Amazon RDS manages the work involved in setting up a relational database, from provisioning the infrastructure capacity you request to installing the database software. Once your database is up and running, Amazon RDS automates common administrative tasks such as performing backups and patching the software that powers your database. With optional Multi-AZ deployments, Amazon RDS also manages synchronous data replication across Availability Zones with automatic failover. Since Amazon RDS provides native database access, you interact with the relational database software as you normally would. This means you're still responsible for managing the database settings that are specific to your application. You'll need to build the relational schema that best fits your use case and are responsible for any performance tuning to optimize your database for your application's workflow.

Inherited Controls – Controls which a customer fully inherits from AWS.

- Physical and Environmental controls

Shared Controls – Controls which apply to both the infrastructure layer and customer layers, but in completely separate contexts or perspectives. In a shared control, AWS provides the requirements for the infrastructure and the customer must provide their own control implementation within their use of AWS services. Examples include:

- Patch Management – AWS is responsible for patching and fixing flaws within the infrastructure, but customers are responsible for patching their guest OS and applications.
- Configuration Management – AWS maintains the configuration of its infrastructure devices, but

a customer is responsible for configuring their own guest operating systems, databases, and applications.

- Awareness & Training - AWS trains AWS employees, but a customer must train their own employees.

Customer Specific – Controls which are solely the responsibility of the customer based on the application they are deploying within AWS services. Examples include:

- Service and Communications Protection or Zone Security which may require a customer to route or zone data within specific security environments.

Installing the database software is AWS' responsibility.

Performing backups is AWS' responsibility.

Patching the database software is AWS' responsibility.

AWS WAF (Web Application Firewall) helps protect your web applications from common web exploits that could affect application availability, compromise security, or consume excessive resources. You can use AWS WAF to create custom rules that block common attack patterns, such as SQL injection or cross-site scripting, and rules that are designed for your specific application.

Amazon Aurora is a database service.

AWS IAM refers to the AWS Identity and Access Management.

Amazon Cognito lets you add user sign-up, sign-in, and access control to your web and mobile apps.

Amazon S3 Transfer Acceleration enables fast, easy, and secure transfers of files over long distances between your client and an S3 bucket. Transfer Acceleration takes advantage of Amazon CloudFront's globally distributed edge locations. As the data arrives at an edge location, data is routed to Amazon S3 over an optimized network path.

AWS Snowball is a petabyte-scale data transport solution that uses devices designed to be secure to transfer large amounts of data into and out of the AWS Cloud.

AWS WAF refers to the AWS Web Application Firewall service.

AWS Snowmobile is an Exabyte-scale data transfer service used to move extremely large amounts of data to AWS.

The pillars of the AWS Well-Architected Framework

Operational Excellence The ability to support development and run workloads effectively, gain insight into their operations, and to continuously improve supporting processes and procedures to deliver business value.

Security The security pillar describes how to take advantage of cloud technologies to protect data, systems, and assets in a way that can improve your security posture.

Reliability The reliability pillar encompasses the ability of a workload to perform its intended function correctly and consistently when it's expected to. This includes the ability to operate and test the workload through its total lifecycle. This paper provides in-depth, best practice guidance for implementing reliable workloads on AWS.

Performance Efficiency The ability to use computing resources efficiently to meet system requirements, and to maintain that efficiency as demand changes and technologies evolve.

Cost Optimization The ability to run systems to deliver business value at the lowest price point.

General Design Principles The Well-Architected Framework identifies a set of general design principles to facilitate good design in the cloud:

• Stop guessing your capacity needs: If you make a poor capacity decision when deploying a workload, you might end up sitting on expensive idle resources or dealing with the performance implications of limited capacity. With cloud computing, these problems can go away. You can use as much or as little capacity as you need, and scale up and down automatically.

• Test systems at production scale: In the cloud, you can create a production-scale test environment on demand, complete your testing, and then decommission the resources. Because you only pay for the test environment when it's running, you can simulate your live environment for a fraction of the cost of testing on premises

Automate to make architectural experimentation easier: Automation allows you to create and replicate your workloads at low cost and avoid the expense of manual effort. You can track changes to your automation, audit the impact, and revert to previous parameters when necessary.

• Allow for evolutionary architectures: In a traditional environment, architectural decisions are often implemented as static, onetime events, with a few major versions of a system during its lifetime. As a business and its context continue to evolve, these initial decisions might hinder the system's ability to deliver changing business requirements. In the cloud, the capability to automate and test on demand lowers the risk of impact from design changes. This allows systems to evolve over time so that businesses can take advantage of innovations as a standard practice.

• Drive architectures using data: In the cloud, you can collect data on how your architectural choices affect the behavior of your workload. This lets you make fact-based decisions on how to improve your workload. Your cloud infrastructure is code, so you can use that data to inform your architecture choices and improvements over time.

• Improve through game days: Test how your architecture and processes perform by regularly scheduling game days to simulate events in production. This will help you understand where improvements can be made and can help develop organizational experience in dealing with events.

There are five design principles for operational excellence in the cloud:

• Perform operations as code: In the cloud, you can apply the same engineering discipline that you use for application code to your entire environment. You can define your entire workload (applications, infrastructure) as code and update it with code. You can implement your operations procedures as code and automate their execution by triggering them in response to events. By performing operations as code, you limit human error and enable consistent responses to events.
• Make frequent, small, reversible changes: Design workloads to allow components to be updated regularly. Make changes in small increments that can be reversed if they fail (without affecting customers when possible).
• Refine operations procedures frequently: As you use operations procedures, look for opportunities to improve them. As you evolve your workload, evolve your procedures appropriately. Set up regular 5 AWS Well-Architected Framework AWS Well-Architected Framework Definition game days to review and validate that all procedures are effective and that teams are familiar with them.
• Anticipate failure: Perform “pre-mortem” exercises to identify potential sources of failure so that they can be removed or mitigated. Test your failure scenarios and validate your understanding of their impact. Test your response procedures to ensure that they are effective, and that teams are familiar with their execution. Set up regular game days to test workloads and team responses to simulated events.
• Learn from all operational failures: Drive improvement through lessons learned from all operational events and failures. Share what is learned across teams and through the entire organization.

There are four best practice areas for operational excellence in the cloud: • Organization • Prepare • Operate • Evolve

There are seven design principles for security in the cloud:

• Implement a strong identity foundation: Implement the principle of least privilege and enforce separation of duties with appropriate authorization for each interaction with your AWS resources. Centralize identity management, and aim to eliminate reliance on long-term static credentials.
• Enable traceability: Monitor, alert, and audit actions and changes to your environment in real time. Integrate log and metric collection with systems to automatically investigate and take action.
• Apply security at all layers: Apply a defense in depth approach with multiple security controls. Apply to all layers (for example, edge of network, VPC, load balancing, every

instance and compute service, operating system, application, and code).
• Automate security best practices: Automated software-based security mechanisms improve your ability to securely scale more rapidly and cost-effectively. Create secure architectures, including the implementation of controls that are defined and managed as code in version-controlled templates.
• Protect data in transit and at rest: Classify your data into sensitivity levels and use mechanisms, such as encryption, tokenization, and access control where appropriate.
• Keep people away from data: Use mechanisms and tools to reduce or eliminate the need for direct access or manual processing of data. This reduces the risk of mishandling or modification and human error when handling sensitive data.
• Prepare for security events: Prepare for an incident by having incident management and investigation policy and processes that align to your organizational requirements. Run incident response simulations and use tools with automation to increase your speed for detection, investigation, and recovery.

