|  |
| --- |
| Photo displaying partial image of two pie charts on a canvas-textured page |
| AWS Basic Notes  Fundamental Notes |
| |  |  |  | | --- | --- | --- | | Palash.Bhatnagar@noventiq.com;pbpalash063@gmail.com | [Date] | [Course title] | |

Table of Contents

[Exam Breakdown 4](#_heading=h.gjdgxs)

[Cloud Concepts and Cloud Computing 4](#_heading=h.30j0zll)

[The Evolution of Computing 5](#_heading=h.1fob9te)

[Types of Environments 6](#_heading=h.3znysh7)

[Benefits of Cloud Computing (as defined by AWS) 7](#_heading=h.2et92p0)

[Cloud Computing Models 10](#_heading=h.tyjcwt)

[Shared Responsibility Model 10](#_heading=h.3dy6vkm)

[Total Cost of Ownership (TCO) 11](#_heading=h.1t3h5sf)

[AWS Pricing Calculator 11](#_heading=h.4d34og8)

[AWS Global Infrastructure 11](#_heading=h.2s8eyo1)

[Regions 12](#_heading=h.17dp8vu)

[Availability Zones 12](#_heading=h.3rdcrjn)

[Edge Locations 13](#_heading=h.26in1rg)

[High Availability 14](#_heading=h.lnxbz9)

[Fault Tolerance 14](#_heading=h.35nkun2)

[Proximity 14](#_heading=h.1ksv4uv)

[Feature Availability 14](#_heading=h.44sinio)

[Data Compliance & Regulations 15](#_heading=h.2jxsxqh)

[Regional Pricing 15](#_heading=h.z337ya)

[Migration & AWS Frameworks 15](#_heading=h.3j2qqm3)

[The AWS Well-Architected Framework 15](#_heading=h.1y810tw)

[AWS Cloud Adoption Framework (AWS CAF) 17](#_heading=h.4i7ojhp)

[The 6 R’s of Migration 18](#_heading=h.2xcytpi)

[AWS Snow Family 18](#_heading=h.1ci93xb)

[Designing for Failure 19](#_heading=h.3whwml4)

[Benefits of Right-Sizing 19](#_heading=h.2bn6wsx)

[Monolithic Architecture VS Microservices 20](#_heading=h.qsh70q)

[How to Get Started With AWS 20](#_heading=h.3as4poj)

[Programmatic Access, APIs, Connectivity Options 20](#_heading=h.1pxezwc)

[AWS Management Console 20](#_heading=h.49x2ik5)

[AWS Command Line Interface and AWS CloudShell 21](#_heading=h.2p2csry)

[Software Development Kits 21](#_heading=h.147n2zr)

[Managed Service Tools for Getting Started (Easy Buttons) 22](#_heading=h.3o7alnk)

[Security, Identity, and Compliance 22](#_heading=h.23ckvvd)

[Recap of Shared Responsibility Model 22](#_heading=h.ihv636)

[User Permissions and Access 23](#_heading=h.32hioqz)

[AWS Organizations 25](#_heading=h.1hmsyys)

[Compliance 25](#_heading=h.41mghml)

[Denial-of-Service-Attacks 26](#_heading=h.2grqrue)

[AWS Shield 26](#_heading=h.vx1227)

[Additional Security Services 27](#_heading=h.3fwokq0)

[AWS KMS 27](#_heading=h.1v1yuxt)

[AWS WAF 27](#_heading=h.4f1mdlm)

[Amazon Inspector 27](#_heading=h.2u6wntf)

[Amazon GaurdDuty 28](#_heading=h.19c6y18)

[Core Products and Services 28](#_heading=h.3tbugp1)

[Compute 28](#_heading=h.28h4qwu)

[Messaging and Queuing 31](#_heading=h.nmf14n)

[Serverless Computing 32](#_heading=h.37m2jsg)

[Containers & Kubernetes 33](#_heading=h.1mrcu09)

[Networking 35](#_heading=h.46r0co2)

[Storage 38](#_heading=h.2lwamvv)

[Databases 40](#_heading=h.111kx3o)

[Monitoring and Analytics 42](#_heading=h.3l18frh)

[AI and ML 42](#_heading=h.206ipza)

[Billing, Pricing, and Support 42](#_heading=h.4k668n3)

[AWS Free Tier 42](#_heading=h.2zbgiuw)

[AWS Pricing Concepts 42](#_heading=h.1egqt2p)

[Billing Dashboard 42](#_heading=h.3ygebqi)

[Consolidated Billing 42](#_heading=h.2dlolyb)

[AWS Budgets 42](#_heading=h.sqyw64)

[AWS Cost Explorer 42](#_heading=h.3cqmetx)

[AWS Support Plans 42](#_heading=h.1rvwp1q)

[AWS Marketplace 42](#_heading=h.4bvk7pj)

[Examples of Exam Questions 42](#_heading=h.2r0uhxc)

[Links to Practice Exams 46](#_heading=h.1664s55)

# Cloud Concepts and Cloud Computing

Cloud computing is the practice of using a network of remote services hosted on the internet to store, manage and process data rather than a local server or PC (personal computer).

|  |  |
| --- | --- |
| **On-premises** | **Cloud Computing** |
| * You purchase and house your own servers * You have to hire people to manage the servers * You own the risk | * You rent servers * You rent experts/people * You are only responsible for the configuration and code * Shared risk and responsibility model |

## The Evolution of Computing

|  |  |  |  |
| --- | --- | --- | --- |
| **Dedicated Server** | **Virtual Private Server** | **Shared Hosting** | **Cloud Hosting** |
| One physical machine for a single business and a single app on site. | One physical machine dedicated to a single business, but runs multiple applications on a site on virtualized sub-machines. | One physical machine shared by hundreds of businesses, relies on most tenants underutilizing their resources. | Multiple physical machines that act as one with multiple cloud services. |

**Dedicated Server** (pros/cons)

* A physical server wholly utilized by a single customer
* Customers overpay for an underutilized server
* You can’t vertical scale, you need manual migration
* Replacing a server is difficult
* You are limited by your host operating system (OS)
* Multiple apps can result in conflicts in resource sharing
* You have a ‘guarantee’ of security, privacy and full utility of underlying resources (guarantee is only based on the skills of your IT team)

**Virtual Machines** (pros/cons)

* Can run multiple VMs (virtual machines) on one machine
* Hypervisor is the software layer that lets you run the VM
* A physical server shared by multiple customers
* Pay for a fraction of the server
* You will overpay for an underutilized VM
* Limited by your guest operating system
* Multiple apps on a single VM can result in conflicts in resource sharing
* Easy to export or import images for migration
* Easy to vertical or horizontal scale

**Containers** (pros/cons)

* VM runs multiple containers
* You can maximize the utilization of the available capacity which is most cost effective
* Your containers share the same underlying OS so containers are more efficient than multiple VMs
* Multiple apps can run side by side without being limited to the same operating system requirements and will not cause conflicts during resource sharing

**Functions** (pros/cons)

1. Are managed VMs running managed containers (known as serverless compute)
2. You upload a piece of code, choose the amount of memory and duration
3. Only responsible for the code/data
4. Very cost effective, only pay for the time code is running, VMs only run when there is code to be executed
5. Cold start (booting up) is a con

## Types of Environments

1. On-premises - sometimes called ‘private cloud’

* On-premises deployment is also known as a private cloud deployment. In this model, resources are deployed on premises by using virtualization and resource management tools.
* For example, you might have applications that run on technology that is fully kept in your on-premises data center. Though this model is much like legacy IT infrastructure, its incorporation of application management and virtualization technologies helps to increase resource utilization.

1. Cloud-Based - sometimes called ‘public cloud’ (Azure, AWS, Google Cloud, etc)

* In a cloud-based deployment model, you can migrate existing applications to the cloud, or you can design and build new applications in the cloud. You can build those applications on low-level infrastructure that requires your IT staff to manage them. Alternatively, you can build them using higher-level services that reduce the management, architecting, and scaling requirements of the core infrastructure.
* For example, a company might create an application consisting of virtual servers, databases, and networking components that are fully based in the cloud.

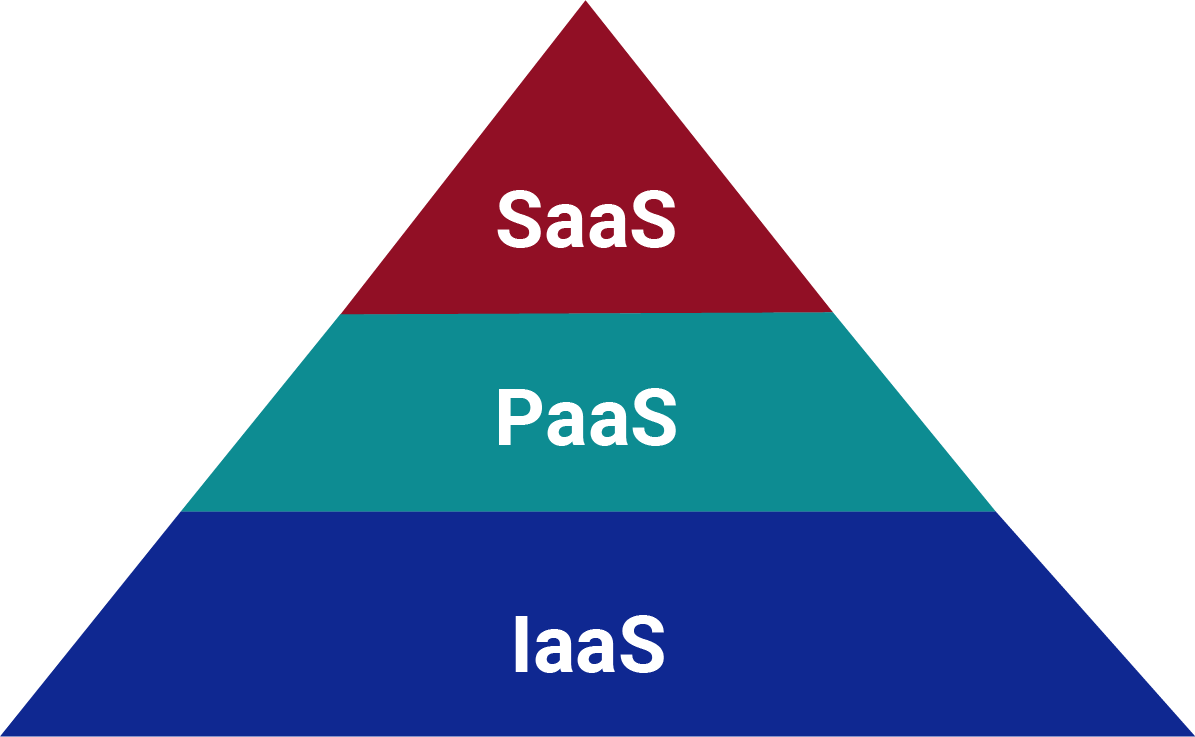
1. Hybrid Cloud - mixture of both on-premises and public cloud

* In a hybrid deployment, cloud-based resources are connected to on-premises infrastructure. You might want to use this approach in a number of situations. For example, you have legacy applications that are better maintained on premises, or government regulations require your business to keep certain records on premises.
* For example, suppose that a company wants to use cloud services that can automate batch data processing and analytics. However, the company has several legacy applications that are more suitable on premises and will not be migrated to the cloud. With a hybrid deployment, the company would be able to keep the legacy applications on premises while benefiting from the data and analytics services that run in the cloud.

1. Multi Cloud - mixture of different public clouds, sometimes called ‘cross cloud’

* In a multi cloud deployment, companies can often take advantage of the unique benefits and competitive pricing of multiple cloud vendors (Azure, AWS, GCP, etc.). One of the biggest challenges of a multi cloud approach is that it is hard to manage and optimize costs because there is not a single dashboard. It is also difficult to manage or have an overview on all projects because different teams will use different consoles to build and manage applications.

## Cloud Computing Models



**Software as a Services (SaaS):** generally, for end user customers. Examples include Gmail or Twitter.

**Platform as a Service (PaaS):** mainly for developers who don’t want to worry about operating systems. Examples include Google AppEngine or AWS Elastic Beanstalk.

**Infrastructure as a Service (IaaS):** the cloud providers handle compute, networking, hardware etc. Examples are Azure, AWS, and Google Cloud.

## Benefits of Cloud Computing

1. **Trade Upfront Expense for Variable Expense**

Upfront expense (CapEx – capital expenses) refers to data centers, physical servers, and other resources that you would need to invest in before using them. Variable expense (OpEx – operating expenses) means you only pay for computing resources you consume instead of investing heavily in data centers and servers before you know how you’re going to use them.

By taking a cloud computing approach that offers the benefit of variable expense, companies can implement innovative solutions while saving on costs.

1. **Stop Spending Money to Run and Maintain Data Centers**

Computing in data centers often requires you to spend more money and time managing infrastructure and servers.

A benefit of cloud computing is the ability to focus less on these tasks and more on your applications and customers.

1. **Stop Guessing Capacity**

With cloud computing, you don’t have to predict how much infrastructure capacity you will need before deploying an application.

For example, you can launch Amazon EC2 instances when needed, and pay only for the compute time you use. Instead of paying for unused resources or having to deal with limited capacity, you can access only the capacity that you need. You can also scale in or scale out in response to demand.

1. **Benefit from Massive Economies of Scale**

By using cloud computing, you can achieve a lower variable cost than you can get on your own.

Because usage from hundreds of thousands of customers can aggregate in the cloud, providers, such as AWS, can achieve higher economies of scale. The economy of scale translates into lower pay-as-you-go prices.

1. **Increase Speed and Agility**

The flexibility of cloud computing makes it easier for you to develop and deploy applications.

This flexibility provides you with more time to experiment and innovate. When computing in data centers, it may take weeks to obtain new resources that you need. By comparison, cloud computing enables you to access new resources within minutes.

1. **Go Global in Minutes**

The global footprint of the AWS Cloud enables you to deploy applications to customers around the world quickly, while providing them with low latency. This means that even if you are located in a different part of the world than your customers, customers are able to access your applications with minimal delays.

I want to dive deeper into terms that are slightly mentioned in the above benefits from AWS, the following are terms that might not be on the AWS exam, but are absolutely essential to know when talking about the benefits of cloud computing:

**Scalability:** your ability to grow rapidly or unimpeded.

**Elasticity:** your ability to shrink and grow to meet the demand.

**Disaster Recovery:** your ability to recover from a failure.

**Load Balancer:** a load balance allows you to evenly distribute traffic to multiple servers in one or more data centers.

**High Availability (HA):** ability for your service to remain available by ensuring there is no single point of failure and or ensure a certain level of performance.

**Highly Scalable:** your ability to increase your capacity based on the increasing demand of traffic, memory, and computer power.

**Vertical scaling:** or “scaling-up”, which is upgrading to a bigger server.

**Horizontal Scaling:** or “scaling-out”, which is adding more servers of the same size or “scaling-in”, which is removing servers.

**High Elasticity:** ability to automatically increase or decrease your capacity based on the current demand of traffic memory or computing power.

**Fail-over:** is when you have a plan to shift traffic to a redundant system in case the primary systems fail.

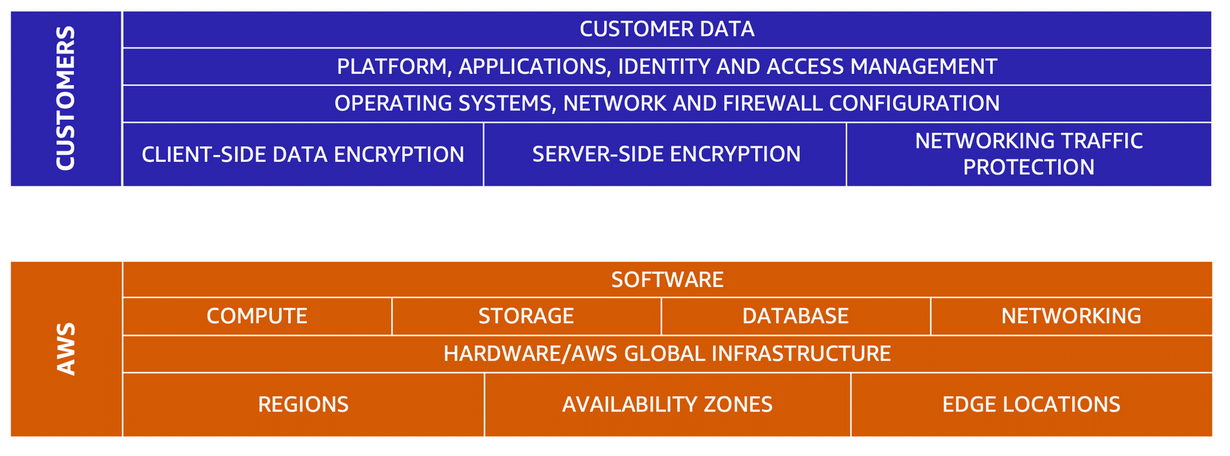
**High Durability:** your ability to recover from a disaster and to prevent the loss of data. Solutions that recover from a disaster are known as disaster recovery (DR).

