Module 1: Cloud Concepts Overview

Section 1: Introduction to cloud computing

After completing this module, you should be able to:
• Define different types of cloud computing
• Describe six advantages of cloud computing
• Recognize the main AWS service categories and core services
• Review the AWS Cloud Adoption Framework (AWS CAF)

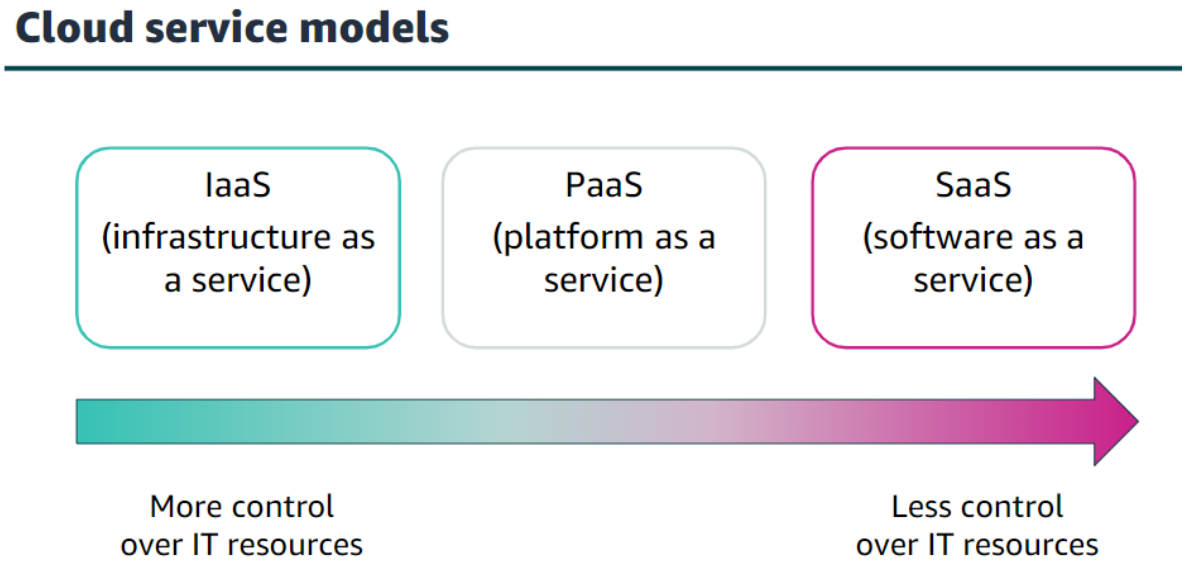
Cloud computing is the **on-demand** delivery of compute power, database, storage, applications, and other IT resources **via the internet** with **pay-as-you-go** pricing.

Cloud computing enables you to stop thinking of your infrastructure as hardware, and instead think of (and use) it as software.

With a hardware solution, you must ask if there is enough resource capacity or sufficient storage to meet your needs, and you provision capacity by guessing theoretical maximum peaks. If you don't meet your projected maximum peak, then you pay for expensive resources that stay idle. If you exceed your projected maximum peak, then you don't have sufficient capacity to meet your needs. And if your needs change, then you must spend the time, effort, and money required to implement a new solution.

- Software solutions:
  - Are flexible
  - Can change more quickly, easily, and cost-effectively than hardware solutions
  - Eliminate the undifferentiated heavy-lifting tasks

Cloud computing helps developers and IT departments avoid undifferentiated work like procurement, maintenance, and capacity planning, thus enabling them to focus on what matters most.

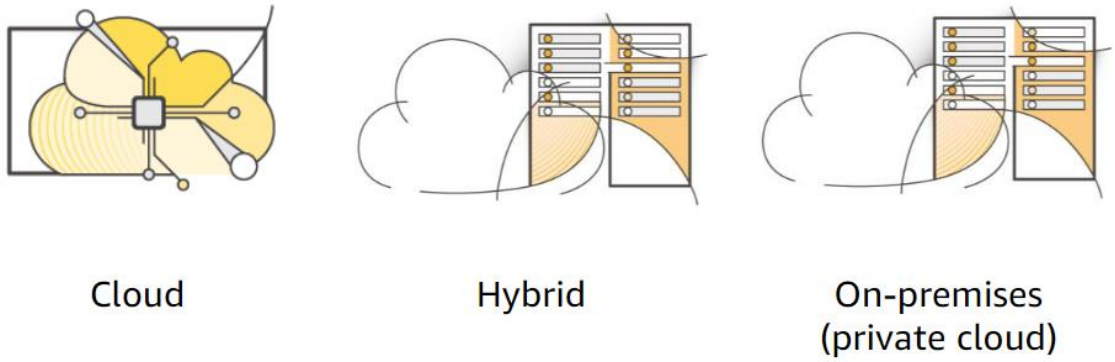


- Infrastructure as a service (IaaS): Services in this category are the basic building blocks for cloud IT and typically provide you with access to networking features, computers (virtual or on dedicated hardware), and data storage space.

- Platform as a service (PaaS): Services in this category reduce the need for you to manage the underlying infrastructure (usually hardware and operating systems) and enable you to focus on the deployment and management of your applications.

- Software as a service (SaaS): Services in this category provide you with a completed product that the service provider runs and manages. In most cases, software as a service refers to end-user applications. With a SaaS offering, you do not have to think about how the service is maintained or how the underlying infrastructure is managed

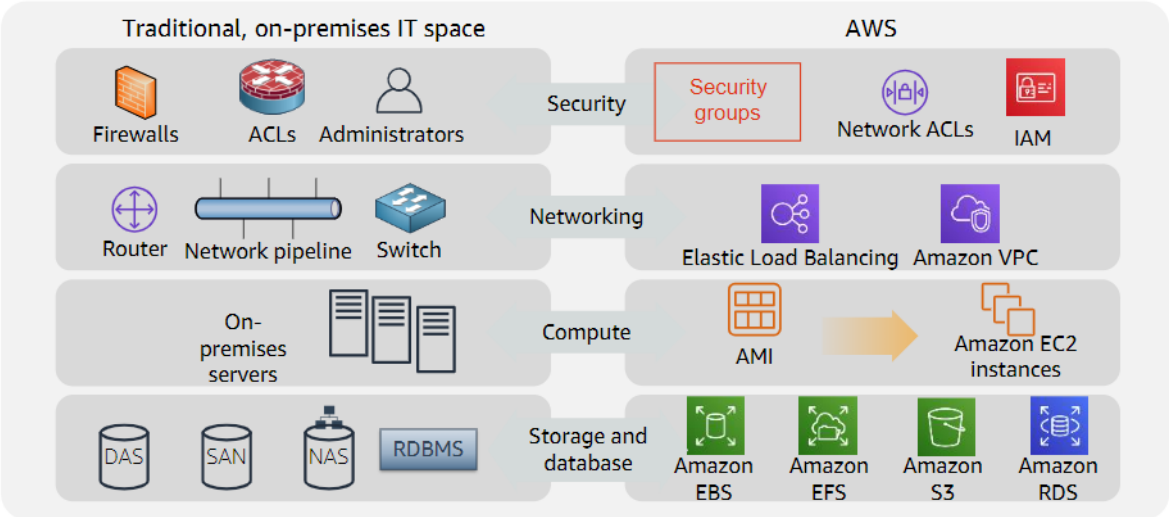
Cloud computing deployment models



Cloud | Hybrid | On-premises (private cloud)

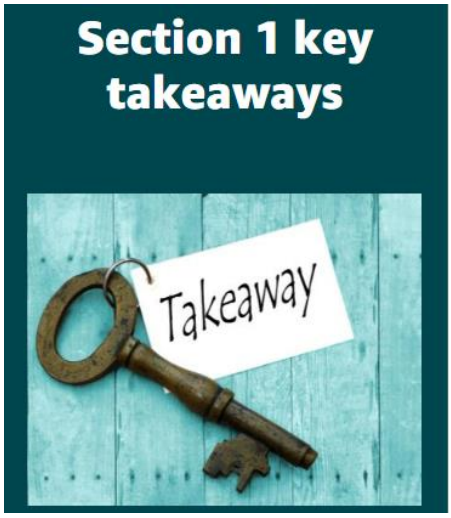
- Cloud: A cloud-based application is fully deployed in the cloud, and all parts of the application run in the cloud. Applications in the cloud have either been created in the cloud or have been migrated from an existing infrastructure to take advantage of the benefits of cloud computing
- Hybrid: The most common method of hybrid deployment is between the cloud and existing on-premises infrastructure. This model enables an organization to extend and grow their infrastructure into the cloud while connecting cloud resources to internal systems.
- On-premises: Deploying resources on-premises, using virtualization and resource management tools, is sometimes called private cloud. While on-premises deployment does not provide many of the benefits of cloud computing, it is sometimes sought for its ability to provide dedicated resources.

Similarities between AWS and traditional IT



There are many similarities between AWS and the traditional, on-premises IT space:

- AWS security groups, network access control lists (network ACLs), and AWS Identity and Access Management (IAM) are similar to firewalls, access control lists (ACLs), and administrators.
- Elastic Load Balancing and Amazon Virtual Private Cloud (Amazon VPC) are similar to routers, network pipelines, and switches.
- Amazon Machine Images (AMIs) and Amazon Elastic Compute Cloud (Amazon EC2) instances are similar to on-premises servers.
- Amazon Elastic Block Store (Amazon EBS), Amazon Elastic File System (Amazon EFS), Amazon Simple Storage Service (Amazon S3), and Amazon Relational Database Service (Amazon RDS) are similar to direct attached storage (DAS), storage area networks (SAN), network attached storage (NAS), and a relational database management service (RDBMS).



- Cloud computing is the on-demand delivery of IT resources via the internet with pay-as-you-go pricing.
- Cloud computing enables you to think of (and use) your infrastructure as software.
- There are three cloud service models: IaaS, PaaS, and SaaS.
- There are three cloud deployment models: cloud, hybrid, and on-premises or private cloud.
- Almost anything you can implement with traditional IT can also be implemented as an AWS cloud computing service.

