="list-style-type: none"><li>1. Democratize advanced technologies</li><li>2. Go global in minutes</li><li>3. Use serverless architectures</li><li>4. Experiment more often</li><li>5. Consider mechanical sympathy</li></ol>
Cost Optimization	<ol style="list-style-type: none"><li>1. Implement cloud financial management</li><li>2. Adopt a consumption model</li><li>3. Measure overall efficiency</li><li>4. Stop spending money on undifferentiated heavy lifting</li><li>5. Analyze and attribute expenditure</li></ol>

## AWS Support Plans

Feature	Basic	Developer	Business	Enterprise
Pricing	Free	Starting at \$29/month	Starting at \$100/month	Starting at \$15,000/month
Support Channels	None	Email during business hours	24/7 email, chat, phone	24/7 email, chat, phone
Response Time (Critical)	No guaranteed response	No guaranteed response	< 1 hour	< 15 minutes
Response Time (High)	No guaranteed response	< 12 business hours	< 4 hours	< 1 hour
Response Time (General Guidance)	No guaranteed response	< 24 business hours	< 24 hours	< 24 hours
Access to Trusted Advisor	No	7 core checks	Full set of checks	Full set of checks
Access to AWS Personal Health Dashboard	Yes	Yes	Yes	Yes
Support for Third-Party Software	No	No	Yes	Yes
Architecture Support	No	General Guidance	Contextual guidance	Dedicated Technical Account Manager (TAM) and concierge support
Training & Documentation	Online resources	Online resources	Online resources	Online resources
Account Management	No	No	No	Dedicated Technical Account Manager (TAM)
Proactive Guidance	No	No	Limited	Comprehensive

### Commented [JN9]: Basic Plan:

- Free with all AWS accounts.
- Access to AWS forums, and support for account and billing issues.

### Commented [JN10]: Developer Plan:

- Designed for testing and development.
- Email support during business hours, with longer response times.

### Commented [JN11]: Business Plan:

- For production workloads.
- 24/7 support with faster response times, access to all Trusted Advisor checks, and support for third-party software.

### Commented [JN12]: Enterprise Plan:

- For mission-critical use.
- Fastest response times, dedicated Technical Account Manager, and comprehensive support including architecture reviews and proactive guidance

## AWS BILLING AND PRICING

Aspect	Description
Pricing Models	<ul style="list-style-type: none"><li>- <b>On-Demand:</b> Pay for compute capacity by the hour or second with no long-term commitments.</li><li>- <b>Reserved Instances:</b> Reserve capacity for a 1- or 3-year term at a lower rate compared to On-Demand.</li><li>- <b>Spot Instances:</b> Bid for unused EC2 capacity at reduced prices.</li><li>- <b>Savings Plans:</b> Commit to a specific amount of usage over 1 or 3 years for flexible compute services at discounted rates.</li></ul>
Free Tier	<ul style="list-style-type: none"><li>- <b>12-Month Free Tier:</b> Limited quantities of AWS resources are free for 12 months following your account sign-up date.</li><li>- <b>Always Free:</b> Some AWS services are always free up to a specific limit (e.g., 1 million requests per month for AWS Lambda).</li><li>- <b>Trials:</b> Temporary free trials for certain services.</li></ul>
Billing Methods	<ul style="list-style-type: none"><li>- <b>Pay-as-You-Go:</b> Pay for only what you use, with no upfront costs or long-term contracts.</li><li>- <b>Tiered Pricing:</b> Discounts as usage increases, with price breaks at certain usage thresholds.</li></ul>
Cost Management Tools	<ul style="list-style-type: none"><li>- <b>AWS Cost Explorer:</b> Visualize and analyze your AWS spending.</li><li>- <b>AWS Budgets:</b> Set custom cost and usage budgets and receive alerts</li><li>- <b>AWS Cost and Usage Reports (CUR):</b> Detailed reports for deeper analysis of your usage and costs.</li></ul>
Pricing Calculation	<ul style="list-style-type: none"><li>- <b>Compute:</b> Pricing based on instance type, size, and usage time.</li><li>- <b>Storage:</b> Charged by the amount of data stored and data transfer.</li><li>- <b>Data Transfer:</b> Costs for data transferred in and out of AWS, with different rates for internet and inter-region data transfer.</li></ul>
Support Costs	<ul style="list-style-type: none"><li>- <b>Basic Support:</b> Free, includes access to AWS documentation and forums.</li><li>- <b>Developer Support:</b> Starting at \$29/month, includes email support during business hours.</li><li>- <b>Business Support:</b> Starting at \$100/month, includes 24/7 email, chat, and phone support.</li><li>- <b>Enterprise Support:</b> Starting at \$15,000/month, includes a dedicated Technical Account Manager (TAM) and concierge support.</li></ul>

## Shared Controls Overview:

Shared controls are security and management tasks that require action from both AWS and the customer. While AWS provides the tools and infrastructure, the customer must configure and use them properly.

## Examples of Shared Controls:

1. **Patch Management:**
  - **AWS:** Manages and applies patches to the infrastructure and underlying hardware.
  - **Customer:** Responsible for patching and updating their operating systems, applications, and software running on AWS services.
2. **Configuration Management:**
  - **AWS:** Provides tools and services (like AWS Config) to help track and manage configuration changes.
  - **Customer:** Must use these tools to ensure their resources are correctly configured and remain compliant with security policies.
3. **Awareness and Training:**
  - **AWS:** Provides security and compliance information, best practices, and training resources.
  - **Customer:** Responsible for ensuring their staff is trained and aware of how to securely use AWS services.

## Summary:

- **AWS:** Provides the necessary tools, infrastructure, and resources.
- **Customer:** Uses these tools and resources to manage and secure their applications and data.

AWS Shared Responsibility

**AWS's Responsibilities:**

- **Security *of* the Cloud:** AWS is responsible for protecting the infrastructure that runs all the services offered in the AWS Cloud. This includes:
  - Physical security of data centers
  - Network Infrastructure
  - Hardware and software that make up the AWS cloud

**Customer's Responsibilities:**

- **Security *in* the Cloud:** Customers are responsible for managing and securing their data and applications in the cloud. This includes:
  - Data encryption and protection
  - Identity and access management (IAM)
  - Application-level security
  - Configuring security groups and network access controls
  - Operating system patches and updates

## Security in AWS

Security Aspect	Description	AWS Services
<b>Identity and Access Management (IAM)</b>	Manages user identities and permissions to control access to AWS resources.	- AWS IAM - AWS IAM Roles - AWS IAM Policies
<b>Data Encryption</b>	Protects data at rest and in transit using encryption methods.	- AWS Key Management Service (KMS) - AWS CloudHSM - AWS Certificate Manager (ACM)
<b>Network Security</b>	Controls and monitors network traffic to and from AWS resources.	- Amazon VPC - AWS Security Groups - AWS Network ACLs - AWS Shield - AWS WAF (Web Application Firewall)
<b>Monitoring and Logging</b>	Tracks and logs activities to ensure compliance and detect issues.	- Amazon CloudWatch - AWS CloudTrail - AWS Config
<b>Compliance</b>	Ensures adherence to regulatory and security standards.	- AWS Artifact - AWS Compliance Center
<b>Incident Response</b>	Provides tools and practices for detecting, managing, and responding to security incidents.	- AWS CloudTrail - Amazon GuardDuty - AWS Security Hub
<b>Application Security</b>	Protects applications from vulnerabilities and attacks.	- AWS Inspector - AWS Secrets Manager - AWS Shield - AWS WAF
<b>Infrastructure Security</b>	Secures the underlying infrastructure that supports AWS services.	- AWS Trusted Advisor - AWS Security Best Practices - AWS Systems Manager
<b>Access Control</b>	Ensures only authorized users and systems can access resources.	- AWS IAM - AWS Organizations - AWS Resource Access Manager (RAM)
<b>Data Backup and Recovery</b>	Manages data backup and recovery to protect against data loss.	- AWS Backup - Amazon S3 Versioning - AWS Snapshots



Here's a table with short descriptions of each AWS service mentioned in the security aspects:

AWS Service	Description
AWS IAM	Manages user identities and permissions, allowing you to control access to AWS resources.
AWS IAM Roles	Allows you to define a set of permissions that can be assumed by users or services.
AWS IAM Policies	Defines permissions that are assigned to users, groups, or roles, specifying what actions are allowed.
AWS Key Management Service (KMS)	Provides managed encryption keys for data encryption and decryption.
AWS CloudHSM	Offers hardware security modules (HSMs) for managing and protecting encryption keys.
AWS Certificate Manager (ACM)	Manages SSL/TLS certificates for securing data in transit between your applications and users.
Amazon VPC	Creates a virtual private cloud, allowing you to define and control a virtualized network environment.
AWS Security Groups	Acts as a virtual firewall to control inbound and outbound traffic to your EC2 instances.
AWS Network ACLs	Provides an additional layer of security by controlling traffic at the subnet level.
AWS Shield	Protects against Distributed Denial of Service (DDoS) attacks.
AWS WAF (Web Application Firewall)	Protects web applications by filtering and monitoring HTTP requests based on customizable rules.
Amazon CloudWatch	Monitors and logs metrics and events from AWS resources and applications.
AWS CloudTrail	Records API calls made on your account, providing visibility into user activity and API usage.
AWS Config	Tracks AWS resource configurations and changes, enabling compliance auditing and security analysis.
Amazon GuardDuty	Monitors and analyzes data from various AWS sources to detect potential security threats.
AWS Inspector	Assesses the security and compliance of your applications by running automated security assessments.
AWS Artifact	Provides on-demand access to AWS compliance reports and security and compliance documentation.
AWS Compliance Center	Centralizes AWS compliance resources and information to help manage regulatory compliance.
AWS Security Hub	Aggregates and prioritizes security findings from various AWS services and partner solutions.

AWS Service	Description
<b>AWS Secrets Manager</b>	Manages and rotates secrets such as database credentials and API keys securely.
<b>AWS Systems Manager</b>	Provides operational data from multiple AWS services and automates tasks for system management.
<b>AWS Trusted Advisor</b>	Offers insights and recommendations for improving your AWS infrastructure's security and performance.
<b>AWS Resource Access Manager (RAM)</b>	Shares AWS resources across accounts or within your organization.
<b>AWS Organizations</b>	Manages multiple AWS accounts centrally, applying policies and controls across accounts.
<b>AWS Backup</b>	Centralizes backup management and automates backup tasks for AWS resources.
<b>Amazon S3 Versioning</b>	Keeps multiple versions of objects in S3, allowing you to recover from unintended deletions or overwrites.
<b>AWS Snapshots</b>	Captures point-in-time snapshots of your EBS volumes, enabling data backup and recovery.

This table provides concise descriptions of each AWS service related to security, giving you an overview of their primary functions and uses.

## AWS Compute Services

Service	Description	Primary Use Cases
<b>Amazon EC2 (Elastic Compute Cloud)</b>	Provides scalable virtual servers with customizable configurations for various applications.	Web hosting, application servers, batch processing.
<b>AWS Lambda</b>	Runs code without provisioning or managing servers, triggered by events or requests.	Serverless applications, real-time data processing, backend services.
<b>Amazon ECS (Elastic Container Service)</b>	Container orchestration service that manages Docker containers on a cluster of EC2 instances or Fargate.	Containerized applications, microservices architecture.
<b>Amazon EKS (Elastic Kubernetes Service)</b>	Managed Kubernetes service that simplifies running Kubernetes on AWS without needing to install and manage your own control plane.	Kubernetes-based container orchestration, complex applications.
<b>AWS Fargate</b>	Serverless compute engine for containers that works with ECS and EKS, eliminating the need to manage servers.	Containerized applications without managing infrastructure.
<b>AWS Batch</b>	Manages and runs batch computing workloads at any scale using AWS services.	High-performance computing (HPC), large-scale data processing.
<b>AWS Elastic Beanstalk</b>	Platform as a Service (PaaS) that simplifies the deployment, management, and scaling of applications.	Web applications, APIs, and services in multiple languages.
<b>Amazon Lightsail</b>	Simplified cloud platform with a straightforward interface, pre-configured instances, and a fixed pricing model.	Simple web applications, development environments, and small-scale production workloads.
<b>AWS Outposts</b>	Fully managed service that extends AWS infrastructure, services, and tools to on-premises locations.	Hybrid cloud applications, data residency, local data processing.
<b>AWS Local Zones</b>	Extends AWS infrastructure to geographic locations close to end-users, providing low-latency access.	Low-latency applications, real-time processing, gaming.
<b>AWS Wavelength</b>	Brings AWS infrastructure and services to telecommunications networks for ultra-low latency applications.	Mobile edge computing, 5G applications.

### Commented [JN13]: Summary of Compute Services:

- Amazon EC2: Scalable virtual servers for various applications.
- AWS Lambda: Serverless compute for event-driven applications.
- Amazon ECS: Container orchestration service for managing Docker containers.
- Amazon EKS: Managed Kubernetes service for container orchestration.
- AWS Fargate: Serverless compute for containers, removing the need to manage servers.
- AWS Batch: Manages and executes batch computing workloads.
- AWS Elastic Beanstalk: Platform as a Service for deploying and managing applications.
- Amazon Lightsail: Simple cloud platform with fixed pricing and pre-configured instances.
- AWS Outposts: Extends AWS infrastructure to on-premises locations for hybrid cloud scenarios.
- AWS Local Zones: Provides low-latency access by extending AWS infrastructure to local geographic locations.
- AWS Wavelength: Delivers AWS services to telecommunications networks for ultra-low latency applications.

## AWS Network Services

Service	Description	Primary Use Cases
<b>Amazon VPC (Virtual Private Cloud)</b>	Creates a logically isolated network within the AWS Cloud, allowing you to define a virtualized network environment.	Custom network configurations, secure network environments.
<b>Amazon Route 53</b>	Scalable DNS and domain name registration service that routes end-users to internet applications.	Domain name resolution, routing traffic to applications, DNS management.
<b>AWS Direct Connect</b>	Establishes a dedicated network connection from your premises to AWS, providing a more consistent network experience.	High-bandwidth, low-latency connections, hybrid cloud environments.
<b>AWS VPN (Virtual Private Network)</b>	Connects your on-premises network to AWS through an encrypted VPN connection.	Secure communication between on-premises data centers and AWS.
<b>Elastic Load Balancing (ELB)</b>	Distributes incoming application or network traffic across multiple targets to ensure high availability and reliability.	Load balancing for web applications, microservices architectures.
<b>AWS Global Accelerator</b>	Improves the availability and performance of your applications with global users by routing traffic through AWS's global network.	Global application performance improvement, traffic acceleration.
<b>AWS Transit Gateway</b>	Connects multiple VPCs and on-premises networks through a central gateway for simplified network management.	Network centralization, multi-VPC connectivity.
<b>Amazon CloudFront</b>	Content Delivery Network (CDN) that delivers data, videos, applications, and APIs to users globally with low latency.	Website acceleration, media delivery, API distribution.
<b>AWS PrivateLink</b>	Provides private connectivity between VPCs and services hosted on AWS, keeping traffic within the AWS network.	Securely access AWS services and third-party applications.
<b>AWS Global Accelerator</b>	Improves global application availability and performance by routing traffic through the AWS global network.	Application performance enhancement, global user base support.
<b>AWS App Mesh</b>	Provides application-level networking to manage microservices communication with consistent visibility and network traffic controls.	Microservices communication, network traffic management.
<b>Amazon VPC Traffic Mirroring</b>	Captures and inspects network traffic in your VPC to monitor and analyze network activity.	Network monitoring, security analysis, troubleshooting.

