**The Value of AWS Cloud**

Amazon Web Services (AWS) offers a multitude of advantages over traditional, internally managed data centers. Here's an improved presentation of these benefits:

**Benefits of AWS Cloud**

1. **Rapid Global Deployment**: AWS boasts a global network of regions, making it possible to deploy applications and services within minutes. This global reach ensures that your resources are closer to your users, reducing latency and enhancing the user experience.
2. **Agility for Speedy Innovation**: With AWS, you can innovate more quickly and efficiently, allowing your organization to deliver new features and solutions to customers at an accelerated pace. This agility is essential in a rapidly evolving digital landscape.
3. **Cost Savings Through Economies of Scale**: AWS leverages economies of scale, spreading infrastructure costs across a vast user base. As a result, you benefit from cost-effective solutions without the need for large upfront capital investments.
4. **No Upfront Costs for Data Center Maintenance**: AWS removes the burden of managing physical data centers. You can focus on your applications and services without worrying about IT infrastructure, as AWS takes care of the maintenance.
5. **Operational Expenses Over Capital Expenses**: AWS follows an operational expenditure (OpEx) model, enabling you to pay for resources and services as you use them. This eliminates the need for substantial upfront capital expenditures (CapEx), making financial planning more predictable.
6. **Elastic Capacity**: AWS offers flexible, pay-as-you-go capacity. There's no need to guess your capacity requirements in advance. You can scale resources up or down as needed, ensuring efficient resource utilization.

**Meeting Non-Functional Requirements in the Public Cloud**

The AWS cloud platform addresses various non-functional requirements effectively, including:

1. **High Availability**: AWS provides redundancy and failover capabilities, ensuring longer uptimes and enhanced reliability for your systems.
2. **Elasticity**: The ability to dynamically scale resources based on demand optimizes resource utilization and minimizes wastage.
3. **Agility**: AWS services empower customers to innovate rapidly, reducing time-to-market for new features and applications.
4. **Durability**: AWS offers data services with long-term data protection and storage, safeguarding your critical data.
5. **Low Latency**: Low latency, or minimal delay between user requests and responses, is a key advantage, ensuring a smooth user experience.

**Cloud Computing Models**

AWS offers a range of cloud computing models, including:

1. **Infrastructure as a Service (IaaS)**: Examples include Amazon EC2, allowing you to manage virtualized infrastructure.
2. **Platform as a Service (PaaS)**: Services like AWS Cloud9 provide a platform for application development, simplifying the development process.
3. **Software as a Service (SaaS)**: Sagemaker is an example of SaaS, offering ready-to-use software solutions.

[For more details, visit AWS Cloud Computing](https://aws.amazon.com/what-is-cloud-computing/?pg=TOCC)

**Cloud Hosting Models**

There are various cloud hosting models, each with its unique characteristics:

1. **Private Cloud**: This includes on-premises virtualization and fully managed off-premises private cloud solutions, such as Amazon Outposts.
2. **Public Cloud**: AWS provides fully publicly hosted and managed cloud services accessible to customers worldwide.
3. **Hybrid Cloud**: AWS Direct Connect service allows seamless integration between a customer's data center and Amazon's cloud infrastructure, enabling hybrid cloud deployments.

**AWS Regions, Availability Zones, and More**

Amazon EC2 operates in multiple global locations, each with distinct components:

1. **Region**: A region represents a separate geographic area, ensuring redundancy and disaster recovery. Regions are fully independent, with varying services and resources.
2. **Availability Zone (AZ)**: An AZ comprises one or more data centers with redundant power, networking, and connectivity. These zones provide high availability and fault tolerance.
3. **Data Center**: Data centers are the building blocks of AZs and come with robust physical and environmental protections.
4. **Local Zones**: Local Zones extend an AWS Region to geographic proximity, serving users with low-latency communication. They are ideal for latency-sensitive applications.
5. **Wavelength Zones**: Wavelength Zones are isolated zones within a carrier location, providing low-latency access for specific applications.
6. **Global Edge Network**: Amazon CloudFront enhances performance and reliability with an extensive network of edge locations, ensuring fast content delivery.

For more information, visit the respective links provided for each component.

**Leveraging the AWS Well-Architected Framework**

The AWS Well-Architected Framework offers guidance for building robust, efficient, and secure cloud infrastructures. Here are the key pillars of the framework:

1. **Operational Excellence**: Plan for failure, deploy smaller reversible changes, script infrastructure as code, and continuously learn from failures. Utilize tools like AWS CodeCommit for versioning.
2. **Security**: Automate security tasks, encrypt data in transit and at rest, follow the principle of least privilege, track and audit actions, and ensure security across all layers. Utilize AWS CloudTrail for comprehensive logging.
3. **Reliability**: Automatically recover from failures, scale horizontally for resilience, avoid guessing capacity needs, automate change management, and test recovery procedures. Consider multi-AZ deployments for high availability.
4. **Performance Efficiency**: Opt for serverless architectures, explore multi-region deployments, delegate tasks to cloud vendors, and experiment with virtual resources. AWS Lambda is an excellent choice for serverless workloads.
5. **Cost Optimization**: Implement consumption-based pricing, adopt Cloud Financial Management, measure overall efficiency, and pay only for the resources your applications require. Utilize features like S3 Intelligent Tiering for cost-effective data management.
6. **Sustainability**: Understand your environmental impact, set sustainability goals, maximize resource utilization, leverage managed services, and reduce downstream impact. EC2 Auto-scaling can help optimize resource usage.

# AWS Shared Responsibility Model

Source: [AWS Shared Responsibility Model](https://aws.amazon.com/compliance/shared-responsibility-model/)

As an AWS customer, understanding the shared responsibility model is essential. AWS and its customers have distinct roles in ensuring the security and compliance of the cloud environment:

**AWS has the responsibility OF the cloud. The customer has responsibilities IN the cloud.**

**AWS Responsibilities**

As an AWS customer, you are not responsible for the hardware, software, networking, and facilities that support AWS Cloud services across its regions, availability zones (AZs), data centers, and edge locations. AWS shoulders the following key responsibilities:

1. **Inherited Controls (AWS only):**
   * These are controls fully inherited from AWS, including physical and environmental controls.
2. **Shared Controls (AWS and Customer):**
   * Responsibilities shared between AWS and the customer include patch management, configuration management, awareness, and training.
3. **Customer Controls (Customer only):**
   * Customer-specific controls encompass service and communications protection, as well as zone security, which may require routing or zoning data within specific security environments.