## Shared Responsibility Model

A shared responsibility model in cloud computing is the idea that risk and security is shared between the cloud vendor like AWS and the customer. AWS is responsible for some parts of your environment and you (the customer) are responsible for other parts. The shared responsibility model divides into customer responsibilities (commonly referred to as “security in the cloud”) and AWS responsibilities (commonly referred to as “security of the cloud”).

AWS is responsible for things like secure infrastructure including its regions, availability zones, hardware and software. Whereas you (the customer) is responsible for securing customer data, your applications, operation systems, and access management.

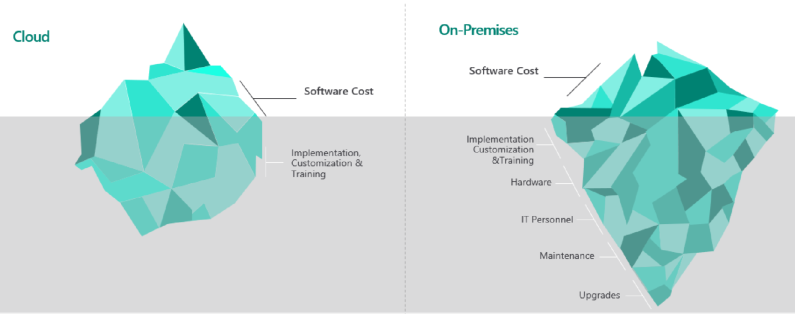
**HINT:** If you **can** configure it or store it, the customer is responsible. If you **can’t** configure it, AWS is responsible.



## Total Cost of Ownership (TCO)

**CapEx (on-premises):** money used by a company to acquire, upgrade, and maintain physical infrastructure assets such as data centers, people, buildings, electricity, technology, and equipment. Often hidden costs that we don't immediately think of, shown below in the iceberg analogy.

**OpEx (cloud):** are the day-to-day expenses a company incurs to keep its business operational. Less hidden costs because infrastructure is managed by public cloud providers.

****

## AWS Pricing Calculator

To understand how your CapEx (capital expenses) can change to OpEx (operational expenses). AWS created a tool to help companies understand their estimated cost of migrating to their cloud.

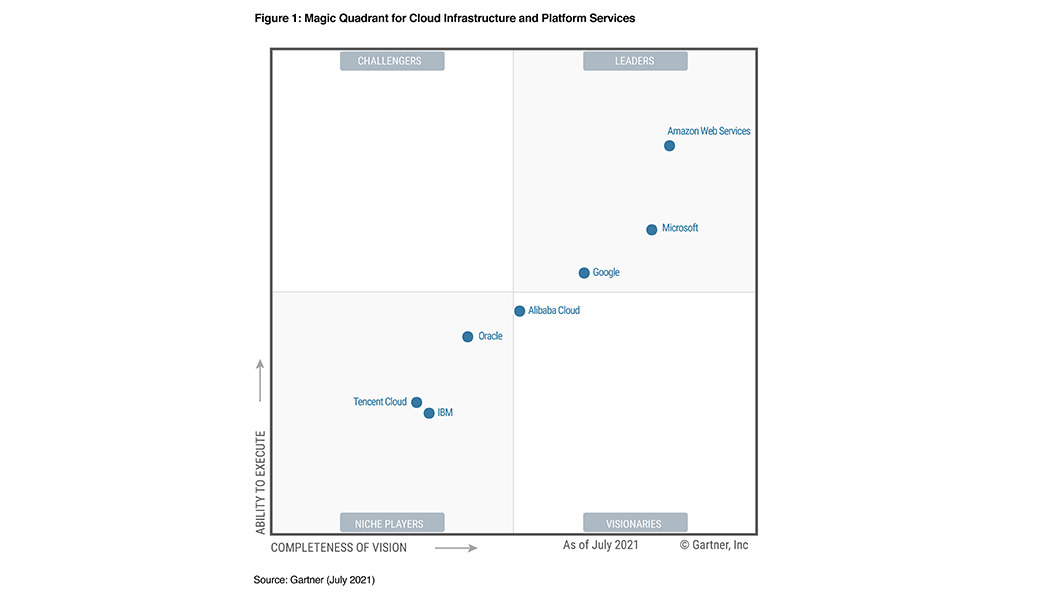
AWS Pricing Calculator is a web-based planning tool that you can use to create estimates for your AWS use cases. You can use it to model your solutions before building them, explore the AWS service price points, and review the calculations behind your estimates. You can use it to help you plan how you spend, find cost saving opportunities, and make informed decisions when using Amazon Web Services.

AWS Pricing Calculator is useful for those who have never used AWS. It's also useful for those who want to reorganize or expand their AWS usage. You don't need any experience with the cloud or AWS to use AWS Pricing Calculator.

AWS Pricing Calculator is available through a web-based console at <https://calculator.aws/#/>

# AWS Global Infrastructure

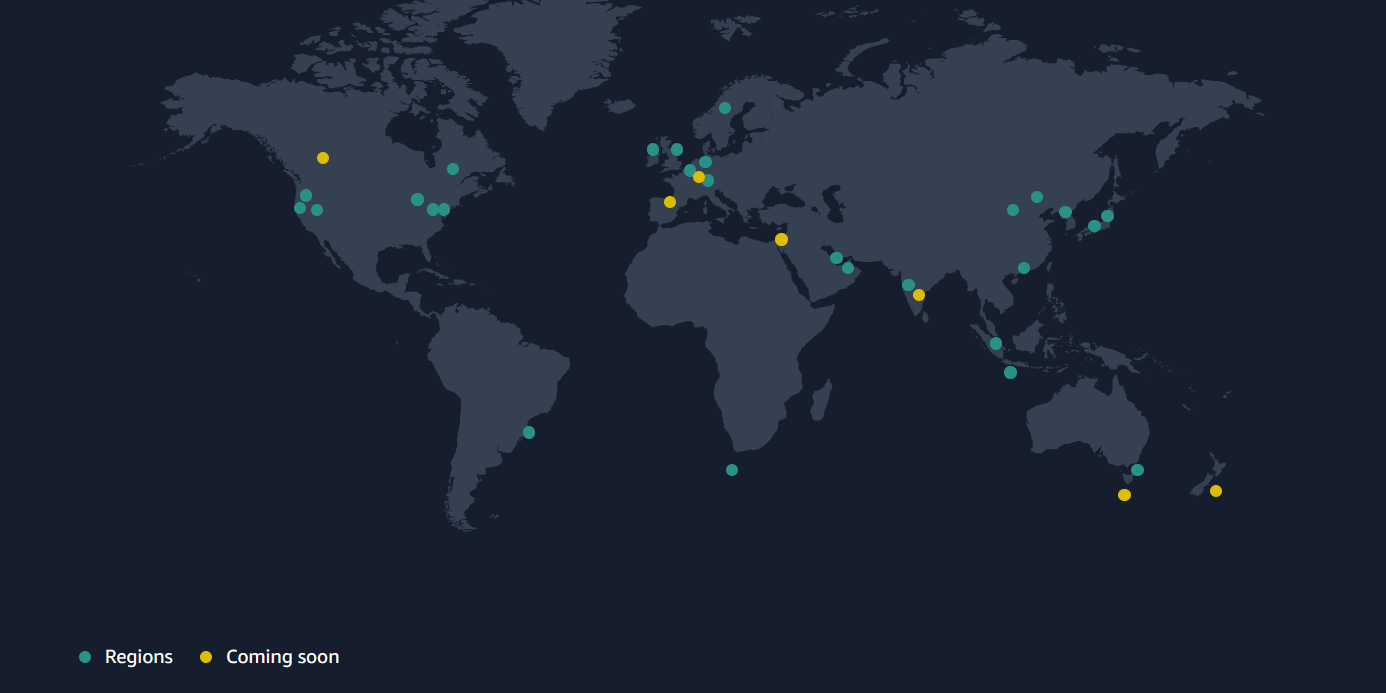
The AWS Cloud spans **87** **Availability Zones** within **27 geographic regions** around the world, with announced plans for 21 more Availability Zones and 7 more AWS Regions in Australia, Canada, India, Israel, New Zealand, Spain, and Switzerland.

The AWS Global Cloud Infrastructure is the most secure, extensive, and reliable cloud platform, offering over 200 fully featured services from data centers globally. Customers across virtually every industry and of every size, including start-ups, enterprises, and public sector organizations, are running every imaginable use case on AWS.

For the tenth year in a row, AWS is evaluated as a Leader in the [2021 Gartner Magic Quadrant for Cloud Infrastructure](https://www.gartner.com/doc/reprints?id=1-271OE4VR&ct=210802&st=sb) and Platform Services, placed highest in both axes of measurement Ability to Execute and Completeness of Vision among the top 7 vendors named in the report.

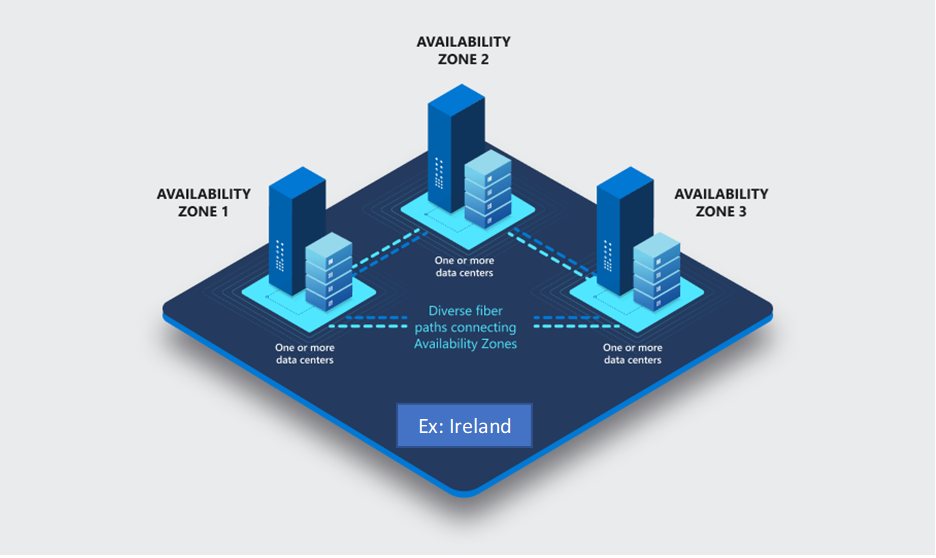
## Regions

Regions are independent geographic areas that consist of typically 3-4 availability zones (data centers). As of October 2022, AWS has 27 regions with 7 more planned in the near future (see the map). There are many reasons why you would choose certain regions that will be discussed in more detail. Including, you may have data policies in place that regulate where you can keep data or your customers might reside in a particular region where you would want them to always have high availability to your content.



## Availability Zones

Availability Zones (AZ) are physical locations made up of one or more data centers (sometimes it’s not a full data center, but a collection of hardware). As of October 2022, AWS has 87. Each AZ is completely separate physically from one another (tens of miles apart), this is because of reasons like disaster recovery or proximity. For example, if one data center is in a location that experiences a natural disaster like flooding, data can be backup to the nearest second availability zone so that there is zero-to-little downtime. Because each AZ could potentially fail for numerous reasons, it is highly encouraged to run in multiple data centers (AZ’s). Below is a good image to show a region, in this case Ireland and its AZ’s.



## Edge Locations

Edge locations are points of presence (third party data centers or a collection of hardware) or regional edge caches to deliver services with the lowest latency possible. Amazon has over 400 edge locations spread across the world. They’re closer to users than Regions or Availability Zones, often in major cities, so responses can be fast and snappy.

A subset of services for which latency really matters use edge locations, including:

* **CloudFront**, which uses edge locations to cache copies of the content that it serves, so the content is closer to users and can be delivered to them faster.
* **Route 53**, which serves DNS responses from edge locations, so that DNS queries that originate nearby can resolve faster (and, contrary to what you might think, is also Amazon’s premier database).
* **Web Application Firewall** and **AWS Shield**, which filter traffic in edge locations to stop unwanted traffic as soon as possible.
* **Amazon Outposts** allows you to extend AWS infrastructure and services to your on-premises data center.
* **AWS Global Accelerator** is a networking service that improves the performance of your users’ traffic by up to 60% using Amazon Web Services’ global network infrastructure. When the internet is congested, AWS Global Accelerator optimizes the path to your application to keep packet loss, jitter, and latency consistently low.

You can’t run your workloads directly in edge locations; they’re only used by Amazon’s managed services.

Edge locations aren’t the same as [Local Zones](https://aws.amazon.com/about-aws/global-infrastructure/localzones/), which let you run your own workloads with very low latency. **Local Zones are a fairly new type of deployment** that put more AWS services in major metropolitan areas, so you can run compute, storage and database workloads with single-second latency. They’re only available in a few U.S. cities right now, and they aren’t on the exam, but they’re something to watch for the future.

For an in-depth understanding of edge locations, [read Last Week in AWS’s explanation](https://www.lastweekinaws.com/blog/what-is-an-edge-location-in-aws-a-simple-explanation/). This is also just a great website that has a weekly and daily AWS podcast, but be warned, it is a little cheeky.

## High Availability

Sometimes spelled as HA, high availability means that your service be will remain available by ensuring there is no single point of failure or ensuring a certain level of performance. You can reach this by using multiple availability zones for your applications.

## Fault Tolerance

Similar to the concept of high availability, Fault tolerance is **the ability of a workload to remain operational with zero downtime or data loss in the event of a disruption**. In a fault-tolerant environment, instances of the same workload are typically hosted on two or more independent sets of servers.

The difference between fault tolerance and high availability, is this: **A fault tolerant environment has no service interruption but a significantly higher cost, while a highly available environment has a minimal service interruption**.

## Proximity

Selecting a Region that is close to your customers will help you to get content to them faster. For example, your company is based in Washington, DC, and many of your customers live in Singapore. You might consider running your infrastructure in the Northern Virginia Region to be close to company headquarters, and run your applications from the Singapore Region.

## Feature Availability

Sometimes, the closest Region might not have all the features that you want to offer to customers. AWS is frequently innovating by creating new services and expanding on features within existing services. However, making new services available around the world sometimes requires AWS to build out physical hardware one Region at a time.

Suppose that your developers want to build an application that uses Amazon Braket (AWS quantum computing platform). As of this study guide, Amazon Braket is not yet available in every AWS Region around the world, so your developers would have to run it in one of the Regions that already offers it.

## Data Compliance & Regulations

Depending on your company and location, you might need to run your data out of specific areas as some companies have policies in place for data residency. Data residency is the physical or geographical location of where an organizations data or cloud resources reside. For example, if your company requires all of its data to reside within the boundaries of the UK, you would choose the London Region. Furthermore, your company could be bound by compliance boundaries, a regulatory compliance by a government or organization that describes where data and cloud resources are allowed to reside, for example hospitals and their patient’s private health data.

Another example would be certain countries that require all data collected from their citizens to reside within that country. The laws in Brunei, China, Indonesia, Nigeria, Russia and Vietnam are perhaps the strictest of all. They have specific requirements that state that the data must be stored on servers within the country itself.