Definition There are six best practice areas for security in the cloud:
• Security
• Identity and Access Management
• Detection • Infrastructure Protection
• Data Protection
• Incident Response

Design Principles There are five design principles for reliability in the cloud:
• Automatically recover from failure: By monitoring a workload for key performance indicators (KPIs), you can trigger automation when a threshold is breached. These KPIs should be a measure of business value, not of the technical aspects of the operation of the service. This allows for automatic notification and tracking of failures, and for automated recovery processes that work around or repair the failure. With more sophisticated automation, it's possible to anticipate and remediate failures before they occur.
• Test recovery procedures: In an on-premises environment, testing is often conducted to prove that the workload works in a particular scenario. Testing is not typically used to validate recovery strategies. In the cloud, you can test how your workload fails, and you can validate your recovery procedures. You can use automation to simulate different failures or to recreate scenarios that led to failures before. This approach exposes failure pathways that you can test and fix before a real failure scenario occurs, thus reducing risk.
• Scale horizontally to increase aggregate workload availability: Replace one large resource with multiple small resources to reduce the impact of a single failure on the overall workload. Distribute requests across multiple, smaller resources to ensure that they don't share a common point of failure.

• Stop guessing capacity: A common cause of failure in on-premises workloads is resource saturation, when the demands placed on a workload exceed the capacity of that workload (this is often the objective of denial of service attacks). In the cloud, you can monitor demand and workload utilization, and automate the addition or removal of resources to maintain the optimal level to satisfy demand without over- or under-provisioning. There are still limits, but some quotas can be controlled and others can be managed (see Manage Service Quotas and Constraints).
• Manage change in automation: Changes to your infrastructure should be made using automation. The changes that need to be managed include changes to the automation, which then can be tracked and reviewed.

There are four best practice areas for reliability in the cloud: • Foundations • Workload Architecture • Change Management • Failure Management

There are five design principles for performance efficiency in the cloud: • Democratize advanced technologies: Make advanced technology implementation easier for your team by delegating complex tasks to your cloud vendor. Rather than asking your IT team to learn about hosting and running a new technology, consider consuming the technology as a service. For example, NoSQL databases, media transcoding, and machine learning are all technologies that require specialized expertise. In the cloud, these technologies become services that your team can consume, allowing your team to focus on product development rather than resource provisioning and management.

• Go global in minutes: Deploying your workload in multiple AWS Regions around the world allows you to provide lower latency and a better experience for your customers at minimal cost.

• Use serverless architectures: Serverless architectures remove the need for you to run and maintain physical servers for traditional compute activities. For example, serverless storage services can act as static websites (removing the need for web servers) and event services can host code. This removes the operational burden of managing physical servers, and can lower transactional costs because managed services operate at cloud scale.

• Experiment more often: With virtual and automatable resources, you can quickly carry out comparative testing using different types of instances, storage, or configurations.

• Consider mechanical sympathy: Understand how cloud services are consumed and always use the technology approach that aligns best with your workload goals. For example, consider data access patterns when you select database or storage approaches.

Definition **There are four best practice areas for performance efficiency in the cloud:** •

Selection • Review • Monitoring • Tradeoffs

There are five design principles for cost optimization in the cloud: • Implement Cloud Financial Management: To achieve financial success and accelerate business value realization in the cloud, you need to invest in Cloud Financial Management /Cost Optimization. Your organization needs to dedicate time and resources to build capability in this new domain of technology and usage management. Similar to your Security or Operational Excellence capability, you need to build capability through knowledge building, programs, resources, and processes to become a cost-efficient organization.

• Adopt a consumption model: Pay only for the computing resources that you require and increase or decrease usage depending on business requirements, not by using elaborate forecasting. For example, development and test environments are typically only used for eight hours a day during the work week. You can stop these resources when they are not in use for a potential cost savings of 75% (40 hours versus 168 hours).

• Measure overall efficiency: Measure the business output of the workload and the costs associated with delivering it. Use this measure to know the gains you make from increasing output and reducing costs.

• Stop spending money on undifferentiated heavy lifting: AWS does the heavy lifting of data center operations like racking, stacking, and powering servers. It also removes the operational burden of managing operating systems and applications with managed services. This allows you to focus on your customers and business projects rather than on IT infrastructure.

• Analyze and attribute expenditure: The cloud makes it easier to accurately identify the usage and cost of systems, which then allows transparent attribution of IT costs to individual workload owners. This helps measure return on investment (ROI) and gives workload owners an opportunity to optimize their resources and reduce costs.

Definition There are five best practice areas for cost optimization in the cloud: • Practice Cloud Financial Management • Expenditure and usage awareness • Cost-effective resources • Manage demand and supply resources • Optimize over time

EC2 instance pricing varies depending on many variables:
- The buying option (On-demand, Savings Plans, Reserved, Spot, Dedicated)
- Selected instance type
- Selected Region

- Number of instances
- Load balancing
- Allocated Elastic IP Addresses

Amazon Inspector is an automated security assessment service that helps you test the network accessibility of your Amazon EC2 instances and the security state of your applications running on the instances. Amazon Inspector allows you to create assessment templates to automate security vulnerability assessments throughout your development and deployment pipelines or for static production systems.

Amazon EMR helps you analyze and process vast amounts of data by distributing the computational work across a cluster of virtual servers running in the AWS Cloud. The cluster is managed using an open-source framework called Hadoop. Amazon EMR lets you focus on crunching or analyzing your data without having to worry about time-consuming setup, management, and tuning of Hadoop clusters or the compute capacity they rely on.

DB instances for Amazon RDS for MySQL, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server use Amazon Elastic Block Store (Amazon EBS) volumes for database and log storage.

Amazon S3 refers to the simple storage service. Amazon S3 is an **object level storage** that cannot be used to store running operating systems or live databases.

Amazon EFS refers to the Amazon Elastic File System. Amazon EFS is a file level storage that provides a scalable, elastic **NFS file system** for Linux-based workloads for use with AWS Cloud services and on-premises resources. Amazon RDS does not use Amazon EFS to store databases.

Amazon Glacier is used for storing backups and long-term data.

AWS Global Accelerator and CloudFront are two separate services that use the AWS global network and its edge locations around the world. Amazon CloudFront improves performance for global applications by caching content at the closest Edge Location to end-users. AWS Global Accelerator improves performance for global applications by routing end-user requests to the closest AWS Region. Amazon CloudFront improves performance for both cacheable (e.g., images and videos) and dynamic content (e.g. dynamic site delivery). Global Accelerator is a good fit for specific use cases, such as gaming, IoT or Voice over IP.

Note: AWS Global accelerator does not cache content at edge locations like Amazon CloudFront. AWS Global accelerator uses the AWS edge locations to receive end-user requests and then routes these requests to the closest AWS Region over the AWS global network.

AWS KMS is a key management service that makes it easy for you to create and manage encryption keys and control their use across a wide range of AWS services and in your applications.

AWS Direct Connect is a cloud service solution that is used to establish a dedicated network connection from your premises to AWS.

AWS Glue is a fully-managed, Extract, Transform, and Load (ETL) service that automates the time-consuming steps of data preparation for analytics.
Extract, Transform, and Load (ETL) is the process of **extracting** (collecting) data from various sources (from different databases for example), **transform** the data depending on business rules/needs (This step helps in preparing the data for analytics and decision making) and **load** the data into a destination database, often a data warehouse.

AWS Cloud9 is a cloud-based integrated development environment (IDE) that lets you write, run, and debug your code with just a browser. It includes a code editor, debugger,

and terminal. Cloud9 comes prepackaged with essential tools for popular programming languages, including JavaScript, Python, PHP, and more, so you don't need to install files or configure your development machine to start new projects.

AWS Artifact provides on-demand access to AWS' compliance reports.

AWS CloudTrail is a web service that tracks and records all user interactions with AWS services.

AWS X-Ray helps developers analyze and debug distributed applications in production or under development, such as those built using microservice architecture. With X-Ray, you can understand how your application and its underlying services are performing so you can identify and troubleshoot the root cause of performance issues and errors. X-Ray provides an end-to-end view of requests as they travel through your application, and shows a map of your application's underlying components. You can use X-Ray to analyze both applications in development and in production, from simple three-tier applications to complex microservices applications consisting of thousands of services.

AWS CodePipeline is a fully managed continuous delivery service that helps you automate your release pipelines for fast and reliable application and infrastructure updates.

Amazon Inspector helps you to identify security vulnerabilities as well as deviations from security best practices in applications NOT for troubleshooting performance issues.

CloudTrail is a service that allows you to track all users' actions that are taken in your AWS account.