**Customer Responsibilities**

Your responsibilities as an AWS customer encompass:

* **Data and Application Security:**
  + Ensuring the security of your data and applications, including encryption options and proper security configurations.
* **Identity and Access Management (IAM):**
  + Managing IAM for application security and controlling identity and access for your systems.
* **Network Traffic and Firewall Configuration:**
  + Managing network traffic and configuring group firewall settings.
* **Patching and Maintenance:**
  + Regularly patching guest operating systems of EC2 instances and maintaining the security of your AWS resources.

**Report AWS Abuse**

If you suspect any abuse of AWS resources, take the following steps:

1. **Rotate Keys and Change Passwords:** Ensure that your AWS access keys and passwords are up to date and secure.
2. **Contact the AWS Trust & Safety Team:** Report the abuse to the AWS Trust & Safety team using the [Report Amazon AWS abuse form](https://aws.amazon.com/premiumsupport/knowledge-center/report-aws-abuse/).

By understanding and fulfilling your responsibilities within the AWS shared responsibility model, you can help maintain the security and compliance of your cloud environment.

**AWS Security Best Practices**

*This is 25% of the weight of the exam*

**Root User**

* Automatically created when you create an AWS account.
* Only root user can delete the account.
* There is just one root user that can exclusively:
  + Change your account settings. This includes the account name, email address, root user password, and root user access keys.
  + Restore IAM user permissions. If the only IAM administrator accidentally revokes their own permissions, you can sign in as the root user to edit policies and restore those permissions.
  + Activate IAM access to the Billing and Cost Management console.
  + View certain tax invoices. An IAM user with the aws-portal:ViewBilling permission can view and download VAT invoices from AWS Europe, but not AWS Inc. or Amazon Internet Services Private Limited (AISPL).
  + Close your AWS account.
  + Register as a seller in the Reserved Instance Marketplace.
  + Configure an Amazon S3 bucket to enable MFA (multi-factor authentication).
  + Edit or delete an Amazon Simple Storage Service (Amazon S3) bucket policy that includes an invalid virtual private cloud (VPC) ID or VPC endpoint ID
  + Sign up for AWS GovCloud (US).
  + Request AWS GovCloud (US) account root user access keys from AWS Support.

*Best Practice:* Identity and Access Management - create a new user and provide a role. Never use the root user unless absolutely required. Protect root account with MFA (Multi-factor authentication).

VPC - Vitual Private Cloud. Default VPC will always be created for you.

* AWS Management Console
  + Easy to navigate via web-browser.
  + Good for non-technical roles. Use the search feature for easy access.
* AWS CLI - same features as the management console
  + New features show up here first.
  + Programmatic access provides access to your AWS resources.
* AWS SDK - can be leveraged to make changes to the environment via programmatic access.

**Concepts**

1. Authentication
   * An identity that is verified.
   * Credentials such as username and password.
2. Authorization
   * Determines which services and resources the idenitity has access to.
   * Permissions are granted via a policy.
3. Least Privilege
   * Give a user the minimum access required to get the job done.

**IAM**

1. [IAM](https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html)
   * A web service that allows you securely control access to AWS resources.
2. Users
   * Entities in IAM to represent a person or application that can be given access to your AWS resources.
   * Applications can be users. This is normally done via access keys.
3. Group
   * Collection of users - conveniently apply common permissions.
   * This is not EC2 Security Group - that is a firewall.
   * Can you nest groups? Can you have group inheritance? Are there unlimited groups that can be created?
4. Roles
   * Roles define access permissions and are temporarily assumed by an IAM user or service.
   * DevOps role, Lambda-Execution role are examples.
   * Access is assigned using policies.
   * You grant users in one AWS account access to resources in another AWS acccount using roles.
   * Attach a role to an EC2 instance for access to S3. Applications running on that instance will have access to S3 via roles. This is useful because the application will not need credentials or access keys. This is most secure.
5. Policies
   * You manage persmissions for IAM users, groups, and roles by creating a policy document in JSON format and attaching it. The policy itself is decoupled from IAM identitieis.
   * User - {Policy:Access} - Resource
   * Developer Group = {Policy: Resource Access} - Resource
   * Role - {Policy:Allow-S3-Access} - S3
   * How to limit access to an Amazeon S3 to specific users only? You can add a bucket access policy directly to an Amazon S3 bucket to grant IAM users accesss. I wonder if there is another way, create a special bucket access group with policy to the group, and then add users to the group. Or add users to the policy directly.
6. IAM Credentials Report
   * Assistance with compliance and auditing by offering a downloadable report that lists all your IAM users in this account and the status of their various credentials including MFA devices in your account.

**Security Services**

1. [WAF](https://aws.amazon.com/waf/) : XSS SQL-Injection
   * WAF is a Web Application Firewall that can protect against common attacks such as XSS or SQL injection.
2. [Shield](https://aws.amazon.com/shield/) DDOS
   * AWS Shielf is a managed DDOS protection service. Sheild standard is free but Sheild Advanced provides access to AWS experts for a fee.
   * DDOS protections from CloudFront, Route53, Elastic Load Balancing, and AWS Global Accelerator.
   * Receive real-time notifications of suspected DDoS incidents via CloudWatch metrics and assistance from AWS during the attack.
   * Automatically scrub bad traffic at specific layers: layer 3,4 and 7. Minimize application downtime and latency. Monitor and protect up to 1000 resource types.
3. [Macie](https://aws.amazon.com/macie/) Sensitive Data
   * hHelps you discover and protect sensitive data. Uses maching learning, evaluates S3 environment, uncovers PII information.
   * Use cases: discover passport numbers stored on S3 using Macie. Find SSNs in S3 files.
4. [Config](https://aws.amazon.com/config/) Audit config
   * Assess, audit, and evaluate configurations of your resources.
   * Record and altert by storing in S3.
   * Use cases: Streamline operational troubleshooting and change management. Deploy a complicant-as-code framework. Continually audit security monitoring and analysis.
5. [GuardDuty](https://aws.amazon.com/guardduty/) Threat detection
   * Protect your AWS accounts with intelligent threat detection.
   * Continuously monitors workload for malicious activity and delivers detailed security findings for visibility and remediation. Network and API calls.
   * Use cases: Improve security operations visibility. Assist security analysts in investigations. Identify files containing malware. Route insightful information on security findings.
6. [Inspector](https://aws.amazon.com/inspector/) Vulnerability (EC2)
   * Automate vulnerability management at scale in EC2, Lambda and ECR container images and network exposure.
   * Automated vulnerability management service that continually scans workloads for software vulnerabilities and unintended network exposure. EC2.
   * Use cases: Quickly discover vulnerabilities in compute workloads. Prioritize patch remediation. Meet compliance requirements. Identify zero-day vulnerabilities sooner.
7. [Artifact](https://aws.amazon.com/artifact/) Compliance Report.
   * Access Independent Software Vendor compliance report.
   * Use artifact to SOC and PCI compliance reports. You can generate the report. Access to the report can be provided. Self-service portal.
8. [Cognito](https://aws.amazon.com/cognito/) CIAM
   * Customer identity and acess management.
   * Delivery frictionless CIAM. Adaptive authentication, support compliance, and data residency requirements. Scale to millions of users with a fully managed, high-performantm and reliable identity store. Federate sign-in using OIDC or SAML 2.0 connect to a broad group of AWS services and products.
   * Use-cases: Social media accounts to log in to your application.