Section 2: Advantages of cloud computing

Advantage #1—Trade capital expense for variable expense: Capital expenses (capex) are funds that a company uses to acquire, upgrade, and maintain physical assets such as property, industrial buildings, or equipment.

By contrast, a variable expense is an expense that the person who bears the cost can easily alter or avoid. Instead of investing heavily in data centers and servers before you know how you will use them, you can pay only when you consume resources and pay only for the amount you consume. Thus, you save money on technology.

Advantage #2—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 is aggregated in the cloud, providers such as AWS can achieve higher economies of scale, which translates into lower pay-as-you-go prices.

Advantage #3—Stop guessing capacity: Eliminate guessing about your infrastructure capacity needs. When you make a capacity decision before you deploy an application, you often either have expensive idle resources or deal with limited capacity. With cloud computing, these problems go away.

Advantage #4—Increase speed and agility: In a cloud computing environment, new IT resources are only a click away, which means that you reduce the time it takes to make those resources available to your developers from weeks to just minutes.

Advantage #5—Stop spending money on running and maintaining data centres: Focus on projects that differentiate your business instead of focusing on the infrastructure.

Advantage #6—Go global in minutes: You can deploy your application in multiple AWS Regions around the world with just a few clicks. As a result, you can provide a lower latency and better experience for your customers simply and at minimal cost

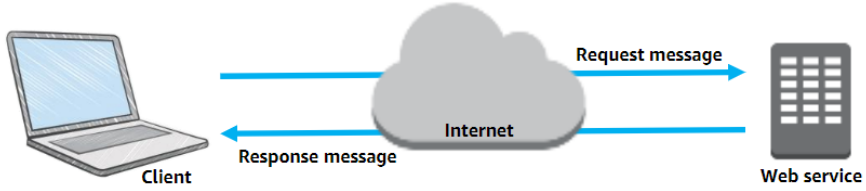
Section 2 key takeaways

- Trade capital expense for variable expense
- Benefit from massive economies of scale
- Stop guessing capacity
- Increase speed and agility
- Stop spending money on running and maintaining data centers
- Go global in minutes

Section 3: Introduction to Amazon Web Services (AWS)

What are web services?

A **web service** is any piece of software that makes itself available over the internet and uses a **standardized format**—such as Extensible Markup Language (XML) or JavaScript Object Notation (JSON)—for the request and the response of an **application programming interface (API) interaction**.



What is AWS?

- AWS is a **secure cloud platform** that offers a **broad set of global cloud-based products**.
- AWS provides you with **on-demand access** to compute, storage, network, database, and other IT resources and management tools.
- AWS offers **flexibility**.
- You **pay only for the individual services you need**, for **as long as you use them**.
- AWS services **work together** like building blocks.

- Amazon EC2 (https://aws.amazon.com/ec2/): You want complete control over your AWS computing resources.
- AWS Lambda (https://aws.amazon.com/lambda/): You want to run your code and not manage or provision servers.
- AWS Elastic Beanstalk (https://aws.amazon.com/elasticbeanstalk/): You want a service that deploys, manages, and scales your web applications for you.
- Amazon Lightsail (https://aws.amazon.com/lightsail/): You need a lightweight cloud platform for a simple web application.
- AWS Batch (https://aws.amazon.com/batch/): You need to run hundreds of thousands of batch workloads.
- AWS Outposts (https://aws.amazon.com/outposts/): You want to run AWS infrastructure in your on-premises data center.
- Amazon Elastic Container Service (Amazon ECS) (https://aws.amazon.com/ecs/)
- Amazon Elastic Kubernetes Service (Amazon EKS) (https://aws.amazon.com/eks/)
- AWS Fargate (https://aws.amazon.com/fargate/): You want to implement a containers or microservices architecture.
- VMware Cloud on AWS (https://aws.amazon.com/vmware/): You have an on-premises server virtualization platform that you want to migrate to AWS.

There are three ways to create and manage resources on the AWS Cloud:
•**AWS Management Console**: The console provides a rich graphical interface to a majority of the features offered by AWS.
•**AWS Command Line Interface (AWS CLI):** The AWS CLI provides a suite of utilities that can be launched from a command script in Linux, macOS, or Microsoft Windows.
•**Software development kits (SDKs):** AWS provides packages that enable accessing AWS in a variety of popular programming languages.

- AWS is a secure cloud platform that offers a broad set of global cloud-based products called services that are designed to work together.
- There are many categories of AWS services, and each category has many services to choose from.
- Choose a service based on your business goals and technology requirements.
- There are three ways to interact with AWS services.

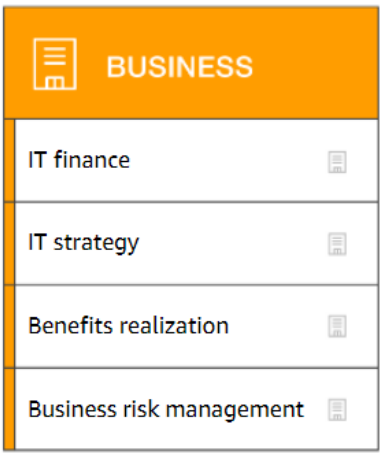
Section 4: Moving to the AWS Cloud –The AWS Cloud Adoption Framework (AWS CAF)

AWS Cloud Adoption Framework (AWS CAF)

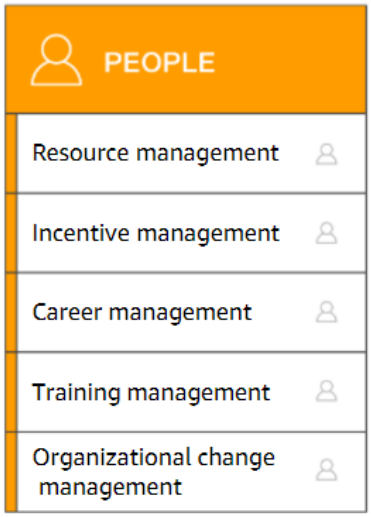
AWS CAF perspectives

- **AWS CAF provides guidance and best practices to help organizations** build a comprehensive approach to cloud computing across the organization and throughout the IT lifecycle to **accelerate successful cloud adoption**.
- AWS CAF is organized into **six *perspectives***.
- Perspectives consist of sets of ***capabilities***.

Business V/S Technical

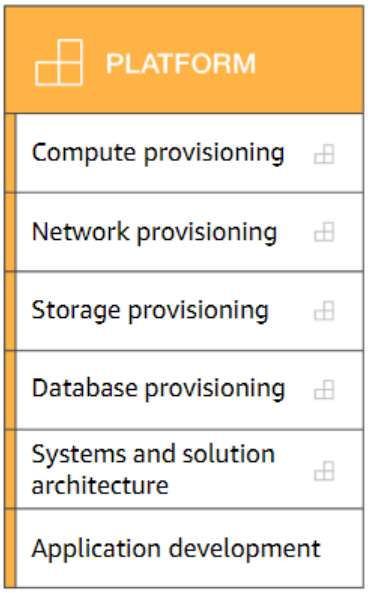


Business perspective capabilities

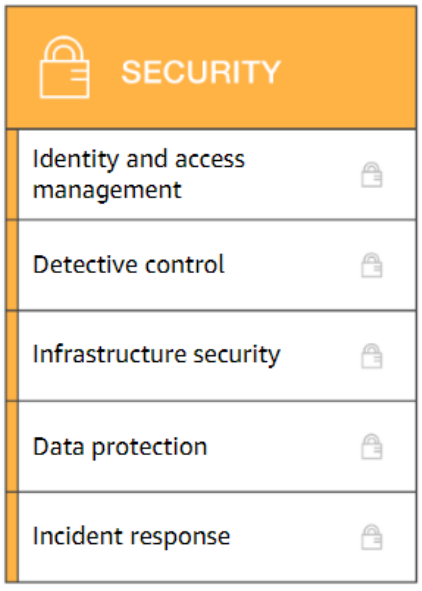


People perspective capabilities

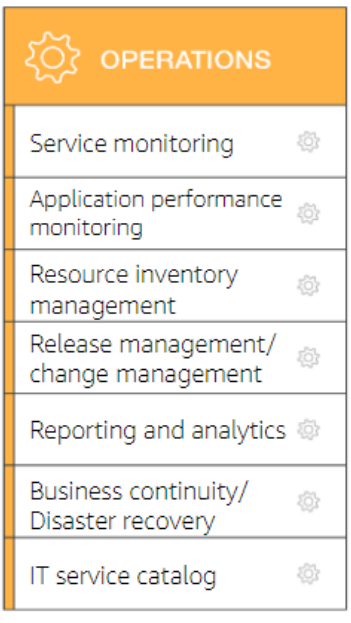
Governance perspective capabilities



Platform perspective capabilities



Security perspective capabilities



Operations perspective capabilities

Module 2: Cloud Economics and Billing

Section 1: Fundamentals of pricing

AWS pricing model

Three fundamental drivers of cost with AWS

Compute
- Charged per hour/second*
- Varies by instance type

*Linux only

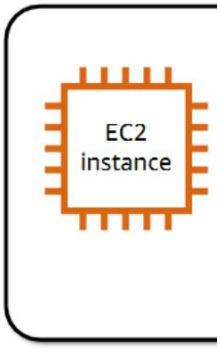
Storage
- Charged typically per GB

Data transfer
- Outbound is aggregated and charged
- Inbound has no charge (with some exceptions)
- Charged typically per GB

AWS offers a range of cloud computing services. For each service, you pay for exactly the amount of resources that you actually need. This utility-style pricing model includes:
•Pay for what you use
•Pay less when you reserve
•Pay less when you use more
•Pay even less as AWS grows

Pay less when you reserve

Invest in Reserved Instances (RIs):
- Save up to 75 percent
- Options:
  - All Upfront Reserved Instance (AURI) → **largest discount**
  - Partial Upfront Reserved Instance (PURI) → **lower discounts**
  - No Upfront Payments Reserved Instance (NURI) → **smaller discount**



Pay less by using more

Realize volume-based discounts:
- **Savings** as usage increases.
- **Tiered pricing** for services like Amazon Simple Storage Service (Amazon S3), Amazon Elastic Block Store (Amazon EBS), or Amazon Elastic File System (Amazon EFS) → the more you use, the less you pay per GB.
- Multiple storage services deliver **lower** storage costs based on needs.