AWS Elastic Beanstalk is an application container on top of Amazon Web Services. Elastic Beanstalk makes it easy for developers to quickly deploy and manage applications in the AWS Cloud. Developers simply upload their application code, and Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, auto-scaling, and application health monitoring.

AWS Elastic Beanstalk is not a database, compute engine nor storage service, AWS Elastic Beanstalk uses proven AWS features and services, such as Amazon EC2, Amazon RDS, Elastic Load Balancing, Auto Scaling, Amazon S3, and Amazon SNS, to create an environment that runs your application.

Change management is defined as "the Process responsible for controlling the Lifecycle of all Changes. The primary objective of Change Management is to enable beneficial changes to be made, with minimum disruption to IT Services.

Despite all of the investments in software and hardware, an erroneous configuration or misstep in a process can frequently undo these efforts and lead to failure.

AWS Config and AWS CloudTrail are change management tools that help AWS customers audit and monitor all resource and configuration changes in their AWS environment

AWS Transit Gateway is a network transit hub that customers can use to interconnect their virtual private clouds (VPCs) and their on-premises networks. AWS transit gateway simplifies how customers interconnect all of their VPCs, across thousands of AWS accounts and into their on-premises networks.

AWS X-Ray is a debugging service that helps developers understand how their application and its underlying services are performing to identify and troubleshoot the root cause of performance issues and errors.

Amazon Comprehend is a **Natural Language Processing (NLP)** service that uses machine learning to find meaning and insights in text. Customers can use Amazon Comprehend to identify the language of the text, extract key phrases,

places, people, brands, or events, understand sentiment about products or services, and identify the main topics from a library of documents. The source of this text could be web pages, social media feeds, emails, or articles. Amazon Comprehend is fully managed, so there are no servers to provision, and no machine learning models to build, train, or deploy.
Note: Natural language processing (NLP) is an artificial intelligence technology that helps computers identify, understand, and manipulate human language.

AWS CloudTrail is designed to log all actions taken in your AWS account. This provides a great resource for governance, compliance, and risk auditing.

Amazon **CloudFront** is a content delivery network (CDN) service.

CloudEndure Migration simplifies the process of migrating applications from physical, virtual, and cloud-based infrastructure, ensuring that they are fully operational in any AWS Region without compatibility issues.

Amazon **CloudWatch** is used to monitor the utilization of AWS resources. CloudWatch provides you with data and actionable insights to monitor your applications, understand and respond to system-wide performance changes, and get a unified view of operational health.

AWS **Direct Connect** is used to establish a dedicated network connection from your premises to AWS. Using AWS Direct Connect, you can establish private connectivity between AWS and your data center, office, or co-location environment, which in many cases can reduce your network costs, increase bandwidth throughput, and provide a more consistent network experience than Internet-based connections.

AWS Snowball is used to physically migrate petabyte-scale data sets into and out of AWS.

Amazon CloudFront is a content delivery network that provides faster response times for your global users.

Amazon Route 53 is a global service that provides a highly available and scalable Domain Name System (DNS) in the Cloud.

The Multi-AZ principle involves deploying an AWS resource in multiple Availability Zones to achieve high availability for that resource.

DynamoDB automatically spreads the data and traffic for your tables over a sufficient number of servers to handle your throughput and storage requirements, while maintaining consistent and fast performance. All of your data is stored on solid-state disks (SSDs) and is automatically replicated across multiple Availability Zones in an AWS Region, providing built-in fault tolerance in the event of a server failure or Availability Zone outage.

Amazon S3 provides durable infrastructure to store important data and is designed for durability of 99.999999999% of objects. Data in all Amazon S3 storage classes is redundantly stored across multiple Availability Zones (except S3 One Zone-IA).

Currently, **Amazon Redshift** only supports Single-AZ deployments.

AWS Snowball is a data transport solution that accelerates moving terabytes to petabytes of data into and out of AWS using storage devices designed to be secure for physical transport.

Amazon EBS volume data is replicated across multiple servers within the same Availability Zone.
Note:
Amazon EFS data is redundantly stored across multiple Availability Zones providing better durability compared to EBS volumes.

Amazon Aurora is a MySQL and PostgreSQL compatible relational database built for the cloud, that combines the performance and availability of high-end commercial databases with the simplicity and cost-effectiveness of open source databases. Aurora is up to five times faster than standard MySQL databases and three times faster than standard PostgreSQL databases. It provides the security, availability, and reliability of commercial-grade databases at 1/10th the cost. Aurora is fully managed by Amazon Relational Database Service (RDS), which automates time-consuming administration tasks like hardware provisioning, database setup, patching, and backups.

Amazon Aurora features "Amazon Aurora Serverless" which is an on-demand, auto-scaling configuration for Amazon Aurora (MySQL-compatible and PostgreSQL-compatible editions), where the database will automatically start up, shut down, and scale capacity up or down based on your application's needs.

Amazon RDS PostgreSQL is used to run PostgreSQL databases NOT MySQL databases.

Amazon RDS for SQL Server is used to run Microsoft SQL Server databases NOT MySQL databases.

Amazon Neptune is a graph database service NOT a MySQL database. Amazon Neptune can be used to build and run applications that work with highly connected datasets, such as social networking, recommendation engines, and knowledge graphs.

In a Multi-AZ deployment, Amazon RDS automatically provisions and maintains a synchronous standby replica in a different Availability Zone. The primary DB instance is synchronously replicated across Availability Zones to a standby replica to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups. Running a DB instance with high availability can enhance availability during planned system maintenance, and help protect your databases against DB instance failure and Availability Zone disruption.

Amazon RDS Read Replicas provide enhanced performance and durability for database (DB) instances. This feature makes it easy to elastically scale out beyond the capacity constraints of a single DB instance for read-heavy database workloads. You can create one or more replicas of a given source DB Instance and serve high-volume application read traffic from multiple copies of your data, thereby increasing aggregate read throughput.

Read replicas provide a complementary availability mechanism to Amazon RDS Multi-AZ Deployments. You can promote a read replica if the source DB instance fails. You can also replicate DB instances across AWS Regions as part of your disaster recovery strategy. This functionality complements the synchronous replication, automatic failure detection, and failover provided with Multi-AZ deployments.

Edge Locations are not a feature of Amazon RDS. Edge locations are used by the CloudFront service to distribute content globally.

The purpose of patching is to resolve functionality issues, improve security or add new features.

AWS Regions are not a feature of Amazon RDS. AWS Regions are separate geographic areas around the world that AWS uses to provide its Cloud Services, including Regions in North America, South America, Europe, Asia Pacific, and the Middle East. Choosing a specific AWS Region depends on its proximity to end-users, data sovereignty, and costs.

For Customers that can commit to using EC2 over a 1 or 3-year term, it is better to use Amazon EC2 Reserved Instances. Reserved Instances provide a

significant discount (up to 75%) compared to On-Demand instance pricing.

Reserved Instances provide a significant discount (up to 75%) compared to On-Demand (pay-as-you-go) instance pricing.

Pay less as AWS grows refers to the discounts that you get over time as AWS grows. This sometimes called “AWS Economies of Scale”. For example, AWS has reduced the per GB storage price of S3 by 80% since the service was first introduced in 2006.

“Pay less by using more” means that you get volume based discounts and as your usage increases. For services such as S3, pricing is tiered, meaning the more you use, the less you pay per GB.

Access keys consist of two parts: an access key ID and a secret access key. You must provide your AWS access keys to make programmatic requests to AWS or to use the AWS Command Line Interface or AWS Tools for PowerShell. Like a user name and password, you must use both the access key ID and secret access key together to authenticate your requests.

MFA is an additional security layer that can be used to secure your AWS console. MFA can also be used to control access to AWS service APIs.

There are no passwords related to the EC2 instances.

The AWS key pair is used to securely connect to your Amazon EC2 instances.

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity while automating time-consuming administration tasks such as hardware provisioning, operating system maintenance, database setup, patching and backups. It frees you to focus on your applications so you can give them the fast performance, high availability, security and compatibility they need.

Amazon RDS can be used to host Amazon Aurora, PostgreSQL, **MySQL**, MariaDB, Oracle, and Microsoft SQL Server databases.