**Data Encryption and Secrets Management Services**

1. [KMS](https://aws.amazon.com/kms/) Key Management
   * Key Management Service is multi-tenant encryption key management service.
   * Create and control encryption keys managed by AWS used to encrypt or digitally sign your data.
   * Centrally manage keys and define policies across integrated services and application from a single point.
   * Encrypt data within your applications with the AWS Encryption SDK data encryption library.
   * Encrypt EBS volume using KMS.
2. [CloudHSM](https://aws.amazon.com/cloudhsm/) Encryption Key Generator.
   * Manage single-tenant hardware security modules (HSMs) on AWS.
   * Use case: Generate and use cryptographic keys on dedicated FIPS 140-2 Level 3 single-tenant HSM instances. Deploy workloads with high reliability and low latency, and help meet regulatory compliance. Pay by the hour, and backup and shut down HSMS when they're not needed. Manage HSM capacity and control your costs by adding and removing HSMs from your cluster.
3. [Secrets Manager](https://aws.amazon.com/secrets-manager/) Secrets Management
   * Use cases: Store secrets securely, manage acess with fine-grained policies, automate secrets rotation, audit and monitor secrets usage.
   * Database credentials, API keys, encrypt secrets at rest, integreates with RDS, DOcumentDB, Redshift.
   * Retrieve database credentials needed for your application code. Secrets Manager allows you to retrieve database credentials with a call to Secrets Manager APIs, removing the need to hardcode sensitive information in plain text within your application code.
4. [AWS Certificate Manager](https://aws.amazon.com/certificate-manager/) Certificate Manager
   * Provisiong public and private certificats for free.
   * SSL/TLS certificates are supported.
   * Use key management for certs and get managed certificate renewal.
   * Integrates with Elastic Load Balancing, API Gateway and more.

**AWS Costs, Economics and Billing Practices**

*16% of the exam\_questions about 8-10 questions*

* [EC2 Instances](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-purchasing-options.html) are [priced](https://aws.amazon.com/ec2/pricing/) as follows
  + On-Demand: EC2 capacity billed to the second.
    - Pay for what you use.
    - Use case: When Applications are under development, or workloads are not expected to run for more than a year, no upfront payment or long-term committment, unpredictable workloads but don't want to be interrupted.
    - On-Demand Capacity Reservation: It is possible to buy upfront capacity to mitigate against capacity contraints in an availability zone (AZ).
  + Spot: unused EC2 capacity on sale.
    - Pay the least but no guarantee of runtimes or interruptions. A 2-minute warning is provided via instance meta-data that your application should check for and prepare for shutdown.
    - Use case: Start and stop time of the workload does not matter. 90% savings over On-Demand. When your workload is feasable only at the lowest price points.
    - Spot price in effect at the beginning of each hour.
  + Reserved: Upfront capacity reservation committment for long running workloads.
    - Pay upfront with a contract to get discounts.
    - Use case: Save 75% versus On-Demand and willing to pay upfront for 1 or 3 year reservation.
    - Flexibility: All upfront, partial upfront or no upfront is possible. A contract is required. Provides convertible types at 54% discount - change tenancy, OS or region.
  + Dedicated Instance and Dedicated Host:
    - [Dedicated Host](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/dedicated-hosts-overview.html): Dedicated bare metal rental and host exclusively for you to install software that have licensing tied to host size.
    - [Dedicated Instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/dedicated-instance.html): Instances run on VPCs on a hardware dedicated to a single customer.
    - Use Case: Save 70% off of On-Demand. Software that is licensed based on per-core, per-socket or per-VM. Regulations that require tenancy exclusivity.
    - Dedidicated host is a physical server, dedicated instance runs on a host.
  + Savings Plan: Compute usage committment for 1 or 3 years applicable across multiple compute services.
    - Save upto 72% off of On-Demand.
    - Use Case: For flexibility across various services like Lambda, Fargate, and EC2.
    - This is a billing convenience nothing to do with a capacity reservation.
* Lambda Pricing
  + Computer Time - no charge for times that code is not running.
  + Duration - duration of compute and memory usage while execution is counted.
  + Free Tier - the free tier includes 1 million free requests each month
* S3 Pricing
  + Storage Class
  + Storage - number of items, and size.
  + Data transfer - outbound.
  + Request and data retrieval - number of requests made.
* RDS Pricing
  + Running Clock Hours
  + Type of Database - brand, size, memory class etc
  + Storage - amount of data
  + Purchase type - on-demand, reserved instance
  + DB count - number of instance
  + API - number of calls
  + Deployment type - is it multi-AZ
  + Outbound - data transfer

**Pricing, Billing and Governance**

Compute, storage and outbound data transfer is where the costs are for AWS. Data in flight moving between system. Data movement within the AWS region are usually not charged. Data out of AWS to end user is where the data transfer costs are. How AWS Pricing Works [whitepaper](https://docs.aws.amazon.com/pdfs/whitepapers/latest/how-aws-pricing-works/how-aws-pricing-works.pdf)