### Commented [JN14]: Summary of Networking Services:

- Amazon VPC: Creates and manages virtual networks within AWS.
- Amazon Route 53: Scalable DNS service for routing and domain management.
- AWS Direct Connect: Dedicated network connection for consistent performance.
- AWS VPN: Secure, encrypted connections between on-premises and AWS.
- Elastic Load Balancing (ELB): Distributes traffic across multiple targets.
- AWS Global Accelerator: Enhances global application performance and availability.
- AWS Transit Gateway: Centralizes network connectivity between VPCs and on-premises networks.
- Amazon CloudFront: CDN for delivering content globally with low latency.
- AWS PrivateLink: Private connectivity between VPCs and AWS services.
- AWS App Mesh: Manages microservices communication with consistent traffic controls.
- Amazon VPC Traffic Mirroring: Monitors and analyzes network traffic.
- AWS Network ACLs: Controls traffic at the subnet level.
- AWS Security Groups: Virtual firewalls for EC2 instance traffic.

Service	Description	Primary Use Cases
<b>AWS Network ACLs</b>	Provides a firewall for controlling inbound and outbound traffic at the subnet level in your VPC.	Network security, traffic filtering at the subnet level.
<b>AWS Security Groups</b>	Acts as a virtual firewall to control inbound and outbound traffic to your EC2 instances.	Instance-level security, traffic control.
<b>AWS Site-to-Site VPN</b>	Establishes a secure connection between your on-premises network and AWS using IPsec.	Secure site-to-site connections, hybrid cloud integration.

## AWS Storage Services

Service	Description	Primary Use Cases
<b>Amazon S3 (Simple Storage Service)</b>	Object storage service that provides scalable, durable, and low-cost storage for a wide variety of data types.	Backup and restore, data archiving, content storage and distribution.
<b>Amazon EBS (Elastic Block Store)</b>	Provides block-level storage volumes for use with Amazon EC2 instances.	Persistent storage for EC2 instances, databases, and applications.
<b>Amazon EFS (Elastic File System)</b>	Fully managed file storage that can be shared across multiple EC2 instances, scalable and elastic.	File storage for applications, shared file systems, content management.
<b>Amazon FSx</b>	Managed file systems that provide highly available and scalable file storage with support for Windows and Lustre.	Windows file systems, high-performance computing workloads.
<b>Amazon S3 Glacier</b>	Low-cost cloud storage service for data archiving and long-term backup with retrieval times ranging from minutes to hours.	Data archiving, compliance and regulatory data retention.
<b>Amazon S3 Glacier Deep Archive</b>	Lowest-cost storage class for long-term data archiving, with retrieval times of up to 12 hours.	Long-term data storage, archival of infrequently accessed data.
<b>AWS Backup</b>	Centralized backup service that automates and centrally manages backups across AWS services and on-premises environments.	Backup management, disaster recovery, compliance.
<b>AWS Snowball</b>	Data transfer service that uses secure devices to move large amounts of data into and out of AWS.	Large-scale data migration, data center decommissioning.
<b>AWS Snowcone</b>	Small, rugged, and portable data transfer device for edge computing and data migration.	Edge data collection, remote data transfer.
<b>AWS Storage Gateway</b>	Hybrid cloud storage service that provides on-premises access to virtually unlimited cloud storage.	Backup and archiving, disaster recovery, cloud integration for on-premises data.
<b>Amazon S3 Intelligent-Tiering</b>	Automatically moves data between different storage tiers based on changing access patterns to optimize costs.	Cost-effective storage management for data with unpredictable access patterns.

### Commented [JN15]: Summary of Storage Services:

- Amazon S3: Scalable object storage for various data types.
- Amazon EBS: Block-level storage for EC2 instances.
- Amazon EFS: Shared file storage across multiple EC2 instances.
- Amazon FSx: Managed Windows and Lustre file systems.
- Amazon S3 Glacier: Low-cost archival storage with variable retrieval times.
- Amazon S3 Glacier Deep Archive: Very low-cost storage for long-term archival.
- AWS Backup: Centralized backup management across AWS and on-premises.
- AWS Snowball: Physical devices for transferring large volumes of data to and from AWS.
- AWS Snowcone: Portable device for edge computing and data migration.
- AWS Storage Gateway: Hybrid cloud storage solution for on-premises and cloud integration.
- Amazon S3 Intelligent-Tiering: Automatically optimizes storage costs by moving data between tiers based on access patterns.

**AWS analytics Services**

Service	Description	Primary Use Cases
<b>Amazon Redshift</b>	Fully managed data warehouse service that allows you to run complex queries and analyze large datasets quickly.	Data warehousing, business intelligence, complex query processing.
<b>Amazon Athena</b>	Serverless interactive query service that allows you to analyze data in Amazon S3 using standard SQL.	Ad-hoc querying, interactive analytics on S3 data.
<b>Amazon EMR (Elastic MapReduce)</b>	Managed Hadoop framework that allows you to process vast amounts of data quickly using open-source tools such as Hadoop, Spark, and HBase.	Big data processing, data transformations, large-scale analytics.
<b>AWS Glue</b>	Fully managed ETL (extract, transform, load) service that simplifies data preparation for analytics.	Data integration, ETL workflows, data cataloging.
<b>Amazon Kinesis</b>	Platform for real-time data streaming and analytics, enabling you to collect, process, and analyze streaming data.	Real-time data processing, log and event data analysis, stream analytics.
<b>Amazon QuickSight</b>	Scalable business intelligence service that provides insights through interactive dashboards and visualizations.	Data visualization, business analytics, interactive reporting.
<b>AWS Data Pipeline</b>	Web service that helps you automate the movement and transformation of data between different AWS compute and storage services.	Data workflows, data processing pipelines, ETL tasks.
<b>Amazon CloudSearch</b>	Managed search service that provides fast and highly scalable search capabilities for your applications.	Website search, application search, log analytics.
<b>AWS OpenSearch Service</b>	Managed search and analytics service that enables you to search, analyze, and visualize data in real-time using OpenSearch (formerly Elasticsearch).	Log analytics, real-time search, application monitoring.
<b>AWS Lake Formation</b>	Simplifies the process of building, securing, and managing data lakes, allowing you to ingest and catalog data from various sources.	Data lake creation, data management, security and access control.
<b>Amazon Managed Streaming for Apache Kafka (MSK)</b>	Fully managed service that makes it easy to build and run applications that use Apache Kafka to process streaming data.	Real-time data processing, event streaming, log aggregation.
<b>Amazon Timestream</b>	Fully managed time series database service optimized for ingesting, storing, and analyzing time-stamped data.	IoT data analysis, monitoring and operational metrics, event tracking.

**Commented [JN16]: Summary of Analytics Services:**

- Amazon Redshift: Managed data warehouse for complex queries and large datasets.
- Amazon Athena: Serverless SQL querying service for analyzing S3 data.
- Amazon EMR: Managed Hadoop framework for big data processing.
- AWS Glue: Managed ETL service for data preparation and cataloging.
- Amazon Kinesis: Real-time data streaming and analytics platform.
- Amazon QuickSight: Business intelligence service for dashboards and visualizations.
- AWS Data Pipeline: Automates data movement and transformation workflows.
- Amazon CloudSearch: Managed search service for applications.
- AWS OpenSearch Service: Search and analytics service using OpenSearch.
- AWS Lake Formation: Simplifies building and managing data lakes.
- Amazon MSK: Managed Apache Kafka service for streaming data.
- Amazon Timestream: Time series database for time-stamped data.
- AWS Glue DataBrew: Visual tool for data cleaning and preparation.

Service	Description	Primary Use Cases
AWS Glue DataBrew	Visual data preparation tool that allows you to clean and normalize data without writing code.	Data cleaning, transformation, data preparation for analysis.



## AWS Machine Learning Services

Service	Description	Primary Use Cases
<b>Amazon SageMaker</b>	Fully managed service that provides tools and workflows for building, training, and deploying machine learning models.	Model development, training, and deployment, end-to-end ML workflows.
<b>AWS Deep Learning AMIs</b>	Pre-built Amazon Machine Images (AMIs) with deep learning frameworks and tools for accelerated development.	Deep learning model development, custom ML model training.
<b>AWS Deep Learning Containers</b>	Docker containers pre-installed with popular deep learning frameworks, enabling flexible ML model development.	Model development and training in containerized environments.
<b>Amazon Comprehend</b>	Natural language processing (NLP) service that extracts insights from text using machine learning.	Text analysis, sentiment analysis, entity recognition.
<b>Amazon Rekognition</b>	Image and video analysis service that uses deep learning to identify objects, people, text, and activities.	Image and video analysis, facial recognition, object detection.
<b>Amazon Textract</b>	Extracts text, forms, and tables from scanned documents using machine learning.	Document analysis, data extraction from forms and tables.
<b>Amazon Translate</b>	Neural machine translation service that provides language translation capabilities.	Language translation for text and documents.
<b>Amazon Polly</b>	Text-to-speech service that converts written text into lifelike speech.	Voiceovers, audiobooks, interactive voice applications.
<b>Amazon Lex</b>	Service for building conversational interfaces using voice and text.	Chatbots, virtual assistants, interactive voice response systems.
<b>Amazon Personalize</b>	Real-time personalization service that uses machine learning to provide personalized recommendations.	Personalized product recommendations, content recommendations.
<b>Amazon Forecast</b>	Time series forecasting service that uses machine learning to predict future values based on historical data.	Demand forecasting, financial forecasting, inventory planning.
<b>Amazon SageMaker Ground Truth</b>	Provides a data labeling service that uses machine learning to improve the efficiency of data labeling tasks.	Labeling datasets for training ML models, creating labeled datasets.
<b>Amazon Augmented AI (A2I)</b>	Human review workflows for ML model predictions, allowing you to integrate human judgment into ML workflows.	Model output verification, data labeling, human-in-the-loop.
<b>AWS Glue DataBrew</b>	Visual data preparation tool that helps you clean and normalize data without writing code.	Data preparation for machine learning, data cleaning.

### Commented [JN17]: Summary of Machine Learning Services:

- Amazon SageMaker: End-to-end service for building, training, and deploying ML models.
- AWS Deep Learning AMIs: Pre-built AMIs with deep learning frameworks for accelerated development.
- AWS Deep Learning Containers: Containerized environments with deep learning frameworks.
- Amazon Comprehend: NLP service for extracting insights from text.
- Amazon Rekognition: Image and video analysis for detecting objects, people, and activities.
- Amazon Textract: Extracts structured data from scanned documents.
- Amazon Translate: Language translation service using neural machine translation.
- Amazon Polly: Converts text into lifelike speech.
- Amazon Lex: Builds conversational interfaces with voice and text.
- Amazon Personalize: Provides personalized recommendations using machine learning.
- Amazon Forecast: Time series forecasting for predicting future values.
- Amazon SageMaker Ground Truth: Data labeling service to create labeled datasets for ML models.
- Amazon Augmented AI (A2I): Human review workflows for integrating human judgment into ML predictions.
- AWS Glue DataBrew: Visual tool for cleaning and preparing data for machine learning.

**AWS Developers Services**

Service	Description	Primary Use Cases
<b>AWS CodeCommit</b>	Fully managed source control service that hosts secure Git repositories.	Version control, code management, collaborative development.
<b>AWS CodeBuild</b>	Fully managed build service that compiles source code, runs tests, and produces software packages.	Continuous integration (CI), automated builds.
<b>AWS CodeDeploy</b>	Automates code deployment to any instance, including EC2 instances and on-premises servers.	Continuous deployment (CD), deployment automation.
<b>AWS CodePipeline</b>	Continuous integration and continuous delivery (CI/CD) service that automates the build, test, and deploy phases.	CI/CD pipeline creation, application delivery automation.
<b>AWS CodeStar</b>	Unified user interface that enables you to manage your software development activities in one place.	Project management, CI/CD setup, team collaboration.
<b>AWS Cloud9</b>	Cloud-based integrated development environment (IDE) that provides a rich code-editing experience.	Cloud-based development, code editing, debugging.
<b>AWS X-Ray</b>	Service that helps developers analyze and debug production applications by providing insights into performance issues.	Distributed tracing, performance monitoring, debugging.
<b>AWS SAM (Serverless Application Model)</b>	Framework for building serverless applications with AWS Lambda and other AWS services.	Serverless application development, deployment, and management.
<b>AWS Amplify</b>	Development platform for building secure, scalable mobile and web applications with ease.	Frontend and backend development for mobile/web apps.
<b>AWS Chalice</b>	Microframework for writing serverless applications in Python.	Serverless application development using Python.
<b>AWS App Runner</b>	Fully managed service for deploying and running containerized web applications and APIs.	Deployment of web applications and APIs from source code or containers.
<b>AWS CDK (Cloud Development Kit)</b>	Open-source software development framework for defining cloud infrastructure using familiar programming languages.	Infrastructure as code (IaC), cloud resource provisioning.

**Commented [JN18]: Summary of Developer Services:**

- AWS CodeCommit: Git-based source control service. For version control
- AWS CodeBuild: Managed build service for compiling code and running tests.
- AWS CodeDeploy: Automates code deployments.
- AWS CodePipeline: CI/CD service for automating build, test, and deploy phases.
- AWS CodeStar: Unified interface for managing development activities.
- AWS Cloud9: Cloud-based IDE for development and debugging.
- AWS X-Ray: Provides insights into application performance and debugging.
- AWS SAM: Framework for building serverless applications.
- AWS Amplify: Platform for building mobile and web applications.
- AWS Chalice: Framework for serverless applications in Python.
- AWS App Runner: Managed service for deploying and running containerized applications.
- AWS CDK: Framework for defining cloud infrastructure using code.
- AWS CloudFormation: IaC service for provisioning AWS resources.
- AWS CLI: Command line tool for managing AWS services.
- AWS SDKs: Libraries for integrating with AWS services from various programming languages.
- AWS Config: Tracks and manages AWS resource configurations and compliance.