"Amazon Redshift" Amazon Redshift is not a MySQL database service. Amazon Redshift is a fully managed data warehouse service that makes it simple and cost-effective to analyze all your data using standard SQL and your existing Business Intelligence (BI) tools.

"Amazon DynamoDB" Amazon DynamoDB is not a MySQL database service. Amazon DynamoDB is a fully managed NoSQL database service.

"Amazon CloudWatch" Amazon CloudWatch is not a database service. Amazon CloudWatch is a monitoring service that gives you complete visibility of your cloud resources and applications

AWS Database Migration Service (DMS) helps you migrate databases to AWS easily and securely. The source database remains fully operational during the migration, minimizing downtime to applications that rely on the database. The AWS Database Migration Service can migrate your data to and from most widely used commercial and open-source databases. The service supports homogeneous migrations such as Oracle to Oracle, as well as heterogeneous migrations between different database platforms, such as Oracle to Amazon Aurora or Microsoft SQL Server to MySQL. It also allows you to stream data to Amazon Redshift from any of the supported sources including Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle, SAP ASE, and SQL Server, enabling consolidation and easy analysis of data in the petabyte-scale data warehouse. AWS Database Migration Service can also be used for continuous data replication with high availability.

AWS OpsWorks is a configuration management service that provides managed instances of Chef and Puppet.

"AWS Server Migration Service" AWS Server Migration Service (SMS) is used to migrate your on-premises workloads to AWS.

"AWS Application Discovery Service" AWS Application Discovery Service helps enterprise customers plan migration projects by gathering information about their on-premises data centers.

Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to send, store, and receive messages between software components at any volume, without losing messages or requiring other services to be available. SQS lets you decouple application components so that they run independently, increasing the overall fault tolerance of the system. Multiple copies of every message are stored redundantly across multiple availability zones so that they are available whenever needed.

Amazon SES (Amazon Simple Email Service) is a flexible, affordable, and highly-scalable email messaging platform for businesses and developers.

Amazon Connect is a cloud-based contact center service that makes it easy for businesses to deliver customer service at low cost.

AWS Direct Connect AWS Direct Connect is a cloud service solution that is used to establish a dedicated network connection between your premises and AWS.

The AWS Abuse team can assist you when AWS resources are being used to engage in the following types of abusive behavior:
I. Spam: You are receiving unwanted emails from an AWS-owned IP address, or AWS resources are being used to spam websites or forums.
II. Port scanning: Your logs show that one or more AWS-owned IP addresses are sending packets to multiple ports on your server, and you believe this is an attempt to discover unsecured ports.
III. Denial of service attacks (DOS): Your logs show that one or more AWS-owned IP addresses are being used to flood ports on your resources with packets, and you believe this is an attempt to overwhelm or crash your server or software running on your server.
IV. Intrusion attempts: Your logs show that one or more AWS-owned IP addresses are being used to attempt to log in to your resources.
V. Hosting objectionable or copyrighted content: You have evidence that AWS resources are being used to host or distribute illegal content or distribute copyrighted content without the consent of the copyright holder.
VI. Distributing malware: You have evidence that AWS resources are being used to distribute software that was knowingly created to compromise or cause harm to computers or machines on which it is installed.

Note: Anyone can report abuse of AWS resources, not just AWS customers.

AWS Security team is responsible for the security of services offered by AWS.

AWS Concierge team can assist you with the issues that are related to your billing and account management.

AWS Customer Service team is at the forefront of this transformational technology assisting a global list of customers that are taking advantage of a growing set of services and features to run their mission-critical applications. The team helps AWS customers understand what Cloud Computing is all about, and whether it can be useful for their business needs.

Amazon DynamoDB is a NoSQL database service. NoSQL databases are used for non-structured data that are typically stored in JSON-like, key-value documents.

Amazon Redshift is a data warehouse service that only supports relational data, NOT key-value data.
Additional information:
Amazon Redshift is a fast, fully managed data warehouse service that is specifically designed for online analytic processing (OLAP) and business intelligence (BI) applications, which require complex queries against large datasets.

Amazon Aurora is a MySQL and PostgreSQL-compatible relational database NOT a key-value database.

Amazon RDS is a relational database NOT a key-value database.

The Well-Architected Framework identifies a set of general design principles to facilitate good design in the cloud:
1- Stop guessing your capacity needs: Eliminate guessing about your infrastructure capacity needs. When you make a capacity decision before you deploy a system, you might end up sitting on expensive idle resources or dealing with the performance implications of limited capacity. With cloud computing, these problems can go away. You can use as much or as little capacity as you need, and scale up and down automatically.
2- Test systems at production scale: In the cloud, you can create a production-scale test environment on demand, complete your testing, and then decommission the resources. Because you only pay for the test environment when it's running, you can simulate your live environment for a fraction of the cost of testing on premises.
3- Automate to make architectural experimentation easier: Automation allows you to create and replicate your systems at low cost and avoid the expense of manual effort. You can track changes to your automation, audit the impact, and revert to previous parameters when necessary.
4- Allow for evolutionary architectures: Allow for evolutionary architectures. In a traditional environment, architectural decisions are often implemented as static, one-time events, with a few major versions of a system during its lifetime. As a business and its context continue to change, these initial decisions might hinder the system's ability to deliver changing business requirements. In the cloud, the capability to automate and test on demand lowers the risk of impact from design changes. This allows systems to evolve over time so that businesses can take advantage of innovations as a standard practice.
5- Drive architectures using data: In the cloud you can collect data on how your architectural choices affect the behavior of your workload. This lets you make fact-based decisions on how to improve your workload. Your cloud infrastructure is code, so you can use that data to inform your architecture choices and improvements over time.
6- Improve through game days: Test how your architecture and processes perform by regularly scheduling game days to simulate events in production. This will help you understand where improvements can be made and can help develop organizational experience in dealing with events.

Horizontal Scaling:
Scaling horizontally takes place through an increase in the number of resources (e.g., adding more hard drives to a storage array or adding more servers to support an application). This is a great way to build Internet-scale applications that leverage the elasticity of cloud computing.

Vertical Scaling:
Scaling vertically takes place through an increase in the specifications of an individual resource (e.g., upgrading a server with a larger hard drive, adding more memory, or provisioning a faster CPU). On Amazon EC2, this can easily be achieved by stopping an instance and resizing it to an instance type that has more RAM, CPU, I/O,or networking capabilities. This way of scaling can eventually hit a limit and it is not always a cost efficient or highly available approach. However, it is very easy to implement and can be sufficient for many use cases especially as a short term solution.

Additional information:
Vertical-scaling is often limited to the capacity constraints of a single machine, scaling beyond that capacity often involves downtime and comes with an upper limit. With horizontal-scaling it is often easier to scale dynamically by adding more machines in parallel. Hence, in most cases, horizontal-scaling is recommended over vertical-scaling.

In CloudWatch, you can set up a billing alarm that triggers if your costs exceed a threshold that you set. This CloudWatch alarm can also be configured to trigger an SNS notification to your email address.

AWS Budgets is another AWS service that can be used in this scenario. AWS Budgets gives you the ability to set custom budgets that alert you when your costs or usage exceed (or are forecasted to exceed) your budgeted amount. The difference between AWS Budgets and Amazon CloudWatch billing alarms is that Amazon CloudWatch billing alarms alert you only when your **actual** cost exceeds a certain threshold, while AWS Budgets can be configured to alert you when the **actual** or **forecasted** cost exceeds a certain threshold.

AWS customers are welcome to carry out security assessments and penetration tests against their AWS infrastructure without prior approval for 8 services:

1- Amazon EC2 instances, NAT Gateways, and Elastic Load Balancers.
2- Amazon RDS.
3- Amazon CloudFront.
4- Amazon Aurora.
5- Amazon API Gateways.
6- AWS Lambda and Lambda Edge functions.
7- Amazon Lightsail resources.
8- Amazon Elastic Beanstalk environments.