* + [TCO](https://docs.aws.amazon.com/whitepapers/latest/how-aws-pricing-works/aws-pricingtco-tools.html)
    - Total Cost of Ownership. Direct and indirect cost of running AWS workloads. How can I reduce my TCO using AWS?
    - Minimize capital expenditures.
    - Utilize reserved instances.
    - Right size your resources.
    - Does not consider Networking or Data costs. No personnel or facilities costs.
  + [AWS Price List API](https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/price-changes.html)
    - Query the price of AWS Services using JSON or CSV. Bulk price or individual APIs.
    - Receive price alerts when prices change.
  + [Application Disovery Service](https://docs.aws.amazon.com/application-discovery/latest/userguide/what-is-appdiscovery.html)
    - Determine the cost of migrating to the cloud.
    - Plan migration projects and estimate TCO.
    - You can view the discovered servers, group them into applications, and then track the migration status of each application from the Migration Hub console in your home Region.
  + [Budgets](https://docs.aws.amazon.com/cost-management/latest/userguide/budgets-managing-costs.html)
    - Set custom budgets for cost and usage tracking. Alerts.
    - Cost, usage and reservation budgets.
    - You can choose to be notified through email and Amazon SNS topics when your utilization drops below 80 percent for a given day.
  + [Cost and Usage Reports](https://docs.aws.amazon.com/cur/latest/userguide/what-is-cur.html)
    - Break down costs by the hour, day, or month, by product or product resource, or by tags that you define yourself.
    - If you get a huge bill - this is where you need to find the needle in the haystack.
    - Downloadable detailed and comprehensive report, list usage for each service category and aggregate usage data on a daily, hourly or monthly level.
    - Cost Allocation Tags
      * Label resources using key-value pairrs.
      * Track costs via the cost allocation report.
  + [Cost Explorer](https://aws.amazon.com/aws-cost-management/aws-cost-explorer/)
    - Visualize, understand, and manage your AWS costs and usage over time.
    - Forecast, build custom apps that use it's apis, and use granular filtering offered by it's analytical engine.
    - [Organizations](https://aws.amazon.com/organizations/)
    - Centrally manage your environment as you scale your AWS resources. Consolidate billing, save costs via volume discounts + reserved instance sharing and govern accounts centrally.
    - Programmatically create AWS accounts as you scale at no additional charge.
    - Centrally secure and audit. Manage and optimize costs centrally. Group accounts and apply policies across.
    - Root Organization is the master payer account that pays for all the accounts.
    - You can apply Service Control Policies (SCPs) across all member accounts within the organization.
  + [Control Tower](https://aws.amazon.com/controltower/)
    - Set up well-architected multi-account environments with pre-configured controls to ensure best practices.
    - Provides dashboard to help manage accounts.
    - Example, if you want to disallow public write access to all S3 buckets across your accounts - you can use Control Tower to enforce this.
  + [Systems Manager](https://aws.amazon.com/systems-manager/)
    - Operation insights into AWS resources, other cloud resources and on-prem resources.
    - Automate configuration and ongoing management including instance compliance relative to patch, configuration and custom policies.
    - Visibility and control. Group resources to take action. Patch and run commands on multiple EC2 and RDS.
    - Usecase: Deploy operating system and software patchs automatically across a large group of instances.
  + [Trusted Advisor](https://aws.amazon.com/premiumsupport/technology/trusted-advisor/)
    - Cost, Performance, Security, Fault Tolerance, and Service Limits.
    - Checks IAM password policy (not free). RDS public snapshot, service usage greater than 80% (available to business or enterprise). Check for exposed access keys (business support) and various other checks.
    - Use case: check read and write capacity service limits for DynamoDB.
  + [Personal Health Dashboard](https://aws.amazon.com/premiumsupport/technology/aws-health-dashboard/)
    - Alerts you on impacts to your AWS environment.
  + [Marketplace](https://aws.amazon.com/marketplace)
    - Digital catalog of prebuilt solutions you can purchase or license.
  + [AWS Partner Network (APN)](https://aws.amazon.com/partners/)
    - Global community of approved partners that offer solutions and consulting services
    - Help design and build a new application.
  + [Managed Services](https://aws.amazon.com/managed-services/)
    - Augment internall staff with additional resources to manage AWS.
    - Patch management, monitoring, event management, cost optimization etc.
    - Will not operate or configur your applications.
  + [Professional Services](https://aws.amazon.com/professional-services/)
    - Move to a cloud based operating model
    - Propose solutions.
    - Architect soutions.
    - You can quickly move from on-prem to cloud.
  + [AWS License Manager](https://aws.amazon.com/license-manager/)
    - AWS and on-premise license manager.
    - Fine-tune your license costs.

[**Support Plans**](https://aws.amazon.com/premiumsupport/plans/)

1. Basic - free. \* Email support only and discussion forums.
2. Developer - $29 pm : \* Fordevelopment and testing. \* 1 contact. \* Cloud support associate via email during business hours.
3. Business - $100 pm : \* Production workloads. \* Unlimited contact. \* Full Trusted Advisory. \* Email, phone and chat 24/7. Production system down - less than one hour.
4. Enterprise - $15k pm \* Mission-critical production workloads. \* Exclusive: Technical Account Manager, Concierge support team, infrastructure event support. \* Less than 15m for business critical system down.

**AWS Core Services (and concepts)**

The following concepts and list of AWS Core Services are essential to understand various layers of an architecture. AWS offers [Trusted Advisor](https://aws.amazon.com/premiumsupport/technology/trusted-advisor/) tool to business and higher subscriptions. \* Provides recommendations that help you follow AWS best practices. \* Benefits: cost optimization, performance, security, fault tolerance and service quotas.

For example, a web-based enterprise application will utilize most if no all the layers and select technologies.

**Architecture**

1. [Elasticity](https://www.ibm.com/cloud/blog/cloud-elasticity-vs-cloud-scalability): The ability to add or remove resources based on demand.
2. [Scalability](https://resources.sei.cmu.edu/asset_files/TechnicalNote/2006_004_001_14681.pdf): Scalability is the ability to handle increased workload by repeatedly applying a cost-effective strategy for extending a system’s capacity
3. [Fault Tolerance](https://en.wikipedia.org/wiki/Fault_tolerance): Is the property that enables a system to continue operating properly in the event of a failure of one or more faults withing some if its components.
4. [High Availability](https://redis.com/blog/high-availability-architecture/): Property of a system to serve the business without failure over a given period of time.
5. [Principle of least priviledge](https://www.cisa.gov/uscert/bsi/articles/knowledge/principles/least-privilege): Every program and every user of the system should operate using the least set of privileges necessary to complete the job. Primarily, this principle limits the damage that can result from an accident or error.