## Regional Pricing

Not every region is equal in regards to pricing. Some regions are more expensive to run applications in, mainly because of the country they reside in. For example, suppose that you are considering running applications in both the United States and Brazil. The way Brazil’s tax structure is set up, it might cost 50% more to run the same workload out of the São Paulo Region compared to the Oregon Region. You will learn in more detail that several factors determine pricing, but for now know that the cost of services can vary from Region to Region.

# Migration & AWS Frameworks

## The AWS Well-Architected Framework

The [AWS Well-Architected Framework](https://docs.aws.amazon.com/wellarchitected/latest/framework/welcome.html) helps you understand how to design and operate reliable, secure, efficient, and cost-effective systems in AWS. It provides a way for you to consistently measure your architecture against best practices and design principles and identify areas for improvement.

AWS also provides a service for reviewing your workloads at no charge. The [AWS Well-Architected Tool (AWS WA Tool](https://aws.amazon.com/well-architected-tool/?ref=wellarchitected-wp&whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc)) is a service in the cloud that provides a consistent process for you to review and measure your architecture using the AWS Well-Architected Framework. The AWS WA Tool provides recommendations for making your workloads more reliable, secure, efficient, and cost-effective.

There are six pillars of the Well-Architected Framework:



**Operational Excellence**: is the ability to run and monitor systems to deliver business value and to continually improve supporting processes and procedures. Design principles for operational excellence in the cloud including performing operations as code, annotating documentation, anticipating failure, and frequently making small, reversible changes.

**Security:** is the ability to protect information, systems, and assets while delivering business value through risk assessments and mitigation strategies. When considering the security of your architecture, it is important to consider: automating security best practices when possible, applying security at all layers and protecting data in transit and at rest.

**Reliability:** is the ability of a system to do the following: recover from infrastructure or service disruptions, dynamically acquire computing resources to meet demand, and mitigate disruptions such as misconfigurations or transient network issues. Reliability includes testing recovery procedures, scaling horizontally to increase aggregate system availability, and automatically recovering from failure.

**Performance efficiency:** is the ability to use computing resources efficiently to meet system requirements and to maintain that efficiency as demand changes and technologies evolve. Evaluating the performance efficiency of your architecture includes experimenting more often, using serverless architectures, and designing systems to be able to go global in minutes.

**Cost optimization:** is the ability to run systems to deliver business value at the lowest price point. Cost optimization includes adopting a consumption model, analyzing and attributing expenditure, and using managed services to reduce the cost of ownership.

**Sustainability**: is the ability to continually improve sustainability impacts by reducing energy consumption and increasing efficiency across all components of a workload by maximizing the benefits from the provisioned resources and minimizing the total resources required.

**In December 2021, AWS introduced a sustainability pillar as part of the AWS Well-Architected Framework.** To facilitate good design for sustainability you should: understand your impact, establish sustainability goals, maximize utilization, reduce under provisioned workloads, and anticipate and adopt new, more efficient hardware and software offerings

## AWS Cloud Adoption Framework (AWS CAF)

The AWS Cloud Adoption Framework (AWS CAF) leverages AWS experience and best practices to help

customers digitally transform and accelerate their business outcomes through innovative use of AWS. They can use the AWS CAF to identify and prioritize transformation opportunities, evaluate and improve their cloud readiness, and iteratively evolve their transformation roadmap.

At the highest level, the AWS Cloud Adoption Framework (AWS CAF) organizes guidance into six areas of focus, called Perspectives. Each Perspective addresses distinct responsibilities. The planning process helps the right people across the organization prepare for the changes ahead.

In general, the **Business, People,** and **Governance** Perspectives focus on **business capabilities**, whereas the **Platform, Security,** and **Operations** Perspectives focus on **technical capabilities**.

The **Business Perspective**: ensures that IT aligns with business needs and that IT investments link to key business results. Use the Business Perspective to create a strong business case for cloud adoption and prioritize cloud adoption initiatives. Ensure that your business strategies and goals align with your IT strategies and goals. Common roles: business managers, finance managers, budget owners, and strategy stakeholders.

The **People Perspective:** supports development of an organization-wide change management strategy for successful cloud adoption. Use the People Perspective to evaluate organizational structures and roles, new skill and process requirements, and identify gaps. This helps prioritize training, staffing, and organizational changes. Common roles: human resources, staffing, and people manager.

The **Governance Perspective:** focuses on the skills and processes to align IT strategy with business strategy. This ensures that you maximize the business value and minimize risks. Use the Governance Perspective to understand how to update the staff skills and processes necessary to ensure business governance in the cloud. Manage and measure cloud investments to evaluate business outcomes. Common roles: CIO, program managers, enterprise architects, business analyst, and portfolio managers.

The **Platform Perspective:** includes principles and patterns for implementing new solutions on the cloud, and migrating on-premises workloads to the cloud. Use a variety of architectural models to understand and communicate the structure of IT systems and their relationships. Describe the architecture of the target state environment in detail. Common roles: CTO, IT managers, and solution architects.

The **Security Perspective:** ensures that the organization meets security objectives for visibility, auditability, control, and agility. Use the AWS CAF to structure the selection and implementation of security controls that meet the organization’s needs. Common roles: CISO, IT security managers, and IT security analysts.

The **Operations Perspective:** helps you to enable, run, use, operate, and recover IT workloads to the level agreed upon with your business stakeholders. Define how day-to-day, quarter-to-quarter, and year-to-year business is conducted. Align with and support the operations of the business. The AWS CAF helps these stakeholders define current operating procedures and identify the process changes and training needed to implement successful cloud adoption. Common roles: IT operation managers and IT support managers.

## The 6 R’s of Migration

When migrating applications to the cloud, six of the most common [migration strategies](https://aws.amazon.com/blogs/enterprise-strategy/6-strategies-for-migrating-applications-to-the-cloud/) that you can implement are:

**Rehosting:** also known as “lift-and-shift” involves moving applications without changes. In the scenario of a large legacy migration, in which the company is looking to implement its migration and scale quickly to meet a business case, the majority of applications are rehosted.

**Replatforming:** also known as “lift, tinker, and shift,” involves making a few cloud optimizations to realize a tangible benefit. Optimization is achieved without changing the core architecture of the application.

**Refactoring:** (also known as re-architecting) involves reimagining how an application is architected and developed by using cloud-native features. Refactoring is driven by a strong business need to add features, scale, or performance that would otherwise be difficult to achieve in the application’s existing environment.

**Repurchasing:** involves moving from a traditional license to a software-as-a-service model. For example, a business might choose to implement the repurchasing strategy by migrating from a customer relationship management (CRM) system to Salesforce.com.

**Retaining:** consists of keeping applications that are critical for the business in the source environment. This might include applications that require major refactoring before they can be migrated, or, work that can be postponed until a later time.

**Retiring:** is the process of removing applications that are no longer needed.

## AWS Snow Family

The AWS Snow Family is a collection of physical devices that help to physically transport up to exabytes of data into and out of AWS. AWS Snow Family is composed of AWS Snowcone, AWS Snowball, and AWS Snowmobile.

These devices offer different capacity points, and most include built-in computing capabilities. AWS owns and manages the Snow Family devices and integrates with AWS security, monitoring, storage management, and computing capabilities.

**AWS Snowcone:** is a small, rugged, and secure edge computing and data transfer device. It features 2 CPUs, 4 GB of memory, and 8 TB of usable storage.

**AWS Snowball:** offers two types of devices:

**Snowball Edge Storage Optimized**: devices are well suited for large-scale data migrations and recurring transfer workflows, in addition to local computing with higher capacity needs.

* Storage: 80 TB of hard disk drive (HDD) capacity for block volumes and Amazon S3 compatible object storage, and 1 TB of SATA solid state drive (SSD) for block volumes.
* Compute: 40 vCPUs, and 80 GiB of memory to support Amazon EC2 sbe1 instances (equivalent to C5).

**Snowball Edge Compute Optimized:** provides powerful computing resources for use cases such as machine learning, full motion video analysis, analytics, and local computing stacks.

* Storage: 42-TB usable HDD capacity for Amazon S3 compatible object storage or Amazon EBS compatible block volumes and 7.68 TB of usable NVMe SSD capacity for Amazon EBS compatible block volumes.
* Compute: 52 vCPUs, 208 GiB of memory, and an optional NVIDIA Tesla V100 GPU. Devices run Amazon EC2 sbe-c and sbe-g instances, which are equivalent to C5, M5a, G3, and P3 instances.

**AWS Snowmobile:** is an exabyte-scale data transfer service used to move large amounts of data to AWS. You can transfer up to 100 petabytes of data per Snowmobile, a 45-foot long ruggedized shipping container, pulled by a semi-trailer truck.



## Designing for Failure

Designing for failure means that your team has automated processes in place for when your system fails, in addition to having as much control as possible over how this failure occurs. A system designed for failure is more capable of self-healing, restarting and maintaining service when the worst happens.

Designing for failure yields a self-healing infrastructure that acts with the maturity that is expected of recent workloads. Preventing emergency calls guarantees a base level of satisfaction for the service-owning team. This also removes a level of stress that can otherwise grow into accelerated attrition. Designing for failure will deliver greater uptime for your products. It can shield a company from outages that could erode customer trust.

## Benefits of Right-Sizing

Before understanding right-sizing, you need to understand what an instance is. More detail can be found in the product section under compute. For the purpose of explaining right-sizing, an instance is running a virtual machine in the cloud or launching an application to the cloud. An example of an AWS instance is EC2.

Right sizing is the process of matching instance types and sizes to your workload performance and capacity requirements at the lowest possible cost. It’s also the process of looking at deployed instances and identifying opportunities to eliminate or downsize without compromising capacity or other requirements, which results in lower costs.

Right sizing is the most effective way to control cloud costs. It involves continually analyzing instance performance and usage needs and patterns—and then turning off idle instances and right sizing instances that are either overprovisioned or poorly matched to the workload.

Automation of right-sizing:

* **Scalability** is the ability to quickly increase or decrease resources like instances or services based on current demand.
* **Elasticity is the automation of scaling** resources in repose to demand.

## Monolithic Architecture VS Microservices

**What is meant by monolithic architecture?**

A monolithic architecture is the traditional unified model for the design of a software program. Monolithic, in this context, means "composed all in one piece." According to the Cambridge dictionary, the adjective monolithic also means both "too large" and "unable to be changed."

**What is microservices?**

Microservices architecture (often shortened to microservices) refers to an architectural style for developing applications. Microservices allow a large application to be separated into smaller independent parts, with each part having its own realm of responsibility.

**While a monolithic application is built as a single unified unit, a microservices architecture is a collection of smaller, independently deployable services.**

# How to Get Started With AWS

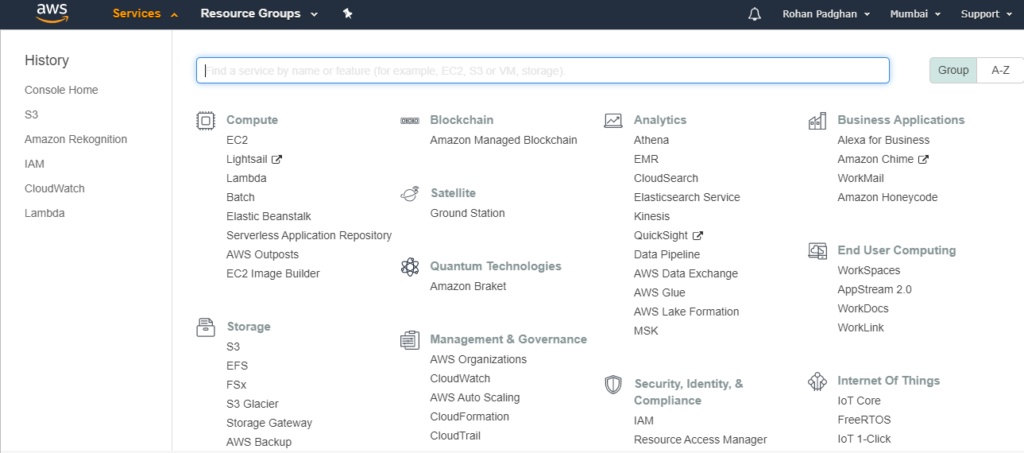
## Programmatic Access, APIs, Connectivity Options

Explain APIs, VPN, AWS Direct Connect, and Public Internet

## AWS Management Console

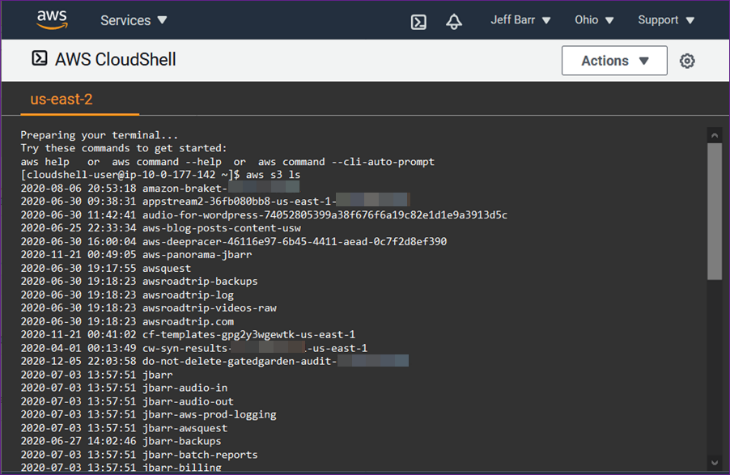
The AWS Management Console is a web-based interface for accessing and managing AWS services. You can quickly access recently used services and search for other services by name, keyword, or acronym. The console includes wizards and automated workflows that can simplify the process of completing tasks.

You can also use the AWS Console mobile application to perform tasks such as monitoring resources, viewing alarms, and accessing billing information. Multiple identities can stay logged into the AWS Console mobile app at the same time.



## AWS Command Line Interface and AWS CloudShell

To save time when making API requests, you can use the AWS Command Line Interface (AWS CLI). It basically processes commands to a computer program in the form of lines and texts. AWS CLI enables you to control multiple AWS services directly from the command line within one tool. AWS CLI is available for users on Windows, macOS, and Linux.

Developers describe AWS CloudShell as "An integrated shell for working with the AWS CLI (by AWS)". The AWS Command Line Interface is a unified tool to manage your AWS services. On the other hand, AWS CLI is detailed as "\* Universal Command Line Interface for Amazon Web Services\*". It is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.

## Software Development Kits

Another option for accessing and managing AWS services is the software development kits (SDKs). SDKs make it easier for you to use AWS services through an API designed for your programming language or platform. SDKs enable you to use AWS services with your existing applications or create entirely new applications that will run on AWS.

To help you get started with using SDKs, AWS provides documentation and sample code for each supported programming language. Supported programming languages include C++, Java, .NET, and more.

## Managed Service Tools for Getting Started (Easy Buttons)

AWS Management Console, the CLI, and the SDKs, are sort of do it yourself ways to provision and manage your AWS environment. There are also other ways you can manage your AWS environment using managed tools like AWS Elastic Beanstalk, and AWS CloudFormation.

**AWS Elastic Beanstalk** is a service that helps you provision Amazon EC2-based environments. Instead of clicking around the console or writing multiple commands to build out your network, you can instead provide your application code and desired configurations to the AWS Elastic Beanstalk service, which then takes that information and builds out your environment for you.

AWS Elastic Beanstalk also makes it easy to save environment configurations, so they can be deployed again easily. AWS Elastic Beanstalk gives you the convenience of not having to provision and manage all of these pieces separately, while still giving you the visibility and control of the underlying resources. You get to focus on your business application, not the infrastructure.