Pay even less as AWS grows

As AWS grows:

- AWS focuses on lowering cost of doing business.
- This practice results in AWS passing savings from economies of scale to you.
- Since 2006, AWS has **lowered pricing 75** times (as of September 2019).
- Future higher-performing resources replace current resources for no extra charge.

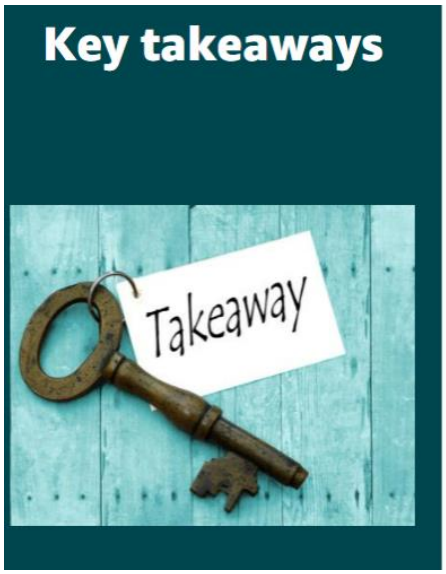
AWS Free Tier

Enables you to gain free hands-on experience with the AWS platform, products, and services. Free for 1 year for new customers.

•Amazon Virtual Private Cloud (Amazon VPC) enables you to provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define.
•AWS Identity and Access Management (IAM) controls your users' access to AWS services and resources.
•Consolidated Billing is a billing feature in AWS Organizations to consolidate payment for multiple AWS accounts or multiple Amazon Internet Services Private Limited (AISPL) accounts*.

Consolidated billing provides:
•One bill for multiple accounts.
•The ability to easily track each account's charges.
•The opportunity to decrease charges as a result of volume pricing discounts from combined usage.
•And you can consolidate all of your accounts using Consolidated Billing and get tiered benefits.

•AWS Elastic Beanstalk is an even easier way for you to quickly deploy and manage applications in the AWS Cloud.
• AWS CloudFormation gives developers and systems administrators an easy way to create a collection of related AWS resources and provision them in an orderly and predictable fashion.
•Automatic Scaling automatically adds or removes resources according to conditions you define. The resources you are using increase seamlessly during demand spikes to maintain performance and decrease automatically during demand lulls to minimize costs.
•AWS OpsWorks is an application management service that makes it easy to deploy and operate applications of all shapes and sizes.



- There is no charge (with some exceptions) for:
  - Inbound data transfer.
  - Data transfer between services within the same AWS Region.
- Pay for what you use.
- Start and stop anytime.
- No long-term contracts are required.
- Some services are free, but the other AWS services that they provision might not be free.

Section 2: Total Cost of Ownership

Total Cost of Ownership (TCO) is the financial estimate to help identify direct and indirect costs of a system

Some of the costs that are associated with data center management include:
•Server costs for both hardware and software, and facilities costs to house the equipment.
•Storage costs for the hardware, administration, and facilities.
•Network costs for hardware, administration, and facilities.
•And IT labour costs that are required to administer the entire solution.

Section 3: AWS Organizations

AWS Organizations is a free account management service that enables you to consolidate multiple AWS accounts into an organization that you create and centrally manage.

Limits		
Limits on Names	Names must be composed of Unicode characters.	
	Names must not exceed 250 characters in length.	
Maximum and Minimum Values	Number of AWS accounts	Varies. Note: An invitation sent to an account counts against this limit.
	Number of roots	1
	Number of OUs	1,000
	Number of policies	1,000
	Maximum size of a service control policy document	5,120 bytes
	Maximum nesting of OUs in a root	5 levels of OUs under a root
	Invitations sent per day	20
	Number of member accounts you can create concurrently	Only five can be in progress at one time
	Number of entities to which you can attach a policy	Unlimited

AWS Organizations can be managed through different interfaces.

The AWS Management Console is a browser-based interface that you can use to manage your organization and your AWS resources. You can perform any task in your organization by using the console.

AWS Command Line Interface(AWS CLI) tools enable you to issue commands at your system's command line to perform AWS Organizations tasks and AWS tasks. This method can be faster and more convenient than using the console.

You can use also AWS software development kits (SDKs) to handle tasks such as cryptographically signing requests, managing errors, and retrying requests automatically. AWS SDKs consist of libraries and sample code for various programming languages and platforms, such as Java, Python, Ruby, .NET, iOS, and Android.

The AWS Organizations HTTPS Query API gives you programmatic access to AWS Organizations and AWS. You can use the API to issue HTTPS requests directly to the service. When you use the HTTPS API, you must include code to digitally sign requests by using your credentials.

Section 4: AWS Billing and Cost Management

AWS Billing and Cost Management is the service that you use to pay your AWS bill, monitor your usage, and budget your costs.
The AWS Cost and Usage Report Tool enables you to identify opportunities for optimization by understanding your cost and usage data trends and how you are using your AWS implementation.
The AWS Billing Dashboard lets you view the status of your month-to-date AWS expenditure, identify the services that account for the majority of your overall expenditure, and understand at a high level how costs are trending.
The AWS Bills page lists the costs that you incurred over the past month for each AWS service, with a further breakdown by AWS Region and linked account.
The AWS Billing and Cost Management console includes the Cost Explorer page for viewing your AWS cost data as a graph.
AWS Budgets uses the cost visualization that is provided by Cost Explorer to show you the status of your budgets and to provide forecasts of your estimated costs.
Budget alerts can be sent via email or via Amazon Simple Notification Service (Amazon SNS).

Section 5: Technical support

AWS support (1 of 2)

- Provide unique combination of tools and expertise:
  - AWS Support
  - AWS Support Plans
- Support is provided for:
  - Experimenting with AWS
  - Production use of AWS
  - Business-critical use of AWS

AWS support (2 of 2)

- Proactive guidance :
  - Technical Account Manager (TAM)
- Best practices :
  - AWS Trusted Advisor
- Account assistance :
  - AWS Support Concierge

The Basic Support Plan offers:
•24/7 access to customer service, documentation, whitepapers and support forums.
•Access to six core Trusted Advisor checks.
•Access to Personal Health Dashboard.

•The Developer Support Plan offers resources for customers that are testing or doing early development on AWS, and any customers who:
•Want access to guidance and technical support.
•Are exploring how to quickly put AWS to work.
•Use AWS for non-production workloads or applications.

The Business Support Plan offers resources for customers that are running production workloads on AWS and any customers who:
•Run one or more applications in production environments.
•Have multiple services activated, or use key services extensively.
•Depend on their business solutions to be available, scalable, and secure.

There are five different severity levels:

•Critical–Your business is at risk. Critical functions of your application are unavailable.
•Urgent–Your business is significantly impacted. Important functions of your application are unavailable.
•High–Important functions of your application are impaired or degraded.
•Normal–Non-critical functions of your application are behaving abnormally, or you have a time-sensitive development question.
•Low–You have a general development question, or you want to request a feature.

Module 3: AWS Global Infrastructure Overview

Section 1: AWS Global Infrastructure

The AWS Cloud infrastructure is built around Regions.
AWS has 22 Regions worldwide. An AWS Region is a physical geographical location with one or more Availability Zones. Availability Zones in turn consist of one or more data centers.
To achieve fault tolerance and stability, Regions are isolated from one another. Resources in one Region are not automatically replicated to other Regions. When you store data in a specific Region, it is not replicated outside that Region. It is your responsibility to replicate data across Regions, if your business needs require it.
AWS Regions that were introduced before March 20, 2019 are enabled by default. Regions that were introduced after March 20, 2019—such as Asia Pacific (Hong Kong) and Middle East (Bahrain)—are disabled by default. You must enable these Regions before you can use them. You can use the AWS Management Console to enable or disable a Region. Some Regions have restricted access. An Amazon AWS (China) account provides access to the Beijing and Ningxia Regions only.
To learn more about AWS in China, see: https://www.amazonaws.cn/en/about-aws/china/. The isolated AWS GovCloud (US) Region is designed to allow US government agencies and customers to move sensitive workloads into the cloud by addressing their specific regulatory and compliance requirements.
For accessibility: Snapshot from the infrastructure. AWS website that shows a picture of downtown London including the Tower Bridge and the Shard. It notes that there are three Availability Zones in the London region. End of accessibility description.

Data centers are securely designed with several factors in mind:

Each location is carefully evaluated to **mitigate environmental risk.**

- Data centers have a **redundant design** that anticipates and tolerates failure while maintaining service levels.
- To ensure availability, **critical system components are backed up** across multiple Availability Zones.
- To ensure capacity, AWS continuously monitors service usage to deploy infrastructure to support availability commitments and requirements.
- Data center **locations are not disclosed** and all access to them is restricted.
- In case of failure, automated processes move data traffic away from the affected area.

AWS uses **custom network equipment** sourced from **multiple original device manufacturers (ODMs).** ODMs design and manufacture products based on specifications from a second company. The second company then rebrands the products for sale.

Amazon CloudFront is **a content delivery network (CDN)** used to distribute content to end users to reduce latency. **Amazon Route 53** is a Domain Name System (DNS) service. Requests going to either one of these services will be routed to the nearest **edge location** automatically in order to lower latency.

AWS Points of Presence are located in most of the major cities around the world. By **continuously measuring internet connectivity, performance and computing to find the best way to route requests,** the Points of Presence deliver a better near real-time user experience. They are used by many AWS services, including Amazon CloudFront, Amazon Route 53, AWS Shield, and AWS Web Application Firewall (AWS WAF) services.

Regional edge caches are used by default with Amazon CloudFront. Regional edge caches are used when you have content that is not accessed frequently enough to remain in an **edge location.** Regional edge caches absorb this content and provide an alternative to that content having to be fetched from the origin server.

Now that you have a good understanding of the major components that comprise the AWS Global Infrastructure, let's consider the benefits provided by this infrastructure.