**Billing**

1. [AWS Organization](https://docs.aws.amazon.com/organizations/latest/userguide/orgs_introduction.html)
   * Account management services that enables you to consolidate multiple AWS Accounts into an organization that you create and centrally manage.
   * Use case: ease of billing, budgetary, security and compliance needs.
2. [Consolidated billing](https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/consolidated-billing.html)
   * AWS Organizations has a management account that pays the charges of all the member accounts.
   * Multiple AWS Accounts can be consolidated for billing and payments.

**Security**

1. [Firewall](https://aws.amazon.com/network-firewall/)
   * AWS Network Firewall automatically scales your network firewall to protect your managed infrastructure.
   * Open source rule formats and underlying rules engine easily implements policies.
2. [IAM](https://aws.amazon.com/iam/)
   * Identity and Access Managment (IAM) sets and manages guardrails and fine-grained access controls for your workforce and workloads.
   * Centrally connect identities to multiple AWS accounts.
   * Grant temporary security credentials for workloads that access your AWS resources.
   * Continually analyze access to right-size permissions on the journey to least privilege.
   * Usecase: "Who can access what" Who=users and workloads. Can access= Permissions with IAM policy. What=Resources within your AWS organization.
3. [Security Group (SG)](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-security-groups.html)
   * Is a virtual firewall for EC2 instances to control incomcing and outgoing traffic.
4. [User Credentials](https://docs.aws.amazon.com/general/latest/gr/aws-sec-cred-types.html)
   * Each identity has unique credentials within AWS.
   * Identity types: Account Root User, AWS Identity and Access Management user, AWS IAM Identity Center user and Federated identity.
5. [Access Control List (ACL)](https://docs.aws.amazon.com/AmazonS3/latest/userguide/acls.html):
   * A firewall layer on the subnet level.
   * ACL cannot grant permissions to users in the account.
   * ACL can grant basic read/write permissions to other AWS accounts to buckets and objects.

**Networking and Content Delivery**

1. [AWS Global Accelerator](https://aws.amazon.com/global-accelerator/) : Global Traffic
   * Improve application availability, performance, and security using the AWS global network.
   * Usecases: global traffic manager, API acceleration, Global static IP, low-latency gaming and media workloads.
   * Global accelerator sends your users through the AWS global network when accessing your content, speeding up delivery.
2. [AWS Transit Gateway](https://aws.amazon.com/transit-gateway/) : No more peering
   * AWS Transit Gateway connects your Amazon Virtual Private Clouds (VPCs) and on-premises networks through a central hub.
   * Put an end to complex peering relationships.
   * Highly scalable cloud router where each new connection is made only once.
3. [Virtual Private Cloud (VPC)](https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html): Slice of the cloud
   * Foundational service that creates a private virtual network to launch resources.
   * Spans AZs in a region.
   * VPC A and VPC B can be [peered](https://docs.aws.amazon.com/vpc/latest/peering/what-is-vpc-peering.html) so they act as one logical VPC.You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account. The VPCs can be in different Regions (also known as an inter-Region VPC peering connection).
   * The default VPC always exists in every region. But all new VPCs are region specific.
4. [Subnet](https://docs.aws.amazon.com/vpc/latest/userguide/configure-subnets.html) : One per AZ
   * A subnet is a range of IP address in the VPC. This is a sub-network which allows you to split the network inside the VPC - it is where resources such as EC2 can be launched.
   * A private subnet is a good choice for hosting a Database - it will not be accessible directly from the Internet
   * A public subnet is a good choice for hosting a WebServer - however it requires a NACL, Router and IG to ensure Internet connectivity
   * Each subnet must reside entirely within one Availability Zone and cannot span zones. For HA, launch EC2 instances into subnets of separate AZs
   * Public subnet: The subnet has a direct route to an internet gateway. Resources in a public subnet can access the public internet.
   * Private subnet: The subnet does not have a direct route to an internet gateway. Resources in a private subnet require a NAT device to access the public internet.
   * VPN-only subnet: The subnet has a route to a Site-to-Site VPN connection through a virtual private gateway. The subnet does not have a route to an internet gateway.
   * A subnet CIDR reservation is a range of IPv4 or IPv6 addresses that you set aside so that AWS can't assign them to your network interfaces.
5. [NACL versus Security Group](https://gocloudtech.medium.com/aws-security-groups-vs-nacl-whats-the-difference-a38b9eb6796b) Subnet and Instance traffic rules.
   * NACL is stateless and allow one-way traffic i.e. separatly specific inbound and outbound traffic to the subnet.
   * NACL allow and deny rules are supported. NACLs have an implicit deny. NACL rules are processed in order.
   * Security Group is stateful i.e. rules for inbound and outbound to EC2 instances are same. They allow return traffic.
   * Security Group only supports allow rules.
6. [CloudFront](https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html): CDN
   * A Content Delivery Network (CDN)
   * Provides low latency for content delivery.
   * Global distribution even when it is hosted in a region.
   * Static and dynamic web content. How? Edge location to cache content.
7. [Route53](https://aws.amazon.com/route53/): DNS
   * DNS Service that routes users to Internet applications.
   * Domain Name System - a DNS server translates a domain name to an IP address.
   * Performs health checks on AWS resources
   * Supports hybrid cloud architectures - makes DNS resolution easier
8. [AWS Direct Connect](https://docs.aws.amazon.com/directconnect/latest/UserGuide/Welcome.html): VLAN
   * Dedicated physical network connection from your on-premises data center to AWS
   * Data travels over a private network - virtual LAN from on-prem data center over ethernet fiber optic cable.
   * Supports hybrid cloud architecture e.g. host database in the private cloud and the application on the public cloud, direct connect ensures the two talk and allows for data sovereignity
   * Use case: Transfer internal data directly to AWS bypassing your ISP, or, build hybrid models or transfer large data sets to AWS.
9. [AWS VPN](https://aws.amazon.com/vpn/): VPN
   * Site-to-site VPN creates a secure connection between your internal network and your AWS VPCs.
   * Similar to Direct Connect - but the data travels through the public internet
   * Cheaper than Direct Connect
   * Customer Gateway hosted on-prem connects with a virtual private gateway to establish a site-to-site VPN over the Internet via an ISP.
10. [API Gateway](https://aws.amazon.com/api-gateway/): API Management
    * Build, manage, secure, and scale APIs.
    * API Gateway can invoke services such as Lambda functions.
    * Support RESTful APIs and WEBSOCKET APIs.