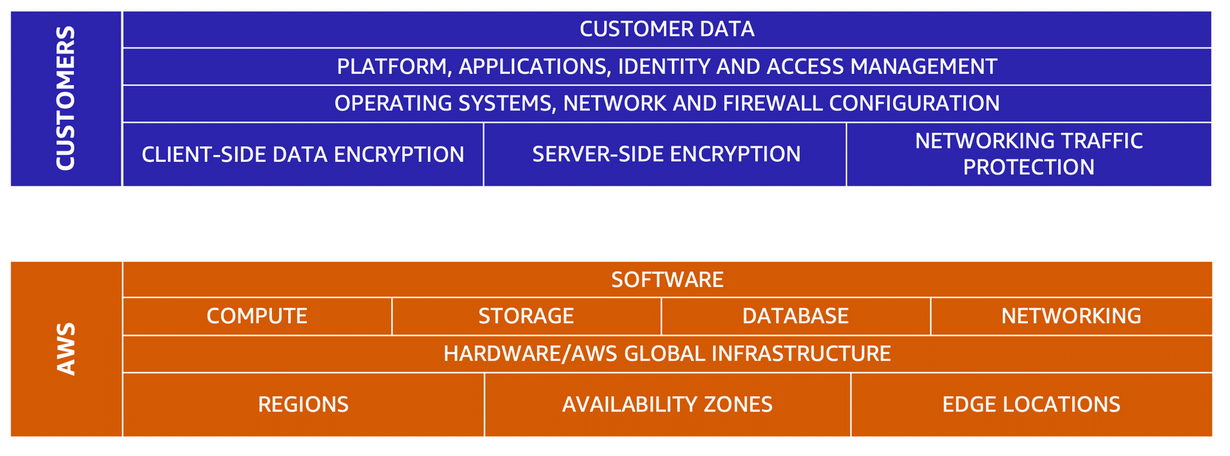
**AWS CloudFormation** is an infrastructure as code tool that allows you to define a wide variety of AWS resources in a declarative way using JSON or YAML text-based documents called CloudFormation templates. A declarative format like this allows you to define what you want to build without specifying the details of exactly how to build it. CloudFormation lets you define what you want and the CloudFormation engine will worry about the details on calling APIs to get everything built out.

It also isn't just limited to EC2-based solutions. CloudFormation supports many different AWS resources from storage, databases, analytics, machine learning, and more. You can run the same CloudFormation template in multiple accounts or multiple regions, and it will create identical environments across them. There is less room for human error as it is a totally automated process.

# Security, Identity, and Compliance

## Recap of Shared Responsibility Model

A shared responsibility model in cloud computing is the idea that risk and security is shared between the cloud vendor like AWS and the customer. The image below shows the split responsibilities. Refer to [page 10](#_heading=h.3dy6vkm) for more information.



## User Permissions and Access

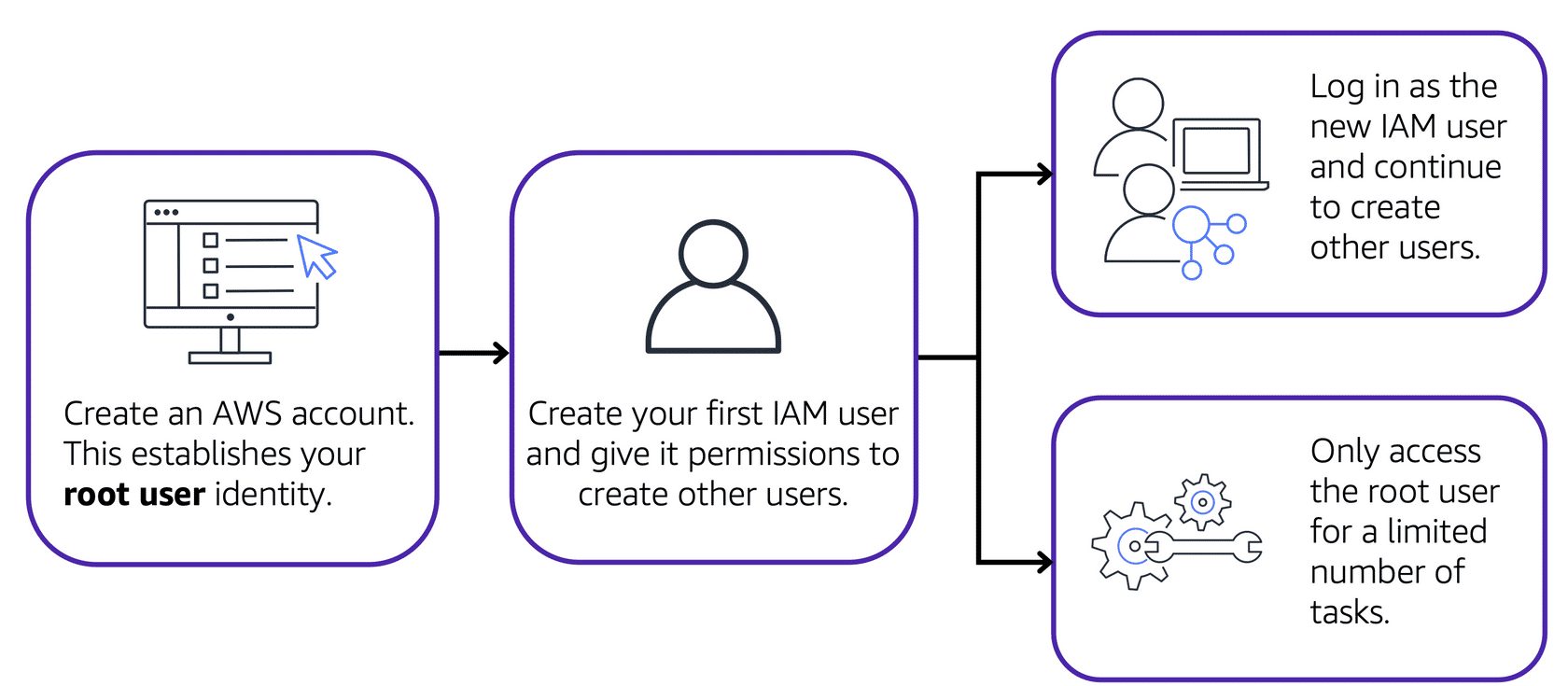
**AWS Identity and Access Management (IAM)**: enables you to manage access to AWS services and resources securely. IAM gives you the flexibility to configure access based on your company’s specific operational and security needs. You do this by using a combination of IAM features:

* IAM users, groups, and roles
* IAM policies
* Multi-factor authentication

**AWS Account Root User:** when you first create an AWS account, you begin with an identity known as the root user. The root user is accessed by signing in with the email address and password that you used to create your AWS account. You can think of the root user as the main owner of access management. It has complete access to all the AWS services and resources in the account.

Do not use the root user for everyday tasks. Instead, use the root user to create your first IAM user and assign it permissions to create other users.

Then, continue to create other IAM users, and access those identities for performing regular tasks throughout AWS. Only use the root user when you need to perform a limited number of tasks that are only available to the root user. Examples of these tasks include changing your root user email address and changing your AWS support plan.



An **IAM user:** is an identity that you create in AWS. It represents the person or application that interacts with AWS services and resources. It consists of a name and credentials. **By default, when you create a new IAM user in AWS, it has no permissions associated with it.** To allow the IAM user to perform specific actions in AWS, such as launching an Amazon EC2 instance or creating an Amazon S3 bucket, you must grant the IAM user the necessary permissions.

AWS recommends that you create individual IAM users for each person who needs to access AWS. Even if you have multiple employees who require the same level of access, you should create individual IAM users for each of them. This provides additional security by allowing each IAM user to have a unique set of security credentials.

An **IAM Policy** is a document that allows or denies permissions to AWS services and resources. IAM policies enable you to customize users’ levels of access to resources. For example, you can allow users to access all of the Amazon S3 buckets within your AWS account, or only a specific bucket.

Follow the security principle of least privilege when granting permissions. By following this principle, you help to prevent users or roles from having more permissions than needed to perform their tasks. For example, if an employee needs access to only a specific bucket, specify the bucket in the IAM policy. Do this instead of granting the employee access to all of the buckets in your AWS account.

An **IAM Group** is a collection of IAM users. When you assign an IAM policy to a group, all users in the group are granted permissions specified by the policy.

**IAM Role:** is an identity that you can give to a user to gain temporary access to permissions, instead of long term. For example, you might need a developer from another project help debug your own project. You would be able to give them temporary access.

**Multi-factor authentication**: is an electronic authentication method in which a user is granted access to a website or application only after successfully presenting two or more pieces of evidence to an authentication mechanism. An example would be entering a password and then entering a code that was sent to your mobile device OR inserting a security key into your laptop. In IAM, multi-factor authentication (MFA) provides an extra layer of security for your AWS account.

## AWS Organizations

If your company has multiple AWS accounts, you can use AWS Organizations to consolidate and manage multiple AWS accounts within a central location.

When you create an organization, AWS Organizations automatically creates a root, which is the parent container for all the accounts in your organization.

In AWS Organizations, you can centrally control permissions for the accounts in your organization by using service control policies (SCPs). SCPs enable you to place restrictions on the AWS services, resources, and individual API actions that users and roles in each account can access. In AWS Organizations, you can group accounts into organizational units (OUs) to make it easier to manage accounts with similar business or security requirements. When you apply a policy to an OU, all the accounts in the OU automatically inherit the permissions specified in the policy.

In AWS Organizations, you can apply service control policies (SCPs) to the organization root, an individual member account, or an OU. An SCP affects all IAM users, groups, and roles within an account, including the AWS account root user. You can apply IAM policies to IAM users, groups, or roles. You cannot apply an IAM policy to the AWS account root user.

## Compliance

See the previous sections of this guide “[Compliance and Data Regulations”](#_heading=h.2jxsxqh) and the “[Shared Responsibility Model](#_heading=h.3dy6vkm)”.

Depending on your company’s industry, you may need to uphold specific standards. An audit or inspection will ensure that the company has met those standards.

AWS Artifact is a service that provides on-demand access to AWS security and compliance reports and select online agreements. AWS Artifact consists of two main sections: AWS Artifact Agreements and AWS Artifact Reports. AWS also have a Customer Compliance Center that contains resources to help you learn more about AWS compliance.

In **AWS Artifact Agreements**, you can review, accept, and manage agreements for an individual account and for all your accounts in AWS Organizations. Different types of agreements are offered to address the needs of customers who are subject to specific regulations, such as the Health Insurance Portability and Accountability Act (HIPAA).

**AWS Artifact Reports** provide compliance reports from third-party auditors. These auditors have tested and verified that AWS is compliant with a variety of global, regional, and industry-specific security standards and regulations. AWS Artifact Reports remains up to date with the latest reports released. You can provide the AWS audit artifacts to your auditors or regulators as evidence of AWS security controls.

In the **Customer Compliance Center**, you can read customer compliance stories to discover how companies in regulated industries have solved various compliance, governance, and audit challenges.

You can also access compliance whitepapers and documentation on topics such as: AWS answers to key compliance questions, an overview of AWS risk and compliance and an auditing security checklist

Additionally, the Customer Compliance Center includes an auditor learning path. This learning path is designed for individuals in auditing, compliance, and legal roles who want to learn more about how their internal operations can demonstrate compliance using the AWS Cloud.

## Denial-of-Service-Attacks

A denial-of-service (DoS) attack is a deliberate attempt to make a website or application unavailable to users. For example, an attacker might flood a website or application with excessive network traffic until the targeted website or application becomes overloaded and is no longer able to respond. If the website or application becomes unavailable, this denies service to users who are trying to make legitimate requests.

In addition to DoS, there is also **distributed denial-of-service attack** (DDos). In a distributed denial-of-service (DDoS) attack, multiple sources are used to start an attack that aims to make a website or application unavailable. This can come from a group of attackers, or even a single attacker. The single attacker can use multiple infected computers (also known as “bots”) to send excessive traffic to a website or application.

Examples of clever DDoS attacks (*below is more of a “cool to know” section, it won’t be on the exam*):

**UDP Flood**: is a type of denial-of-service attack in which a large number of User Datagram Protocol (UDP) packets are sent to a targeted server with the aim of overwhelming that device’s ability to process and respond. For example, there are many services on the internet where you can request megabytes worth of data, like a weather network. In a UDP flood attack, the attacker requests megabytes of information to “your” server instead of their own, which brings your server to a standstill as it tries to process a ton of information it never asked for.

**HTTP Level Attack**: is a volumetric attack, often using a botnet “zombie army”, a group of internet-connects computers, each of which has been maliciously taken over by malware virus. For example, a hacker might use an HTTP Level Attack to get the zombie army to request a ton of information about a product on your website, that will overload your system.

**Slowloris Attack:** is when the attacker pretends to have a slow bandwidth connection and your servers are waiting for the attackers request or “full packet” before it can service the next customer. A few Slowloris attackers can exhaust the capacity of your entire front end with almost no effort at all.

To help minimize the effect of DoS and DDoS attacks on your applications, you can use **AWS Shield.**

## AWS Shield

**AWS Shield** is a service that protects applications against DDoS attacks. AWS Shield provides two levels of protection: Standard and Advanced.

**AWS Shield Standard** automatically protects all AWS customers at no cost. It protects your AWS resources from the most common, frequently occurring types of DDoS attacks.

As network traffic comes into your applications, AWS Shield Standard uses a variety of analysis techniques to detect malicious traffic in real time and automatically mitigates it.

**AWS Shield Advanced** is a paid service that provides detailed attack diagnostics and the ability to detect and mitigate sophisticated DDoS attacks.

It also integrates with other services such as Amazon CloudFront, Amazon Route 53, and Elastic Load Balancing. Additionally, you can integrate AWS Shield with AWS WAF by writing custom rules to mitigate complex DDoS attacks.

Note\* **AWS WAF** is a web application firewall that lets you monitor the HTTP(S) requests that are forwarded to your protected web application resources.

## Additional Security Services

Like your physical business location or home, you must ensure that your applications’ data is secure, whether they’re sitting in the storage room or being transported between locations.  There are two ways we communicate this, for data in storage it is encryption at rest and while it is transmitted, it is known as encryption in transit.

### AWS KMS

**AWS Key Management Service (AWS KMS)** enables you to perform encryption operations through the use of cryptographic keys. A cryptographic key is a random string of digits used for locking (encrypting) and unlocking (decrypting) data. You can use AWS KMS to create, manage, and use cryptographic keys. You can also control the use of keys across a wide range of services and in your applications.

With AWS KMS, you can choose the specific levels of access control that you need for your keys. For example, you can specify which IAM users and roles are able to manage keys. Alternatively, you can temporarily disable keys so that they are no longer in use by anyone. Your keys never leave AWS KMS, and you are always in control of them

### AWS WAF

**AWS WAF is a web application firewall** that lets you monitor network requests that come into your web applications. AWS WAF works together with Amazon CloudFront and an Application Load Balancer. AWS WAF works to block or allow traffic. When a request comes into AWS WAF, it checks against the list of rules that you have configured in the web ACL (access control list). If a request did not come from one of the blocked IP addresses, it allows access to the application.

### Amazon Inspector

**Amazon Inspector** helps to improve the security and compliance of applications by running automated security assessments. It checks applications for security vulnerabilities and deviations from security best practices, such as open access to Amazon EC2 instances and installations of vulnerable software versions.

After Amazon Inspector has performed an assessment, it provides you with a list of security findings. The list prioritizes by severity level, including a detailed description of each security issue and a recommendation for how to fix it.

### Amazon GaurdDuty

**Amazon GuardDuty** is a service that provides intelligent threat detection for your AWS infrastructure and resources. It identifies threats by continuously monitoring the network activity and account behavior within your AWS environment. It uses integrated threat intelligence such as known malicious IP addresses, anomaly detection, and machine learning to identify threats more accurately. If GuardDuty detects any threats, you can review detailed findings about them from the AWS Management Console.

# Core Products and Services

## Compute

**Amazon Elastic Compute Cloud** or **Amazon EC2** allows users to rent virtual computers on which to run their own computer applications.

Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers (also know as **instances**) as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

**Different types of EC2 Instances**

Amazon EC2 instance types are optimized for different tasks. When selecting an instance type, consider the specific needs of your workloads and applications. This might include requirements for compute, memory, or storage capabilities.

**1. General purpose instances** provide a balance of compute, memory, and networking resources. You can use them for a variety of workloads, such as:

* application servers
* gaming servers
* backend servers for enterprise applications
* small and medium databases

Suppose that you have an application in which the resource needs for compute, memory, and networking are roughly equivalent. You might consider running it on a general-purpose instance because the application does not require optimization in any single resource area.