Service	Description	Primary Use Cases
<b>AWS CloudFormation</b>	Infrastructure as code (IaC) service that provides a common language for describing and provisioning AWS infrastructure.	Cloud resource provisioning, infrastructure management.
<b>AWS CLI (Command Line Interface)</b>	Unified tool to manage AWS services from the command line.	AWS service management and automation via CLI.
<b>AWS SDKs (Software Development Kits)</b>	Libraries and tools that make it easier to interact with AWS services from various programming languages.	Service integration, application development with AWS services.
<b>AWS Config</b>	Service that provides AWS resource inventory, configuration history, and configuration change notifications to enable security and governance.	Resource tracking, compliance auditing, configuration management.

#### Ways to access and interact with AWS services

Access Method	Description	Primary Use Cases
<b>AWS Management Console</b>	Web-based user interface that provides access to all AWS services and resources for management and configuration.	Service management, resource configuration, monitoring.
<b>AWS Command Line Interface (CLI)</b>	Unified tool to manage AWS services from the command line using commands in shell scripts.	Scripting, automation, and service management via CLI.
<b>AWS SDKs (Software Development Kits)</b>	Libraries and tools for integrating AWS services into applications using various programming languages.	Application development, service integration, programmatic access.
<b>AWS CloudFormation</b>	Infrastructure as Code (IaC) service that allows you to define and provision AWS infrastructure using templates.	Infrastructure provisioning and management through templates.
<b>AWS CDK (Cloud Development Kit)</b>	Framework for defining cloud infrastructure using familiar programming languages, generating CloudFormation templates.	Infrastructure as Code (IaC), cloud resource provisioning.
<b>AWS CloudTrail</b>	Service that records API calls and activity made on your account, allowing you to track changes and access patterns.	Security auditing, compliance, monitoring API activity.
<b>AWS Systems Manager</b>	Provides operational data and automation to manage AWS resources and on-premises servers.	Resource management, operational tasks automation.
<b>AWS Config</b>	Tracks and manages AWS resource configurations and compliance with governance rules.	Configuration tracking, compliance auditing, change management.
<b>AWS Service Catalog</b>	Enables organizations to create and manage approved catalogs of resources for consistent provisioning.	Resource cataloging, centralized resource management.
<b>AWS Console Mobile Application</b>	Mobile app for managing AWS services and resources on the go.	Remote management, on-the-go monitoring.
<b>AWS Systems Manager Session Manager</b>	Provides secure and auditable instance management without needing SSH or RDP access.	Instance management, secure remote access.
<b>AWS Lambda Function URLs</b>	Allows direct access to AWS Lambda functions via HTTP endpoints, facilitating serverless application interactions.	Serverless application integrations, direct function access.

#### Commented [JN19]: Summary of Access Methods:

- AWS Management Console: Web UI for service management and configuration.
- AWS CLI: Command line tool for managing services and scripting.
- AWS SDKs: Libraries for integrating AWS services into applications.
- AWS CloudFormation: IaC service for defining and provisioning infrastructure.
- AWS CDK: Framework for defining cloud infrastructure using code.
- AWS CloudTrail: Tracks and logs API calls and activities.
- AWS Systems Manager: Manages AWS resources and servers with operational data.
- AWS Config: Tracks and manages resource configurations and compliance.
- AWS Service Catalog: Manages and provisions approved resource catalogs.
- AWS Console Mobile App: Mobile app for AWS resource management.
- AWS Systems Manager Session Manager: Provides secure instance management without SSH/RDP.
- AWS Lambda Function URLs: Direct HTTP access to Lambda functions.
- Amazon API Gateway: API management and gateway for secure API interactions.
- AWS App Runner: Deploys and manages containerized web applications and APIs.

#### Commented [JN20]: Access the AWS service

Access Method	Description	Primary Use Cases
Amazon API Gateway	Managed service for creating, publishing, maintaining, monitoring, and securing APIs at any scale.	API management, integration, and routing.
AWS App Runner	Service for deploying containerized web applications and APIs without managing the underlying infrastructure.	Application deployment, API hosting.

**TCO Considerations-** considerations for AWS services involve evaluating various factors that impact the overall cost of using AWS resources over time.

Consideration	Description	Impact on TCO
<b>Initial Setup Costs</b>	Costs associated with setting up AWS resources and infrastructure, including configuration and deployment.	Initial expenses, potential setup and configuration fees.
<b>Ongoing Operational Costs</b>	Recurring costs of running AWS resources, including compute, storage, and data transfer.	Monthly or annual charges based on usage.
<b>Scaling Costs</b>	Costs associated with scaling resources up or down based on demand, including autoscaling and load balancing.	Variable costs that depend on resource utilization.
<b>Data Transfer Costs</b>	Charges for data transferred between AWS services, between AWS and the internet, or between regions.	Impact on costs based on data movement and transfer volume.
<b>Storage Costs</b>	Costs for storing data in various AWS storage services, including standard, infrequent access, and archival options.	Costs vary by storage type and data access frequency.
<b>Compute Costs</b>	Charges for using compute services such as EC2 instances, Lambda functions, and container services.	Costs depend on instance types, usage patterns, and duration.
<b>Licensing Costs</b>	Costs for software licenses required to run on AWS, including third-party software and AWS marketplace products.	Additional charges for licensed software or products.
<b>Management and Administration Costs</b>	Costs related to managing and administrating AWS resources, including monitoring, maintenance, and updates.	Costs associated with operational overhead and staff.
<b>Support Costs</b>	Expenses for AWS support plans, which provide varying levels of technical support and guidance.	Costs depend on the level of support chosen (Basic, Developer, Business, Enterprise).
<b>Training and Development Costs</b>	Costs for training staff to effectively use AWS services and tools, including courses and certifications.	Investment in skills development and training programs.
<b>Security and Compliance Costs</b>	Costs associated with implementing security measures, compliance, and data protection.	Expenses for security tools, audits, and compliance measures.
<b>Disaster Recovery Costs</b>	Costs for setting up and maintaining disaster recovery solutions and backups.	Costs for backup storage, replication, and failover solutions.

**Commented [JN21]: Summary of TCO Considerations:**

- Initial Setup Costs:** One-time expenses for resource setup and deployment.
- Ongoing Operational Costs:** Recurring charges based on resource usage.
- Scaling Costs:** Variable costs related to scaling resources as needed.
- Data Transfer Costs:** Charges for data movement within and outside AWS.
- Storage Costs:** Costs for data storage based on access needs and storage class.
- Compute Costs:** Expenses for using compute resources like EC2 and Lambda.
- Licensing Costs:** Additional charges for software licenses and marketplace products.
- Management and Administration Costs:** Operational costs for managing AWS resources.
- Support Costs:** Charges for AWS support plans and technical assistance.
- Training and Development Costs:** Expenses for training staff on AWS technologies.
- Security and Compliance Costs:** Costs for implementing security and compliance measures.
- Disaster Recovery Costs:** Costs for backup and disaster recovery solutions.
- Cost Optimization Efforts:** Investments in strategies to optimize and reduce AWS costs.

Consideration	Description	Impact on TCO
Cost Optimization Efforts	Efforts to manage and reduce AWS costs, such as reserved instances, savings plans, and resource right-sizing.	Potential savings from optimized resource usage and cost management strategies.

**AWS costing services**

Service	Description	Primary Use Cases
<b>AWS Cost Explorer</b>	Tool that allows you to view and analyze your AWS spending patterns, providing detailed insights into your costs.	Cost analysis, budget tracking, spending reports.
<b>AWS Budgets</b>	Service that lets you set custom cost and usage budgets and receive alerts when your usage exceeds the defined thresholds.	Budget management, cost monitoring, alerting on budget breaches.
<b>AWS Cost and Usage Report (CUR)</b>	Detailed report that provides comprehensive data about your AWS usage and spending.	In-depth cost analysis, financial reporting, data integration.
<b>AWS Cost Anomaly Detection</b>	Uses machine learning to automatically detect unusual spending patterns and alert you to potential anomalies.	Anomaly detection, cost monitoring, identifying unexpected charges.
<b>AWS Savings Plans</b>	Flexible pricing model that provides significant savings over on-demand pricing in exchange for a commitment to a consistent amount of usage over a one- or three-year term.	Cost savings through usage commitment, flexible cost management.
<b>Reserved Instances (RIs)</b>	Provides significant savings compared to on-demand pricing in exchange for committing to use specific instance types for a one- or three-year term.	Long-term cost savings, capacity reservation.
<b>AWS Compute Optimizer</b>	Analyzes your usage patterns and recommends optimal instance types for your workloads based on historical data.	Instance right-sizing, cost optimization, performance improvement.
<b>AWS Trusted Advisor</b>	Provides real-time guidance to help you provision your resources following AWS best practices, including cost optimization.	Cost optimization recommendations, resource utilization analysis.
<b>AWS Cost Categories</b>	Allows you to categorize your AWS costs and usage for more granular cost tracking and reporting.	Cost allocation, detailed financial reporting.
<b>AWS Pricing Calculator</b>	Tool for estimating the cost of AWS services based on your specific usage patterns and requirements.	Cost estimation, budgeting, pricing planning.
<b>AWS Cost Management Dashboard</b>	Centralized view of your cost and usage data, providing insights and actionable recommendations for cost management.	Dashboard for monitoring costs, identifying savings opportunities.
<b>AWS Personal Health Dashboard</b>	Provides alerts and remediation guidance when AWS is experiencing events that may impact your resources.	Incident response, proactive issue management.

**Commented [JN22]: Summary of Cost Management Services:**

- AWS Cost Explorer:** Analyzes spending patterns and provides insights.
- AWS Budgets:** Sets and tracks custom budgets with alerts.
- AWS Cost and Usage Report (CUR):** Detailed data on AWS usage and spending.
- AWS Cost Anomaly Detection:** Detects and alerts on unusual spending patterns.
- AWS Savings Plans:** Provides savings through usage commitment.
- Reserved Instances (RIs):** Offers savings for committed instance usage.
- AWS Compute Optimizer:** Recommends optimal instance types based on usage.
- AWS Trusted Advisor:** Provides best practices and cost optimization recommendations.
- AWS Cost Categories:** Categorizes costs for detailed tracking and reporting.
- AWS Pricing Calculator:** Estimates service costs based on usage.
- AWS Cost Management Dashboard:** Centralized view for cost monitoring and management.
- AWS Personal Health Dashboard:** Alerts and guidance for AWS service impacts.



**AWS compliant environment**

Compliance Standard/Framework	Description	Primary Use Cases
General Data Protection Regulation (GDPR)	European Union regulation for data protection and privacy for all individuals within the EU and EEA.	Data protection, privacy compliance for EU and EEA data.
Health Insurance Portability and Accountability Act (HIPAA)	US law providing data privacy and security provisions for safeguarding medical information.	Compliance for handling and storing protected health information (PHI).
Payment Card Industry Data Security Standard (PCI DSS)	Security standard for organizations that handle credit card information to protect against data breaches.	Compliance for processing, storing, and transmitting payment card information.
Federal Risk and Authorization Management Program (FedRAMP)	US government program providing a standardized approach to security assessment, authorization, and continuous monitoring for cloud products and services.	Compliance for US federal agencies and contractors.
Federal Information Processing Standard (FIPS)	Standards for federal computer systems in the US, including encryption and security protocols.	Compliance for federal information systems and data protection.
ISO/IEC 27001	International standard for information security management systems (ISMS).	Certification for managing and protecting information security.
ISO/IEC 27018	International standard focusing on protecting personal data in the cloud.	Data protection and privacy in cloud computing environments.
Sarbanes-Oxley Act (SOX)	US law aimed at protecting investors by improving the accuracy and reliability of corporate disclosures.	Compliance for financial reporting and internal controls.
International Traffic in Arms Regulations (ITAR)	US regulations controlling the export and import of defense-related articles and services.	Compliance for handling and exporting defense-related information.
Controlled Unclassified Information (CUI)	Standards for handling sensitive but unclassified information within US government agencies.	Compliance for managing sensitive government information.
European Union Agency for Cybersecurity (ENISA)	EU agency that provides guidelines for cybersecurity and data protection.	Compliance with EU cybersecurity practices and recommendations.
Cybersecurity Maturity Model Certification (CMMC)	Framework for ensuring the cybersecurity of defense contractors in the US.	Compliance for defense industry contractors and suppliers.

**Commented [JN23]: Key Compliance Considerations for AWS:**

- 1.Data Protection and Privacy:** Ensure compliance with GDPR, HIPAA, and other data protection regulations for managing and safeguarding sensitive data.
- 2.Financial Reporting:** Adhere to SOX for financial accuracy and internal controls.
- 3.Security Standards:** Implement security practices and controls to meet ISO/IEC 27001, PCI DSS, and other relevant standards.
- 4.Government Regulations:** Comply with FedRAMP, FIPS, ITAR, and CUI for handling government and defense-related information.
- 5.Cybersecurity Best Practices:** Follow guidelines from NIST CSF and ENISA for robust cybersecurity practices.
- 6.Cloud-Specific Compliance:** Utilize AWS compliance features and tools to maintain adherence to required standards and frameworks.

Compliance Standard/Framework	Description	Primary Use Cases
NIST Cybersecurity Framework (CSF)	Framework providing guidelines for improving cybersecurity practices and managing cyber risks.	Cybersecurity risk management, framework adoption for security practices.

AWS provides several tools to help manage and optimize costs, including AWS Budgets, AWS Cost Explorer, and the Total Cost of Ownership (TCO) Calculator. Here's a breakdown of each tool, including its primary functions and use cases:

1. AWS Budgets

Feature	Description	Use Cases
Cost and Usage Budgets	Allows you to set custom budgets for cost, usage, or reserved instances (RIs) and track your progress against them.	Budget planning and tracking Cost control Resource usage monitoring
Alerts and Notifications	Sends notifications via email or SNS when your costs or usage exceed (or are forecasted to exceed) your budget thresholds.	Proactive cost management Alerting for budget breaches
Forecasting	Uses historical data to predict future costs and usage, helping you anticipate budget needs.	Financial forecasting Resource allocation planning

AWS Budgets is particularly useful for setting up proactive cost management strategies, allowing you to define spending thresholds and get alerted when you approach or exceed them.

2. AWS Cost Explorer

Feature	Description	Use Cases
Cost and Usage Analysis	Provides detailed insights into your AWS spending patterns, including historical data and trends.	Cost optimization Spend analysis Usage pattern identification
Filtering and Grouping	Allows you to filter and group costs by various dimensions (e.g., service, linked account, region, tag) for detailed analysis.	Cost allocation Resource optimization
Forecasting and Recommendations	Offers cost forecasts based on historical data and suggests cost-saving measures, such as purchasing Reserved Instances.	Budget planning Resource planning and optimization

AWS Cost Explorer is ideal for analyzing your AWS spending in detail, identifying trends, and understanding where your money is being spent. It provides a comprehensive view of your costs, helping you make informed decisions.