**Compute**

1. [Lambda](https://aws.amazon.com/lambda/): Serverless Compute
   * Serverless: write functions and deploy. AWS manages the servers but no direct access.
   * Scales automatically - no need to configure, patch or manage.
   * Use case: Real-time file processing, sending email notifications, Backend business logic
   * Supports Java, Go, PoweShell, Node.Js, C#, Python, and Ruby. Executes code in response to events, timers or other triggers. Lambda has ’re actively hiring engineers as we respond to changing market conditr - deploy a db or web server whatever you need
   * SSH securly connects with a key pair. SSH Client uses private key, the EC2 instance uses a public key
   * EIC is EC2 Instance Connect - uses IAM polices to control SSH access to your instances
   * AWS Systems Manager- use a web browser, or AWS CLI to manage EC2 instances directly
2. [ELB](https://aws.amazon.com/elasticloadbalancing/): Block Storage
   * Distribute network traffic to improve application scalability.
   * Elastic Load Balancing and Auto-Scaling is offered by EC2.
   * Automatically distribute load across servers - classic, application, gateway and network load balancers.
3. [Fargate](https://aws.amazon.com/fargate/): Containers
   * Serverless compute engine for containers.
   * Manages containers like Docker.
   * Scaled automatically - and serverless.
4. [ECS](https://aws.amazon.com/ecs/): Containers
   * Elastic container service.
   * Run highly secure, reliable, and scalable containers.
5. [EKS](https://aws.amazon.com/eks/): Containers
   * Amazone Elastic Kubernetes Service.
   * Start, run, and scale Kubernetes.
6. [Lightsail](https://aws.amazon.com/lightsail/): IAC
   * Quickly lauch all resources you need for small projects.
   * Simple for folks with no cloud experience.
   * Low and predictable fees.
7. [Outpost](https://aws.amazon.com/outposts/): Hybrid Cloud
   * Run AWS Infrastructure and services on premises for a consistent hybrid cloud architecture.
   * Allows cloud services in the internal data ’re actively hiring engineers as we respond to changing market conditcenter
   * Useful for latency or data sovereignty needs
   * Used for hybrid experience
8. [Batch](https://aws.amazon.com/batch/): IAC Spot
   * Process large workloads in smaller chunks.
   * Dynamically provisions instances based on volume.
   * Example - send high volume email 1000 emails at a time or process ML.

**Operational Databases and Caching**

1. [Relational Database Service (RDS)](https://aws.amazon.com/rds/)
   * Launch, manage and scale relational databases on the cloud. Supports Aurora, PostgreSQL, MySQL, MariaDB, Oracle, SQLServer.
   * Offers HA, fault tolerance using Multi-AZ deployment option.
   * AWS manages the database with automatic software patching, automated backups, OS updates.
   * Launch read-repliccas across Regions in order to provide enhanced performance and durability.
   * Does not automatically add capacity or storage.
2. [Aurora](https://aws.amazon.com/rds/aurora/)
   * AWS build Aurora for the cloud compatible with MySQL and PostgreSQL - created by AWS.
   * Supported MySQL and PostgreSQL database enginges. 5x and 3x faster that native.
   * Scales automatically by adding capacity and storage while providing durability and high availability.
   * Backs up to S3, replication to multiple region and storage across 6 stores.
3. [DynamoDB](https://docs.aws.amazon.com/dynamodb/index.html)
   * Fully managed serverless NoSQL key-value and document database.
   * Scales automatically to massive workloads.
   * Adds capacity automatically.
4. [DocumentDB](https://aws.amazon.com/documentdb/)
   * Fully managed document database that supports MongoDB.
   * Serverless, scales enterprise workloads using a fully managed native JSON document database.
5. [Neptune](https://aws.amazon.com/neptune/)
   * Graph database service, fully managed and serverless.
   * Fast, reliable and durable.
   * User profiles and social connections.
   * Usecases: Customer360, Detect fraud patterns, machine learning predictions, IT security detection and investigation.
6. [ElastiCache](https://aws.amazon.com/elasticache/)
   * Microsecond latency and scale with in-memory caching.
   * In-memory data cache compatible with [Redis and Memcache](https://aws.amazon.com/elasticache/redis-vs-memcached/).
   * High-performance, low latency and no durability.
   * Usecases: Application performance, ease backend database load, low latency data retrieval needs.

**Data Migration and Transfer**

1. [Database Migration Service](https://aws.amazon.com/dms/) \*
   * [Feature rich tool](https://aws.amazon.com/dms/features/) that helps you migrate databases to or within AWS.
   * Homogenous and hetrogenous databases can be migrated with virtually no downtime.
   * Data is synchronized between the source and target continuously.
2. [Server Migration Service](https://aws.amazon.com/server-migration-service/): Deprecated in favor of AWS MGN (AWS Application Migration Services)
   * AWS Server Migration Service will automatically replicate live server volumes to AWS and create Amazon Machine Images (AMI) as needed.
   * This is being discontiuned in favor of AWS Application Migration Service.
3. [Application Migration Service](https://aws.amazon.com/application-migration-service/): Lift and Shift
   * Migrate applications from any source infrastructure that runs supported operating systems.
   * Application Migration Service is the next generation of CloudEndure Migration
4. [Snow Family](https://aws.amazon.com/snow/)
   * Move large amounts of data to and from AWS physically or process data at the edge.
   * Snowcone: Smallest member holds 8TB of usable storage, collect process
   * Snowball: 80TB. Cheaper And Snowball Edge used for petabyte scale data migration and has local processing when in a remote environment - supports EC2 and lambda.
   * Snowmobile: 100PB. Multi-perabyte or exabyte scale. Data loaded to S3 - securely transported with escort vehicle.
5. [Data Sync](https://docs.aws.amazon.com/datasync/latest/userguide/what-is-datasync.html) Data Transfer Service
   * Data transfer online with speeds are 10x faster.
   * Data replication cross-region and cross-account.