**2. Compute optimized instances** are ideal for compute-bound applications that benefit from high-performance processors. Like general purpose instances, you can use compute optimized instances for workloads such as web, application, and gaming servers.

However, the difference is compute optimized applications are ideal for high-performance web servers, compute-intensive applications servers, and dedicated gaming servers. You can also use compute optimized instances for batch processing workloads that require processing many transactions in a single group.

**3. Memory optimized instances** are designed to deliver fast performance for workloads that process large datasets in memory. In computing, memory is a temporary storage area. It holds all the data and instructions that a central processing unit (CPU) needs to be able to complete actions.

Before a computer program or application is able to run, it is loaded from storage into memory. This preloading process gives the CPU direct access to the computer program. Memory optimized instances enable you to run workloads with high memory needs and receive great performance.

**4. Accelerated computing instances** use hardware accelerators, or coprocessors, to perform some functions more efficiently than is possible in software running on CPUs. Examples of these functions include floating-point number calculations, graphics processing, and data pattern matching.

In computing, a hardware accelerator is a component that can expedite data processing. Accelerated computing instances are ideal for workloads such as graphics applications, game streaming, and application streaming.

**5. Storage optimized instances** are designed for workloads that require high, sequential read and write access to large datasets on local storage. Examples of workloads suitable for storage optimized instances include distributed file systems, data warehousing applications, and high-frequency online transaction processing (OLTP) systems.

**Amazon EC2 Pricing**

**With Amazon EC2, you pay only for the compute time that you use**. Amazon EC2 offers a variety of pricing options for different use cases.

**1. On-Demand Instances** are ideal for short-term, irregular workloads that cannot be interrupted. No upfront costs or minimum contracts apply. The instances run continuously until you stop them, and you pay for only the compute time you use.

Sample use cases for On-Demand Instances include developing and testing applications and running applications that have unpredictable usage patterns. On-Demand Instances are not recommended for workloads that last a year or longer because these workloads can experience greater cost savings using Reserved Instances.

**2. AWS offers Savings Plans** for several compute services, including Amazon EC2. Amazon EC2 Savings Plans enable you to reduce your compute costs by committing to a consistent amount of compute usage for a 1-year or 3-year term. This term commitment results in savings of up to 72% over On-Demand costs. This is also referred to as a “**customers commit**” or a common phrase when customers start using their compute towards this is called “**drawing down on their commit**”.

**3. Reserved Instances** are a billing discount applied to the use of On-Demand Instances in your account. You can purchase Standard Reserved and Convertible Reserved Instances for a 1-year or 3-year term, and Scheduled Reserved Instances for a 1-year term. You realize greater cost savings with the 3-year option.

At the end of a Reserved Instance term, you can continue using the Amazon EC2 instance without interruption. However, you are charged On-Demand rates until you do one of the following:

* Terminate the instance.
* Purchase a new Reserved Instance that matches the instance attributes (instance type, Region, tenancy, and platform).

**What’s the difference between Reserved Instances vs AWS Savings Plans?** To begin with, Reserved Instances are based on the commitment to use an instance at a particular price over a specific period, while Savings Plans are based on the commitment to spend a particular dollar amount per hour over a specific period. Amazon EC2 Savings Plans are ideal for workloads that involve a consistent amount of computer usage over a 1-year or 3-year term.

**4.** **Spot Instances** are ideal for workloads with flexible start and end times, or that can withstand interruptions. Spot Instances use unused Amazon EC2 computing capacity and offer you cost savings at up to 90% off of On-Demand prices. You should only use spot instances for workloads that can be interpreted if there is unavailable capacity at the time of deploying your instance.

**5. Dedicated Hosts** are physical servers with Amazon EC2 instance capacity that is fully dedicated to your use. Of all the Amazon EC2 options that were covered, Dedicated Hosts are the **most expensive**.

**Scaling and Auto Scaling EC2 Instances**

Scalability involves beginning with only the resources you need and designing your architecture to automatically respond to changing demand by scaling out or in. As a result, you pay for only the resources you use. You don’t have to worry about a lack of computing capacity to meet your customers’ needs.

If you’ve tried to access a website that wouldn’t load and frequently timed out, the website might have received more requests than it was able to handle.

**Amazon EC2 Auto Scaling** enables you to automatically add or remove Amazon EC2 instances in response to changing application demand. By automatically scaling your instances in and out as needed, you are able to maintain a greater sense of application availability. You would need to set your own parameters (EC2 instances) such as, minimum capacity, desired capacity, and maximum capacity.

Within Amazon EC2 Auto Scaling, you can use two approaches: dynamic scaling and predictive scaling.

* Dynamic scaling responds to changing demand.
* Predictive scaling automatically schedules the right number of Amazon EC2 instances based on predicted demand.

**Elastic Load Balancing**

Elastic Load Balancing is the AWS service that automatically distributes incoming application traffic across multiple resources, such as Amazon EC2 instances.

A load balancer acts as a single point of contact for all incoming web traffic to your Auto Scaling group. This means that as you add or remove Amazon EC2 instances in response to the amount of incoming traffic, these requests route to the load balancer first. Then, the requests spread across multiple resources that will handle them. For example, if you have multiple Amazon EC2 instances, Elastic Load Balancing distributes the workload across the multiple instances so that no single instance has to carry the bulk of it.

Although Elastic Load Balancing and Amazon EC2 Auto Scaling are separate services, they work together to help ensure that applications running in Amazon EC2 can provide high performance and availability.

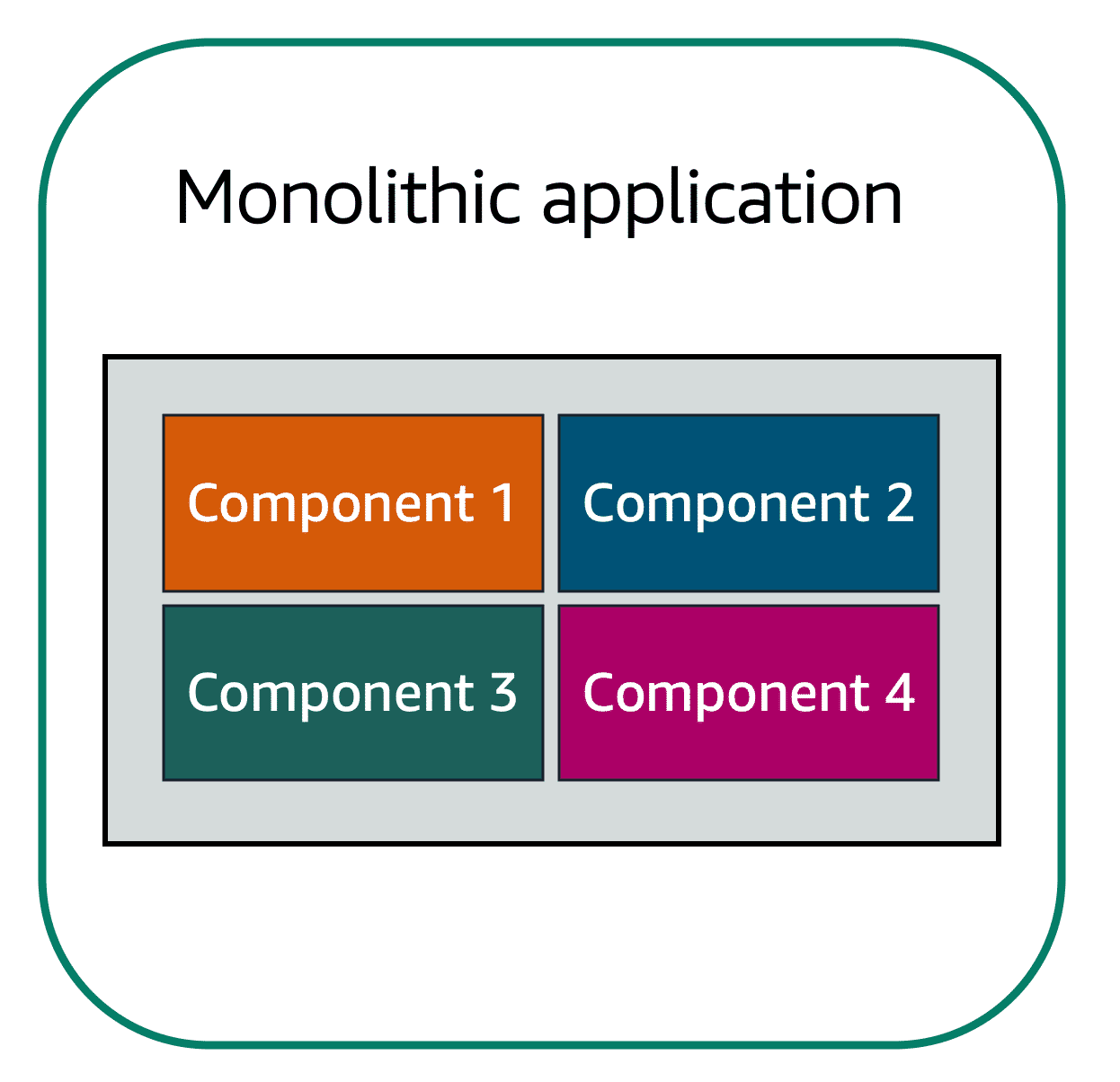
A real-life example of this would be waiting in line at Walmart or Homesense where they direct customers in line to the next available cashier with either a light up sign or a recording that says “cashier number 6”.

**NOTE:** If you are still trying to understand the difference between auto scaling and load balancing, here is a [great article by Anish Antony on Medium](https://medium.com/programmingnotes/aws-auto-scaling-and-load-balancing-b1c6eeb4d074) that explains the two services and there differences in detail.

## Messaging and Queuing

In order to understand the services for messaging and queuing, we first need to dive deeper into two different environments.

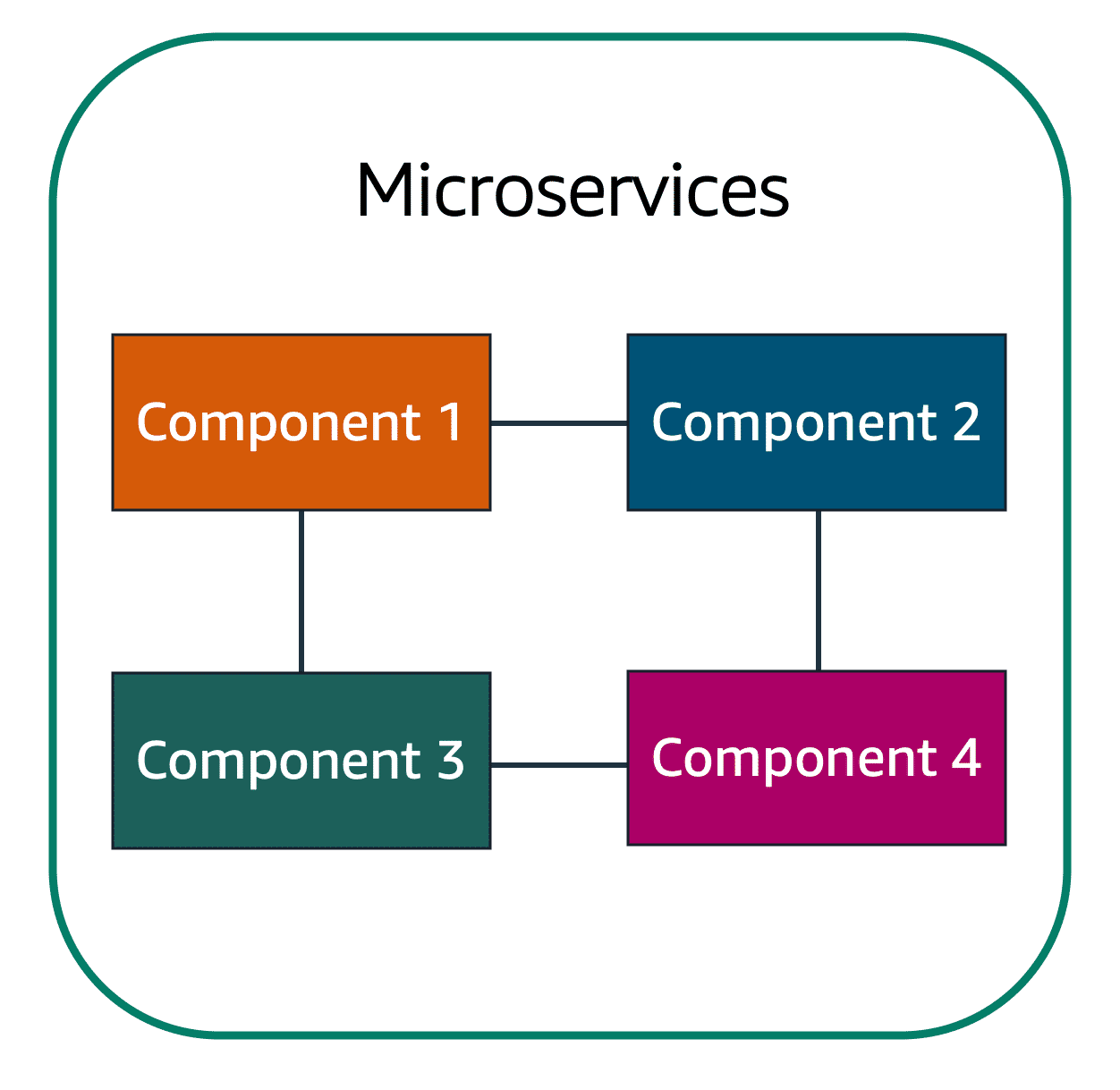
**Monolithic Applications and Microservices**

Applications are made of multiple components. The components communicate with each other to transmit data, fulfill requests, and keep the application running.

Suppose that you have an application with tightly coupled components. These components might include databases, servers, the user interface, business logic, and so on. This type of architecture can be considered a monolithic application.

In this approach to application architecture, if a single component fails, other components fail, and possibly the entire application fails.

**Microservices and Loosely Coupled Components**

In a **microservices approach**, application components are **loosely coupled**. In this case, if a single component fails, the other components continue to work because they are communicating with each other. The loose coupling prevents the entire application from failing.

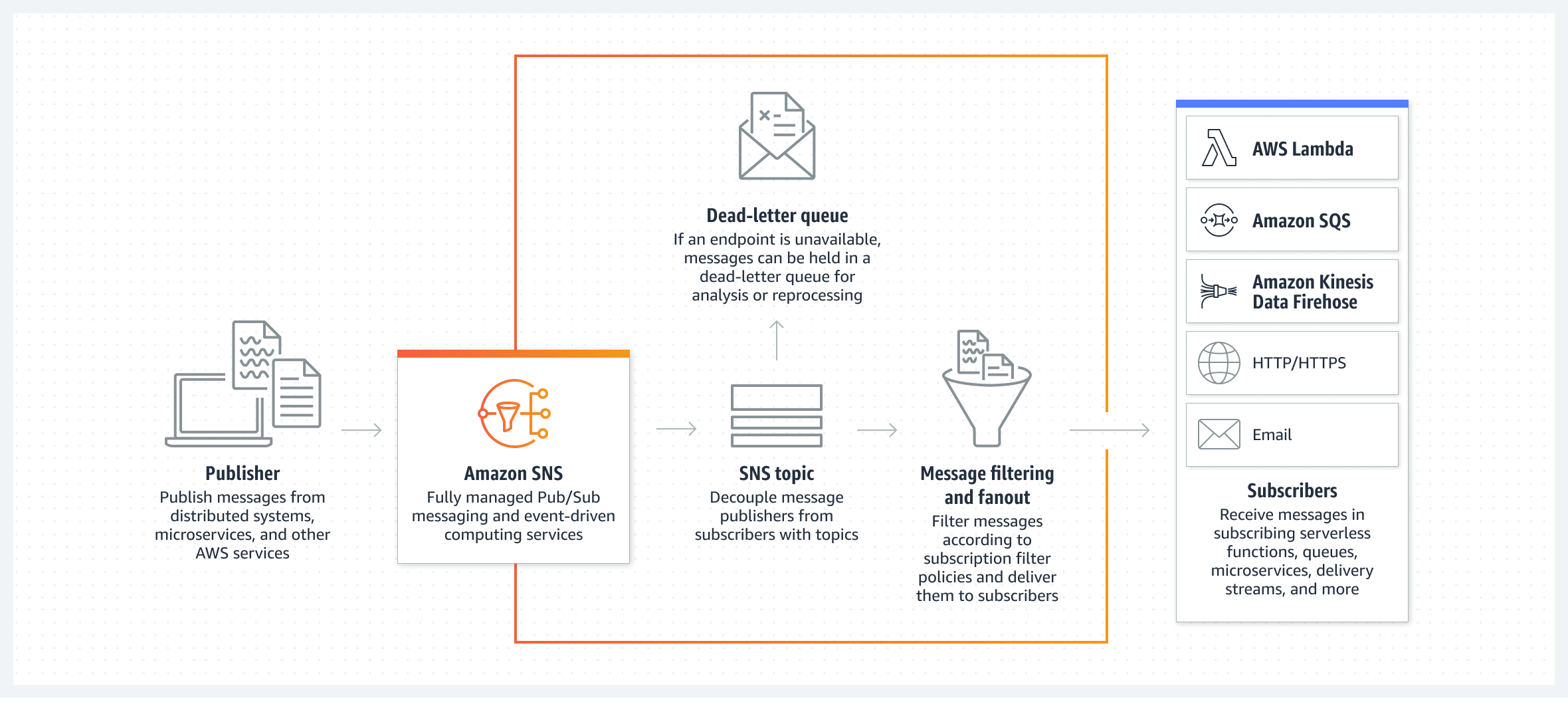
When designing applications on AWS, you can take a microservices approach with services and components that fulfill different functions. Two services facilitate application integration: **Amazon Simple Notification Service (Amazon SNS)** **and Amazon Simple Queue Service (Amazon SQS)**.