3. Total Cost of Ownership (TCO) Calculator

Feature	Description	Use Cases
Cost Comparison	Compares the costs of running applications in an on-premises or traditional data center environment versus in AWS.	Financial justification for cloud migration Cost-benefit analysis
Customizable Inputs	Allows you to input detailed information about your on-premises setup, including hardware, software, and operational costs.	Tailored cost comparison Scenario analysis
Detailed Reports	Provides detailed reports breaking down costs by category, helping to highlight potential savings from moving to AWS.	Decision-making support Budget planning for migration

The TCO Calculator is primarily used during the planning phase of cloud adoption or migration. It helps businesses understand the potential cost savings and benefits of moving workloads to AWS compared to maintaining on-premises infrastructure.

## Resource tags

Resource tags in AWS are key-value pairs that you can assign to AWS resources to help you organize, manage, and track them more effectively. Tags are metadata, and while they do not affect the resource's behavior, they are extremely useful for several purposes, such as cost allocation, resource management, and access control.

## Key Components of a Tag

- **Key:** A case-sensitive string that identifies the tag. Each resource can have a unique set of keys.
- **Value:** A case-sensitive string that can be used to describe the resource. Each key can have only one value assigned to it.

## Benefits and Use Cases of Resource Tags

1. **Cost Allocation and Management**
  - Tags can be used to categorize and track AWS costs by project, department, or team, enabling more precise cost management and allocation.
  - AWS Cost Explorer and AWS Budgets can use tags to provide cost and usage reports based on tagged resources.
2. **Resource Organization and Management**
  - Tags help organize resources by adding descriptive information, such as environment (e.g., "production", "staging", "development"), owner, or project name.
  - This is particularly useful in large-scale environments with many resources, helping teams quickly identify and manage resources.
3. **Access Control and Security**
  - AWS Identity and Access Management (IAM) policies can use tags to define permissions, allowing more granular control over who can access specific resources.
  - For example, you can restrict access to resources with certain tags to specific users or groups.
4. **Automation**
  - Tags can be used in automation scripts and workflows to dynamically manage resources. For instance, auto-scaling policies can be configured to target resources with specific tags.
  - AWS services like AWS Lambda, AWS Systems Manager, and others can use tags to automate resource management tasks.
5. **Compliance and Auditing**
  - Tags can be used to label resources with information required for compliance, such as data classification (e.g., "sensitive", "public").
  - They assist in maintaining governance and compliance by ensuring that resources are appropriately classified and managed.

## Best Practices for Using Tags

- **Consistent Tagging Strategy:** Establish a consistent tagging strategy across your organization to ensure tags are meaningful and standardized. This includes using standardized keys and values.
- **Tagging Enforcement:** Use AWS Config and AWS Tag Policies to enforce tagging compliance and ensure that all resources are tagged according to the organization's policies.
- **Limit Tag Key and Value Lengths:** AWS recommends limiting the length of tag keys and values to ensure they are manageable. AWS has a maximum limit of 128 characters for keys and 256 characters for values.
- **Document Tags:** Maintain documentation on the purpose and usage of each tag to avoid confusion and ensure proper usage across teams.

### Example Tags

- **Project:** Project: WebsiteRedesign
- **Environment:** Environment: Production
- **Owner:** Owner: JohnDoe
- **CostCenter:** CostCenter: 12345
- **Compliance:** Compliance: GDPR

Tags are a powerful feature in AWS that, when used effectively, can greatly enhance resource management, cost control, and operational efficiency.

**Routing Policies** available in Amazon Route 53

Routing Policy	Description	Use Cases
<b>Simple Routing</b>	Routes traffic to a single resource.	Single web server, single application instance.
<b>Failover Routing</b>	Routes traffic to a primary resource, and if it is unavailable, to a secondary resource.	High-availability and disaster recovery scenarios.
<b>Geolocation Routing</b>	Routes traffic based on the geographic location of the request.	Providing localized content, complying with regulatory requirements, regional services.
<b>Geoproximity Routing</b>	Routes traffic based on the geographic location of resources and users, with optional bias.	Directing traffic to the nearest resource, weighted resource distribution.
<b>Latency Routing</b>	Routes traffic to the resource that provides the lowest latency to the user.	Improving user experience by reducing latency, global applications with multiple endpoints.
<b>Multivalue Answer</b>	Provides multiple IP addresses in response to DNS queries, and allows health checks on each.	Load balancing, fault tolerance for applications requiring multiple IP addresses.
<b>Weighted Routing</b>	Routes traffic to multiple resources based on assigned weights.	A/B testing, load distribution, staged deployments.

**Commented [JN24]:**

- **Simple Routing:** Best for straightforward use cases where a single resource is sufficient.
- **Failover Routing:** Ensures high availability by providing a backup resource.
- **Geolocation Routing:** Directs users to resources based on their geographic location, useful for localized services.
- **Geoproximity Routing:** Similar to geolocation but allows for biasing traffic towards certain resources, providing more control.
- **Latency Routing:** Optimizes user experience by minimizing latency, ideal for applications with global users.
- **Multivalue Answer:** Provides multiple answers with health checks, adding fault tolerance and simple load balancing.
- **Weighted Routing:** Distributes traffic based on weights, useful for load balancing, testing, and phased rollouts.

## AWS Managed Services

Managed services in AWS are fully managed by AWS, meaning AWS handles the underlying infrastructure, patching, updates, and maintenance. These services allow you to focus on your applications and business logic without worrying about infrastructure management.

Service	Description
Amazon RDS	Managed relational database service for MySQL, PostgreSQL, MariaDB, Oracle, and SQL Server.
Amazon ElastiCache	Managed in-memory caching service for Redis and Memcached.
Amazon Redshift	Managed data warehouse service.
Amazon Elasticsearch Service	Managed Elasticsearch service for search, logging, and analytics.
Amazon EMR	Managed Hadoop framework for processing big data.
Amazon Managed Streaming for Apache Kafka (MSK)	Managed service for Apache Kafka.
AWS OpsWorks	Managed instances of Chef and Puppet for configuration management.
AWS Managed Microsoft AD	Managed Active Directory service.
Amazon DocumentDB	Managed NoSQL database service designed for JSON data, compatible with MongoDB.
AWS Backup	Centralized backup service to automate and manage backups across AWS services.



## AWS Serverless Services

Serverless services in AWS allow you to run code, manage data, or integrate applications without provisioning or managing servers. These services automatically scale and only charge you for the resources used.

Service	Description
AWS Lambda	Run code in response to events without provisioning or managing servers.
Amazon S3	Object storage service with a simple web service interface to store and retrieve any amount of data.
Amazon DynamoDB	Fully managed NoSQL database service that provides fast and predictable performance.
Amazon API Gateway	Fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs.
Amazon SNS	Managed service for sending notifications from the cloud.
Amazon SQS	Fully managed message queuing service for decoupling and scaling microservices, distributed systems, and serverless applications.
AWS Step Functions	Coordinate multiple AWS services into serverless workflows so you can build and update apps quickly.
Amazon EventBridge	Serverless event bus service that makes it easy to connect applications using data from your own apps, integrated SaaS applications, and AWS services.
Amazon Kinesis Data Firehose	Fully managed service for real-time streaming data delivery to destinations like Amazon S3, Amazon Redshift, Amazon Elasticsearch Service, and Splunk.
Amazon Athena	Serverless query service that makes it easy to analyze data in Amazon S3 using standard SQL.
AWS Fargate	Serverless compute engine for containers that works with both Amazon ECS and Amazon EKS.

### Comparison Table

Type	Service	Description
Managed	Amazon RDS	Managed relational database service.
	Amazon ElastiCache	Managed in-memory caching service.
	Amazon Redshift	Managed data warehouse service.
	Amazon Elasticsearch Service	Managed Elasticsearch service for search, logging, and analytics.
	Amazon EMR	Managed Hadoop framework for big data processing.
	Amazon Managed Streaming for Apache Kafka	Managed service for Apache Kafka.
	AWS OpsWorks	Managed Chef and Puppet instances for configuration management.
	AWS Managed Microsoft AD	Managed Active Directory service.
	Amazon DocumentDB	Managed NoSQL database service compatible with MongoDB.
	AWS Backup	Centralized backup service to automate and manage backups across AWS services.
Serverless	AWS Lambda	Run code without provisioning or managing servers.
	Amazon S3	Object storage service with a web service interface.
	Amazon DynamoDB	Fully managed NoSQL database service.
	Amazon API Gateway	Create, publish, maintain, monitor, and secure APIs.
	Amazon SNS	Managed service for sending notifications.
	Amazon SQS	Fully managed message queuing service.
	AWS Step Functions	Coordinate multiple AWS services into serverless workflows.
	Amazon EventBridge	Serverless event bus service for connecting applications using data from apps, integrated SaaS applications, and AWS services.

Type	Service	Description
	Amazon Kinesis Data Firehose	Fully managed service for real-time streaming data delivery.
	Amazon Athena	Serverless query service to analyze data in Amazon S3 using SQL.
	AWS Fargate	Serverless compute engine for containers.

This table provides a clear distinction between AWS managed services and serverless services, helping you understand their use cases and functionalities.

## AWS Services by Scope

### Global Services

Service	Description
Amazon Route 53	DNS and domain name registration service, operates globally.
AWS CloudFront	Content delivery network (CDN) service, delivers content globally.
AWS Identity and Access Management (IAM)	Manages users and permissions globally.
AWS Key Management Service (KMS)	Manages encryption keys, with global scope for multi-region access.
AWS WAF & AWS Shield	Web application firewall and DDoS protection service, operates globally.
Amazon S3 (with cross-region replication)	Object storage service, can be accessed globally, with cross-region replication capabilities.

### Regional Services

Service	Description
Amazon EC2	Virtual servers for running applications, deployed in specific regions.
Amazon RDS	Managed relational database service, deployed in specific regions.
Amazon DynamoDB	Fully managed NoSQL database service, with regional endpoints.
Amazon S3	Object storage service, with data residing in specific regions.
Amazon VPC	Virtual Private Cloud, allowing the creation of logically isolated networks within specific regions.
AWS Lambda	Serverless compute service, with regional endpoints.
Amazon Redshift	Data warehouse service, deployed in specific regions.
Amazon Elasticsearch Service	Managed Elasticsearch clusters, deployed in specific regions.
AWS Glue	Managed ETL service, deployed in specific regions.

Service	Description
AWS Step Functions	Serverless orchestration service, with regional endpoints.
AWS Elastic Beanstalk	Service for deploying and scaling web applications, deployed in specific regions.

**Zonal Services**

Service	Description
Amazon EC2 Instances	Virtual servers deployed within specific Availability Zones.
Amazon EBS	Block storage volumes attached to EC2 instances within a specific Availability Zone.
Amazon RDS (Multi-AZ)	Managed relational database service, with data replicated across multiple Availability Zones.
Amazon ElastiCache	In-memory caching service, deployed within specific Availability Zones.
Amazon DynamoDB (with DAX)	Managed NoSQL database with DynamoDB Accelerator (DAX), offering high availability within specific AZs.
Amazon ECS (with EC2 launch type)	Container orchestration service, deploying containers within specific Availability Zones.
Amazon EFS	Managed file storage service, accessible within specific Availability Zones.

**Comparison Table**

Scope	Service	Description
Global	Amazon Route 53	DNS and domain name registration service, operates globally.
	AWS CloudFront	Content delivery network (CDN) service, delivers content globally.
	AWS IAM	Manages users and permissions globally.
	AWS KMS	Manages encryption keys, with global scope for multi-region access.

Scope	Service	Description
	AWS WAF & AWS Shield	Web application firewall and DDoS protection service, operates globally.
	Amazon S3 (cross-region replication)	Object storage service, with global access and cross-region replication.
Regional	Amazon EC2	Virtual servers for running applications, deployed in specific regions.
	Amazon RDS	Managed relational database service, deployed in specific regions.
	Amazon DynamoDB	Fully managed NoSQL database service, with regional endpoints.
	Amazon S3	Object storage service, with data residing in specific regions.
	Amazon VPC	Virtual Private Cloud, creating isolated networks within specific regions.
	AWS Lambda	Serverless compute service, with regional endpoints.
	Amazon Redshift	Data warehouse service, deployed in specific regions.
	Amazon Elasticsearch Service	Managed Elasticsearch clusters, deployed in specific regions.
	AWS Glue	Managed ETL service, deployed in specific regions.
	AWS Step Functions	Serverless orchestration service, with regional endpoints.
	AWS Elastic Beanstalk	Service for deploying and scaling web applications, deployed in specific regions.
Zonal	Amazon EC2 Instances	Virtual servers deployed within specific Availability Zones.
	Amazon EBS	Block storage volumes attached to EC2 instances within a specific Availability Zone.
	Amazon RDS (Multi-AZ)	Managed relational database service, with data replicated across multiple Availability Zones.
	Amazon ElastiCache	In-memory caching service, deployed within specific Availability Zones.

Scope	Service	Description
	Amazon DynamoDB (with DAX)	Managed NoSQL database with DynamoDB Accelerator (DAX), high availability within specific AZs.
	Amazon ECS (EC2 launch type)	Container orchestration service, deploying containers within specific Availability Zones.
	Amazon EFS	Managed file storage service, accessible within specific Availability Zones.

This table categorizes AWS services by their scope, helping you understand where and how these services operate within AWS's global infrastructure.

## AWS Services That Can Be Reserved

Service	Description
Amazon EC2 Reserved Instances (RIs)	Provides a significant discount (up to 75%) compared to On-Demand pricing for EC2 instances. Available in Standard and Convertible types.
Amazon RDS Reserved Instances	Offers cost savings on Amazon RDS (Relational Database Service) instances for MySQL, PostgreSQL, Oracle, SQL Server, and MariaDB.
Amazon Redshift Reserved Nodes	Provides discounts for reserving nodes in Amazon Redshift, a fully managed data warehouse service.
Amazon ElastiCache Reserved Nodes	Offers discounts on Amazon ElastiCache nodes for Redis and Memcached by committing to a one or three-year term.
Amazon DynamoDB Reserved Capacity	Provides cost savings on DynamoDB read and write capacity units (RCUs and WCUs) by committing to a one or three-year term.
Amazon Elasticsearch Service Reserved Instances	Offers discounts on instance usage for Amazon Elasticsearch Service by committing to a one or three-year term.
Amazon EMR Reserved Instances	Provides cost savings on Amazon EMR (Elastic MapReduce) instances by committing to a specific usage term.
Amazon CloudFront Reserved Capacity	Offers discounts on data transfer and request pricing for Amazon CloudFront by committing to a specific amount of data transfer.