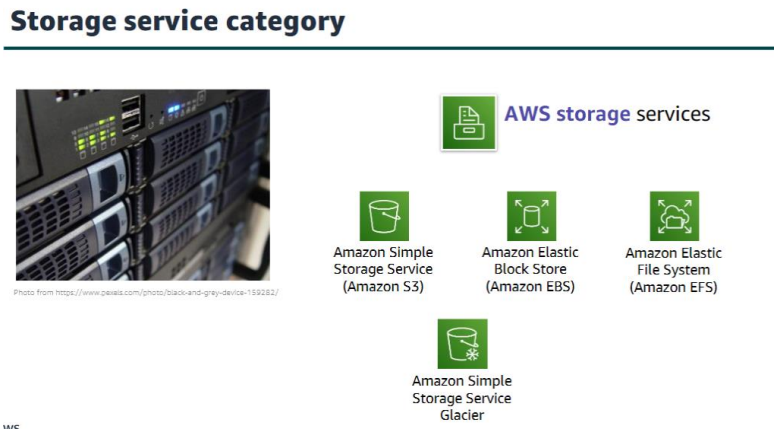
The AWS Global Infrastructure has several valuable features:

- First, it is **elastic** and **scalable.** This means resources can dynamically adjust to increases or decreases in capacity requirements. It can also rapidly adjust to accommodate growth.
- Second, this infrastructure is **fault tolerant,** which means it has built-in component redundancy which enables it to continue operations despite a failed component.
- Finally, it requires minimal to no human intervention, while providing **high availability** with minimal down time.

Some key takeaways from this section of the module include:

- The AWS Global Infrastructure consists of Regions and Availability Zones.
- Your choice of a Region is typically based on compliance requirements or to reduce latency.
- Each Availability Zone is physically separate from other Availability Zones and has redundant power, networking, and connectivity.
- Edge locations, and Regional edge caches improve performance by caching content closer to users.

Section 2: AWS services and service category overview



Amazon Simple Storage Service (Amazon S3) is an object storage service that offers scalability, data availability, security, and performance. Use it to store and protect any amount of data for websites, mobile apps, backup and restore, archive, enterprise applications, Internet of Things (IoT) devices, and big data analytics.

Amazon Elastic Block Store (Amazon EBS) is high-performance block storage that is designed for use with Amazon EC2 for both throughput and transaction-intensive workloads. It is used for a broad range of workloads, such as relational and non-relational databases, enterprise applications, containerized applications, big data analytics engines, file systems, and media workflows.

Amazon Elastic File System (Amazon EFS) provides a scalable, fully managed elastic Network File System (NFS) file system for use with AWS Cloud services and on-premises resources. It is built to scale on demand to petabytes, growing and shrinking automatically as you add and remove files. It reduces the need to provision and manage capacity to accommodate growth.

Amazon Simple Storage Service Glacier is a secure, durable, and extremely low-cost Amazon S3 cloud storage class for data archiving and long-term backup. It is designed to deliver 11 9s of durability and to provide comprehensive security and compliance capabilities to meet stringent regulatory requirements.

Compute service category

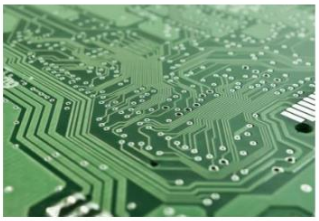


Photo from https://www.pexels.com/photo/technology-computer-lines-board-50711/

AWS compute services include the services listed here, and many others.

Amazon Elastic Compute Cloud (Amazon EC2) provides resizable compute capacity as virtual machines in the cloud. Amazon EC2 Auto Scaling enables you to automatically add or remove EC2 instances according to conditions that you define.

Amazon Elastic Container Service (Amazon ECS) is a highly scalable, high-performance container orchestration service that supports Docker containers.

Amazon Elastic Container Registry (Amazon ECR) is a fully-managed Docker container registry that makes it easy for developers to store, manage, and deploy Docker container images.

AWS Elastic Beanstalk is a service for deploying and scaling web applications and services on familiar servers such as Apache and Microsoft Internet Information Services (IIS).

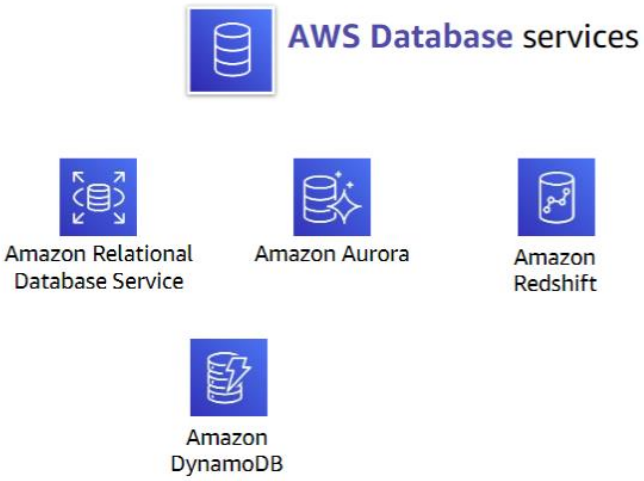
AWS Lambda enables you to run code without provisioning or managing servers. You pay only for the compute time that you consume. There is no charge when your code is not running.

Amazon Elastic Kubernetes Service (Amazon EKS) makes it easy to deploy, manage, and scale containerized applications that use Kubernetes on AWS.

AWS Fargate is a compute engine for Amazon ECS that allows you to run containers without having to manage servers or clusters.

Database service category

Photo from https://aws.amazon.com/compliance/data-center/data-centers/



AWS database services include the services listed here, and many others.

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud.
It provides resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching, and backups.

Amazon Aurora is a MySQL and PostgreSQL-compatible relational database. It is up to five times faster than standard MySQLdatabases and three times faster than standard PostgreSQL databases.

Amazon Redshift enables you to run analytic queries against petabytes of data that is stored locally in Amazon Redshift, and directly against exabytes of data that are stored in Amazon S3. It delivers fast performance at any scale.

Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale, with built-in security, backup and restore, and in-memory caching

Networking and content delivery service category

Amazon Virtual Private Cloud (Amazon VPC) enables you to provision logically isolated sections of the AWS Cloud.

Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, and Lambda functions.

Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and application programming interfaces (APIs) to customers globally, with low latency and high transfer speeds.

AWS Transit Gateway is a service that enables customers to connect their Amazon Virtual Private Clouds (VPCs) and their on-premises networks to a single gateway.

Amazon Route 53 is a scalable cloud Domain Name System (DNS) web service designed to give you a reliable way to route end users to internet applications. It translates names (like www.example.com) into the numeric IP addresses (like 192.0.2.1) that computers use to connect to each other.

AWS Direct Connect provides a way to establish a dedicated private network connection from your data center or office to AWS, which can reduce network costs and increase bandwidth throughput.

AWS VPN provides a secure private tunnel from your network or device to the AWS global network.

Security, identity, and compliance service category

Photo by Paweł Czerwiński on Unsplash

aws

AWS Identity and Access Management (IAM) enables you to manage access to AWS services and resources securely. By using IAM, you can create and manage AWS users and groups. You can use IAM permissions to allow and deny user and group access to AWS resources.

AWS Organizations allows you to restrict what services and actions are allowed in your accounts.

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

AWS Artifact provides on-demand access to AWS security and compliance reports and select online agreements.

AWS Key Management Service (AWS KMS) enables you to create and manage keys. You can use AWS KMS to control the use of encryption across a wide range of AWS services and in your applications.

AWS Shield is a managed Distributed Denial of Service (DDoS) protection service that safeguards applications running on AWS.

AWS cost management service category

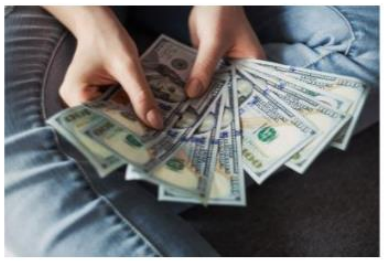
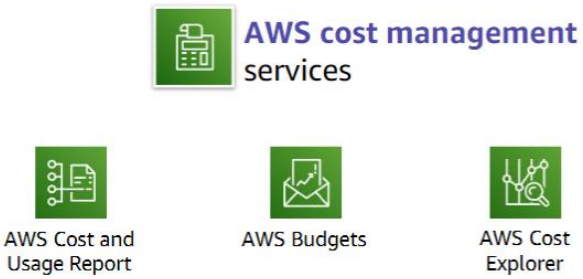


Photo by Alexander Mils on Unsplash



The AWS Cost and Usage Report contains the most comprehensive set of AWS cost and usage data available, including additional metadata about AWS services, pricing, and reservations.

AWS Budgets enables you to set custom budgets that alert you when your costs or usage exceed (or are forecasted to exceed) your budgeted amount.

AWS Cost Explorer has an easy-to-use interface that enables you to visualize, understand, and manage your AWS costs and usage over time.

Photo by Marta Branco from Pexels

The AWS Management Console provides a web-based user interface for accessing your AWS account.

AWS Config provides a service that helps you track resource inventory and changes.

Amazon CloudWatch allows you to monitor resources and applications.

AWS Auto Scaling provides features that allow you to scale multiple resources to meet demand.

AWS Command Line Interface provides a unified tool to manage AWS services.

AWS Trusted Advisor helps you optimize performance and security.

AWS Well-Architected Tool provides help in reviewing and improving your workloads.

AWS CloudTrail tracks user activity and API usage.

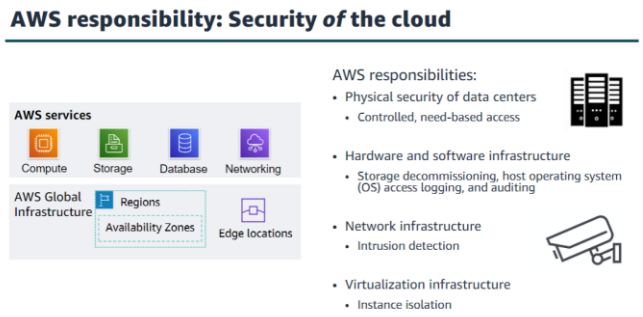
Module 4: AWS Cloud Security

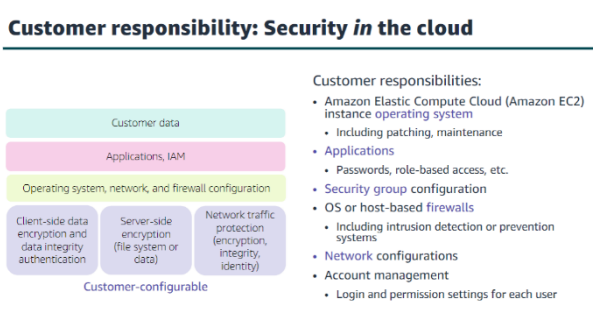
Section 1: AWS shared responsibility model

The differentiation of who is responsible for what is commonly referred to as security “of” the cloud versus security “in” the cloud.