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

Amazon Simple Notification Service (Amazon SNS) is a publish/subscribe service. Using Amazon SNS topics, a publisher publishes messages to subscribers. In Amazon SNS, subscribers can be web servers, email addresses, AWS Lambda functions, or several other options.

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

Amazon Simple Queue Service (Amazon SQS) is a message queuing service. Using Amazon SQS, you can send, store, and receive messages between software components, without losing messages or requiring other services to be available. In Amazon SQS, an application sends messages into a queue. A user or service retrieves a message from the queue, processes it, and then deletes it from the queue.



## Serverless Computing

Unlike EC2 instances where you need to provision or manage the servers, AWS offers serverless computing solutions. The term “serverless” means that your code runs on servers, but you do not need to provision or manage these servers. With serverless computing, you can focus more on innovating new products and features instead of maintaining servers.

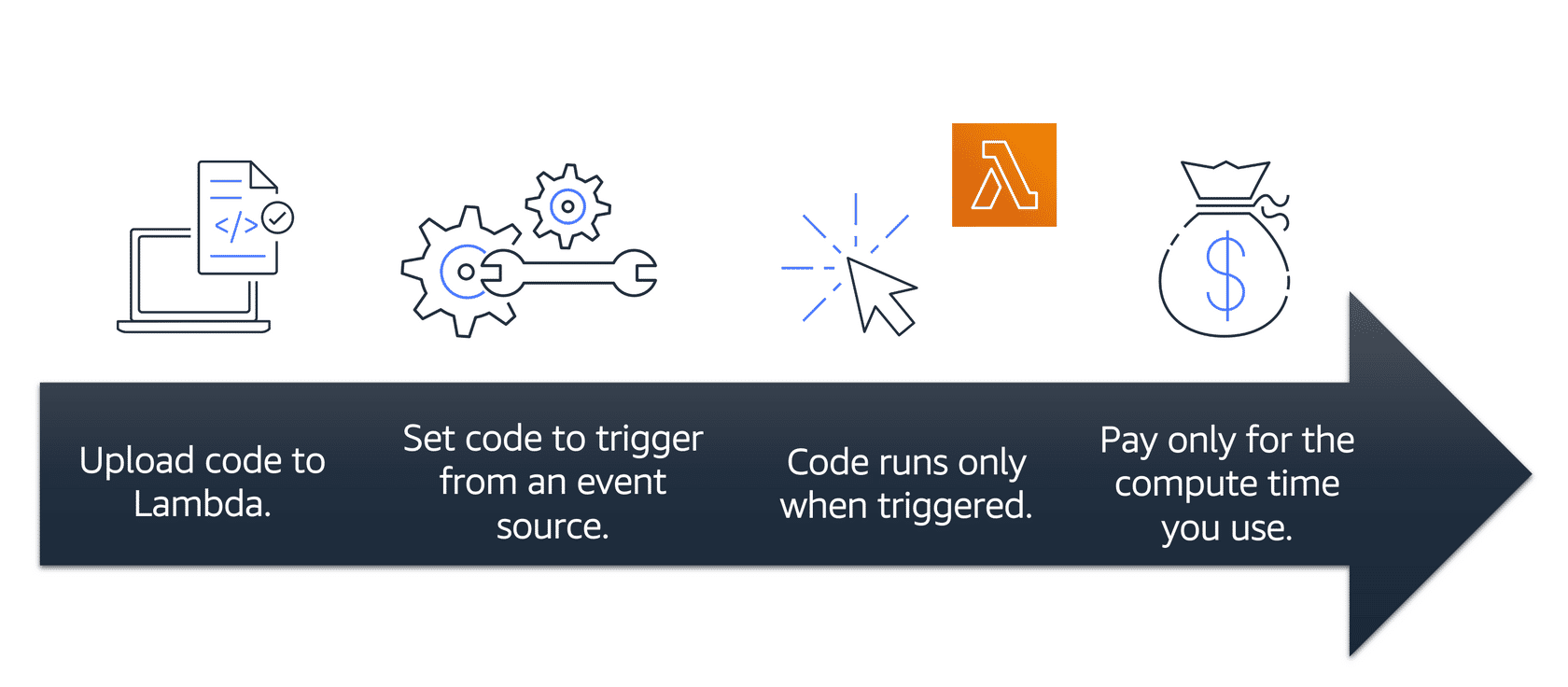
Another benefit of serverless computing is the flexibility to scale serverless applications automatically. Serverless computing can adjust the applications' capacity by modifying the units of consumptions, such as throughput and memory.

**AWS Lambda**

AWS Lambda is a serverless computing service that lets you run code without needing to provision or manage servers. You write the code and then upload it to AWS Lambda, it’s that simple, hence the “Lamb-Duh” favorite laptop sticker of mine.

While using AWS Lambda, you pay only for the compute time that you consume. Charges apply only when your code is running. You can also run code for virtually any type of application or backend service, all with zero administration.

For example, a simple Lambda function might involve automatically resizing uploaded images to the AWS Cloud. In this case, the function triggers when uploading a new image.



## Containers & Kubernetes

Containers provide you with a standard way to package your application's code and dependencies into a single object. You can also use containers for processes and workflows in which there are essential requirements for security, reliability, and scalability.

To understand containers and Kubernetes – a container orchestration service – check out my humorous article, [Marketing Tries to Explain Kubernetes](https://www.linkedin.com/pulse/marketing-girl-tries-explain-kubernetes-sarah-kymberlee-walker-/), that explains these concepts in a VERY basic and introductory way to the average non-technical layperson.

**Amazon Elastic Container Service (Amazon ECS)**

Amazon Elastic Container Service (Amazon ECS) is a highly scalable, high-performance container management system that enables you to run and scale containerized applications on AWS.

Amazon ECS supports Docker containers. Docker is a software platform that enables you to build, test, and deploy applications quickly. AWS supports the use of open-source Docker Community Edition and subscription-based Docker Enterprise Edition. With Amazon ECS, you can use API calls to launch and stop Docker-enabled applications.

**Amazon Elastic Kubernetes Service (Amazon EKS)**

Amazon Elastic Kubernetes Service (Amazon EKS) is a fully managed service that you can use to run Kubernetes on AWS.

Kubernetes is open-source software that enables you to deploy and manage containerized applications at scale. A large community of volunteers maintains Kubernetes, and AWS actively works together with the Kubernetes community. As new features and functionalities release for Kubernetes applications, you can easily apply these updates to your applications managed by Amazon EKS.

**AWS Fargate**

AWS Fargate is a serverless compute engine for containers. It works with both Amazon ECS and Amazon EKS. When using AWS Fargate, you do not need to provision or manage servers. AWS Fargate manages your server infrastructure for you. You can focus more on innovating and developing your applications, and you pay only for the resources that are required to run your containers.

## Networking

Let’s first establish the difference between a Virtual Private Network (VPN) and a Virtual Private Cloud (VPC). All businesses need to be able to transfer information, today we mainly use the internet like email or document storage vs fax or physically mailing documents through the post office. In order for businesses to be able to transfer information over the internet, they need to connect to a VPN, which provides secure data transfer over the public internet. The difference, is that a VPC provides secure data transfer between a private enterprise and a public cloud provider, like AWS.

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

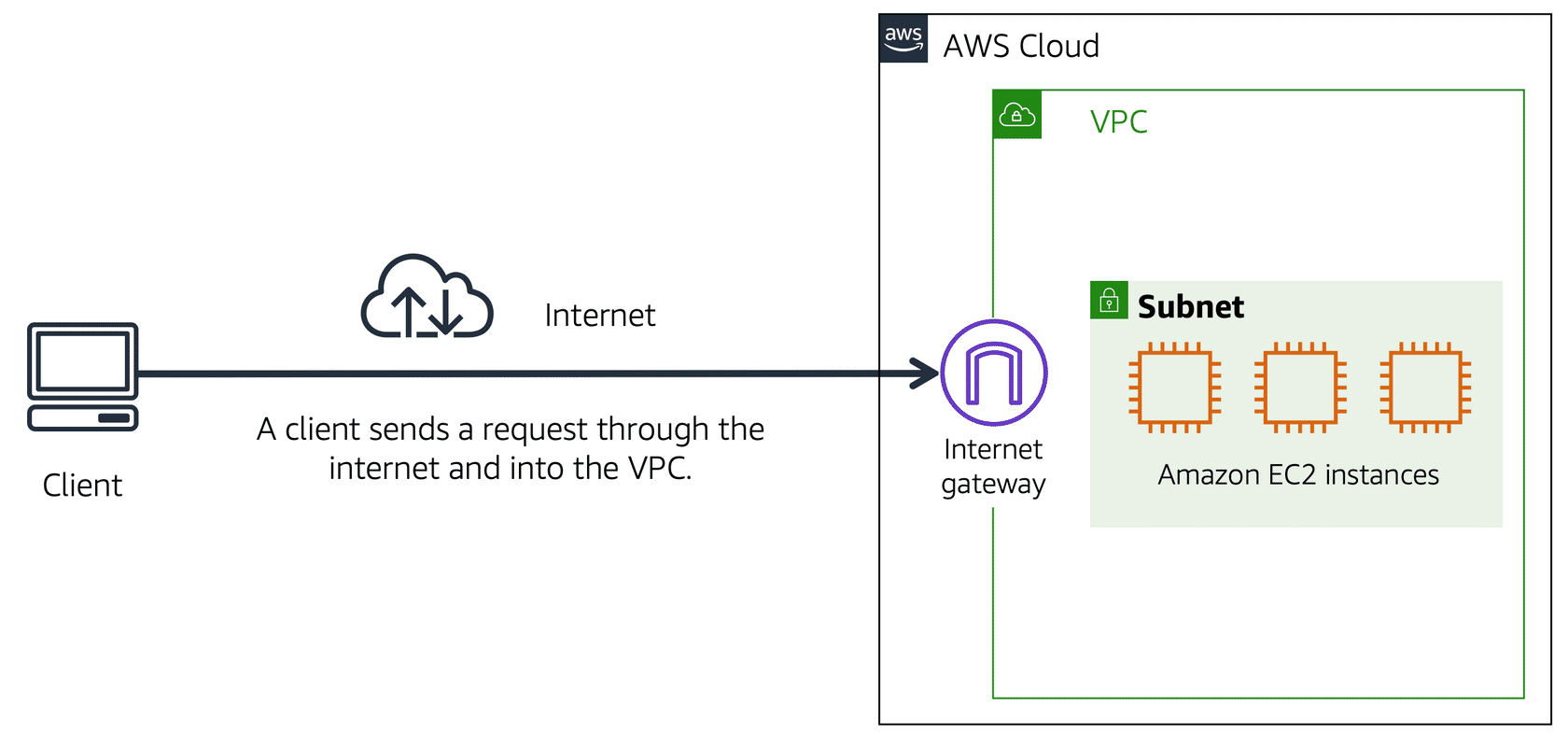
Imagine the millions of customers who use AWS services. Also, imagine the millions of resources that these customers have created, such as Amazon EC2 instances. Without boundaries around all of these resources, network traffic would be able to flow between them unrestricted.

A networking service that you can use to establish boundaries around your AWS resources is Amazon Virtual Private Cloud (Amazon VPC).

Amazon VPC enables you to provision an isolated section of the AWS Cloud. In this isolated section, you can launch resources in a virtual network that you define. Within a virtual private cloud (VPC), you can organize your resources into subnets. A subnet is a section of a VPC that can contain resources such as Amazon EC2 instances.

**Internet gateway**

To allow public traffic from the internet to access your VPC (ie to download images or make purchases), you attach an internet gateway to the VPC. An internet gateway is a connection between a VPC and the internet. You can think of an internet gateway as being similar to a doorway that customers use to enter the coffee shop. Without an internet gateway, no one can access the resources within your VPC.

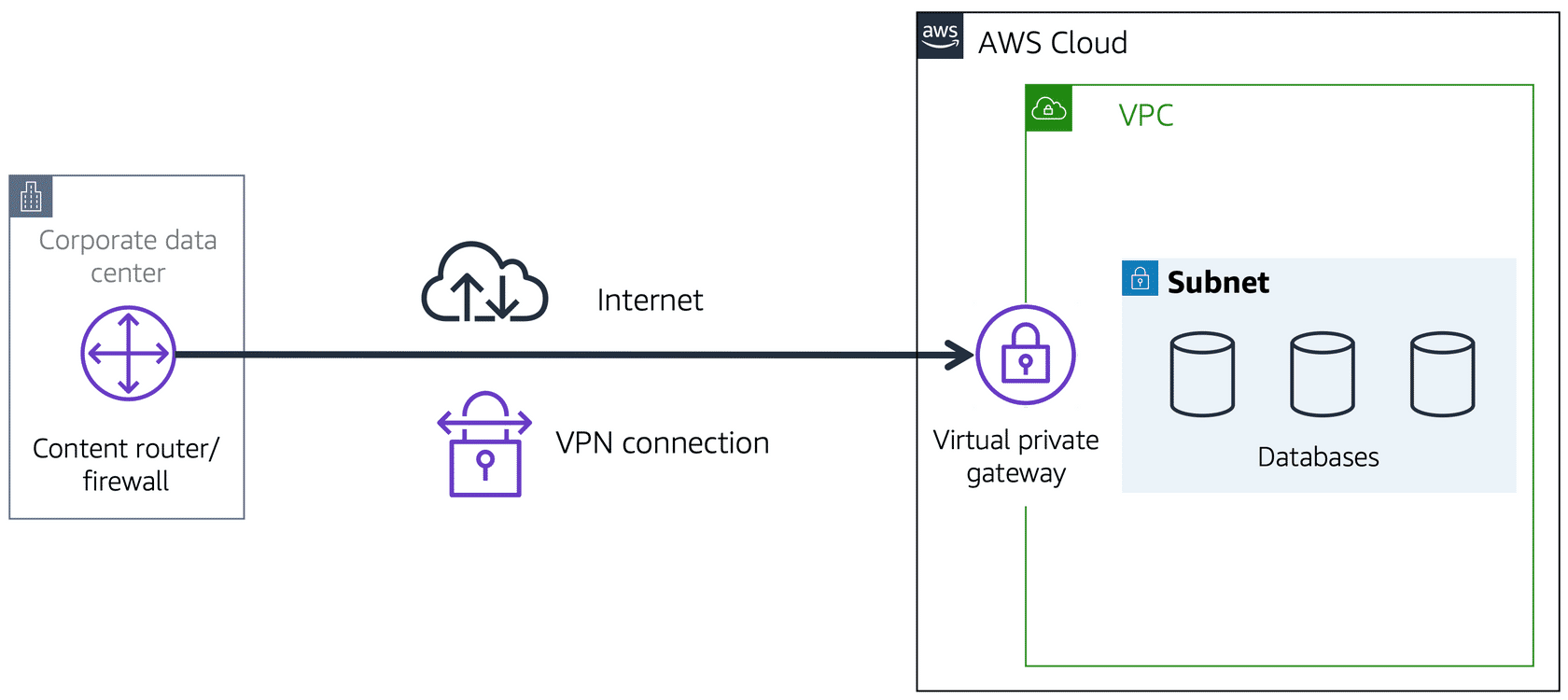


**Virtual private gateway**

To access private resources in a VPC (ie remote employees who need access to confidential internal documents), you can use a virtual private gateway.