## Types of Reserved Pricing

1. **Standard Reserved Instances:** Offer the highest discount but are less flexible. Cannot be modified, only in terms of availability zone and instance size.
2. **Convertible Reserved Instances:** Offer a slightly lower discount compared to standard RIs but provide the flexibility to change the instance type, operating system, and tenancy during the term.
3. **Scheduled Reserved Instances:** Designed for workloads that run on a predictable schedule, such as every Saturday for a few hours. These offer some flexibility and cost savings compared to on-demand instances.

## Payment Options

1. **All Upfront:** Pay the entire cost of the reservation upfront for the highest discount.
2. **Partial Upfront:** Pay a portion of the cost upfront, with the remaining amount spread across the term.
3. **No Upfront:** Pay nothing upfront, with the entire cost spread across the term. This option offers the least discount compared to the other payment options.

**Workloads to Include in a TCO Analysis**

Workload Type	Example Services	Description
Compute	EC2, ECS, Lambda	Running virtual machines, containers, and serverless functions.
Storage	S3, EBS, EFS	Persistent storage solutions for various data types and uses.
Networking	VPC, ELB, Route 53	Networking components like data transfer, load balancing, and DNS.
Databases	RDS, DynamoDB, Redshift, ElastiCache	Managed relational and NoSQL databases, data warehouses, in-memory caching.
Security	IAM, KMS, Security Groups	Services for managing access, encryption, and firewalls.
Monitoring and Management	CloudWatch, CloudTrail, AWS Config	Tools for monitoring, logging, and managing AWS resources.
Development and Test	EC2, Cloud9, CodeBuild	Environments for developing and testing applications.
Backup and Disaster Recovery	AWS Backup, Cross-Region Replication	Solutions for data backup, disaster recovery, and business continuity.
Content Delivery	CloudFront, S3	CDN services for distributing content globally.
AI/ML	SageMaker, Rekognition	Machine learning services for building and deploying models.
Analytics	EMR, Glue, Athena	Services for data processing, ETL, and big data analytics.
Compliance and Governance	Artifact, Organizations	Tools for managing compliance documentation and governance across AWS accounts.

Including these workloads in your TCO analysis will help you get a complete picture of your potential costs when moving to or operating within the AWS cloud.

## AWS Recovery Types

Recovery Type	Description	Typical RTO/RPO	Example Services
Backup and Restore	Simple and cost-effective strategy where data is regularly backed up and can be restored in case of failure.	RTO: Hours to days, RPO: Hours	AWS Backup, Amazon S3, AWS Storage Gateway
Pilot Light	A minimal version of an environment is always running in the cloud, which can be rapidly scaled to full production in case of a disaster.	RTO: Minutes to hours, RPO: Minutes	EC2, Auto Scaling, Amazon RDS
Warm Standby	A scaled-down version of a fully functional environment is running in the cloud, ready to scale up to full production during a disaster.	RTO: Minutes to hours, RPO: Minutes	EC2, RDS, Elastic Load Balancing, Route 53
Multi-Site (Active-Active)	Full production workloads are running simultaneously in two or more locations. Data and traffic are balanced between the sites.	RTO: Near-zero, RPO: Near-zero	Route 53, Global Accelerator, RDS Multi-AZ
Replication (Cross-Region/AZ)	Data is replicated across multiple AWS regions or Availability Zones (AZs) to ensure high availability and data durability.	RTO: Seconds to minutes, RPO: Seconds	S3 Cross-Region Replication, RDS Multi-AZ, DynamoDB Global Tables
Disaster Recovery as a Service (DRaaS)	Managed disaster recovery service provided by AWS or third-party vendors that handles the entire recovery process on behalf of the customer.	RTO: Varies, RPO: Varies	CloudEndure Disaster Recovery, AWS Managed Services

comparison and brief descriptions of AWS EC2, Lambda, ECS, ECR, Fargate, EMR, and Batch:

Service	Description	Purpose	Use Cases
EC2 (Elastic Compute Cloud)	Provides resizable compute capacity in the cloud as virtual machines (VMs).	Offers flexible compute capacity, allowing users to launch and manage VMs.	Running applications that require consistent, dedicated compute resources, hosting websites, running databases, or performing data analysis.
Lambda	A serverless compute service that automatically runs code in response to events and scales automatically.	Enables event-driven compute without managing servers, paying only for compute time used.	Running backend services, processing S3 objects, real-time file processing, and automating operational tasks.
ECS (Elastic Container Service)	A fully managed container orchestration service that supports Docker containers.	Simplifies running, stopping, and managing containers on a cluster of EC2 instances.	Deploying microservices, running distributed applications, or hosting containers.
ECR (Elastic Container Registry)	A fully managed Docker container registry that stores, manages, and deploys Docker container images.	Simplifies storing and managing container images securely, integrated with ECS and Fargate.	Storing and sharing container images used in ECS, EKS, or Fargate deployments.
Fargate	A serverless compute engine for containers that works with ECS and EKS.	Runs containers without needing to manage the underlying infrastructure.	Running microservices and containers in a serverless environment with ECS or EKS.
EMR (Elastic MapReduce)	A managed big data processing service that processes vast amounts of data using Hadoop, Spark, and other frameworks.	Simplifies big data processing by provisioning and managing the infrastructure.	Processing large datasets, running big data frameworks like Hadoop, Spark, or Presto, and performing data transformations and analytics.
Batch	A fully managed service for running batch computing workloads at any scale.	Automates the execution of batch jobs, handling the underlying compute resources.	Running large-scale parallel jobs, high-performance computing tasks, or data processing jobs.

Here's a comparison and brief descriptions of AWS CloudFormation, AWS Systems Manager, and AWS OpsWorks

Service	Description	Purpose	Use Cases
CloudFormation	Automates provisioning and management of AWS resources using IaC.	Infrastructure as Code (IaC) for consistent and repeatable infrastructure deployment.	Deploying and managing complex cloud infrastructures.
Systems Manager	Centralized management tool for AWS resources, providing operational insights and automation.	Centralizes and automates operations and management tasks across AWS environments.	Infrastructure management, patching, monitoring, automation.
OpsWorks	Configuration management service using Chef and Puppet for automated server management.	Continuous configuration management and automation using Chef or Puppet.	Managing complex application stacks and server configurations.

These services cater to different aspects of managing and automating AWS infrastructure, from infrastructure as code with CloudFormation to configuration management with OpsWorks, and centralized operations with Systems Manager.

A comparison and brief descriptions of AWS Lightsail, Elastic Beanstalk, AppSync, and the AWS Cloud Development Kit (CDK):

Service	Description	Purpose	Use Cases
Lightsail	A simplified cloud platform to launch and manage virtual private servers (VPS), databases, and networking.	Provides an easy-to-use platform for developers who need a simplified, cost-effective way to deploy and manage applications.	Hosting simple websites, small web applications, and development environments.
Elastic Beanstalk	An orchestration service for deploying and scaling web applications and services.	Automates the deployment process, including capacity provisioning, load balancing, and auto-scaling.	Rapidly deploying web applications in familiar languages and environments with minimal management overhead.
AppSync	A fully managed service for building scalable GraphQL APIs that securely connect to multiple data sources.	Simplifies the creation of APIs by allowing developers to use GraphQL to access and manipulate data from multiple sources.	Building data-driven apps that require real-time data updates and efficient API management.
AWS CDK	A framework for defining cloud infrastructure using programming languages such as TypeScript, Python, Java, and C#.	Enables infrastructure as code (IaC) by allowing developers to define AWS infrastructure using familiar programming languages.	Automating the provisioning of AWS resources and managing infrastructure using code.

A comparison of AWS CodeDeploy, CodeCommit, CodeStar, and CodeBuild, along with brief descriptions of each service:

Service	Description	Purpose	Use Cases
CodeDeploy	Automates the deployment of applications to various compute services like EC2, Lambda, and on-premises servers.	Facilitates automated deployments to ensure consistent and reliable updates.	Deploying updates to applications across EC2 instances, AWS Lambda, or on-premises servers.
CodeCommit	A fully managed source control service that hosts Git repositories.	Provides secure and scalable Git repositories for source code management.	Hosting private Git repositories for source code and version control.
CodeStar	An integrated development environment (IDE) for building, testing, and deploying applications on AWS.	Simplifies the setup of an entire CI/CD pipeline with integrations for various AWS services.	Quickly setting up a CI/CD pipeline for developing, building, and deploying applications.
CodeBuild	A fully managed build service that compiles source code, runs tests, and produces software packages.	Automates the build process, including compiling code, running tests, and generating artifacts.	Building and testing applications in a continuous integration pipeline.

Differences between S3 Transfer Acceleration, AWS Global Accelerator, and DynamoDB Accelerator (DAX):

Feature	S3 Transfer Acceleration	AWS Global Accelerator	DynamoDB Accelerator (DAX)
Purpose	Speeds up uploads to Amazon S3 from distant locations.	Improves the performance and availability of global applications by routing traffic through the AWS global network.	Caches DynamoDB queries to reduce response times and improve performance.
Use Case	Ideal for quickly uploading large files to S3 from around the world.	Enhances the performance of global applications by directing traffic to the optimal endpoint.	Speeds up read-intensive workloads for DynamoDB by caching results.
Key Benefit	Faster uploads to S3 by leveraging Amazon CloudFront's globally distributed edge locations.	Provides low-latency, high-availability access by using the closest AWS region.	Reduces latency for DynamoDB read operations by serving cached responses.
How it Works	Uses edge locations to accelerate data transfer between your client and S3 bucket.	Uses Anycast IP addresses to route traffic to the best-performing endpoint.	Deploys a cache cluster that sits in front of DynamoDB, serving results from cache instead of the database.



Difference between IGW and NAT gateway

Feature	Internet Gateway (IGW)	NAT Gateway (NAT)
Primary Function	Provides internet access and allows inbound traffic from the internet to public resources.	Allows instances in private subnets to access the internet, but blocks inbound traffic from the internet.
Use Case	For instances that need to be publicly accessible.	For instances that need outbound internet access but should not be publicly accessible.
Traffic Direction	Supports both inbound and outbound traffic.	Supports outbound traffic only; no inbound traffic from the internet.
Subnets	Typically used with public subnets.	Typically used with private subnets.

Scalability and elasticity are key concepts in cloud computing, especially within AWS. They both refer to the ability of a system to handle changes in demand, but they focus on different aspects of how resources are managed.

## Scalability

**Definition:** Scalability is the ability of a system to increase its capacity to handle a growing amount of work by adding resources. In AWS, this typically involves adding more instances (vertical scaling) or distributing the load across more instances (horizontal scaling).

### Key Points:

- **Vertical Scaling:** Increasing the capacity of an existing resource (e.g., upgrading an EC2 instance from a smaller type to a larger one).
- **Horizontal Scaling:** Adding more instances of a resource (e.g., adding more EC2 instances to a load balancer).
- **Use Cases:** Scalability is crucial when you anticipate a steady increase in demand, such as a growing user base or expanding application features.
- **Example in AWS:** Auto Scaling Groups allow you to automatically scale the number of EC2 instances based on demand. Amazon RDS allows you to scale up or out by increasing the size of the database instance or adding read replicas.

## Elasticity

**Definition:** Elasticity is the ability of a system to automatically expand and contract its capacity based on the current load. In AWS, elasticity means that resources can be dynamically allocated or de-allocated in real-time based on demand.

### Key Points:

- **Automatic Adjustment:** Elasticity involves the automatic addition or removal of resources in response to changes in demand.
- **Use Cases:** Elasticity is important for applications with unpredictable or fluctuating workloads, such as online stores with seasonal traffic spikes.
- **Example in AWS:** AWS Lambda automatically scales in response to incoming requests, and EC2 Auto Scaling adjusts the number of instances based on the current load, scaling out during peak times and scaling in when demand decreases.

## High Availability (HA)

**Definition:** High availability refers to a system's ability to remain operational and accessible for a high percentage of time, usually measured as uptime (e.g., 99.9%, 99.99%, etc.). It ensures that services are continuously available with minimal downtime.

### Key Points:

- **Redundancy:** High availability is typically achieved by having redundant components, so if one fails, another can take over.
- **Load Balancing:** Distributing workloads across multiple resources to avoid any single point of failure.
- **Automatic Failover:** If a component fails, the system automatically switches to a backup or standby component with minimal disruption.
- **Example in AWS:** Deploying applications across multiple Availability Zones (AZs) using an Elastic Load Balancer (ELB) ensures high availability. If one AZ goes down, traffic is routed to healthy instances in other AZs.

## Fault Tolerance

**Definition:** Fault tolerance is the ability of a system to continue operating correctly even if one or more components fail. A fault-tolerant system is designed to provide uninterrupted service, with no perceived downtime, even in the event of a failure.

### Key Points:

- **Continuous Operation:** Fault-tolerant systems can handle failures without any interruption in service.
- **No Single Point of Failure:** Fault tolerance involves designing systems so that no single component failure can cause the system to go down.
- **Redundant Systems:** Fault tolerance often involves having completely redundant systems (e.g., active-active configurations) that can seamlessly take over if a failure occurs.
- **Example in AWS:** Using services like Amazon S3 or Amazon Aurora, which replicate data across multiple AZs, ensures that data is still accessible even if an entire AZ fails. Another example is using AWS Auto Scaling in conjunction with multiple instances across different AZs to maintain service continuity.

In AWS, **Operational Excellence** and **Performance Efficiency** are two pillars of the AWS Well-Architected Framework, which provides best practices for designing and operating reliable, secure, efficient, and cost-effective systems in the cloud. Both are focused on optimizing different aspects of cloud operations, but they address different goals and practices.

## Operational Excellence

**Definition:** Operational excellence in AWS refers to the ability to run and monitor systems to deliver business value and to continually improve supporting processes and procedures.

### Key Focus Areas:

- **Organizational Culture:** Encourages a culture of continuous improvement, learning, and sharing of best practices.
- **Design for Operations:** Ensures that the architecture is designed for effective operation and supports ongoing improvements.
- **Automate Processes:** Emphasizes automating operational processes to improve efficiency and reduce the chance of human error.
- **Monitor and Respond:** Involves setting up monitoring and logging to detect issues and respond to them quickly.
- **Manage Change:** Ensures that changes are managed in a controlled and planned way to avoid unintended consequences.

**Example in AWS:** Using AWS Systems Manager to automate routine tasks, monitor operations, and manage changes across AWS resources, ensuring smooth and efficient operations.

## Performance Efficiency

**Definition:** Performance efficiency in AWS refers to the ability to use computing resources efficiently to meet system requirements and to maintain that efficiency as demand changes and technologies evolve.