AWS operates, manages, and controls the components from the software virtualization layer down to the physical security of the facilities where AWS services operate. AWS is responsible for protecting the infrastructure that runs all the services that are offered in the AWS Cloud. This infrastructure is composed of the hardware, software, networking, and facilities that run the AWS Cloud services.

The customer is responsible for the encryption of data at rest and data in transit. The customer should also ensure that the network is configured for security and that security credentials and logins are managed safely. Additionally, the customer is responsible for the configuration of security groups and the configuration of the operating system that run on compute instances that they launch (including updates and security patches.





1. Upgrades and patches to the operating system on the EC2 instance?
   - ANSWER: The customer
2. Physical security of the data center?
   - ANSWER: AWS
3. Virtualization infrastructure?
   - ANSWER: AWS
4. EC2 security group settings?
   - ANSWER: The customer
5. Configuration of applications that run on the EC2 instance?
   - ANSWER: The customer
6. Oracle upgrades or patches If the Oracle instance runs as an Amazon RDS instance?
   - ANSWER: AWS
7. Oracle upgrades or patches If Oracle runs on an EC2 instance?
   - ANSWER: The customer
8. S3 bucket access configuration?
   - ANSWER: The customer

1. Ensuring that the AWS Management Console is not hacked?
   - ANSWER: AWS
2. Configuring the subnet?
   - ANSWER: The customer
3. Configuring the VPC?
   - ANSWER: The customer
4. Protecting against network outages in AWS Regions?
   - ANSWER: AWS
5. Securing the SSH keys
   - ANSWER: The customer
6. Ensuring network isolation between AWS customers' data?
   - ANSWER: AWS
7. Ensuring low-latency network connection between the web server and the S3 bucket?
   - ANSWER: AWS
8. Enforcing multi-factor authentication for all user logins?
   - ANSWER: The customer

Section 2: AWS Identity and Access Management (IAM)

When you define an **IAM user**, you select what ***types of access***

Programmatic access

- Authenticate using:
  - Access key ID
  - Secret access key
- Provides AWS CLI and AWS SDK access

AWS Management Console access

- Authenticate using:
  - 12-digit Account ID *or* alias
  - IAM user name
  - IAM password

IAM: Authorization

- Assign permissions by creating an IAM policy.
- Permissions determine **which resources and operations** are allowed:
  - All permissions are implicitly denied by default.
  - If something is explicitly denied, it is never allowed.

Best practice: Follow the **principle of least privilege**.

IAM permissions

Note: The scope of IAM service configurations is **global**. Settings apply across all AWS Regions.

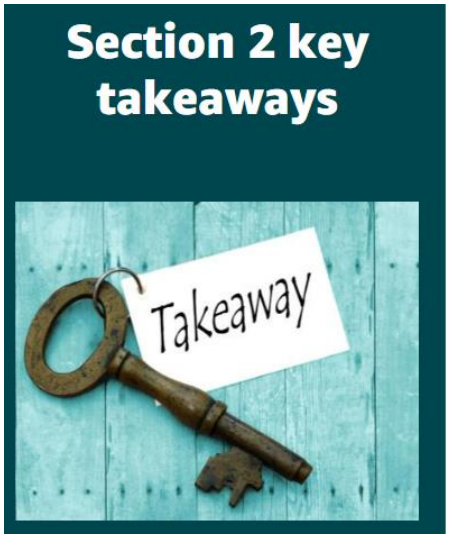
An explicit deny statement takes precedence over an allow statement. Resource-based policies are defined inline only, which means that you define the policy on the resource itself, instead of creating a separate IAM policy document that you attach.

An IAM Group is a collection of IAM users. IAM groups offer a convenient way to specify permissions for a collection of users, which can make it easier to manage the permissions for those users.

Important characteristics of IAM groups:
• A group can contain many users, and a user can belong to multiple groups.

• Groups cannot be nested. So a group can contain only users, and a group cannot contain other groups.
• There is no default group that automatically includes all users in the AWS account. If you want
to have a group with all account users in it, you need to create the group and add each new user to it.

An **IAM role** is an IAM identity you can create in your account that has specific permissions. An IAM role is **similar to an IAM user** because it is also an AWS identity that you can attach permissions policies to, and those permissions determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, the role provides you with temporary security credentials for your role session.

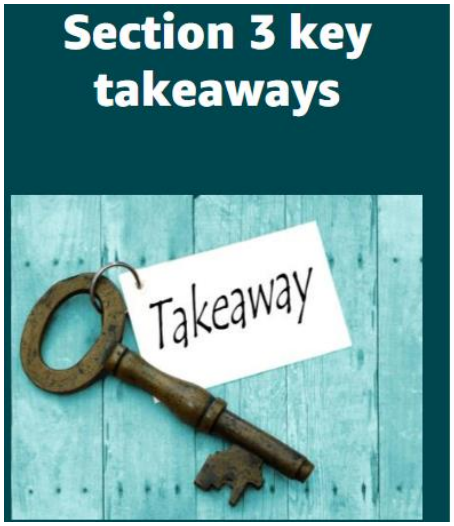


- **IAM policies** are constructed with JavaScript Object Notation (JSON) and define permissions.
  - IAM policies can be attached to any **IAM entity**.
  - Entities are IAM users, IAM groups, and IAM roles.
- An **IAM user** provides a way for a person, application, or service to authenticate to AWS.
- An **IAM group** is a simple way to attach the same policies to multiple users.
- An **IAM role** can have permissions policies attached to it and can be used to delegate temporary access to users or applications.

Section 3: Securing a new AWS account

- **Best practice**: Do not use the AWS account root user except when necessary.
  - Access to the **account root user** requires logging in with the *email address* (and password) that you used to create the account.
- Example actions that can only be done with the account root user:
  - Update the account root user password
  - Change the AWS Support plan
  - Restore an IAM user's permissions
  - Change account settings (for example, contact information, allowed Regions)

AWS CloudTrail is enabled on account creation by default on all AWS accounts, and it keeps a record of the last 90 days of account management event activity.
You can view and download the last 90 days of your account activity for creating, modifying, and deleting related operations of services that are supported by CloudTrail without needing to manually create another trail.



Best practices to secure an AWS account:

- **Secure** logins with multi-factor authentication (MFA).
- **Delete** account root user access keys.
- **Create** individual IAM users and grant permissions according to the principle of least privilege.
- **Use groups** to assign permissions to IAM users.
- **Configure** a strong password policy.
- **Delegate** using roles instead of sharing credentials.
- **Monitor** account activity by using AWS CloudTrail.

Section 4: Securing accounts

AWS Organizations is an account management service that enables you to consolidate multiple AWS accounts into an organization that you create and centrally manage. Here, the focus is on the security features that AWS Organizations provides. One helpful security feature is that you can group accounts into organizational units(OUs) and attach different access policies to each OU.
Another security feature is that AWS organizations integrate with and supports IAM. AWS Organizations expands that control to the account level by giving you control over what users and roles in an account or a group of accounts can do.
Finally, AWS Organizations provides service control policies (SCPs) that enable you to specify the maximum permissions that member accounts in the organization can have. In SCPs, you can restrict which AWS services, resources, and individual actions the users and roles in each member account can access. These restrictions even override the administrators of member accounts. When AWS Organizations blocks access to a service, resource, or API action, a user or role in that account can't access it, even if an administrator of a member account explicitly grants such permissions.

SCPs are similar to IAM permissions policies –

- They use similar syntax.
- However, an SCP never grants permissions.
- Instead, SCPs specify the maximum permissions for an organization.

AWS Key Management Service (AWS KMS) features:

- Enables you to **create and manage encryption keys**
- Enables you to control the use of encryption across AWS services and in your applications.
- Integrates with AWS CloudTrail to log all key usage.
- Uses hardware security modules (HSMs) that are validated by Federal Information Processing Standards (FIPS) 140-2 to protect keys

AWS Key Management Service (AWS KMS)

Amazon Cognito features:

- **Adds user sign-up, sign-in, and access control to your web and mobile applications.**
- Scales to millions of users.
- Supports sign-in with social identity providers, such as Facebook, Google, and Amazon; and enterprise identity providers, such as Microsoft Active Directory via Security Assertion Markup Language (SAML) 2.0.

Amazon Cognito

- **AWS Shield** features:
  - Is a managed distributed denial of service (DDoS) protection service
  - Safeguards applications running on AWS
  - Provides always-on detection and automatic inline mitigations
  - *AWS Shield Standard* enabled for at no additional cost. *AWS Shield Advanced* is an optional paid service.
- Use it to **minimize application downtime and latency.**

AWS Shield

Section 5: Securing data on AWS

- Encryption of **data in transit** (data moving across a network)
  - **Transport Layer Security (TLS)**—formerly SSL—is an open standard protocol
  - **AWS Certificate Manager** provides a way to manage, deploy, and renew TLS or SSL certificates
- Secure HTTP (HTTPS) creates a secure tunnel
  - Uses TLS or SSL for the bidirectional exchange of data
- **AWS services support data in transit encryption.**

- Tools and options for controlling access to S3 data include –
  - Amazon S3 Block Public Access feature: Simple to use.
  - IAM policies: A good option when the user can authenticate using IAM.
  - Bucket policies
  - Access control lists (ACLs): A legacy access control mechanism.
  - AWS Trusted Advisor bucket permission check: A free feature.