**Data Analytics**

1. [RedShift](https://aws.amazon.com/redshift/) : Data warehouse
   * Data warehouse: data storage solution with historical data from disparate sources.
   * Business intelligence, querying and business intelligence.
   * Handles exabyte-scale data.
   * Use case: Data consolidation. Run a database when it doesn't require CRUD.
   * Analytics - allows querying to gain business insights.
2. [Glue](https://aws.amazon.com/glue/) : ETL \*
   * Discover, prepare, and integrate all your data at any scale.
   * ETL Service.
   * Prepare to better understand your data.
3. [Lake Formation](https://aws.amazon.com/lake-formation/) : Data Lake
   * Build, manage, and secure data lakes in days.
   * Create, administer, and protect data lakes using familiar database-like features quickly.
4. [QuickSight](https://aws.amazon.com/quicksight/): BI
   * Business Analytics visualization of data with interactive dashboards that can be embedded in your applications
5. [Athena](https://aws.amazon.com/athena/) : SQL for S3
   * Analyze petabyte-scale data where it lives with ease and flexibility.
   * S3 SQL. Pre-configured to work with Glue.
   * Query service to analyze data using SQL. It is serverless.
   * Use cases: run federated queries across relational, nonrelational, object, and custom data sources running on premises or in the cloud. Use ML models in SQL queries or Python. Build distributed big data reconciliation engines. Analyze google analytics data by using AppFlow to store in S3 to query it.
6. [Data Pipeline](https://aws.amazon.com/datapipeline/) :
   * Helps you move data between compute and storage services running either AWS or on-premises
   * Move data based on conditions, intervals and sends notifactions
   * Move from S3 to Redshift.

**Big Data and Search**

1. [EMR](https://aws.amazon.com/emr/) Map Reduce
   * Process large amounts of data via map reduce.
   * Analyze data using Hadoop and Apache Spark.
   * Usecase: Perform big data analytics, build scalable data piplelines, process real-time data streams, accelerate data science and ML adoption.
2. [OpenSearch](https://aws.amazon.com/opensearch-service/) Interactive Log Analytics
   * Search petabytes of unstructured data.
   * Open source Elastic Search, Open Search Dashboard and Kibana.

**Streams**

1. [Kinesis](https://aws.amazon.com/kinesis/): Stream proecessor
   * Easily collect, process, and analyze video and data streams in real time.
   * Usecase: Real-time video and data streams, IoT Data, Click Log, Web Stream logs are good use-cases.
   * Evolve from batch to real-time analytics.
2. [MSK](https://aws.amazon.com/msk/): Kafka
   * Managed Streaming for Apache Kafka.
   * Usecase: Ingest and process log and event streams, run centralized state or data buses, power your event-driven systems.

**Artificial Intelligence and Machine Learning**

1. [Rekognition](https://aws.amazon.com/rekognition/)usecases: Computer Vision
   * Automate image and video analysis
   * Identify custom labels in image and video
   * Use cases: Analyze pizza images to ensure toppings
2. [Comprehend](https://aws.amazon.com/comprehend/): NLP
   * Natural Language Processing (NLP) Service that finds relationships in text
   * Customer sentiment analysis on social media
3. [Polly](https://aws.amazon.com/polly/): Speech-to-text
   * High quality natural sounding human voices in dozens of languages.
   * Customize Text to speech output with Speech Synthesis Markup Language tags.
   * Usecases: Generate speech in dozens of languages, engage customers with a natural-sounding voice, adjust speaking style, speech rate, pitch and loudness.
4. [SageMaker](https://aws.amazon.com/sagemaker/): ML
   * Machine Learning service.
   * Helps you build, train and deploy machine learning models quickly.
   * Prepare data for models, train and deploy models, provides deep learning AMIs.
   * Recommendation engine for movies, music etc.
5. [Translate](https://aws.amazon.com/translate/): Translate
   * Provides language tanslation and support many languages and content formats.
   * Use case: Add localization to websites and applications.
6. [Lex](https://aws.amazon.com/lex/): Chatbot
   * Chatbots with conversational AI.
   * Helps you build conversational interfaces like chatbots.
   * Recognize speech and understand language.
   * Powers Amazon Alexa.
   * Integrate voice into device.
   * Usecases: Build virtual agents and voice assistants, automate informational responses, improve productivity with application bots, maxminize the information trapped in transcripts.

**Storage**

1. [Simple Storage Service S3](https://docs.aws.amazon.com/cli/latest/reference/s3/) - Regional Service with global namespace and bucket policies
   * Unique name across all buckets in AWS
   * 11 9s of durability: regional level redundancy
   * 4 9s of availability
   * S3 does not automatically replicate across regions - it can be setup.
   * Usecase: Host static websites, data archivale, analytics such as redshift and athena. Upload with S3 transfer acceleration for file uploads from mobile applications.
   * S3 Storage Class
     + Standard: Durable 11-9s. 4-9s available.
     + Intelligent Tiering: Unknown or changing access. Standard durability with 3-9s availability
     + Infrequent Access: For Long-Lived, Infrequently Accessed, Millisecond access when needed. Durable with 3 9s availability.
     + One-Zone Infrequent Access: Cost 20% less than IA. Use if data is recreatable, infrequent millisecond access, availability is 99.5%.
     + Glacier: Data retrieval options 1-5 minutes, 3-5 hours, 5-12 hours. Multiple AZs. Standard durability. Cheap storage options.
     + Glacier Deep Archieve: 12hrs or 48hr retrieval options. Cheapest. Long-term data archivale accessed once or twice a year. No availability - but standard durability.
     + Outposts: Data that needs to be kept local. Demanding application performance needs.
   * Buckets: Root level 'folders' for file storage
     + Folder
     + Object Durability
     + Object Availability
     + Object Lifecycle
     + Object sharing
     + Object versioning
   * [S3 Transfer Acceleration](https://aws.amazon.com/s3/transfer-acceleration/)
   * S3TA improves uploads and downloads to and from S3 buckets between 50% and 500%.
   * Moves data faster over longer distances.
   * Shorten distance to S3 via CloudFront.
2. [EC2 Instance Storage](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Storage.html)
   * Emphemeral storage that is temporary block-level for your instance.
   * Lasts during the life of the instance.
   * It is temporary block-level storage for instances.
   * Provides local fastest I/O.
3. [EBS - Elastic Block Storage](https://aws.amazon.com/ebs/)
   * Scalable block storage at any scale. Raw volume.
   * Good for database storage.
   * HDD with an independent life from the instance it is attached to.
   * Only one per instance.
   * Use cases: Build SAn in the cloud for I/O intensive applications, Run relational or NoSQL databases, reight-size your big-data analytic engine.
4. [EFS - Elastic File System](https://aws.amazon.com/efs/) : Shared file system.
   * EFS file system as a common data source for workloads and applications running on multiple instances
   * Regional serverless network file system. Like dropbox.
   * Only for Linux filesystems.
   * Shared directories. Expensive option.
   * 11-9s durability and 4-9s availability.
5. [Storage Gateway](https://aws.amazon.com/storagegateway/): Hybrid storage
   * On-prem extends storage to cloud.
   * Some on the cloud, some local. File directory - some hosted locally some on the cloud.
   * Moving backups to the cloud.
   * Reduce costs by being selective, opt for low latency local files.
6. [AWS Backup](https://aws.amazon.com/backup/): Backup and recovery
   * Create a backup plan for all storage.