The virtual private gateway is the component that allows protected internet traffic to enter into the VPC. A virtual private gateway enables you to establish a virtual private network (VPN) connection between your VPC and a private network, such as an on-premises data center or internal corporate network. A virtual private gateway allows traffic into the VPC only if it is coming from an approved network.

Here’s an example of how a virtual private gateway works:

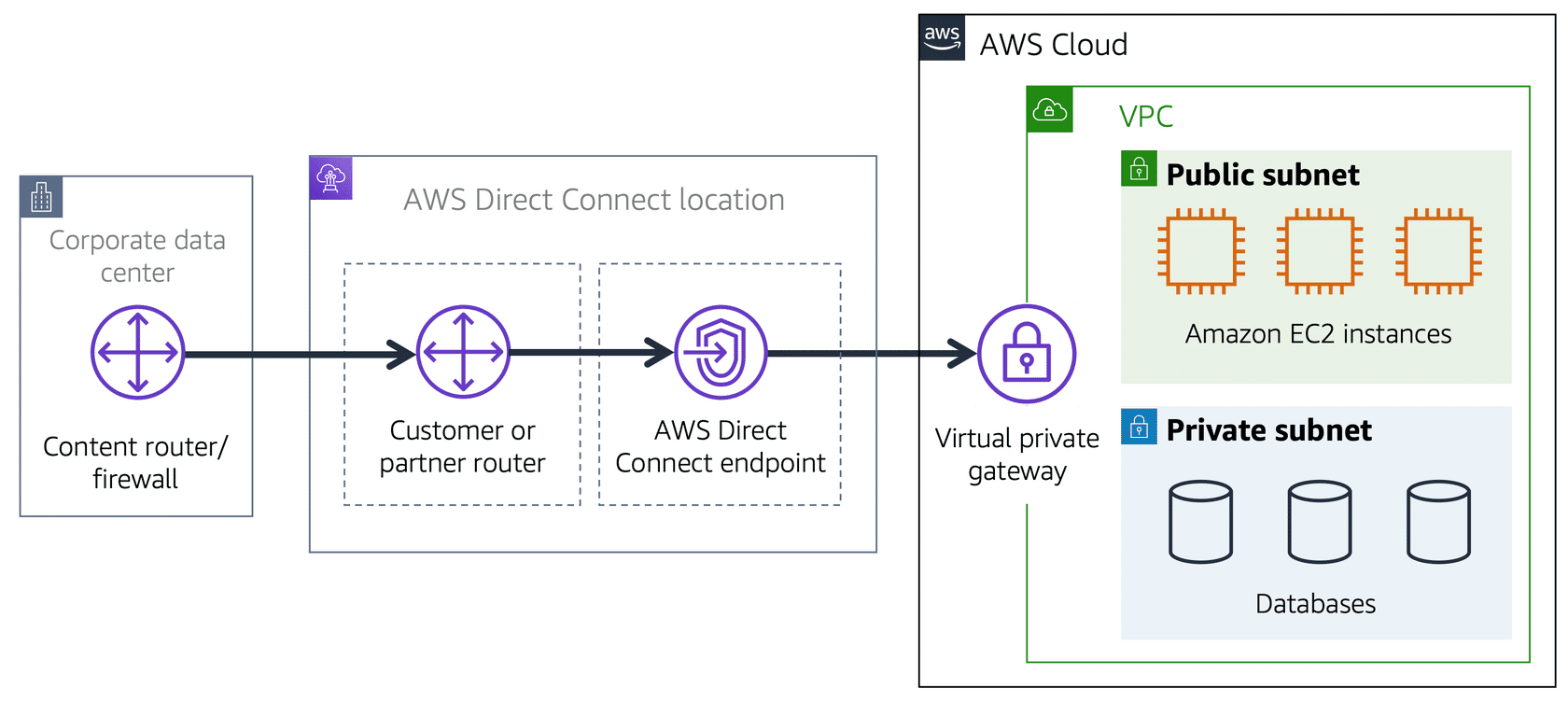


**AWS Direct Connect**

Some businesses require a dedicated private connection between their data center and a VPC, which is established with AWS Direct Connect.

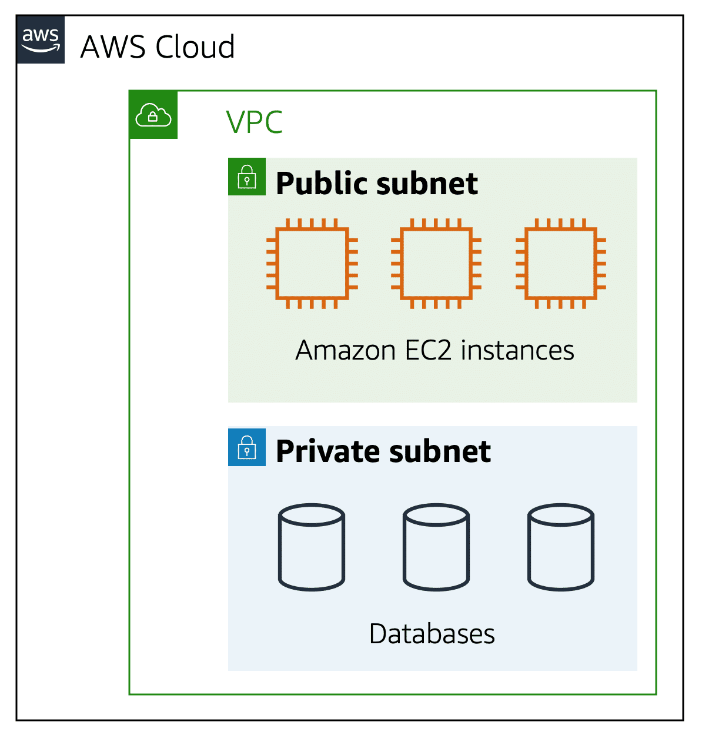
The private connection that AWS Direct Connect provides helps you to reduce network costs and increase the amount of bandwidth that can travel through your network.

Architecture diagram showing traffic flowing from a corporate data center, across an AWS Direct Connect location, and into a VPC through a virtual private gateway



**Subnets**

A subnet is a section of a VPC in which you can group resources based on security or operational needs. Subnets can be public or private.

Architecture diagram of a VPC with three Amazon EC2 instances in a public subnet and three databases in a private subnet

**Public subnets** contain resources that need to be accessible by the public, such as an online store’s website.

**Private subnets** contain resources that should be accessible only through your private network, such as a database that contains customers’ personal information and order histories.

In a VPC, subnets can communicate with each other. For example, you might have an application that involves Amazon EC2 instances in a public subnet communicating with databases that are located in a private subnet.

**Network access control lists (ACLs)**

A network access control list (ACL) is a virtual firewall that controls inbound and outbound traffic at the subnet level. Network ACLs perform stateless packet filtering. They remember nothing and check packets that cross the subnet border each way: inbound and outbound.

**Domain Name System (DNS)**

Suppose that a business has a website hosted in the AWS Cloud. Customers enter the web address into their browser, and they are able to access the website. This happens because of Domain Name System (DNS) resolution. DNS resolution involves a customer DNS resolver communicating with a company DNS server.

You can think of DNS as being the phone book of the internet. DNS resolution is the process of translating a domain name to an IP address.

**Amazon Route 53**

Amazon Route 53 is a DNS web service. It gives developers and businesses a reliable way to route end users to internet applications hosted in AWS.

Amazon Route 53 connects user requests to infrastructure running in AWS (such as Amazon EC2 instances and load balancers). It can route users to infrastructure outside of AWS.

## Storage

To understand AWS’s storage solutions, we first need to understand what block storage vs object storage.

**Object Storage**

Object storage treats any file as a complete and discrete object, great for documents, images and video files. Think of a YouTube video, and if you want to make a change to that video, you need to reupload the entire file again.

**Block Storage**

Block storage differs from object storage because block storage breaks down the files into small parts or ‘blocks’. Think of if you were editing your copy of that YouTube video and you only wanted to make a change to a small section, the engine would only update the blocks where that section lives.

**File storage**

In file storage, multiple clients (such as users, applications, servers, and so on) can access data that is stored in shared file folders. In this approach, a storage server uses block storage with a local file system to organize files. Clients access data through file paths.

Compared to block storage and object storage, file storage is ideal for use cases in which a large number of services and resources need to access the same data at the same time.

**Instance Stores for Temporary Block Storage**

Block-level storage volumes behave like physical hard drives. An **instance store** provides temporary block-level storage for an Amazon EC2 instance. An instance store is disk storage that is physically attached to the host computer for an EC2 instance, and therefore has the same lifespan as the instance. When the instance is terminated, you lose any data in the instance store.

**Amazon Elastic Block Store (Amazon EBS) – for Block Storage**

Amazon Elastic Block Store (Amazon EBS) is a service that provides block-level storage volumes that you can use with Amazon EC2 instances. If you stop or terminate an Amazon EC2 instance, all the data on the attached EBS volume remains available. **EBS volumes store data on a single Availability Zone**.

Because EBS volumes are for data that needs to persist, it’s important to back up the data. You can take incremental backups of EBS volumes by creating Amazon EBS snapshots.

**Amazon EBS snapshots – for Block Storage**

The first EBS snapshot backs up all source data from the EBS volume. Additional snapshots add or remove data that has changed since the previous snapshot, referred to as incremental backups.

Incremental backups are different from full backups, in which all the data in a storage volume copies each time a backup occurs. The full backup includes data that has not changed since the most recent backup.

**Amazon Simple Storage Service (Amazon S3) – for Object Storage**

Amazon Simple Storage Service (Amazon S3) is a service that provides object-level storage. Amazon S3 stores data as objects in buckets.

You can upload any type of file to Amazon S3, such as images, videos, text files, and so on. For example, you might use Amazon S3 to store backup files, media files for a website, or archived documents. Amazon S3 offers unlimited storage space. The maximum file size for an object in Amazon S3 is 5 TB.

**Amazon S3 Options**

You can choose from a range of storage classes to select a fit for your business and cost needs. When selecting an Amazon S3 storage class, consider these two factors, how often you plan to retrieve your data and how available you need your data to be.

|  |  |
| --- | --- |
| **Amazon S3 Standard** | * Designed for frequently accessed data * Stores data in a minimum of three Availability Zones |
| **Amazon S3 Standard-Infrequent Access**  **(S3 Standard-IA)** | * Ideal for infrequently accessed data * Similar to Amazon S3 Standard but has a lower storage price and higher retrieval price |
| **Amazon S3 One Zone-Infrequent Access**  **(S3 One-Zone-IA)** | * Stores data in a single Availability Zone * Has a lower storage price than Amazon S3 Standard-IA |
| **Amazon Intelligent-Tiering** | * Ideal for data with changing access patterns as it automates which tier your objects are store in based on access requests * Requires a small monthly monitoring and automation fee per object |
| **Amazon S3 Glacier Instant Retrieval** | * Works well for archived data that requires immediate access * Can retrieve objects within a few milliseconds |
| **Amazon S3 Glacier Flexible Retrieval** | * Low-cost storage designed for data archiving * Able to retrieve objects within a few minutes to hours |
| **Amazon S3 Glacier Deep Archive** | * Lowest-cost object storage class ideal for archiving * Able to retrieve objects within 12 hours |
| **Amazon S3 Outposts**  **(An on-premises solution)** | * Creates S3 buckets on Amazon S3 Outposts * Makes it easier to retrieve, store, and access data on AWS Outposts |

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

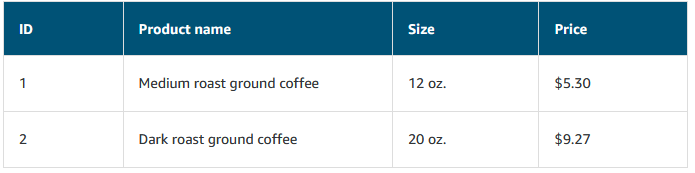
Amazon Elastic File System (Amazon EFS) is a scalable file system used with AWS Cloud services and on-premises resources. As you add and remove files, Amazon EFS grows and shrinks automatically. It can scale on demand to petabytes without disrupting applications. **Amazon EFS file systems store data across multiple Availability Zones.**

## Databases

There are two different types of databases, relational and non-relational.

**Relational Databases**

In a relational database, data is stored in a way that relates it to other pieces of data. Relational databases use structured query language (SQL) to store and query data. This approach allows data to be stored in an easily understandable, consistent, and scalable way.

Example of data in a relational database:

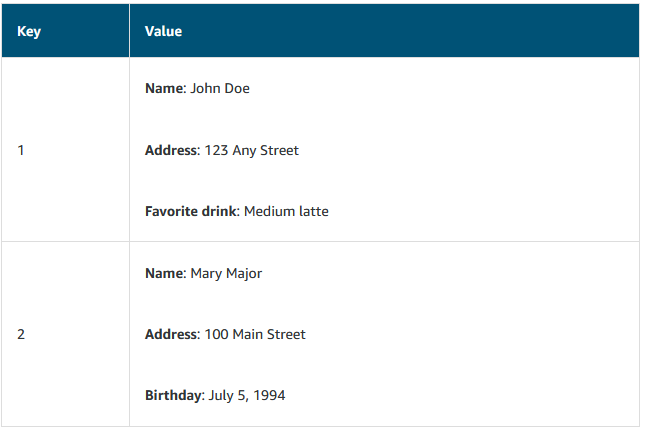
**Non-Relational Database**

In a non-relational database, you create tables. A table is a place where you can store and query data.

Non-relational databases are sometimes referred to as “NoSQL databases” because they use structures (ie key-value database as seen below) other than rows and columns to organize data.

Example of data in a nonrelational database:

**Amazon Relational Database Service**

Amazon Relational Database Service (Amazon RDS) is a service that enables you to run relational databases in the AWS Cloud.

Amazon RDS is a managed service that automates tasks such as hardware provisioning, database setup, patching, and backups. With these capabilities, you can spend less time completing administrative tasks and more time using data to innovate your applications. You can integrate Amazon RDS with other services to fulfill your business and operational needs, such as using AWS Lambda to query your database from a serverless application.

Amazon RDS provides a number of different security options. Many Amazon RDS database engines offer encryption at rest (protecting data while it is stored) and encryption in transit (protecting data while it is being sent and received)

**Amazon Aurora – for an Enterprise Grade Relational Database Service**

Amazon Aurora is an enterprise-class relational database. It is compatible with MySQL and PostgreSQL relational databases. It is up to five times faster than standard MySQL databases and up to three times faster than standard PostgreSQL databases. Consider Amazon Aurora if your workloads require high availability. It replicates six copies of your data across three Availability Zones and continuously backs up your data to Amazon S3.

**Amazon DynamoDB – for a Non-Relational Database**

Amazon DynamoDB is a key-value database service. It delivers single-digit millisecond performance at any scale. DynamoDB is **serverless**, which means that you do not have to provision, patch, or manage servers. DynamoDB is also uses **automatic scaling** as your database shrinks or grows, it adjusts its capacity while maintaining performance.

**Amazon Redshift – Data Warehouse for Data Analytics**

Amazon Redshift is a data warehousing service that you can use for big data analytics. It offers the ability to collect data from many sources and helps you to understand relationships and trends across your data.