Section 6: Working to ensure compliance

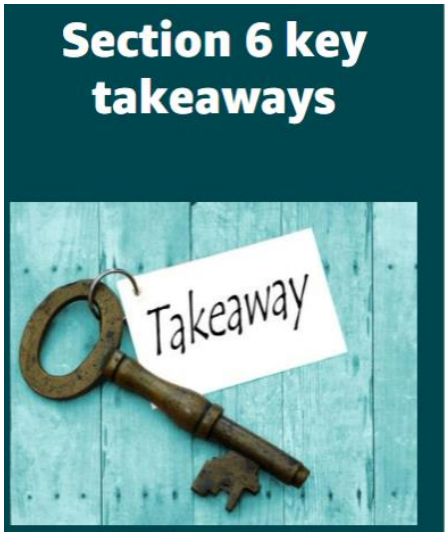
- Compliance programs can be broadly categorized –
  - Certifications and attestations
    - Assessed by a third-party, independent auditor
    - Examples: ISO27001,27017,27018, and ISO/IEC9001
  - Laws, regulations, and privacy
    - AWS provides security features and legal agreements to support compliance
    - Examples: EU General Data Protection Regulation (GDPR), HIPAA
  - Alignments and frameworks
    - Industry-or function-specific security or compliance requirements
    - Examples: Center for Internet Security (CIS), EU-US Privacy Shield certified

AWS Config

•Assess, audit, and evaluate the configurations of AWS resources.

AWS Artifact

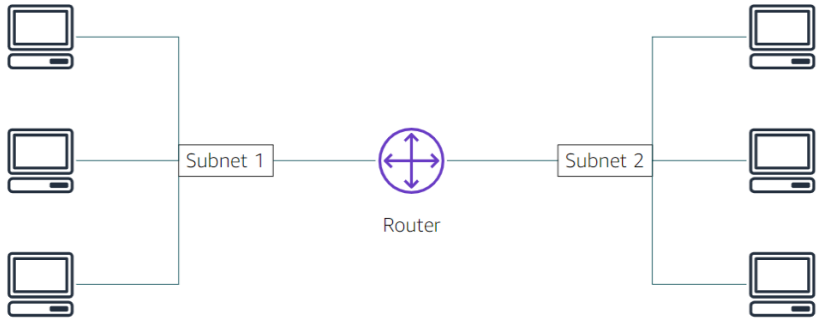
•Is a resource for compliance-related information



- **AWS security compliance programs** provide information about the policies, processes, and controls that are established and operated by AWS.
- **AWS Config** is used to assess, audit, and evaluate the configurations of AWS resources.
- **AWS Artifact** provides access to security and compliance reports.

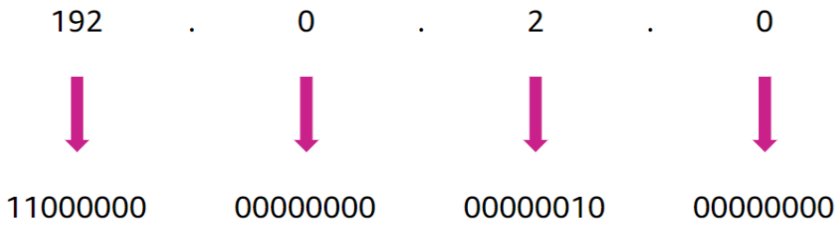
Module 5: Networking and Content Delivery

Section 1: Networking basics



Each client machine in a network has a unique Internet Protocol (IP) address that identifies it. An IP address is a numerical label in decimal format. Machines convert that decimal number to a binary format.
In this example, the IP address is 192.0.2.0.

Each of the four dot (.)-separated numbers of the IP address represents 8 bits in octal number format. That means each of the four numbers can be anything from 0 to 255. The combined total of the four numbers for an IP address is 32 bits in binary format.

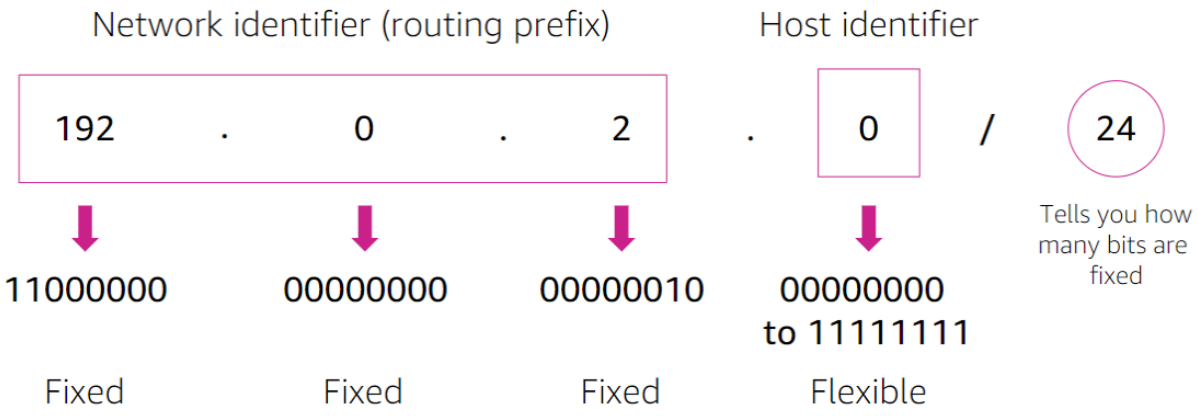


IPv4 (32-bit) address: 192.0.2.0

IPv6 (128-bit) address:
2600:1f18:22ba:8c00:ba86:a05e:a5ba:00FF

An IPv6 address is composed of eight groups of four letters and numbers that are separated by colons (:). In this example, the IPv6 address is 2600:1f18:22ba:8c00:ba86:a05e:a5ba:00FF.
Each of the eight colon-separated groups of the IPv6 address represents 16 bits in hexadecimal number format. That means each of the eight groups can be anything from 0 to FFFF. The combined total of the eight groups for an IPv6 address is 128 bits in binary format

Classless Inter-Domain Routing (CIDR)



The internet, in which every bit is flexible, is represented as 0.0.0.0/0

Open Systems Interconnection (OSI) model

Layer	Number	Function	Protocol/Address
Application	7	Means for an application to access a computer network	HTTP(S), FTP, DHCP, LDAP
Presentation	6	• Ensures that the application layer can read the data • Encryption	ASCI, ICA
Session	5	Enables orderly exchange of data	NetBIOS, RPC
Transport	4	Provides protocols to support host-to-host communication	TCP, UDP
Network	3	Routing and packet forwarding (routers)	IP
Data link	2	Transfer data in the same LAN network (hubs and switches)	MAC
Physical	1	Transmission and reception of raw bitstreams over a physical medium	Signals (1s and 0s)

Please Do Not Touch Selvi’s Pet Anish

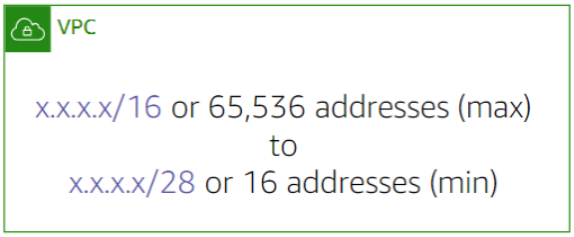
Section 2: Amazon VPC

Amazon Virtual Private Cloud (Amazon VPC) is a service that lets you provision a logically isolated section of the AWS Cloud (called a virtual private cloud, or VPC) where you can launch your AWS resources.

Amazon VPC gives you control over your virtual networking resources, including the selection of your own IP address range, the creation of subnets, and the configuration of route tables and network gateways. You can use both IPv4 and IPv6 in your VPC for secure access to resources and applications. You can use multiple layers of security, including security groups and network access control lists (networkACLs), to help control access to Amazon Elastic Compute Cloud (Amazon EC2) instances in each subnet.

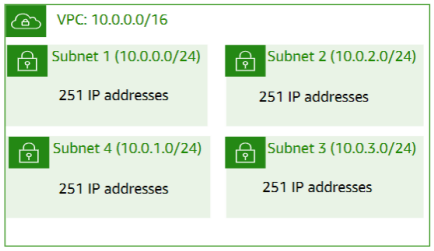
IP addressing

- When you create a VPC, you assign it to an IPv4 CIDR block (range of private IPv4 addresses).
- You cannot change the address range after you create the VPC.
- The largest IPv4 CIDR block size is /16.
- The smallest IPv4 CIDR block size is /28.
- IPv6 is also supported (with a different block size limit).
- CIDR blocks of subnets cannot overlap.



Reserved IP addresses

Example: A VPC with an IPv4 CIDR block of 10.0.0.0/16 has 65,536 total IP addresses. The VPC has four equal-sized subnets. Only 251 IP addresses are available for use by each subnet.



IP Addresses for CIDR block 10.0.0.0/24	Reserved for
10.0.0.0	Network address
10.0.0.1	Internal communication
10.0.0.2	Domain Name System (DNS) resolution
10.0.0.3	Future use
10.0.0.255	Network broadcast address

Public IP address types

Public IPv4 address

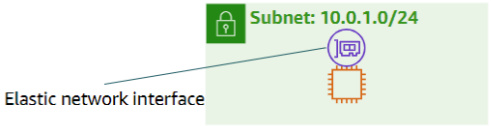
- Manually assigned through an Elastic IP address
- Automatically assigned through the auto-assign public IP address settings at the subnet level

Elastic IP address

- Associated with an AWS account
- Can be allocated and remapped anytime
- Additional costs might apply

Elastic network interface

- An elastic network interface is a virtual network interface that you can:
  - Attach to an instance.
  - Detach from the instance, and attach to another instance to redirect network traffic.
- Its attributes follow when it is reattached to a new instance.
- Each instance in your VPC has a default network interface that is assigned a private IPv4 address from the IPv4 address range of your VPC.



Route tables and routes

- A route table contains a set of rules (or routes) that you can configure to direct network traffic from your subnet.
- Each route specifies a destination and a target.
- By default, every route table contains a local route for communication within the VPC.
- Each subnet must be associated with a route table (at most one).