For AWS-managed services such as Amazon RDS and Amazon DynamoDB, AWS is responsible for performing all the operations needed to keep the service running.
The AWS-managed services automate time-consuming administration tasks such as hardware provisioning, software setup, patching and backups. The AWS-managed services free customers to focus on their applications so they can give them the fast performance, high availability, security and compatibility they need.
Examples of AWS-managed services include Amazon RDS, Amazon DynamoDB, Amazon Redshift, Amazon WorkSpaces, Amazon CloudFront, Amazon CloudSearch, and several other services.
On the other hand, customer-managed services are services that are completely managed by the customer. For example, a service such as Amazon Elastic Compute Cloud (Amazon EC2) is categorized as Infrastructure as a Service (IaaS) and, as such, requires the customer to perform all of the necessary security configuration and management tasks. Customers that deploy an Amazon EC2 instance are responsible for the management of the guest operating system (including updates and security patches), any application software or utilities installed by the customer on the instances, and the configuration of the AWS-provided firewall (called a security group) on each instance.
Examples of customer-managed services include Amazon Elastic Compute Cloud (Amazon EC2), Amazon Virtual Private Cloud (Amazon VPC), and AWS Identity And Access Management (AWS IAM).

EC2 instance pricing varies depending on many variables:
- The buying option (On-demand, Savings Plans, Reserved, Spot, Dedicated)
- Selected instance type
- Selected Region
- Number of instances
- Load balancing
- Allocated Elastic IP Addresses

These are things that traditional web hosting cannot provide:
**High-availability (eliminating single points of failure): A system is highly available when it can withstand the failure of an individual component or multiple components, such as hard disks, servers, and network links. The best way to understand and avoid the single point of failure is to begin by making a list of all major points of your architecture. You need to break the points down and understand them further. Then, review each of these points and think what would happen if any of these failed. AWS gives you the opportunity to automate recovery and reduce disruption at every layer of your architecture.

Additionally, AWS provides fully managed services that enable customers to offload the administrative burdens of operating and scaling the infrastructure to AWS so that they don't have to worry about high availability or Single Point of Failures. For example, AWS Lambda and DynamoDB are serverless services; there are no servers to provision, patch, or manage and no software to install, maintain, or operate. Availability and fault tolerance are built-in, eliminating the need to architect your applications for these capabilities.

**Distributed infrastructure: The AWS Cloud operates in over 75 Availability Zones within over 20 geographic Regions around the world, with announced plans for more Availability Zones and Regions, allowing you to reduce latency to users from all around the world.

**On-demand infrastructure for scaling applications or tasks: AWS allows you to provision the required resources for your application in minutes and also allows you to stop them when you don't need them.

**Cost savings: You don't have to run your own data center for internal or private servers, so your IT department doesn't have to make bulk purchases of servers which may never get used, or may be inadequate. The "pay as you go" model from AWS allows you to pay only for what you use and the ability to scale down to avoid over-spending. With AWS you don't have to pay an entire IT department to maintain that hardware -- you don't even have to pay an accountant to figure out how much hardware you can afford or how much you need to purchase.

Support Concierge is only available for the AWS Enterprise support plan. The Concierge Team are AWS billing and account experts that specialize in working with enterprise accounts. They will quickly and efficiently assist you with your billing and account inquiries, and work with you to implement billing and account best practices so that you can focus on what matters: running your business.

The AWS Management Console lets you create new RDS instances through a web-based user interface.
You can also use AWS CloudFormation to create new RDS instances using the CloudFormation template language.

AWS DMS (database migration service) is used to migrate databases to AWS.

AWS Quick Starts are built by AWS solutions architects and partners to help you deploy popular technologies on AWS, based on AWS best practices for security and high availability. These accelerators reduce hundreds of manual procedures into just a few steps, so you can build your production environment quickly and start using it

immediately.

AWS CodeDeploy is a fully managed deployment service that automates software deployments to a variety of compute services such as Amazon EC2, AWS Fargate, AWS Lambda, and your on-premises servers.

To enable HTTPS connections to your website or application in AWS, you need an SSL/TLS server certificate. You can use a server certificate provided by **AWS Certificate Manager (ACM)** or one that you obtained from an external provider. You can use ACM or IAM to store and deploy server certificates. Use IAM as a certificate manager only when you must support HTTPS connections in a region that is not supported by ACM. IAM supports deploying server certificates in all regions, but you must obtain your certificate from an external provider for use with AWS. Amazon Route 53 is used to register domain names or use your own domain name to route your end users to Internet applications. Route 53 is not responsible for creating SSL certifications.

AWS Directory Service is a managed Microsoft Active Directory in the AWS Cloud. Customers can use it to manage users and groups, provide single sign-on (SSO) to applications and services, as well as create and apply group policies.
Note: What is Single sign-on (SSO)? Single sign-on (SSO) enables a company's employees to sign in to AWS using their existing corporate Microsoft Active Directory credentials.

Amazon Route 53 can be used for registering domain names, routing end users to Internet applications, configuring DNS health checks to route traffic to healthy endpoints, managing traffic globally through a variety of routing types etc.

AWS Data Pipeline is a web service that helps you reliably process and move data between different AWS compute and storage services, as well as on-premises data sources. AWS Data Pipeline integrates with on-premise and cloud-based storage systems to allow developers to use their data when they need it, where they want it, and in the required format.

Amazon S3 provides a number of security features for the protection of data at rest, which you can use or not depending on your threat profile:
1- Permissions: Use bucket-level or object-level permissions alongside IAM policies to protect resources from unauthorized access and to prevent information disclosure, data integrity compromise or deletion.
2- Versioning: Amazon S3 supports object versions. Versioning is disabled by default. Enable versioning to store a new version for every modified or deleted object from which you can restore compromised objects if necessary.
3- Replication: Although Amazon S3 stores your data across multiple geographically diverse Availability Zones by default, compliance requirements might dictate that you store data at even greater distances. Cross-region replication (CRR) allows you to replicate data between distant AWS Regions to help satisfy these requirements. CRR enables automatic, asynchronous copying of objects across buckets in different AWS Regions.
4- Encryption – server side: Amazon S3 supports server-side encryption of user data. Server-side encryption is transparent to the end user. AWS generates a unique encryption key for each object, and then encrypts the object using AES-256.

5- Encryption – client side: With client-side encryption you create and manage your own encryption keys. Keys you create are not exported to AWS in clear text. Your applications encrypt data before submitting it to Amazon S3, and decrypt data after receiving it from Amazon S3. Data is stored in an encrypted form, with keys and algorithms only known to you.

Additional information: (IMPORTANT)

AWS also provides a fully managed security service called AWS Macie to help protect your sensitive data in Amazon S3. Amazon Macie uses machine learning to automatically discover, classify, and protect sensitive data in Amazon S3. Amazon Macie recognizes sensitive data such as personally identifiable information (PII) or intellectual property, and provides you with dashboards and alerts that give visibility into how this data is being accessed or moved. The fully managed service continuously monitors data access activity for anomalies, and generates detailed alerts when it detects risk of unauthorized access or inadvertent data leaks. Today, Amazon Macie is available to protect data stored in Amazon S3, with support for additional AWS data stores coming later this year.

AWS will charge the user once the AWS resource is allocated (even if it is not used). Thus, it is advised that once the user's work is completed they should:
1- Delete all Elastic Load Balancers.
2- Terminate all unused EC2 instances.
3- Delete the attached EBS volumes that they don’t need.
4- Release any unused Elastic IPs.

Amazon EC2 Auto Scaling offers the following benefits:
1- Better fault tolerance. Amazon EC2 Auto Scaling can detect when an instance is unhealthy, terminate it, and launch an instance to replace it. Also, Amazon EC2 Auto Scaling enables you to take advantage of the safety and reliability of geographic redundancy by spanning Auto Scaling groups across multiple Availability Zones within a Region. When one Availability Zone becomes unhealthy or unavailable, Auto Scaling launches new instances in an unaffected Availability Zone. When the unhealthy Availability Zone returns to a healthy state, Auto Scaling automatically redistributes the application instances evenly across all of the designated Availability Zones.
2- Better availability. Amazon EC2 Auto Scaling helps ensure that your application always has the right amount of capacity to handle the current traffic demand.
3- Better cost management. Amazon EC2 Auto Scaling can dynamically increase and decrease capacity as needed. Because you pay for the EC2 instances you use, you save money by launching instances when they are needed and terminating them when they aren't.