### Key Focus Areas:

- **Selection:** Choose the right type, size, and configuration of resources based on the workload requirements.
- **Review:** Regularly review and adjust your choices based on changes in the system or new AWS offerings.
- **Monitor:** Continuously monitor the system to ensure that it meets performance objectives.
- **Optimize:** Optimize resources by taking advantage of the elasticity and scalability of the cloud to match demand.
- **Evolve:** Continuously evaluate and implement the latest AWS services and features to improve performance.

**Example in AWS:** Using Amazon EC2 Auto Scaling to automatically adjust the number of instances to match the current load, ensuring optimal performance without over-provisioning.

## Components of Risk and Compliance Program in AWS

Component	Description
Governance Framework	<ul style="list-style-type: none"><li>- Policies and Procedures: Establish cloud-specific policies and procedures that align with industry standards (e.g., GDPR, HIPAA).</li><li>- Roles and Responsibilities: Define roles and responsibilities for compliance and risk management within the organization.</li></ul>
Risk Management	<ul style="list-style-type: none"><li>- Risk Assessment: Identify, assess, and prioritize risks related to cloud adoption and operation.</li><li>- Risk Mitigation: Develop strategies to mitigate identified risks, such as implementing security controls or redundancy.</li><li>- Continuous Monitoring: Regularly monitor and review risks as part of ongoing risk management activities.</li></ul>
Compliance Mapping	<ul style="list-style-type: none"><li>- Regulatory Requirements: Map AWS services and configurations to regulatory requirements (e.g., PCI DSS, ISO 27001).</li><li>- Compliance Controls: Implement and document controls to meet these regulatory requirements, using AWS services like AWS Config, AWS Artifact, and AWS Security Hub.</li></ul>
Security and Access Control	<ul style="list-style-type: none"><li>- Identity and Access Management (IAM): Enforce least privilege access, manage roles and policies, and monitor access logs.</li><li>- Data Encryption: Implement encryption for data at rest and in transit using AWS services like AWS KMS (Key Management Service) and SSL/TLS.</li></ul>
Audit and Reporting	<ul style="list-style-type: none"><li>- Audit Trails: Use AWS CloudTrail to capture detailed logs of API calls and monitor changes to resources.</li><li>- Compliance Reporting: Generate and maintain reports that demonstrate compliance with internal policies and external regulations.</li></ul>
Training and Awareness	<ul style="list-style-type: none"><li>- Staff Training: Regularly train staff on AWS security practices, compliance requirements, and the organization's risk management framework.</li><li>- Awareness Programs: Develop awareness programs to keep staff informed about the importance of compliance and risk management.</li></ul>
Incident Response and Management	<ul style="list-style-type: none"><li>- Incident Response Plan: Develop and test incident response plans that detail actions to take in case of security breaches or compliance violations.</li><li>- Forensic Capabilities: Use AWS services like Amazon GuardDuty and AWS Security Hub to detect, investigate, and respond to security incidents.</li></ul>
Vendor and Third-Party Management	<ul style="list-style-type: none"><li>- Third-Party Risk Management: Assess and manage risks related to third-party vendors and services integrated with AWS.</li></ul>

Component	Description
	- Service Level Agreements (SLAs): Ensure that SLAs with vendors meet the organization's compliance and risk management needs.
Continuous Improvement	- Regular Reviews and Audits: Conduct regular reviews and audits of the compliance program to ensure it remains effective and up to date. - Feedback Loop: Implement a feedback loop to learn from past incidents and continuously improve the risk and compliance program.
Automation and Tooling	- Automated Compliance Checks: Use AWS Config, AWS Security Hub, and AWS Audit Manager to automate compliance checks and ensure continuous compliance. - Infrastructure as Code (IaC): Use tools like AWS CloudFormation to manage and enforce compliance in cloud infrastructure automatically.
Documentation and Record Keeping	- Documentation: Maintain comprehensive documentation of all compliance activities, policies, and procedures. - Record Keeping: Ensure that all records are securely stored and easily accessible for audits and compliance checks.

### AWS Services with Built-in DDoS Protection

Service	Description	DDoS Protection Features
Amazon CloudFront	A content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency.	Integrated with AWS Shield Standard for automatic protection against common DDoS attacks at the edge locations.
AWS Shield	A managed DDoS protection service that safeguards applications running on AWS.	AWS Shield Standard is automatically included at no additional cost, protecting against most common DDoS attacks.
Elastic Load Balancing (ELB)	Automatically distributes incoming application traffic across multiple targets, such as EC2 instances, in one or more Availability Zones.	Protects applications by distributing traffic and mitigating the impact of DDoS attacks through load balancing.
Amazon Route 53	A scalable Domain Name System (DNS) web service designed to route end users to internet applications.	Provides DNS-based protection by absorbing DDoS attacks and using AWS Shield for enhanced protection.
AWS WAF (Web Application Firewall)	Protects web applications by filtering and monitoring HTTP/S requests.	Integrated with AWS Shield to protect against DDoS attacks at the application layer, with rules to block malicious traffic.
Amazon CloudFront + AWS WAF + Shield Advanced	A combination of services that provides comprehensive protection against complex DDoS attacks.	Provides enhanced protection with detailed attack diagnostics and 24/7 access to the AWS DDoS Response Team (DRT).
Amazon S3	A scalable object storage service for data storage and retrieval.	Benefits from AWS Shield and can handle large volumes of traffic, reducing the impact of DDoS attacks.



AWS provides several tools to help manage and optimize costs, including AWS Budgets, AWS Cost Explorer, and the Total Cost of Ownership (TCO) Calculator. Here's a breakdown of each tool, including its primary functions and use cases:

1. AWS Budgets

Feature	Description	Use Cases
Cost and Usage Budgets	Allows you to set custom budgets for cost, usage, or reserved instances (RIs) and track your progress against them.	Budget planning and tracking Cost control Resource usage monitoring
Alerts and Notifications	Sends notifications via email or SNS when your costs or usage exceed (or are forecasted to exceed) your budget thresholds.	Proactive cost management Alerting for budget breaches
Forecasting	Uses historical data to predict future costs and usage, helping you anticipate budget needs.	Financial forecasting Resource allocation planning

AWS Budgets is particularly useful for setting up proactive cost management strategies, allowing you to define spending thresholds and get alerted when you approach or exceed them.

2. AWS Cost Explorer

Feature	Description	Use Cases
Cost and Usage Analysis	Provides detailed insights into your AWS spending patterns, including historical data and trends.	Cost optimization Spend analysis Usage pattern identification
Filtering and Grouping	Allows you to filter and group costs by various dimensions (e.g., service, linked account, region, tag) for detailed analysis.	Cost allocation Resource optimization
Forecasting and Recommendations	Offers cost forecasts based on historical data and suggests cost-saving measures, such as purchasing Reserved Instances.	Budget planning Resource planning and optimization

AWS Cost Explorer is ideal for analyzing your AWS spending in detail, identifying trends, and understanding where your money is being spent. It provides a comprehensive view of your costs, helping you make informed decisions.

3. Total Cost of Ownership (TCO) Calculator

Feature	Description	Use Cases
Cost Comparison	Compares the costs of running applications in an on-premises or traditional data center environment versus in AWS.	Financial justification for cloud migration Cost-benefit analysis
Customizable Inputs	Allows you to input detailed information about your on-premises setup, including hardware, software, and operational costs.	Tailored cost comparison Scenario analysis
Detailed Reports	Provides detailed reports breaking down costs by category, helping to highlight potential savings from moving to AWS.	Decision-making support Budget planning for migration

The TCO Calculator is primarily used during the planning phase of cloud adoption or migration. It helps businesses understand the potential cost savings and benefits of moving workloads to AWS compared to maintaining on-premises infrastructure.

## Resource tags

Resource tags in AWS are key-value pairs that you can assign to AWS resources to help you organize, manage, and track them more effectively. Tags are metadata, and while they do not affect the resource's behavior, they are extremely useful for several purposes, such as cost allocation, resource management, and access control.

## Key Components of a Tag

- **Key:** A case-sensitive string that identifies the tag. Each resource can have a unique set of keys.
- **Value:** A case-sensitive string that can be used to describe the resource. Each key can have only one value assigned to it.

## Benefits and Use Cases of Resource Tags

1. **Cost Allocation and Management**
  - Tags can be used to categorize and track AWS costs by project, department, or team, enabling more precise cost management and allocation.
  - AWS Cost Explorer and AWS Budgets can use tags to provide cost and usage reports based on tagged resources.
2. **Resource Organization and Management**
  - Tags help organize resources by adding descriptive information, such as environment (e.g., "production", "staging", "development"), owner, or project name.
  - This is particularly useful in large-scale environments with many resources, helping teams quickly identify and manage resources.
3. **Access Control and Security**
  - AWS Identity and Access Management (IAM) policies can use tags to define permissions, allowing more granular control over who can access specific resources.
  - For example, you can restrict access to resources with certain tags to specific users or groups.
4. **Automation**
  - Tags can be used in automation scripts and workflows to dynamically manage resources. For instance, auto-scaling policies can be configured to target resources with specific tags.
  - AWS services like AWS Lambda, AWS Systems Manager, and others can use tags to automate resource management tasks.
5. **Compliance and Auditing**
  - Tags can be used to label resources with information required for compliance, such as data classification (e.g., "sensitive", "public").
  - They assist in maintaining governance and compliance by ensuring that resources are appropriately classified and managed.

## Best Practices for Using Tags

- **Consistent Tagging Strategy:** Establish a consistent tagging strategy across your organization to ensure tags are meaningful and standardized. This includes using standardized keys and values.
- **Tagging Enforcement:** Use AWS Config and AWS Tag Policies to enforce tagging compliance and ensure that all resources are tagged according to the organization's policies.
- **Limit Tag Key and Value Lengths:** AWS recommends limiting the length of tag keys and values to ensure they are manageable. AWS has a maximum limit of 128 characters for keys and 256 characters for values.
- **Document Tags:** Maintain documentation on the purpose and usage of each tag to avoid confusion and ensure proper usage across teams.

### Example Tags

- **Project:** Project: WebsiteRedesign
- **Environment:** Environment: Production
- **Owner:** Owner: JohnDoe
- **CostCenter:** CostCenter: 12345
- **Compliance:** Compliance: GDPR

Tags are a powerful feature in AWS that, when used effectively, can greatly enhance resource management, cost control, and operational efficiency.

**Routing Policies** available in Amazon Route 53

Routing Policy	Description	Use Cases
Simple Routing	Routes traffic to a single resource.	Single web server, single application instance.
Failover Routing	Routes traffic to a primary resource, and if it is unavailable, to a secondary resource.	High-availability and disaster recovery scenarios.
Geolocation Routing	Routes traffic based on the geographic location of the request.	Providing localized content, complying with regulatory requirements, regional services.
Geoproximity Routing	Routes traffic based on the geographic location of resources and users, with optional bias.	Directing traffic to the nearest resource, weighted resource distribution.
Latency Routing	Routes traffic to the resource that provides the lowest latency to the user.	Improving user experience by reducing latency, global applications with multiple endpoints.
Multivalue Answer	Provides multiple IP addresses in response to DNS queries, and allows health checks on each.	Load balancing, fault tolerance for applications requiring multiple IP addresses.
Weighted Routing	Routes traffic to multiple resources based on assigned weights.	A/B testing, load distribution, staged deployments.

**Commented [JN25]:**

- **Simple Routing:** Best for straightforward use cases where a single resource is sufficient.
- **Failover Routing:** Ensures high availability by providing a backup resource.
- **Geolocation Routing:** Directs users to resources based on their geographic location, useful for localized services.
- **Geoproximity Routing:** Similar to geolocation but allows for biasing traffic towards certain resources, providing more control.
- **Latency Routing:** Optimizes user experience by minimizing latency, ideal for applications with global users.
- **Multivalue Answer:** Provides multiple answers with health checks, adding fault tolerance and simple load balancing.
- **Weighted Routing:** Distributes traffic based on weights, useful for load balancing, testing, and phased rollouts.

## AWS Managed Services

Managed services in AWS are fully managed by AWS, meaning AWS handles the underlying infrastructure, patching, updates, and maintenance. These services allow you to focus on your applications and business logic without worrying about infrastructure management.

Service	Description
Amazon RDS	Managed relational database service for MySQL, PostgreSQL, MariaDB, Oracle, and SQL Server.
Amazon ElastiCache	Managed in-memory caching service for Redis and Memcached.
Amazon Redshift	Managed data warehouse service.
Amazon Elasticsearch Service	Managed Elasticsearch service for search, logging, and analytics.
Amazon EMR	Managed Hadoop framework for processing big data.
Amazon Managed Streaming for Apache Kafka (MSK)	Managed service for Apache Kafka.
AWS OpsWorks	Managed instances of Chef and Puppet for configuration management.
AWS Managed Microsoft AD	Managed Active Directory service.
Amazon DocumentDB	Managed NoSQL database service designed for JSON data, compatible with MongoDB.
AWS Backup	Centralized backup service to automate and manage backups across AWS services.

## AWS Serverless Services

Serverless services in AWS allow you to run code, manage data, or integrate applications without provisioning or managing servers. These services automatically scale and only charge you for the resources used.

Service	Description
AWS Lambda	Run code in response to events without provisioning or managing servers.
Amazon S3	Object storage service with a simple web service interface to store and retrieve any amount of data.
Amazon DynamoDB	Fully managed NoSQL database service that provides fast and predictable performance.
Amazon API Gateway	Fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs.
Amazon SNS	Managed service for sending notifications from the cloud.
Amazon SQS	Fully managed message queuing service for decoupling and scaling microservices, distributed systems, and serverless applications.
AWS Step Functions	Coordinate multiple AWS services into serverless workflows so you can build and update apps quickly.
Amazon EventBridge	Serverless event bus service that makes it easy to connect applications using data from your own apps, integrated SaaS applications, and AWS services.
Amazon Kinesis Data Firehose	Fully managed service for real-time streaming data delivery to destinations like Amazon S3, Amazon Redshift, Amazon Elasticsearch Service, and Splunk.
Amazon Athena	Serverless query service that makes it easy to analyze data in Amazon S3 using standard SQL.
AWS Fargate	Serverless compute engine for containers that works with both Amazon ECS and Amazon EKS.