**Messaging and Integration Services**

1. [SQS](https://aws.amazon.com/sqs/): Queue
   * Fully managed message queuing for microservices, distributed systems, and servlerless applications.
   * Sends messages on a queue between publisher and a single subscriber.
   * Securely send sensitive data between applications and centrally manage your keys using AWS Key Management.
   * Reliably deliver large volumes of data, at any level of throughput, without losing messages or needing other services to be available.
   * Usecase: architect a loosely coupled system architecture such as money transfer application. Improve performance and scalability. Requests are processed in FIFO.
2. [SNS](https://aws.amazon.com/sns/): Topic
   * Simple Notification Service - Fully managed Pub/Sub service for A2A and A2P messaging.
   * A2P with SMS, texts, push notifactions and email (plain text).
3. [SES](https://aws.amazon.com/ses/): Email
   * Sends rich text HTML Emails from your applications.
   * Get reliable, scalable email to communicate with customers at the lowest industry prices.
   * Marketing campaigns, and professional richly formatted HTML text.

**Developer Tools**

1. [Cloud9](https://aws.amazon.com/cloud9/): IDE
   * IDE write and debug code in your browser
   * Build serverless applications - preconfigures environment.
2. [CodeCommit](https://aws.amazon.com/codecommit/) : Git
   * Source Control system for private Git repositories.
3. [CodeBuild](https://aws.amazon.com/codebuild/): Build Server
   * Allows you to build and test your applicaton source code.
   * Compiles source code and runs tests.
   * Enables CI-CD
   * Produces build artifacts ready to be deployed
4. [CodeDeploy](https://aws.amazon.com/codedeploy/) : Delivery Server
   * Automate code edeployment to maintain application uptime.
   * Manage the deployment of code to on-premises as well as cloud.
   * Use prepackaged build environments or your own, and encrypt artifacts with your own keys.
   * Maintain application uptime, deploy to EC2, lambda, fargate and others.
   * Supports rolling deployments - it minimizes application downtime.
5. [CodePipeline](https://aws.amazon.com/codepipeline/): Release Server
   * Automate release pipelines with CI-CD.
   * AWS offers continuous integration and continuous delivery service.
6. [CodeStar](https://aws.amazon.com/codestar/features/): Pre-configured CI-CD with CodeCommit, CodeBuild, CodeDeploy and CodePipeline out of the box.
   * AWS CodeStar allows you to accelerate application delivery by providing a pre-configured continuous delivery toolchain for developing, building, testing, and deploying your projects on AWS.
7. [X-Ray](https://docs.aws.amazon.com/xray/latest/devguide/aws-xray.html): NDC Logs
   * X-Ray uses trace data from the AWS resources that power your cloud applications to generate a detailed service map. Typically, applications use nested diagnostic context (NDC) for distributed tracing for microservices.
   * The service map shows the client, your front-end service, and backend services that your front-end service calls to process requests and persist data.
   * Use the service map to identify bottlenecks, latency spikes, and other issues to solve to improve the performance of your applications.

**Deployment and Infrastructure Management Service**

1. [CloudFormation](https://aws.amazon.com/cloudformation/): IaC
   * Speed up cloud provisioning with infrastructure as code (IaC).
   * A CloudFormation template describes your desired resources and their dependencies so you can launch and configure them together as a stack.
   * JSON and YAML are supported - define templates to create stacks.
   * Repeatable process for provisioning of resources.
   * Usecase: automate the infrastructure-provisiong for EC2 servers
2. [Elastic Beanstalk](https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/Welcome.html): IaC for dummies
   * Deploy your web applications and services to AWS and not on-prem.
   * Orchestration service that provisions resources.
   * Automatically handles deployments, handles capacity provisioning, load balancing and auto-scaling.
   * Monitors application health via a health dashboard.
   * Usecase: Quickly deploy a scalable Java-based web application to AWS.
3. [OpsWorks](https://aws.amazon.com/opsworks/): DevSecOps
   * Automate operations with Chef and Puppet on-premises or AWS.
   * OpsWorks has three offerings, AWS Opsworks for Chef Automate, AWS OpsWorks for Puppet Enterprise, and AWS OpsWorks Stacks.

**Auditing, Monitoring and Logging**

1. [CloudTrail](https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-user-guide.html): Audit Trails
   * Log and retain account activity as well as unusual activity - enable operational and risk auditing, governance, and compliance of your AWS account
   * Actions taken by a user, role, or an AWS service are recorded as events in CloudTrail.
   * Events include actions taken in the AWS Management Console, AWS Command Line Interface, and AWS SDKs and APIs.
   * If a user terminates an EC2 instance via an API. Cloudtrail will be able to tell which user took that action.
   * Username, event time and name, IP address, access key, region, and error code can be tracked.
2. [CloudWatch](https://aws.amazon.com/cloudwatch/): Logs
   * Observe and monitor resources and applications on AWS, on premises, and on other clouds e.g. EC2 on AWS can be watched.
   * Amazon CloudWatch is a monitoring and management service for AWS, hybrid, and on-premises applications and infrastructure resources.
   * Performance and operational data in the form of logs and metrics.
   * Use to detect anomalies in your environment. Set alarms.
   * Use cases: Monitor full stack (applications, infrastructure, network, and services) and use alarms, logs, and events data to take automated actions and reduce mean time to resolution (MTTR).
3. [Amazon Workspace](https://clients.amazonworkspaces.com/) : VDI
   * Allows you to host virtual desktops in the cloud.
   * Enables employees to work from Home with no data stored on local devices.
   * Use cases: Desktop as a service, Virtual Desktop (VDI).
4. [Amazon Connect](https://aws.amazon.com/connect/) : Contact Center
   * Provide customer service at a lower cost with a cloud contact center.
   * Cloud contact center service.
   * Provides customer service functionality.
   * Improves productivity of help desk.
   * Use cases: omnichannel self-service experience, agent productivity with AI, optimize from insights.