When you begin to estimate the cost of using Amazon EC2, consider the following:
1- Clock hours of server time: The amount of time that the instances will be running has a direct bearing on the overall price, as EC2 instances are charged either by the hour or by the second, depending on which AMI is used.
2- Instance type: Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory,

storage, and networking capacity.
3- Pricing model: On-Demand, Reserved, Spot, Savings Plans, and Dedicated
4- Number of instances: You can provision multiple instances of your Amazon EC2 and Amazon EBS resources to handle peak loads.
5- Load balancing: The number of hours the Elastic Load Balancer runs and the amount of data it processes contribute to the EC2 monthly cost.
6- Elastic IP addresses: You can have one Elastic IP (EIP) address associated with a running instance at no charge. Additional Elastic IPs are not free.
7- Operating systems and software packages: Operating system prices are included in instance prices, unless you choose to bring your own licenses.

Customers can work with AWS Identity and Access Management in any of the following ways:
1- AWS Management Console: The console is a browser-based interface that can be used to manage IAM and AWS resources.
2- AWS Command Line Tools: Customers can use the AWS command line tools to issue commands at your system's command line to perform IAM and AWS tasks. Using the command line can be faster and more convenient than the console. The command line tools are also useful if you want to build scripts that perform AWS tasks. AWS provides two sets of command line tools: the AWS Command Line Interface (AWS CLI) and the AWS Tools for Windows PowerShell.
3- AWS SDKs: AWS provides SDKs (software development kits) that consist of libraries and sample code for various programming languages and platforms (Java, Python, Ruby, .NET, iOS, Android, etc.). The SDKs provide a convenient way to create programmatic access to IAM and AWS. For example, the SDKs take care of tasks such as cryptographically signing requests, managing errors, and retrying requests automatically.

AWS Lambda is a compute service that lets customers run code without provisioning or managing servers. AWS Lambda executes code only when needed and scales automatically, from a few requests per day to thousands per second.

With DynamoDB, there are no servers to provision, patch, or manage and no software to install, maintain, or operate. DynamoDB automatically scales tables up and down to adjust for capacity and maintain performance.

AWS Serverless Services include:
Compute: AWS Lambda, AWS Fargate
Messaging: Amazon SNS, Amazon SQS
Database: Amazon DynamoDB**,** Amazon Aurora Serverless
Orchestration: AWS Step Functions

Amazon Cognito lets customers add user sign-up, sign-in, and access control to their web and mobile apps quickly and easily. Amazon Cognito scales to millions of users and supports sign-in with social identity providers, such as Facebook, Google, and Amazon, and enterprise identity providers via SAML 2.0.

Per AWS Best Practices, proximity to your end users, regulatory compliance, data residency constraints, and cost are all factors you have to consider when choosing the most suitable AWS Region.

"Security **IN** the Cloud" refers to the Customer's responsibility in the Shared Responsibility Model. Customers are responsible for items such as building application schema, monitoring server and application performance, configuring security groups and network ACLs, and encrypting their data.

"Security **OF** the Cloud" refers to the AWS' responsibility in the Shared Responsibility Model. AWS is responsible for items such as the physical security of the DC (data center), creating hypervisors, replacement of old disk drives, and patch management of the infrastructure.
NOTE:
For "Patch Management", AWS is responsible for patching the underlying hosts and fixing flaws within the infrastructure, but customers are responsible for patching their guest OS and applications.

When a new IAM user is created, that user has NO access to any AWS service. This is called a non-explicit deny. For that user, access must be explicitly allowed via IAM permissions.

The **AWS Pricing Calculator** helps you estimate your monthly AWS bill more efficiently. The calculator can be used to determine your best and worst case scenarios and identify areas of development to reduce your monthly costs. The AWS Pricing Calculator is continuously updated with the latest pricing for all AWS services in all Regions. The AWS Pricing Calculator is available at: https://calculator.aws/

AWS Budgets gives you the ability to set custom budgets that alert you when your costs or usage exceed (or are forecasted to exceed) your budgeted amount. You can also set up AWS Budgets to alert you when your reservation utilization drops below the threshold you define.

AWS Cost & Usage Report does not estimate costs. The AWS Cost & Usage Report enables customers to access detailed information related to their AWS costs and usage. This information can help them analyze their cost drivers and usage trends.

AWS Cost Explore is used to explore and analyze your historical spend and usage. AWS Cost Explorer allows you to have visibility into your consumption patterns, such as, mapping the most commonly used services, and identifying unexpected anomalies or expenses.

AWS Cost Explorer can also be used to estimate AWS services costs, but it calculates these estimates based on your previous AWS consumption (meaning AWS Cost Explorer is suitable for **existing projects only**). In the above scenario, AWS Pricing Calculator is the right choice because it can be used to estimate the costs of **both existing and new projects** (in our case, it is a new project).

AWS Pricing Calculator enables you to estimate the monthly cost of AWS services for your use case based on your expected usage (not based on previous consumption as is the case with AWS Cost Explorer). For example, if you expect to use 500 GB of S3 Standard storage, you can simply enter this value in the appropriate field and the calculator provides an estimate of your monthly bill.

Additional information:

AWS Cost Explorer Forecasting provides an estimate of what your AWS bill will be, based on your past usage. AWS Cost Explorer segments your historical data based on distinct charge types (e.g., on-demand usage, reserved instance usage, and more) and uses a combination of machine learning and rules-based models to predict spend across all of those charge types individually.

A monolithic application is designed to be self-contained; components of the application are interconnected and interdependent rather than loosely coupled as is the case with Microservices applications.

With monolithic architectures, all processes are **tightly-coupled** and run as a single service. This means that if one process of the application experiences a spike in demand, the entire architecture must be scaled. Adding or improving a monolithic application's features becomes more complex as the code base grows. This complexity limits experimentation and makes it difficult to implement new ideas. Monolithic architectures add risk for application availability because many dependent and tightly coupled processes increase the impact of a single process failure.

With a microservices architecture, an application is built as **loosely-coupled** components that run each application process as a service. These services communicate via a well-defined interface using lightweight APIs. Services are built for business capabilities and each service performs a single function. Because they are independently run, each service can be updated, deployed, and scaled to meet demand for specific functions of an application. Microservices architectures make applications easier to scale and faster to develop, enabling innovation and accelerating time-to-market for new features.

Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. Amazon SQS offers a reliable, highly-scalable hosted queue for storing messages as they travel between applications or microservices. It moves data between distributed application components and helps you decouple these components.

AWS-managed databases are a database as a service offering from AWS where AWS manages the underlying hardware, storage, networking, backups, and patching. Users of AWS-managed databases simply connect to the database endpoint, and do not have to concern themselves with any aspects of managing the database. Examples of AWS-managed databases include: Amazon RDS (Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database, and Microsoft SQL Server), Amazon Neptune, Amazon DocumentDB, Amazon Redshift, and Amazon DynamoDB.

Amazon Neptune is a fully-managed **graph database service** that makes it easy to build and run applications that work with highly connected datasets, such as social networking, recommendation engines, and knowledge graphs. Amazon Neptune is fully managed and handles the time-consuming tasks such as provisioning, patching, backup, recovery, failure detection and repair.

Amazon RDS for MySQL is a managed service that makes it easy to set up, operate, and scale a **MySQL database** in the cloud. Amazon RDS for MySQL frees you up to focus on application development by managing time-consuming database administration tasks including backups, software patching, monitoring, scaling and replication.

Amazon EC2 Auto Scaling offers the following benefits:
1- Better fault tolerance. Amazon EC2 Auto Scaling can detect when an instance is unhealthy, terminate it, and launch an instance to replace it. Also, Amazon EC2 Auto Scaling enables you to take advantage of the safety and reliability of geographic redundancy by spanning Auto Scaling groups across multiple Availability Zones within a

Region. When one Availability Zone becomes unhealthy or unavailable, Auto Scaling launches new instances in an unaffected Availability Zone. When the unhealthy Availability Zone returns to a healthy state, Auto Scaling automatically redistributes the application instances evenly across all of the designated Availability Zones.
2- Better availability. Amazon EC2 Auto Scaling helps ensure that your application always has the right amount of capacity to handle the current traffic demand.
3- Better cost management. Amazon EC2 Auto Scaling can dynamically increase and decrease capacity as needed. Because you pay for the EC2 instances you use, you save money by launching instances when they are needed and terminating them when they aren't.

AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS. Developers simply upload their application, and Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, auto-scaling, and application health monitoring.

Amazon Elastic File System (Amazon EFS) provides a fully managed **NFS file system** for use with AWS Cloud services and on-premises resources.

Amazon EFS supports the latest version of the Network File System (NFS) protocol, so the applications and tools that you use today work seamlessly with Amazon EFS. Multiple compute instances, including Amazon EC2, Amazon ECS, and AWS Lambda, can access an Amazon EFS file system at the same time, providing a common data source for workloads and applications running on more than one compute instance or server.

With Amazon EFS, storage capacity is elastic, growing and shrinking automatically as you add and remove files, so your applications have the storage they need, when they need it.

Amazon Route 53 provides highly available and scalable Domain Name System (DNS), domain name registration, and health-checking web services. It is designed to give developers and businesses an extremely reliable and cost effective way to route end users to Internet applications by translating names like example.com into the numeric IP addresses, such as 192.0.2.1, that computers use to connect to each other. Route 53 also offers health checks to monitor the health and performance of your application as well as your web servers and other resources. Route 53 can be configured to route traffic only to the healthy endpoints to achieve greater levels of fault tolerance in your applications.

Note: The Elastic Load Balancing service also performs health checks on Amazon EC2 instances and distribute traffic only to the healthy ones.

In AWS, each Region has multiple, isolated locations known as Availability Zones. Availability Zones consist of one or more discrete data centers, each with redundant power, networking, and connectivity, housed in separate facilities.

Edge locations may or may not exist within a region. They are located in most

major cities around the world. Edge locations are specifically used by CloudFront (CDN) to distribute content to global users with low latency.

AWS OpsWorks is a configuration management service that provides managed instances of Chef and Puppet. Chef and Puppet are automation platforms that allow you to use code to automate the configurations of your servers. OpsWorks lets you use Chef and Puppet to automate how servers are configured, deployed, and managed across your Amazon EC2 instances or on-premises compute environments.

A Spot Instance is an unused EC2 instance that is available for less than the On-Demand price. Because Spot Instances enable customers to request unused EC2 instances at steep discounts, customers can lower their Amazon EC2 costs significantly. Spot Instances run whenever capacity is available, and the maximum price per hour for the request exceeds the Spot price. The risk with Spot instances is that a running instance can be interrupted due to changes in demand and pricing for a specific class of Spot instances, as there is no guarantee of availability at any time. Spot Instances are well-suited for data analysis, batch jobs, background processing, and optional tasks, as well as for workloads that are not time critical.

If you suspect that your account has been compromised, or if you have received a notification from AWS that the account has been compromised, perform the following tasks:
1- Change your AWS root account password and the passwords of any IAM users.
2- Delete or rotate all root and AWS Identity and Access Management (IAM) access keys.
3- Delete any potentially compromised IAM users.
4- Delete any resources on your account you didn't create, such as EC2 instances and AMIs, EBS volumes and snapshots, and IAM users.
5- Respond to any notifications you received from AWS Support through the AWS Support Center.

Customers with AWS Business or Enterprise support plans can open a "Production System Down" support case. The response time for this type of support case is one hour.
Similarly, the response time for the "Business-critical system down" support case is 15 minutes. But, AWS customers must have an Enterprise support plan to be able to open this support case.

AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS. Developers simply upload their application, and Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, auto-scaling, and application health monitoring.

Elastic Beanstalk makes it easy for developers to quickly deploy and manage applications in the AWS Cloud. Developers simply upload their application code, and Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, auto-scaling, and application health monitoring. AWS Elastic Beanstalk uses proven AWS features and services, such as Amazon EC2, Amazon RDS, Elastic Load Balancing, Auto Scaling, Amazon S3, and Amazon SNS, to create an environment that runs your application. At the same time, you retain full control over the AWS resources powering your application and can access the underlying resources at any time.

AWS CodeDeploy does not create resources for you like AWS Elastic Beanstalk. AWS CodeDeploy is used to **deploy** application code to a variety of compute services such as Amazon EC2, AWS Fargate, AWS Lambda, and your on-premises servers.

AWS CodeCommit vs. AWS CodeBuild vs. AWS CodeDeploy vs. AWS CodePipeline:
- AWS CodeCommit is used to **store and version** source code.
- AWS CodeBuild is used to **compile and test** source code, helping you find and fix bugs early in the development process when they are easy to fix.
- AWS CodeDeploy is used to **deploy** application code to a variety of compute services such as Amazon EC2, AWS Fargate, AWS Lambda, and your on-premises servers.
- AWS CodePipeline is the glue that builds these steps together. AWS CodePipeline enables you to **automate all phases of your release process**, from committing the code into AWS CodeCommit all the way to deploying it with AWS CodeDeploy. You can also integrate your own custom tools into any stage of the release process to form an end-to-end continuous delivery solution. This enables you to deliver new features and updates rapidly and reliably.

In relation to Amazon RDS databases:

AWS is responsible for:
1- Managing the underlying infrastructure and foundation services.
2- Managing the operating system.
3- Database setup.
4- Patching and backups.

The customer is still responsible for:
1- Protecting the data stored in databases (through encryption and IAM access control).
2- Managing the database settings that are specific to the application.
3- Building the relational schema.
4- Network traffic protection.

Amazon EBS pricing has two factors:
1- Volumes: Volume storage for all EBS volume types is charged by the amount of GB you provision per month, until you release the storage.
2- Snapshots: Snapshot storage is based on the amount of space your data consumes in Amazon S3. Because Amazon EBS does not save empty blocks, it is likely that the snapshot size will be considerably less than your volume size. Copying EBS snapshots is charged based on the volume of data transferred across regions. For the first snapshot of a volume, Amazon EBS saves a full copy of your data to Amazon S3. For each incremental snapshot, only the changed part of your Amazon EBS volume is saved. After the snapshot is copied, standard EBS snapshot charges apply for storage in the destination region.

The factors that have the greatest impact on cost include: Compute, Storage and Data Transfer Out. Their pricing differs according to the service you use.

The benefits of using AWS CloudFormation include:
1- CloudFormation allows you to model your entire infrastructure in a text file. This template becomes the single source of truth for your infrastructure. This helps you to standardize infrastructure components used across your organization, enabling configuration compliance and faster troubleshooting.
2- AWS CloudFormation provisions your resources in a safe, repeatable manner, allowing you to build and rebuild your infrastructure and applications, without having to perform manual actions or write custom scripts. CloudFormation takes care of determining the right operations to perform when managing your stack, and rolls back changes automatically if errors are detected.
3- Codifying your infrastructure allows you to treat your infrastructure as just code. You can author it with any code editor, check it into a version control system, and review the files with team members before deploying into production.
4- CloudFormation allows you to model and provision, in an automated and secure manner, all the resources needed for your applications across all regions and accounts.

AWS X-Ray helps you identify performance bottlenecks. X-Ray's service maps let you see relationships between services and resources in your application in real time. You can easily detect where high latencies are occurring, visualize node and edge latency distribution for services, and then drill down into the specific services and paths impacting application performance.

Amazon Detective is a security service that allows customers to analyze, investigate, and quickly identify the root cause of potential **security** issues or suspicious activities. Amazon Detective cannot detect **performance** issues.

AWS Security Hub aggregates, organizes, and prioritizes security alerts and findings from multiple AWS security services, such as Amazon GuardDuty, Amazon Inspector, and Amazon Macie, and supported third-party partners to help you analyze your security trends and identify the **highest priority** security issues.

AWS Shield is a managed Distributed Denial of Service (DDoS) protection service that safeguards applications running on AWS. AWS Shield Standard is automatically enabled to all AWS customers and provides always-on detection and automatic inline mitigations that minimize application downtime and latency.