Main (Default) Route Table

Destination	Target
10.0.0.0/16	local

VPC CIDR block

- A VPC is a logically isolated section of the AWS Cloud.
- A VPC belongs to one Region and requires a CIDR block.
- A VPC is subdivided into subnets.
- A subnet belongs to one Availability Zone and requires a CIDR block.
- Route tables control traffic for a subnet.
- Route tables have a built-in local route.
- You add additional routes to the table.
- The local route cannot be deleted.

Section 3: VPC networking

An internet gatewayis a scalable, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet. An internet gateway serves two purposes: to provide a target in your VPC route tables for internet-routable traffic, and to perform network address translation for instances that were assigned public IPv4 addresses.

To make a subnet public, you attach an internet gateway to your VPC and add a route to the route table to send non-local traffic through the internet gateway to the internet (0.0.0.0/0).

A network address translation (NAT) gatewayenables instances in a private subnet to connect to the internet or other AWS services, but prevents the internet from initiating a connection with those instances.

To create a NAT gateway, you must specify the public subnet in which the NAT gateway should reside. You must also specify an Elastic IP address to associate with the NAT gateway when you create it. After you createa NAT gateway, you must update the route table thatis associated with one or more of your private subnets to point internet-bound traffic to the NAT gateway.

Thus, instances in your private subnets can communicate with the internet. You can also use a NAT instance in a public subnet in your VPC instead of a NAT gateway. However, a NAT gateway is a managed NAT service that provides better availability, higher bandwidth, and less administrative effort. For common use cases, AWS recommends that you use a NAT gateway instead of a NAT instance.

A *VPC peering connection* is a networking connection between two VPCs that enables you to route traffic between them privately. Instances in either VPC can communicate with each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, with a VPC in another AWS account, or with a VPC in a different AWS Region.

When you set up the peering connection, you create rules in your route table to allow the VPCs to communicate with each other through the peering resource. For example, suppose that you have two VPCs. In the route table for VPC A, you set the destination to be the IP address of VPC B and the target to be the peering resource ID. In the route table for VPC B, you set the destination to be the IP address of VPC A and the target to be the peering resource ID.

VPC peering has some restrictions:

- IP address ranges cannot overlap.
- Transitive peering is not supported. For example, suppose that you have three VPCs: A, B, and C. VPC A is connected to VPC B, and VPC A is connected to VPC C. However, VPC B is *not* connected to VPC C implicitly. To connect VPC B to VPC C, you must explicitly establish that connectivity.
- You can only have one peering resource between the same two VPCs.

By default, instances that you launch into a VPC cannot communicate with a remote network. To connect your VPC to your remote network (that is, create a virtual private network or VPN connection), you:

1. Create a new virtual gateway device (called a *virtual private network (VPN) gateway*) and attach it to your VPC.
2. Define the configuration of the VPN device or the *customer gateway*. The customer gateway is not a device but an AWS resource that provides information to AWS about your VPN device.
3. Create a custom route table to point corporate data center-bound traffic to the VPN gateway. You also must update security group rules. (You will learn about security groups in the next section.)
4. Establish an *AWS Site-to-Site VPN (Site-to-Site VPN) connection* to link the two systems together.
5. Configure routing to pass traffic through the connection.

AWS Direct Connect enables you to establish a dedicated, private network connection between your network and one of the DX locations.

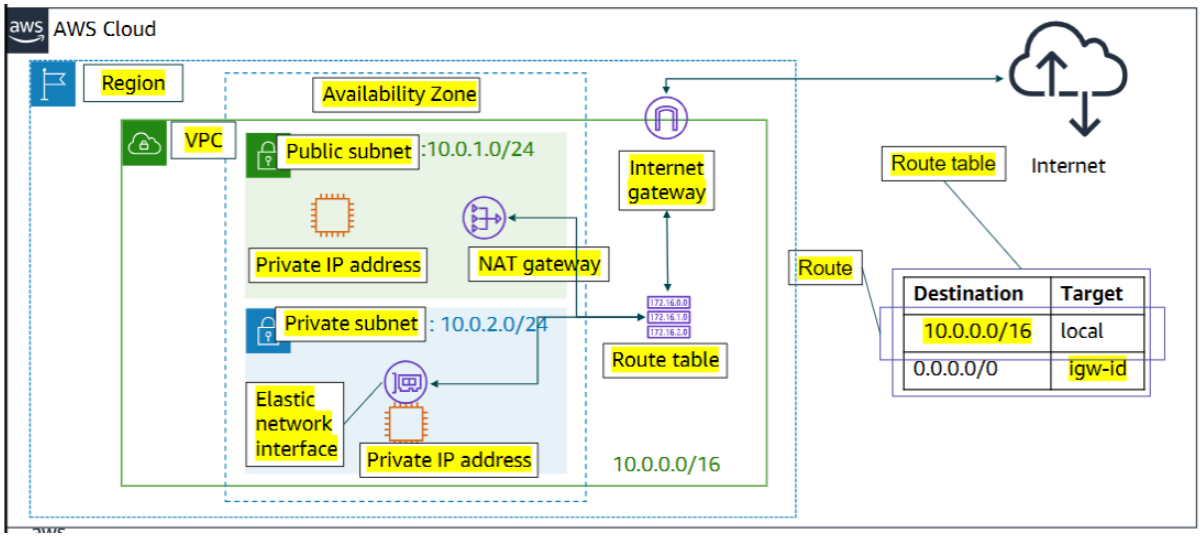
A VPC endpointis a virtual device that enables you to privately connect your VPC to supported AWS services and VPC endpoint services that are powered by AWS PrivateLink. Connection to these services does not require an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection.

There are two types of VPC endpoints:

•An interface VPC endpoint(interface endpoint) enables you to connect to services that are powered by AWS PrivateLink. These services include some AWS services, services that are hosted by other AWS customers and AWS Partner Network (APN) Partners in their own VPCs (referred to as endpoint services), and supported AWS Marketplace APN Partner services. The owner of the service is the service provider, and you—as the principal who creates the interface endpoint—are the service consumer. You are charged for creating and using an interface endpoint to a service. Hourly usage rates and data processing rates apply. See the AWS Documentation for a list of supported interface

endpoints and for more information about the example shown here at https://docs.aws.amazon.com/vpc/latest/privatelink/create-interface-endpoint.html.

•Gateway endpoints: The use of gateway endpoints incurs no additional charge. Standard charges for data transfer and resource usage apply.



- There are several VPC networking options, which include:
  - Internet gateway
  - NAT gateway
  - VPC endpoint
  - VPC peering
  - VPC sharing
  - AWS Site-to-Site VPN
  - AWS Direct Connect
  - AWS Transit Gateway
- You can use the VPC Wizard to implement your design.

Section 4: VPC security

A security groupacts as a virtual firewall for your instance, and it controls inbound and outbound traffic. Security groups act at the instance level, not the subnet level. Therefore, each instance in a

subnet in your VPC can be assigned to a different set of security groups.
At the most basic level, a security group is a way for you to filter traffic to your instances

Security groups have rules that control inbound and outbound instance traffic.

•Default security groups deny all inbound traffic and allow all outbound traffic.
•Security groups are stateful, which means that state information is kept even after a request is processed.

A network access control list (network ACL)is an optional layer of security for your Amazon VPC. It acts as a firewall for controlling traffic in and out of one or more subnets.

•A network ACL has separate inbound and outbound rules, and each rule can either allow or deny traffic.
•Default network ACLs allow all inbound and outbound IPv4 traffic.
•Network ACLs are stateless, which means that no information about a request is maintained after a request is processed.

Custom network ACLs deny all inbound and outbound traffic until you add rules.
•You can specify both allow and deny rules.
•Rules are evaluated in number order, starting with the lowest number.

Attribute	Security Groups	Network ACLs
Scope	Instance level	Subnet level
Supported Rules	Allow rules only	Allow and deny rules
State	Stateful (return traffic is automatically allowed, regardless of rules)	Stateless (return traffic must be explicitly allowed by rules)
Order of Rules	All rules are evaluated before decision to allow traffic	Rules are evaluated in number order before decision to allow traffic

- Build security into your VPC architecture:
  - Isolate subnets if possible.
  - Choose the appropriate gateway device or VPN connection for your needs.
  - Use firewalls.
- Security groups and network ACLs are firewall options that you can use to secure your VPC.

Section 5: Amazon Route 53

Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. It is designed to give developers and businesses areliable and cost-effective way to route users to internet applications by translating names (like www.example.com) into the numeric IP addresses (like 192.0.2.1) that computers use to connect to each other. In addition, Amazon Route 53 is fully compliant with IPv6.

Amazon Route 53 supported routing

- Simple routing – Use in single-server environments
- Weighted round robin routing – Assign weights to resource record sets to specify the frequency
- Latency routing – Help improve your global applications
- Geolocation routing – Route traffic based on location of your users
- Geoproximity routing – Route traffic based on location of your resources
- Failover routing – Fail over to a backup site if your primary site becomes unreachable
- Multivalue answer routing – Respond to DNS queries with up to eight healthy records selected at random

Simple routing is based on round robin

Amazon Route 53 enables you to improve the availability of your applications that run on AWS by:
•Configuring backup and failover scenarios for your own applications.
•Enabling highly available multi-Region architectures on AWS.
•Creating health checks to monitor the health and performance of your web applications, web servers, and other resources.

Section 6: Amazon CloudFront

Content delivery network (CDN)

- Is a globally distributed system of caching servers
- Caches copies of commonly requested files (static content)
- Delivers a local copy of the requested content from a nearby cache edge or Point of Presence
- Accelerates delivery of dynamic content
- Improves application performance and scaling

Amazon CloudFront

Amazon CloudFront

- Fast, global, and secure CDN service
- Global network of edge locations and Regional edge caches
- Self-service model
- Pay-as-you-go pricing

- **Edge locations** – Network of data centers that CloudFront uses to serve popular content quickly to customers.
- **Regional edge cache** – CloudFront location that caches content that is popular enough to stay at an edge l It is located between the origin serv the global edge location.

Amazon CloudFront pricing

Data transfer out

- Charged for the volume of data transferred out from Amazon CloudFront edge location to the internet or to your origin.

HTTP(S) requests

- Charged for number of HTTP(S) requests.

Invalidation requests

- No additional charge for the first 1,000 paths that are requested for invalidation each month. Thereafter, $0.005 per path that is requested for invalidation.