### Comparison Table

Type	Service	Description
Managed	Amazon RDS	Managed relational database service.
	Amazon ElastiCache	Managed in-memory caching service.
	Amazon Redshift	Managed data warehouse service.
	Amazon Elasticsearch Service	Managed Elasticsearch service for search, logging, and analytics.
	Amazon EMR	Managed Hadoop framework for big data processing.
	Amazon Managed Streaming for Apache Kafka	Managed service for Apache Kafka.
	AWS OpsWorks	Managed Chef and Puppet instances for configuration management.
	AWS Managed Microsoft AD	Managed Active Directory service.
	Amazon DocumentDB	Managed NoSQL database service compatible with MongoDB.
	AWS Backup	Centralized backup service to automate and manage backups across AWS services.
Serverless	AWS Lambda	Run code without provisioning or managing servers.
	Amazon S3	Object storage service with a web service interface.
	Amazon DynamoDB	Fully managed NoSQL database service.
	Amazon API Gateway	Create, publish, maintain, monitor, and secure APIs.
	Amazon SNS	Managed service for sending notifications.
	Amazon SQS	Fully managed message queuing service.
	AWS Step Functions	Coordinate multiple AWS services into serverless workflows.
	Amazon EventBridge	Serverless event bus service for connecting applications using data from apps, integrated SaaS applications, and AWS services.



Type	Service	Description
	Amazon Kinesis Data Firehose	Fully managed service for real-time streaming data delivery.
	Amazon Athena	Serverless query service to analyze data in Amazon S3 using SQL.
	AWS Fargate	Serverless compute engine for containers.

This table provides a clear distinction between AWS managed services and serverless services, helping you understand their use cases and functionalities.

## AWS Services by Scope

### Global Services

Service	Description
Amazon Route 53	DNS and domain name registration service, operates globally.
AWS CloudFront	Content delivery network (CDN) service, delivers content globally.
AWS Identity and Access Management (IAM)	Manages users and permissions globally.
AWS Key Management Service (KMS)	Manages encryption keys, with global scope for multi-region access.
AWS WAF & AWS Shield	Web application firewall and DDoS protection service, operates globally.
Amazon S3 (with cross-region replication)	Object storage service, can be accessed globally, with cross-region replication capabilities.

### Regional Services

Service	Description
Amazon EC2	Virtual servers for running applications, deployed in specific regions.
Amazon RDS	Managed relational database service, deployed in specific regions.
Amazon DynamoDB	Fully managed NoSQL database service, with regional endpoints.
Amazon S3	Object storage service, with data residing in specific regions.
Amazon VPC	Virtual Private Cloud, allowing the creation of logically isolated networks within specific regions.
AWS Lambda	Serverless compute service, with regional endpoints.
Amazon Redshift	Data warehouse service, deployed in specific regions.
Amazon Elasticsearch Service	Managed Elasticsearch clusters, deployed in specific regions.
AWS Glue	Managed ETL service, deployed in specific regions.

Service	Description
AWS Step Functions	Serverless orchestration service, with regional endpoints.
AWS Elastic Beanstalk	Service for deploying and scaling web applications, deployed in specific regions.

**Zonal Services**

Service	Description
Amazon EC2 Instances	Virtual servers deployed within specific Availability Zones.
Amazon EBS	Block storage volumes attached to EC2 instances within a specific Availability Zone.
Amazon RDS (Multi-AZ)	Managed relational database service, with data replicated across multiple Availability Zones.
Amazon ElastiCache	In-memory caching service, deployed within specific Availability Zones.
Amazon DynamoDB (with DAX)	Managed NoSQL database with DynamoDB Accelerator (DAX), offering high availability within specific AZs.
Amazon ECS (with EC2 launch type)	Container orchestration service, deploying containers within specific Availability Zones.
Amazon EFS	Managed file storage service, accessible within specific Availability Zones.

**Comparison Table**

Scope	Service	Description
Global	Amazon Route 53	DNS and domain name registration service, operates globally.
	AWS CloudFront	Content delivery network (CDN) service, delivers content globally.
	AWS IAM	Manages users and permissions globally.
	AWS KMS	Manages encryption keys, with global scope for multi-region access.

Scope	Service	Description
	AWS WAF & AWS Shield	Web application firewall and DDoS protection service, operates globally.
	Amazon S3 (cross-region replication)	Object storage service, with global access and cross-region replication.
Regional	Amazon EC2	Virtual servers for running applications, deployed in specific regions.
	Amazon RDS	Managed relational database service, deployed in specific regions.
	Amazon DynamoDB	Fully managed NoSQL database service, with regional endpoints.
	Amazon S3	Object storage service, with data residing in specific regions.
	Amazon VPC	Virtual Private Cloud, creating isolated networks within specific regions.
	AWS Lambda	Serverless compute service, with regional endpoints.
	Amazon Redshift	Data warehouse service, deployed in specific regions.
	Amazon Elasticsearch Service	Managed Elasticsearch clusters, deployed in specific regions.
	AWS Glue	Managed ETL service, deployed in specific regions.
	AWS Step Functions	Serverless orchestration service, with regional endpoints.
	AWS Elastic Beanstalk	Service for deploying and scaling web applications, deployed in specific regions.
Zonal	Amazon EC2 Instances	Virtual servers deployed within specific Availability Zones.
	Amazon EBS	Block storage volumes attached to EC2 instances within a specific Availability Zone.
	Amazon RDS (Multi-AZ)	Managed relational database service, with data replicated across multiple Availability Zones.
	Amazon ElastiCache	In-memory caching service, deployed within specific Availability Zones.

Scope	Service	Description
	Amazon DynamoDB (with DAX)	Managed NoSQL database with DynamoDB Accelerator (DAX), high availability within specific AZs.
	Amazon ECS (EC2 launch type)	Container orchestration service, deploying containers within specific Availability Zones.
	Amazon EFS	Managed file storage service, accessible within specific Availability Zones.

This table categorizes AWS services by their scope, helping you understand where and how these services operate within AWS's global infrastructure.

## AWS Services That Can Be Reserved

Service	Description
Amazon EC2 Reserved Instances (RIs)	Provides a significant discount (up to 75%) compared to On-Demand pricing for EC2 instances. Available in Standard and Convertible types.
Amazon RDS Reserved Instances	Offers cost savings on Amazon RDS (Relational Database Service) instances for MySQL, PostgreSQL, Oracle, SQL Server, and MariaDB.
Amazon Redshift Reserved Nodes	Provides discounts for reserving nodes in Amazon Redshift, a fully managed data warehouse service.
Amazon ElastiCache Reserved Nodes	Offers discounts on Amazon ElastiCache nodes for Redis and Memcached by committing to a one or three-year term.
Amazon DynamoDB Reserved Capacity	Provides cost savings on DynamoDB read and write capacity units (RCUs and WCUs) by committing to a one or three-year term.
Amazon Elasticsearch Service Reserved Instances	Offers discounts on instance usage for Amazon Elasticsearch Service by committing to a one or three-year term.
Amazon EMR Reserved Instances	Provides cost savings on Amazon EMR (Elastic MapReduce) instances by committing to a specific usage term.
Amazon CloudFront Reserved Capacity	Offers discounts on data transfer and request pricing for Amazon CloudFront by committing to a specific amount of data transfer.

## Types of Reserved Pricing

1. **Standard Reserved Instances:** Offer the highest discount but are less flexible. Cannot be modified, only in terms of availability zone and instance size.
2. **Convertible Reserved Instances:** Offer a slightly lower discount compared to standard RIs but provide the flexibility to change the instance type, operating system, and tenancy during the term.
3. **Scheduled Reserved Instances:** Designed for workloads that run on a predictable schedule, such as every Saturday for a few hours. These offer some flexibility and cost savings compared to on-demand instances.

## Payment Options

1. **All Upfront:** Pay the entire cost of the reservation upfront for the highest discount.
2. **Partial Upfront:** Pay a portion of the cost upfront, with the remaining amount spread across the term.
3. **No Upfront:** Pay nothing upfront, with the entire cost spread across the term. This option offers the least discount compared to the other payment options.

### Workloads to Include in a TCO Analysis

Workload Type	Example Services	Description
Compute	EC2, ECS, Lambda	Running virtual machines, containers, and serverless functions.
Storage	S3, EBS, EFS	Persistent storage solutions for various data types and uses.
Networking	VPC, ELB, Route 53	Networking components like data transfer, load balancing, and DNS.
Databases	RDS, DynamoDB, Redshift, ElastiCache	Managed relational and NoSQL databases, data warehouses, in-memory caching.
Security	IAM, KMS, Security Groups	Services for managing access, encryption, and firewalls.
Monitoring and Management	CloudWatch, CloudTrail, AWS Config	Tools for monitoring, logging, and managing AWS resources.
Development and Test	EC2, Cloud9, CodeBuild	Environments for developing and testing applications.
Backup and Disaster Recovery	AWS Backup, Cross-Region Replication	Solutions for data backup, disaster recovery, and business continuity.
Content Delivery	CloudFront, S3	CDN services for distributing content globally.
AI/ML	SageMaker, Rekognition	Machine learning services for building and deploying models.
Analytics	EMR, Glue, Athena	Services for data processing, ETL, and big data analytics.
Compliance and Governance	Artifact, Organizations	Tools for managing compliance documentation and governance across AWS accounts.

Including these workloads in your TCO analysis will help you get a complete picture of your potential costs when moving to or operating within the AWS cloud.



## AWS Recovery Types

Recovery Type	Description	Typical RTO/RPO	Example Services
Backup and Restore	Simple and cost-effective strategy where data is regularly backed up and can be restored in case of failure.	RTO: Hours to days, RPO: Hours	AWS Backup, Amazon S3, AWS Storage Gateway
Pilot Light	A minimal version of an environment is always running in the cloud, which can be rapidly scaled to full production in case of a disaster.	RTO: Minutes to hours, RPO: Minutes	EC2, Auto Scaling, Amazon RDS
Warm Standby	A scaled-down version of a fully functional environment is running in the cloud, ready to scale up to full production during a disaster.	RTO: Minutes to hours, RPO: Minutes	EC2, RDS, Elastic Load Balancing, Route 53
Multi-Site (Active-Active)	Full production workloads are running simultaneously in two or more locations. Data and traffic are balanced between the sites.	RTO: Near-zero, RPO: Near-zero	Route 53, Global Accelerator, RDS Multi-AZ
Replication (Cross-Region/AZ)	Data is replicated across multiple AWS regions or Availability Zones (AZs) to ensure high availability and data durability.	RTO: Seconds to minutes, RPO: Seconds	S3 Cross-Region Replication, RDS Multi-AZ, DynamoDB Global Tables
Disaster Recovery as a Service (DRaaS)	Managed disaster recovery service provided by AWS or third-party vendors that handles the entire recovery process on behalf of the customer.	RTO: Varies, RPO: Varies	CloudEndure Disaster Recovery, AWS Managed Services

comparison and brief descriptions of AWS EC2, Lambda, ECS, ECR, Fargate, EMR, and Batch:

Service	Description	Purpose	Use Cases
EC2 (Elastic Compute Cloud)	Provides resizable compute capacity in the cloud as virtual machines (VMs).	Offers flexible compute capacity, allowing users to launch and manage VMs.	Running applications that require consistent, dedicated compute resources, hosting websites, running databases, or performing data analysis.
Lambda	A serverless compute service that automatically runs code in response to events and scales automatically.	Enables event-driven compute without managing servers, paying only for compute time used.	Running backend services, processing S3 objects, real-time file processing, and automating operational tasks.
ECS (Elastic Container Service)	A fully managed container orchestration service that supports Docker containers.	Simplifies running, stopping, and managing containers on a cluster of EC2 instances.	Deploying microservices, running distributed applications, or hosting containers.
ECR (Elastic Container Registry)	A fully managed Docker container registry that stores, manages, and deploys Docker container images.	Simplifies storing and managing container images securely, integrated with ECS and Fargate.	Storing and sharing container images used in ECS, EKS, or Fargate deployments.
Fargate	A serverless compute engine for containers that works with ECS and EKS.	Runs containers without needing to manage the underlying infrastructure.	Running microservices and containers in a serverless environment with ECS or EKS.
EMR (Elastic MapReduce)	A managed big data processing service that processes vast amounts of data using Hadoop, Spark, and other frameworks.	Simplifies big data processing by provisioning and managing the infrastructure.	Processing large datasets, running big data frameworks like Hadoop, Spark, or Presto, and performing data transformations and analytics.
Batch	A fully managed service for running batch computing workloads at any scale.	Automates the execution of batch jobs, handling the underlying compute resources.	Running large-scale parallel jobs, high-performance computing tasks, or data processing jobs.

Here's a comparison and brief descriptions of AWS CloudFormation, AWS Systems Manager, and AWS OpsWorks

Service	Description	Purpose	Use Cases
CloudFormation	Automates provisioning and management of AWS resources using IaC.	Infrastructure as Code (IaC) for consistent and repeatable infrastructure deployment.	Deploying and managing complex cloud infrastructures.
Systems Manager	Centralized management tool for AWS resources, providing operational insights and automation.	Centralizes and automates operations and management tasks across AWS environments.	Infrastructure management, patching, monitoring, automation.
OpsWorks	Configuration management service using Chef and Puppet for automated server management.	Continuous configuration management and automation using Chef or Puppet.	Managing complex application stacks and server configurations.

These services cater to different aspects of managing and automating AWS infrastructure, from infrastructure as code with CloudFormation to configuration management with OpsWorks, and centralized operations with Systems Manager.

A comparison and brief descriptions of AWS Lightsail, Elastic Beanstalk, AppSync, and the AWS Cloud Development Kit (CDK):

Service	Description	Purpose	Use Cases
Lightsail	A simplified cloud platform to launch and manage virtual private servers (VPS), databases, and networking.	Provides an easy-to-use platform for developers who need a simplified, cost-effective way to deploy and manage applications.	Hosting simple websites, small web applications, and development environments.
Elastic Beanstalk	An orchestration service for deploying and scaling web applications and services.	Automates the deployment process, including capacity provisioning, load balancing, and auto-scaling.	Rapidly deploying web applications in familiar languages and environments with minimal management overhead.
AppSync	A fully managed service for building scalable GraphQL APIs that securely connect to multiple data sources.	Simplifies the creation of APIs by allowing developers to use GraphQL to access and manipulate data from multiple sources.	Building data-driven apps that require real-time data updates and efficient API management.
AWS CDK	A framework for defining cloud infrastructure using programming languages such as TypeScript, Python, Java, and C#.	Enables infrastructure as code (IaC) by allowing developers to define AWS infrastructure using familiar programming languages.	Automating the provisioning of AWS resources and managing infrastructure using code.