AWS Cost Governance Best Practices:
1- Resource controls (policy-based and automated) govern who can deploy resources and the process for identifying, monitoring, and categorizing these new resources. These controls can use tools such as AWS Service Catalog, AWS Identity and Access Management (IAM) roles and permissions, and AWS Organizations, as well as third-party tools such as ServiceNow.
2- Cost allocation applies to teams using resources, shifting the emphasis from the IT-as-cost-center

mentality to one of shared responsibility.
3- Budgeting processes include reviewing budgets and realized costs, and then acting on them.
4- Architecture optimization focuses on the need to continually refine workloads to be more cost-conscious to create better architected systems.
5- Tagging and tagging enforcement ensure cost tracking and visibility across organization lines.

Having effective processes in place ensures that the right information and controls are available to the right people. This reinforces channels of communication for cost-related inquiries, which strengthens your cost-conscious culture.

Tags are key-value pairs that allow you to organize your AWS resources into groups. Implementing a tagging strategy will help you track usage and spending across different departments, applications, or Development/Production environments. For example, if you tag resources with an application name, you can track the total cost of a single application that runs on those resources.
You can use tags to:
1- Visualize information about tagged resources in one place.
2- View billing information using Cost Explorer and the AWS Cost and Usage report.
3- Create separate invoices for each project or work environment.
It is recommended that you use logical groupings of your resources that make sense for your infrastructure or business. For example, you could organize your resources by:
- Project
- Environment (Development - Testing - Production)
- Cost center
- Application
- Department

Server-based services include: Amazon EC2, Amazon RDS, Amazon Redshift and Amazon EMR.
Serverless services include: AWS Lambda, AWS Fargate, Amazon SNS, Amazon SQS and Amazon DynamoDB.

When you purchase a Reserved Instance, you can choose between a Standard or Convertible offering class.
Standard RIs: These provide the most significant discount (up to 72% off On-Demand) and are best suited for steady-state usage.
Convertible RIs: These provide a discount (up to 54% off On-Demand) and the capability to change the attributes of the RI as long as the exchange results in the creation of Reserved Instances of equal or greater value. Like Standard RIs, Convertible RIs are best suited for steady-state usage.

Services like **AWS Config, Amazon Inspector, and AWS Trusted Advisor** continually monitor for compliance or vulnerabilities in your AWS environment which gives you a clear overview of which resources are in compliance, and which are not. With AWS Config rules you can also see if a component was out of compliance

even for a brief period of time in the past, making both point-in-time and period-in-time audits very effective.

AWS OpsWorks is a configuration management service that provides managed instances of Chef and Puppet. Chef and Puppet are automation platforms that allow you to use code to automate the configurations of your servers. OpsWorks lets you use Chef and Puppet to automate how servers are configured, deployed, and managed across your Amazon EC2 instances or on-premises compute environments.

The AWS Cloud includes many design patterns and architectural options that you can apply to a wide variety of use cases. Some key design principles of the AWS Cloud include scalability, disposable resources, automation, loose coupling, managed services instead of servers, and flexible data storage options.

AWS CodeBuild is a fully managed continuous integration service that compiles source code, runs tests, and produces software packages that are ready to deploy.

AWS CodeCommit vs. AWS CodeBuild vs. AWS CodeDeploy vs. AWS CodePipeline:
- AWS CodeCommit is used to **store and version** source code.
- AWS CodeBuild is used to **compile and test** source code, helping you find and fix bugs early in the development process when they are easy to fix.
- AWS CodeDeploy is used to **deploy** application code to a variety of compute services such as Amazon EC2, AWS Fargate, AWS Lambda, and your on-premises servers.
- AWS CodePipeline is the glue that builds these steps together. AWS CodePipeline enables you to **automate all phases of your release process**, from committing the code into AWS CodeCommit all the way to deploying it with AWS CodeDeploy. You can also integrate your own custom tools into any stage of the release process to form an end-to-end continuous delivery solution. This enables you to deliver new features and updates rapidly and reliably.

Amazon Route 53 can be used for:
• Registering domain names
• DNS configuration and management
• Configuring health checks to route traffic only to healthy endpoints
• Managing global application traffic (cross-regions) through a variety of routing types.

Amazon Route53 allows for registration of new domain names in AWS. Amazon Route 53 is a global service that provides a highly available and scalable Domain Name System (DNS) in the Cloud. It is designed to give developers and businesses an extremely reliable and cost effective way to route end users to Internet applications by translating names like www.example.com into the numeric IP addresses like 192.0.2.1 that computers use to connect to each other.

Amazon Route 53 also offers health checks to monitor the health and performance of your application as well as your web servers and other resources. Route 53 can be configured to route traffic only to the healthy endpoints to achieve greater levels of fault tolerance in your applications.

Amazon Route 53 provides many routing types to help AWS Customers improve their application's performance for a global audience. For example, Amazon Route 53

latency-based policy routes user requests to the closest AWS Region, which reduces latency and improves application performance.

Amazon Route 53 also simplifies the hybrid cloud by providing recursive DNS for your Amazon VPC and on-premises networks over AWS Direct Connect or AWS VPN.

There are three Cloud Computing Deployment Models:

1- Cloud:

A cloud-based application is fully deployed in the cloud and all parts of the application run in the cloud. This Cloud Computing deployment model eliminates the need to run and maintain physical data centers.

2- Hybrid:

A hybrid deployment is a way to connect infrastructure and applications between cloud-based resources and existing resources that are not located in the cloud (On-premises data centers).

3- On-premises:

Deploying resources on-premises, using virtualization and resource management tools, is sometimes called “private cloud”. On-premises deployment does not provide many of the benefits of cloud computing but is sometimes sought for its ability to provide dedicated resources.

To protect your AWS infrastructure you should:

1- Change the email address and the password of the root user account
2- Enable MFA on the root user account
4- Rotate (change) all access keys for all accounts
3- Change the user name and password of all IAM users
5- Enable MFA on all IAM user accounts

AWS Organizations has five main benefits:

1) Centrally manage access polices across multiple AWS accounts.
2) Automate AWS account creation and management.
3) Control access to AWS services.
4) Consolidate billing across multiple AWS accounts.
5) Configure AWS services across multiple accounts.

Amazon ElastiCache improves the performance of web applications by allowing you to retrieve information from a fast, managed, in-memory data store, instead of relying entirely on slower disk-based databases. Querying a database is always slower and more expensive than locating a copy of that data in a cache. By caching (storing) common database query results, you can quickly retrieve the data multiple times without having to re-execute the query.

Security Groups and Network Access Control Lists (Network ACLs) are the two parts of the VPC Security Layer. Security Groups are a firewall at the instance layer, and Network ACLs are a firewall at the subnet layer.

Objects stored in Glacier take time to retrieve. You can pay for expedited retrieval, which will take several minutes or wait several hours for normal retrieval.

Amazon EBS provides durable, block-level storage volumes that you can attach to a running EC2 instance. You can use Amazon EBS as a primary storage device for data that requires frequent and granular updates. Amazon EBS is the recommended storage option when you run a database on an EC2 instance.

Amazon CloudFront is a fast Content Delivery Network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency and high transfer speeds.
CloudFront is the best solution to reduce latency if you have users from different places around the world.
Storing media assets in a region closer to the end-users can help reduce latency for those users. This is because these assets will travel a shorter distance over the network.

The AWS free security resources include the AWS Security Blog, Whitepapers, AWS Developer Forums, Articles and Tutorials, Training, Security Bulletins, Compliance Resources and Testimonials.

AWS Global Accelerator uses the AWS global network to optimize the path from your users to your applications, improving the performance of your traffic by as much as 60%. AWS Global Accelerator continually monitors the health of your application endpoints and redirects traffic to healthy endpoints in less than 30 seconds.

CPU utilization is the percentage of allocated EC2 compute units that are currently in use on the instance. This metric measures the percentage of allocated CPU cycles that are being utilized on an instance. The CPU Utilization CloudWatch metric shows CPU usage per instance and not CPU usage per core