**AWS Database Migration Service**

AWS Database Migration Service (AWS DMS) enables you to migrate relational databases, nonrelational databases, and other types of data stores. During the migration, your source database remains operational, reducing downtime for any applications that rely on the database.

## Monitoring and Analytics

**Amazon CloudWatch**

Amazon CloudWatch is a web service that enables you to monitor and manage various metrics and configure alarm actions based on data from those metrics. CloudWatch uses metrics to represent the data points for your resources. AWS services send metrics to CloudWatch. CloudWatch then uses these metrics to create graphs automatically that show how performance has changed over time.

**CloudWatch Alarms**

With CloudWatch, you can create alarms that automatically perform actions if the value of your metric has gone above or below a predefined threshold.

For example, suppose that your company’s developers use Amazon EC2 instances for application development or testing purposes. If the developers occasionally forget to stop the instances, the instances will continue to run and incur charges. In this scenario, you could create a CloudWatch alarm that automatically stops an Amazon EC2 instance when the CPU utilization percentage has remained below a certain threshold for a specified period. When configuring the alarm, you can specify to receive a notification whenever this alarm is triggered.

**AWS CloudTrail**

AWS CloudTrail records API calls for your account. The recorded information includes the identity of the API caller, the time of the API call, the source IP address of the API caller, and more. You can think of CloudTrail as a “trail” of breadcrumbs (or a log of actions) that someone has left behind them.

**AWS Trusted Advisor**

AWS Trusted Advisor is a web service that inspects your AWS environment and provides real-time recommendations in accordance with AWS best practices.

Trusted Advisor compares its findings to AWS best practices in five categories: cost optimization, performance, security, fault tolerance, and service limits. For the checks in each category, Trusted Advisor offers a list of recommended actions and additional resources to learn more about AWS best practices.



## AI and ML

Amazons Artificial Intelligence (AI) and Machine Learning (ML) solutions are not on the exam, and there are many products available including AWS pre-trained AI Services that provide ready-made intelligence.

Amazons AI and ML services include analyzing videos or images, control quality for sensitive data, computer vision to track manufacturing defects on the plant floor, build chatbots, translate text and even forecast business metrics.

The most notable ML service that I want to highlight is Amazon SageMaker. Amazon SageMaker is a cloud machine-learning platform that was launched in November 2017. SageMaker enables developers to create, train, and deploy machine-learning models in the cloud. SageMaker also enables developers to deploy ML models on embedded systems and edge-devices. [Watch this video to learn more](https://www.youtube.com/watch?v=Qv_Tr_BCFCQ).

# Billing, Pricing, and Support

## AWS Free Tier

There are three AWS Free Tiers available that enables you to begin using certain services without having to worry about incurring costs for the specified period.

**Always Free**

These offers do not expire and are available to all AWS customers. For example, AWS Lambda allows 1 million free requests and up to 3.2 million seconds of compute time per month. Amazon DynamoDB allows 25 GB of free storage per month.

**12 Months Free**

These offers are free for 12 months following your initial sign-update to AWS. Examples include specific amounts of Amazon S3 Standard Storage, thresholds for monthly hours of Amazon EC2 compute time, and amounts of Amazon CloudFront data transfer out.

**Trials**

Short-term free trial offers start from the date you activate a particular service, the length of trials vary by product.

## AWS Pricing Concepts

AWS offers a range of cloud computing services with pay-as-you-go pricing:

**Pay for What You Use**

Fir each service, you pay exactly the amount of resources that you actually use, without requiring long-term contracts or complex licensing.

**Pay Less When You Reserve**

Some services offer reservation options that provide a significant discount compared to On-Demand Instance pricing, for example Amazon EC2 Instance Savings Plan would be beneficial to choose for a workload that runs continuously.

**AWS Pricing Calculator**

The AWS Pricing Calculator lets you explore AWS services and create an estimate for the cost of your use cases on AWS. You can organize your AWS estimates by groups that you define. A group can reflect how your company is organized, such as providing estimates by cost center.

## Billing Dashboard

**AWS Billing & Cost Management Dashboard**

Use the AWS Billing & Cost Management dashboard to pay your AWS bill, monitor your usage, and analyze and control your costs.

Compare your current month-to-date balance with the previous month, and get a forecast of the next month based on current usage.

* View month-to-date spend by service.
* View Free Tier usage by service.
* Access Cost Explorer and create budgets.
* Purchase and manage Savings Plans.
* Publish AWS Cost and Usage Reports.

## Consolidated Billing

In an earlier module, you learned about AWS Organizations, a service that enables you to manage multiple AWS accounts from a central location. AWS Organizations also provides the option for **consolidated billing**.

The consolidated billing feature of AWS Organizations enables you to **receive a single bill for all AWS accounts in your organization**. By consolidating, you can easily track the combined costs of all the linked accounts in your organization. **The default maximum number of accounts allowed for an organization is 4, but you can contact AWS Support to increase your quota**, if needed.

On your monthly bill, you **can review itemized charges incurred by each account**. This enables you to have greater transparency into your organization’s accounts while still maintaining the convenience of receiving a single monthly bill.

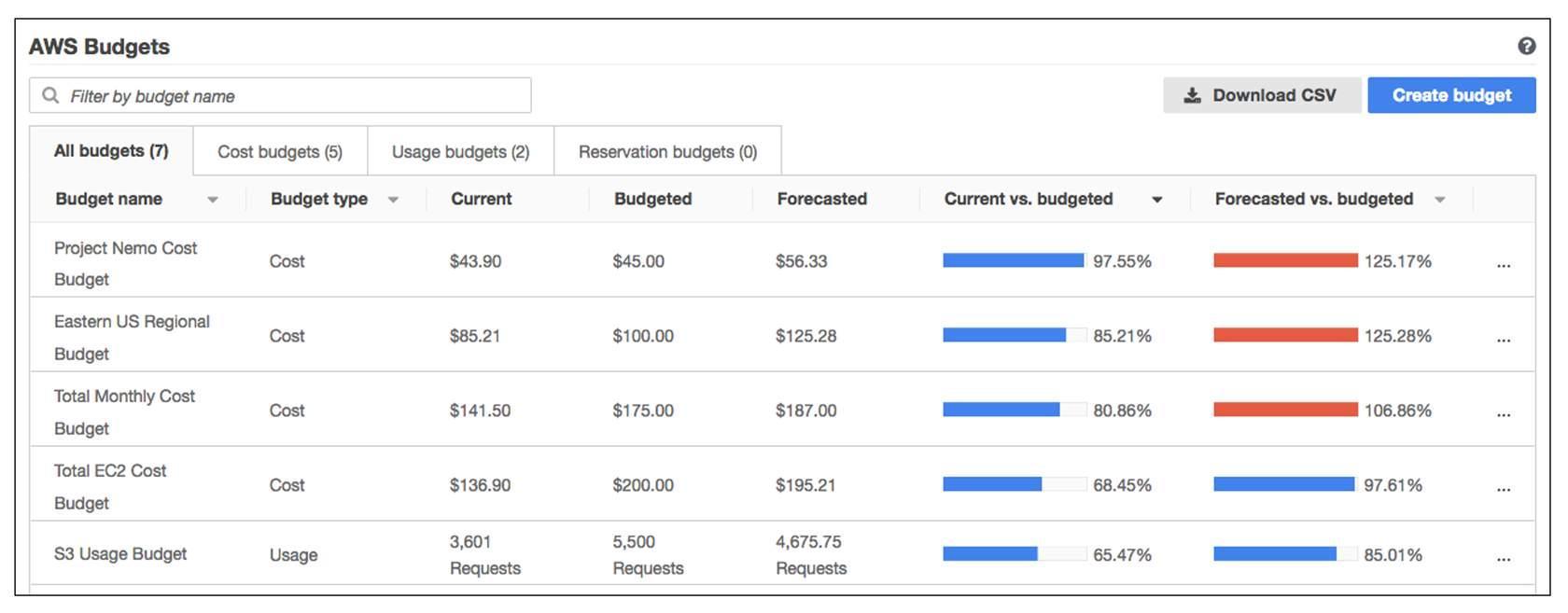
Another benefit of consolidated billing is **the ability to share bulk discount pricing, Savings Plans, and Reserved Instances** across the accounts in your organization. For instance, one account might not have enough monthly usage to qualify for discount pricing. However, when multiple accounts are combined, their aggregated usage may result in a benefit that applies across all accounts in the organization.

## AWS Budgets

In **AWS Budgets**, you can create budgets to plan your service usage, service costs, and instance reservations.

The information in AWS Budgets updates three times a day. This helps you to accurately determine how close your usage is to your budgeted amounts or to the AWS Free Tier limits.

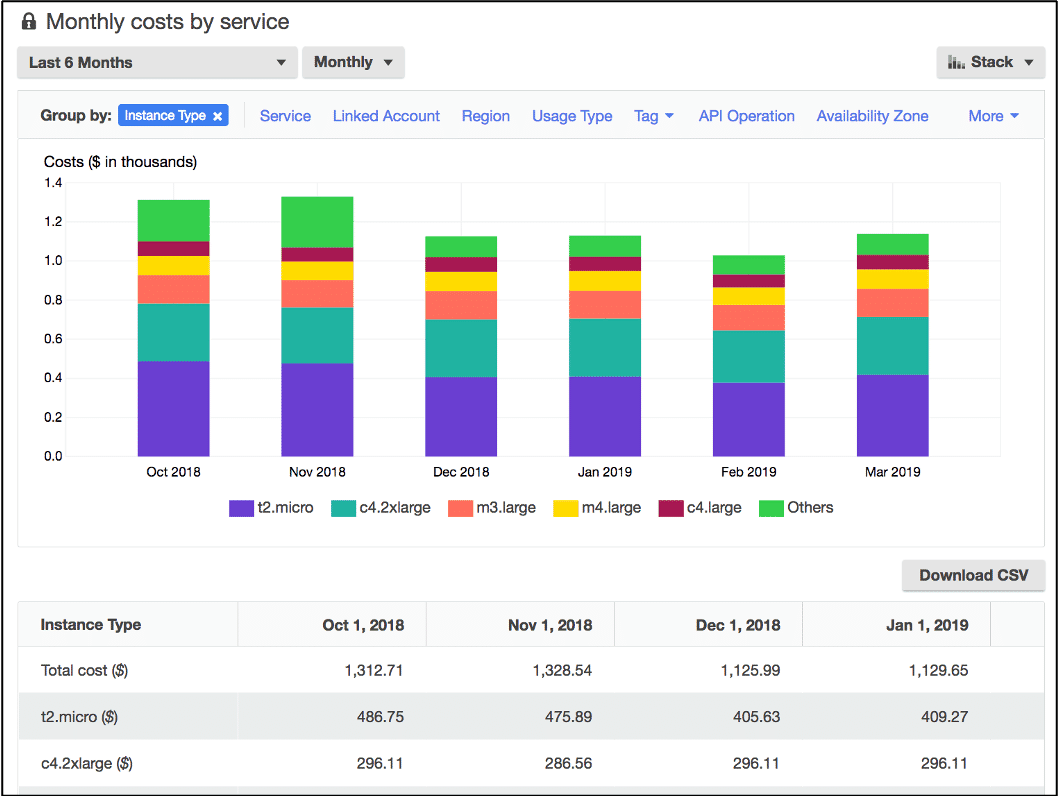
In AWS Budgets, you can also set custom alerts when your usage exceeds (or is forecasted to exceed) the budgeted amount.



## AWS Cost Explorer

**AWS Cost Explorer** is a tool that enables you to visualize, understand, and manage your AWS costs and usage over time.

AWS Cost Explorer includes a default report of the costs and usage for your top five cost-accruing AWS services. You can apply custom filters and groups to analyze your data. For example, you can view resource usage at the hourly level.



## AWS Support Plans

AWS offers four different support plans to help you troubleshoot issues, lower costs, and efficiently use AWS services.

**You can choose from the following 5 support plans to meet your company’s needs:**

* Basic
* Developer
* Business
* Enterprise On-Ramp
* Enterprise

**Basic Support**

Basic Support is free for all AWS customers. It includes access to whitepapers, documentation, and support communities. With Basic Support, you can also contact AWS for billing questions and service limit increases.

With Basic Support, you have access to a limited selection of AWS Trusted Advisor checks. Additionally, you can use the AWS Personal Health Dashboard, a tool that provides alerts and remediation guidance when AWS is experiencing events that may affect you.

**Developer, Business, Enterprise On-Ramp, and Enterprise Support**

The Developer, Business, Enterprise On-Ramp, and Enterprise Support plans include all the benefits of Basic Support, in addition to the ability to open an unrestricted number of technical support cases. These Support plans have pay-by-the-month pricing and require no long-term contracts.

|  |  |
| --- | --- |
| **Developer Support**  Customers in the Developer Support plan have access to features such as:   * Best practice guidance * Client-side diagnostic tools * Building-block architecture support, which consists of guidance for how to use AWS offerings, features, and services together   For example, suppose that your company is exploring AWS services. You’ve heard about a few different AWS services. However, you’re unsure of how to potentially use them together to build applications that can address your company’s needs. In this scenario, the building-block architecture support that is included with the Developer Support plan could help you to identify opportunities for combining specific services and features. | **Business Support**  Customers with a Business Support plan have access to additional features, including:   * Use-case guidance to identify AWS offerings, features, and services that can best support your specific needs * All AWS Trusted Advisor checks * Limited support for third-party software, such as common operating systems and application stack components   Suppose that your company has the Business Support plan and wants to install a common third-party operating system onto your Amazon EC2 instances. You could contact AWS Support for assistance with installing, configuring, and troubleshooting the operating system. For advanced topics such as optimizing performance, using custom scripts, or resolving security issues, you may need to contact the third-party software provider directly. |
| **Enterprise Support On-Ramp Support**  In November 2021, AWS opened enrollment into AWS Enterprise On-Ramp Support plan. In addition to all the features included in the Basic, Developer, and Business Support plans, customers with an Enterprise On-Ramp Support plan have access to:   * A pool of Technical Account Managers to provide proactive guidance and coordinate access to programs and AWS experts * A Cost Optimization workshop (one per year) * A Concierge support team for billing and account assistance * Tools to monitor costs and performance through Trusted Advisor and Health API/Dashboard   Enterprise On-Ramp Support plan also provides access to a specific set of proactive support services, which are provided by a pool of Technical Account Managers.   * Consultative review and architecture guidance (one per year) * Infrastructure Event Management support (one per year) * Support automation workflows | **Enterprise Support**  In addition to all features included in the Basic, Developer, Business, and Enterprise On-Ramp support plans, customers with Enterprise Support have access to:   * A designated Technical Account Manager to provide proactive guidance and coordinate access to programs and AWS experts * A Concierge support team for billing and account assistance * Operations Reviews and tools to monitor health * Training and Game Days to drive innovation * Tools to monitor costs and performance through Trusted Advisor and Health API/Dashboard   The Enterprise plan also provides full access to proactive services, which are provided by a designated Technical Account Manager:   * Consultative review and architecture guidance * Infrastructure Event Management support * Cost Optimization Workshop and tools * Support automation workflows * 15 minutes or less response time for business-critical issues |

**Technical Account Manager (TAM)**

The Enterprise On-Ramp and Enterprise Support plans include access to a Technical Account Manager (TAM).

The TAM is your primary point of contact at AWS. If your company subscribes to Enterprise Support or Enterprise On-Ramp, your TAM educates, empowers, and evolves your cloud journey across the full range of AWS services. TAMs provide expert engineering guidance, help you design solutions that efficiently integrate AWS services, assist with cost-effective and resilient architectures, and provide direct access to AWS programs and a broad community of experts.

For example, suppose that you are interested in developing an application that uses several AWS services together. Your TAM could provide insights into how to best use the services together. They achieve this, while aligning with the specific needs that your company is hoping to address through the new application.

## AWS Marketplace

AWS Marketplace is a digital catalog that includes thousands of software listings from independent software vendors. You can use AWS Marketplace to find, test, and buy software that runs on AWS.

For each listing in AWS Marketplace, you can access detailed information on pricing options, available support, and reviews from other AWS customers.