A comparison of AWS CodeDeploy, CodeCommit, CodeStar, and CodeBuild, along with brief descriptions of each service:

Service	Description	Purpose	Use Cases
CodeDeploy	Automates the deployment of applications to various compute services like EC2, Lambda, and on-premises servers.	Facilitates automated deployments to ensure consistent and reliable updates.	Deploying updates to applications across EC2 instances, AWS Lambda, or on-premises servers.
CodeCommit	A fully managed source control service that hosts Git repositories.	Provides secure and scalable Git repositories for source code management.	Hosting private Git repositories for source code and version control.
CodeStar	An integrated development environment (IDE) for building, testing, and deploying applications on AWS.	Simplifies the setup of an entire CI/CD pipeline with integrations for various AWS services.	Quickly setting up a CI/CD pipeline for developing, building, and deploying applications.
CodeBuild	A fully managed build service that compiles source code, runs tests, and produces software packages.	Automates the build process, including compiling code, running tests, and generating artifacts.	Building and testing applications in a continuous integration pipeline.

Differences between S3 Transfer Acceleration, AWS Global Accelerator, and DynamoDB Accelerator (DAX):

Feature	S3 Transfer Acceleration	AWS Global Accelerator	DynamoDB Accelerator (DAX)
Purpose	Speeds up uploads to Amazon S3 from distant locations.	Improves the performance and availability of global applications by routing traffic through the AWS global network.	Caches DynamoDB queries to reduce response times and improve performance.
Use Case	Ideal for quickly uploading large files to S3 from around the world.	Enhances the performance of global applications by directing traffic to the optimal endpoint.	Speeds up read-intensive workloads for DynamoDB by caching results.
Key Benefit	Faster uploads to S3 by leveraging Amazon CloudFront’s globally distributed edge locations.	Provides low-latency, high-availability access by using the closest AWS region.	Reduces latency for DynamoDB read operations by serving cached responses.
How it Works	Uses edge locations to accelerate data transfer between your client and S3 bucket.	Uses Anycast IP addresses to route traffic to the best-performing endpoint.	Deploys a cache cluster that sits in front of DynamoDB, serving results from cache instead of the database.

Difference between IGW and NAT gateway

Feature	Internet Gateway (IGW)	NAT Gateway (NAT)
Primary Function	Provides internet access and allows inbound traffic from the internet to public resources.	Allows instances in private subnets to access the internet, but blocks inbound traffic from the internet.
Use Case	For instances that need to be publicly accessible.	For instances that need outbound internet access but should not be publicly accessible.
Traffic Direction	Supports both inbound and outbound traffic.	Supports outbound traffic only; no inbound traffic from the internet.
Subnets	Typically used with public subnets.	Typically used with private subnets.

Scalability and elasticity are key concepts in cloud computing, especially within AWS. They both refer to the ability of a system to handle changes in demand, but they focus on different aspects of how resources are managed.

## Scalability

Definition: Scalability is the ability of a system to increase its capacity to handle a growing amount of work by adding resources. In AWS, this typically involves adding more instances (vertical scaling) or distributing the load across more instances (horizontal scaling).

Key Points:

- Vertical Scaling: Increasing the capacity of an existing resource (e.g., upgrading an EC2 instance from a smaller type to a larger one).
- Horizontal Scaling: Adding more instances of a resource (e.g., adding more EC2 instances to a load balancer).
- Use Cases: Scalability is crucial when you anticipate a steady increase in demand, such as a growing user base or expanding application features.
- Example in AWS: Auto Scaling Groups allow you to automatically scale the number of EC2 instances based on demand. Amazon RDS allows you to scale up or out by increasing the size of the database instance or adding read replicas.

## Elasticity

Definition: Elasticity is the ability of a system to automatically expand and contract its capacity based on the current load. In AWS, elasticity means that resources can be dynamically allocated or de-allocated in real-time based on demand.

Key Points:

- Automatic Adjustment: Elasticity involves the automatic addition or removal of resources in response to changes in demand.
- Use Cases: Elasticity is important for applications with unpredictable or fluctuating workloads, such as online stores with seasonal traffic spikes.
- Example in AWS: AWS Lambda automatically scales in response to incoming requests, and EC2 Auto Scaling adjusts the number of instances based on the current load, scaling out during peak times and scaling in when demand decreases.



## High Availability (HA)

Definition: High availability refers to a system's ability to remain operational and accessible for a high percentage of time, usually measured as uptime (e.g., 99.9%, 99.99%, etc.). It ensures that services are continuously available with minimal downtime.

Key Points:

- **Redundancy:** High availability is typically achieved by having redundant components, so if one fails, another can take over.
- **Load Balancing:** Distributing workloads across multiple resources to avoid any single point of failure.
- **Automatic Failover:** If a component fails, the system automatically switches to a backup or standby component with minimal disruption.
- **Example in AWS:** Deploying applications across multiple Availability Zones (AZs) using an Elastic Load Balancer (ELB) ensures high availability. If one AZ goes down, traffic is routed to healthy instances in other AZs.

## Fault Tolerance

Definition: Fault tolerance is the ability of a system to continue operating correctly even if one or more components fail. A fault-tolerant system is designed to provide uninterrupted service, with no perceived downtime, even in the event of a failure.

Key Points:

- **Continuous Operation:** Fault-tolerant systems can handle failures without any interruption in service.
- **No Single Point of Failure:** Fault tolerance involves designing systems so that no single component failure can cause the system to go down.
- **Redundant Systems:** Fault tolerance often involves having completely redundant systems (e.g., active-active configurations) that can seamlessly take over if a failure occurs.
- **Example in AWS:** Using services like Amazon S3 or Amazon Aurora, which replicate data across multiple AZs, ensures that data is still accessible even if an entire AZ fails. Another example is using AWS Auto Scaling in conjunction with multiple instances across different AZs to maintain service continuity.

In AWS, Operational Excellence and Performance Efficiency are two pillars of the AWS Well-Architected Framework, which provides best practices for designing and operating reliable, secure, efficient, and cost-effective systems in the cloud. Both are focused on optimizing different aspects of cloud operations, but they address different goals and practices.

## **Operational Excellence**

Definition: Operational excellence in AWS refers to the ability to run and monitor systems to deliver business value and to continually improve supporting processes and procedures.

Key Focus Areas:

- **Organizational Culture:** Encourages a culture of continuous improvement, learning, and sharing of best practices.
- **Design for Operations:** Ensures that the architecture is designed for effective operation and supports ongoing improvements.
- **Automate Processes:** Emphasizes automating operational processes to improve efficiency and reduce the chance of human error.
- **Monitor and Respond:** Involves setting up monitoring and logging to detect issues and respond to them quickly.
- **Manage Change:** Ensures that changes are managed in a controlled and planned way to avoid unintended consequences.

Example in AWS: Using AWS Systems Manager to automate routine tasks, monitor operations, and manage changes across AWS resources, ensuring smooth and efficient operations.

## **Performance Efficiency**

Definition: Performance efficiency in AWS refers to the ability to use computing resources efficiently to meet system requirements and to maintain that efficiency as demand changes and technologies evolve.

Key Focus Areas:

- **Selection:** Choose the right type, size, and configuration of resources based on the workload requirements.
- **Review:** Regularly review and adjust your choices based on changes in the system or new AWS offerings.
- **Monitor:** Continuously monitor the system to ensure that it meets performance objectives.
- **Optimize:** Optimize resources by taking advantage of the elasticity and scalability of the cloud to match demand.
- **Evolve:** Continuously evaluate and implement the latest AWS services and features to improve performance.

Example in AWS: Using Amazon EC2 Auto Scaling to automatically adjust the number of instances to match the current load, ensuring optimal performance without over-provisioning.

## **Components of Risk and Compliance Program in AWS**

Component	Description
Governance Framework	<ul style="list-style-type: none"> <li>- Policies and Procedures: Establish cloud-specific policies and procedures that align with industry standards (e.g., GDPR, HIPAA).</li> <li>- Roles and Responsibilities: Define roles and responsibilities for compliance and risk management within the organization.</li> </ul>
Risk Management	<ul style="list-style-type: none"> <li>- Risk Assessment: Identify, assess, and prioritize risks related to cloud adoption and operation.</li> <li>- Risk Mitigation: Develop strategies to mitigate identified risks, such as implementing security controls or redundancy.</li> <li>- Continuous Monitoring: Regularly monitor and review risks as part of ongoing risk management activities.</li> </ul>
Compliance Mapping	<ul style="list-style-type: none"> <li>- Regulatory Requirements: Map AWS services and configurations to regulatory requirements (e.g., PCI DSS, ISO 27001).</li> <li>- Compliance Controls: Implement and document controls to meet these regulatory requirements, using AWS services like AWS Config, AWS Artifact, and AWS Security Hub.</li> </ul>
Security and Access Control	<ul style="list-style-type: none"> <li>- Identity and Access Management (IAM): Enforce least privilege access, manage roles and policies, and monitor access logs.</li> <li>- Data Encryption: Implement encryption for data at rest and in transit using AWS services like AWS KMS (Key Management Service) and SSL/TLS.</li> </ul>
Audit and Reporting	<ul style="list-style-type: none"> <li>- Audit Trails: Use AWS CloudTrail to capture detailed logs of API calls and monitor changes to resources.</li> <li>- Compliance Reporting: Generate and maintain reports that demonstrate compliance with internal policies and external regulations.</li> </ul>
Training and Awareness	<ul style="list-style-type: none"> <li>- Staff Training: Regularly train staff on AWS security practices, compliance requirements, and the organization's risk management framework.</li> <li>- Awareness Programs: Develop awareness programs to keep staff informed about the importance of compliance and risk management.</li> </ul>
Incident Response and Management	<ul style="list-style-type: none"> <li>- Incident Response Plan: Develop and test incident response plans that detail actions to take in case of security breaches or compliance violations.</li> <li>- Forensic Capabilities: Use AWS services like Amazon GuardDuty and AWS Security Hub to detect, investigate, and respond to security incidents.</li> </ul>
Vendor and Third-Party Management	<ul style="list-style-type: none"> <li>- Third-Party Risk Management: Assess and manage risks related to third-party vendors and services integrated with AWS.</li> <li>- Service Level Agreements (SLAs): Ensure that SLAs with vendors meet the organization's compliance and risk management needs.</li> </ul>

Component	Description
Continuous Improvement	<ul style="list-style-type: none"> <li>- Regular Reviews and Audits: Conduct regular reviews and audits of the compliance program to ensure it remains effective and up to date.</li> <li>- Feedback Loop: Implement a feedback loop to learn from past incidents and continuously improve the risk and compliance program.</li> </ul>
Automation and Tooling	<ul style="list-style-type: none"> <li>- Automated Compliance Checks: Use AWS Config, AWS Security Hub, and AWS Audit Manager to automate compliance checks and ensure continuous compliance.</li> <li>- Infrastructure as Code (IaC): Use tools like AWS CloudFormation to manage and enforce compliance in cloud infrastructure automatically.</li> </ul>
Documentation and Record Keeping	<ul style="list-style-type: none"> <li>- Documentation: Maintain comprehensive documentation of all compliance activities, policies, and procedures.</li> <li>- Record Keeping: Ensure that all records are securely stored and easily accessible for audits and compliance checks.</li> </ul>

## AWS Services with Built-in DDoS Protection

Service	Description	DDoS Protection Features
Amazon CloudFront	A content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency.	Integrated with AWS Shield Standard for automatic protection against common DDoS attacks at the edge locations.
AWS Shield	A managed DDoS protection service that safeguards applications running on AWS.	AWS Shield Standard is automatically included at no additional cost, protecting against most common DDoS attacks.
Elastic Load Balancing (ELB)	Automatically distributes incoming application traffic across multiple targets, such as EC2 instances, in one or more Availability Zones.	Protects applications by distributing traffic and mitigating the impact of DDoS attacks through load balancing.
Amazon Route 53	A scalable Domain Name System (DNS) web service designed to route end users to internet applications.	Provides DNS-based protection by absorbing DDoS attacks and using AWS Shield for enhanced protection.
AWS WAF (Web Application Firewall)	Protects web applications by filtering and monitoring HTTP/S requests.	Integrated with AWS Shield to protect against DDoS attacks at the application layer, with rules to block malicious traffic.
Amazon CloudFront + AWS WAF + Shield Advanced	A combination of services that provides comprehensive protection against complex DDoS attacks.	Provides enhanced protection with detailed attack diagnostics and 24/7 access to the AWS DDoS Response Team (DRT).
Amazon S3	A scalable object storage service for data storage and retrieval.	Benefits from AWS Shield and can handle large volumes of traffic, reducing the impact of DDoS attacks.

## The 7 R's of AWS Migration Strategies

Strategy	Description
1. Rehost (Lift and Shift)	<ul style="list-style-type: none"><li>- Overview: Moving applications to the cloud with minimal changes. Often used for applications that are already virtualized or don't require cloud-native features.</li><li>- Use Case: Quick migration with minimal modification, typically to reduce costs or move out of a data center.</li></ul>
2. Replatform (Lift, Tinker, and Shift)	<ul style="list-style-type: none"><li>- Overview: Making a few cloud optimizations to achieve benefits without changing the core architecture of the application.</li><li>- Use Case: Applications that can benefit from cloud features like managed databases or improved performance without needing significant changes.</li></ul>
3. Repurchase (Drop and Shop)	<ul style="list-style-type: none"><li>- Overview: Moving to a new product, often a SaaS (Software as a Service) solution that replaces the current application.</li><li>- Use Case: Legacy applications or software that are outdated or where the cost of upgrading is higher than switching to a cloud-based service.</li></ul>
4. Refactor / Re-architect	<ul style="list-style-type: none"><li>- Overview: Re-imagining how the application is architected and developed using cloud-native features. This involves significant changes to the code and architecture.</li><li>- Use Case: Applications that require major scalability or performance improvements, or when moving to a microservices architecture.</li></ul>
5. Retire	<ul style="list-style-type: none"><li>- Overview: Identifying and turning off applications that are no longer needed. This reduces the IT footprint and cuts costs.</li><li>- Use Case: Applications or workloads that are obsolete, redundant, or replaced by other solutions.</li></ul>
6. Retain	<ul style="list-style-type: none"><li>- Overview: Choosing to keep applications in their current environment, usually because they are still relevant or there's no immediate benefit to migrating them.</li><li>- Use Case: Applications that are not ready for migration, or where the cost and risk of migration outweigh the benefits.</li></ul>
7. Relocate	<ul style="list-style-type: none"><li>- Overview: Moving workloads to the cloud without purchasing new hardware, similar to rehosting but may involve moving entire data centers as is. This is often used in VMware Cloud on AWS scenarios.</li><li>- Use Case: Scenarios where rehosting is required at scale, or for highly regulated environments where it's essential to maintain existing configurations.</li></ul>

### Commented [JN26]: Summary

These 7 R's represent different strategies organizations can use depending on their specific needs, existing infrastructure, and cloud goals. Some applications might be best suited for a simple lift-and-shift, while others might benefit from a full re-architecture to take advantage of cloud-native features. The choice of strategy often depends on factors like cost, time, complexity, and the specific goals of the migration.