Dedicated IP custom SSL

- $600 per month for each custom SSL certificate that is associated with one or more CloudFront distributions that use the Dedicated IP version of custom SSL certificate support.

Section 6 key takeaways

- A CDN is a globally distributed system of caching servers that accelerates delivery of content.
- Amazon CloudFront is a fast CDN service that securely delivers data, videos, applications, and APIs over a global infrastructure with low latency and high transfer speeds.
- Amazon CloudFront offers many benefits.

Module 6: Compute

Section 1: Compute services overview

- **Amazon Elastic Compute Cloud (Amazon EC2)** provides resizable virtual machines.
- **Amazon EC2 Auto Scaling** supports application availability by allowing you to define conditions that will automatically launch or terminate EC2 instances.
- **Amazon Elastic Container Registry (Amazon ECR)** is used to store and retrieve Docker images.
- **Amazon Elastic Container Service (Amazon ECS)** is a container orchestration service that supports Docker.
- **VMware Cloud on AWS** enables you to provision a hybrid cloud without custom hardware.
- **AWS Elastic Beanstalk** provides a simple way to run and manage web applications.
- **AWS Lambda** is a serverless compute solution. You pay only for the compute time that you use.
- **Amazon Elastic Kubernetes Service (Amazon EKS)** enables you to run managed Kubernetes on AWS.
- **Amazon Lightsail** provides a simple-to-use service for building an application or website.
- **AWS Batch** provides a tool for running batch jobs at any scale.
- **AWS Fargate** provides a way to run containers that reduce the need for you to manage servers or clusters.
- **AWS Outposts** provides a way to run select AWS services in your on-premises data center.
- **AWS Serverless Application Repository** provides a way to discover, deploy, and publish serverless applications.

Services	Key Concepts	Characteristics	Ease of Use
• Amazon EC2	• Infrastructure as a service (IaaS) • Instance-based • Virtual machines	• Provision virtual machines that you can manage as you choose	A familiar concept to many IT professionals.
• AWS Lambda	• Serverless computing • Function-based • Low-cost	• Write and deploy code that runs on a schedule or that can be triggered by events • Use when possible (architect for the cloud)	A relatively new concept for many IT staff members, but easy to use after you learn how.
• Amazon ECS • Amazon EKS • AWS Fargate • Amazon ECR	• Container-based computing • Instance-based	• Spin up and run jobs more quickly	AWS Fargate reduces administrative overhead, but you can use options that give you more control.
• AWS Elastic Beanstalk	• Platform as a service (PaaS) • For web applications	• Focus on your code (building your application) • Can easily tie into other services—databases, Domain Name System (DNS), etc.	Fast and easy to get started.

Section 2: Amazon EC2

- **Amazon Elastic Compute Cloud (Amazon EC2)**
  - Provides virtual machines—referred to as EC2 instances—in the cloud.
  - Gives you *full control* over the guest operating system (Windows or Linux) on each instance.
- You can launch instances of any size into an Availability Zone anywhere in the world.
  - Launch instances from **Amazon Machine Images (AMIs)**.
  - Launch instances with a few clicks or a line of code, and they are ready in minutes.
- You can control traffic to and from instances.

This section of the module walks through nine key decisions to make when you create an EC2 instance by using the AWS Management Console Launch Instance Wizard

Choices made using the Launch Instance Wizard:

1. AMI
2. Instance Type
3. Network settings
4. IAM role
5. User data
6. Storage options
7. Tags
8. Security group
9. Key pair

1. Select an AMI

Choices made using the Launch Instance Wizard:

1. AMI
2. Instance Type
3. Network settings
4. IAM role
5. User data
6. Storage options
7. Tags
8. Security group
9. Key pair

- Amazon Machine Image (AMI)
  - Is a template that is used to create an EC2 instance (which is a **virtual machine, or VM,** that runs in the AWS Cloud)
  - Contains a **Windows** or **Linux** operating system
  - Often also has some **software** pre-installed
- AMI choices:
  - Quick Start – *Linux and Windows AMIs that are provided by AWS*
  - My AMIs – *Any AMIs that you created*
  - AWS Marketplace – *Pre-configured templates from third parties*
  - Community AMIs – *AMIs shared by others; use at your own risk*

2. Select an instance type

Choices made using the Launch Instance Wizard:

1. AMI
2. Instance Type
3. Network settings
4. IAM role
5. User data
6. Storage options
7. Tags
8. Security group
9. Key pair

- Consider your use case
  - How will the EC2 instance you create be used?
- The **instance type** that you choose determines –
  - Memory (RAM)
  - Processing power (CPU)
  - Disk space and disk type (Storage)
  - Network performance
- Instance type categories –
  - General purpose
  - Compute optimized
  - Memory optimized
  - Storage optimized
  - Accelerated computing
- Instance types offer *family*, *generation*, and *size*

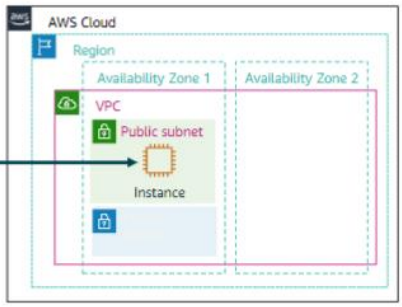
3. Specify network settings

Choices made by using the Launch Instance Wizard:

1. **AMI**
2. **Instance Type**
3. **Network settings**
4. **IAM role**
5. **User data**
6. **Storage options**
7. **Tags**
8. **Security group**
9. **Key pair**

- Where should the instance be deployed?
  - Identify the **VPC** and optionally the **subnet**
- Should a **public IP address** be automatically assigned?
  - To make it internet-accessible

Example: specify to deploy the instance here



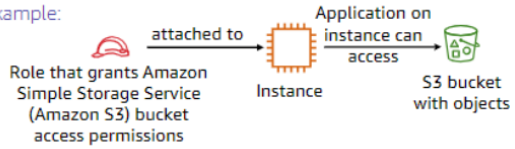
4. Attach IAM role (optional)

Choices made by using the Launch Instance Wizard:

1. AMI
2. Instance Type
3. Network settings
4. IAM role
5. User data
6. Storage options
7. Tags
8. Security group
9. Key pair

- Will software on the EC2 instance need to interact with other AWS services?
  - If yes, attach an appropriate **IAM Role**.
- An AWS Identity and Access Management (IAM) role that is attached to an EC2 instance is kept in an **instance profile**.
- You are *not* restricted to attaching a role only at instance launch.
  - You can also attach a role to an instance that already exists.

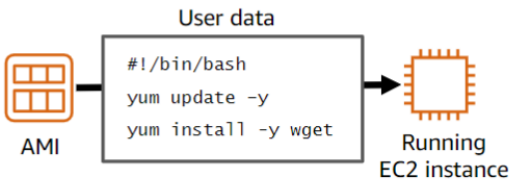
Example:



5. User data script (optional)

Choices made by using the Launch Instance Wizard:

1. AMI
2. Instance Type
3. Network settings
4. IAM role
5. User data
6. Storage options
7. Tags
8. Security group
9. Key pair



- Optionally specify a user data script at instance launch
- Use **user data** scripts to customize the runtime environment of your instance
  - Script runs the first time the instance starts
- Can be used strategically
  - For example, reduce the number of custom AMIs that you build and maintain

6. Specify storage

Choices made by using the Launch Instance Wizard:

1. AMI
2. Instance Type
3. Network settings
4. IAM role
5. User data
6. Storage options
7. Tags
8. Security group
9. Key pair

- Configure the root volume
  - Where the guest operating system is installed
- Attach additional storage volumes (optional)
  - AMI might already include more than one volume
- For each volume, specify:
  - The **size** of the disk (in GB)
  - The **volume type**
    - Different types of solid state drives (SSDs) and hard disk drives (HDDs) are available
  - If the volume will be deleted when the instance is terminated
  - If **encryption** should be used

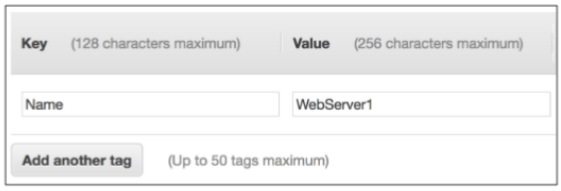
7. Add tags

Choices made by using the Launch Instance Wizard:

1. AMI
2. Instance Type
3. Network settings
4. IAM role
5. User data
6. Storage options
7. Tags
8. Security group
9. Key pair

- A tag is a label that you can assign to an AWS resource.
  - Consists of a *key* and an optional *value*.
- Tagging is how you can attach **metadata** to an EC2 instance.
- Potential benefits of tagging—Filtering, automation, cost allocation, and access control.

Example:



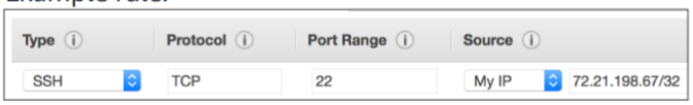
8. Security group settings

Choices made by using the Launch Instance Wizard:

1. AMI
2. Instance Type
3. Network settings
4. IAM role
5. User data
6. Storage options
7. Tags
8. Security group
9. Key pair

- A security group is a **set of firewall rules** that control traffic to the instance.
  - It exists *outside* of the instance's guest OS.
- Create **rules** that specify the **source** and which **ports** that network communications can use.
  - Specify the **port** number and the **protocol**, such as Transmission Control Protocol (TCP), User Datagram Protocol (UDP), or Internet Control Message Protocol (ICMP).
  - Specify the **source** (for example, an IP address or another security group) that is allowed to use the rule.

Example rule:



9. Identify or create the key pair

Choices made by using the Launch Instance Wizard:

1. AMI
2. Instance Type
3. Network settings
4. IAM role
5. User data
6. Storage options
7. Tags
8. Security group
9. Key pair

- At instance launch, you specify an existing key pair *or* create a new key pair.
- A key pair consists of –
  - A **public key** that AWS stores.
  - A **private key** file that you store.

mykey.pem

- It enables secure connections to the instance.

- For **Windows AMIs –**
  - Use the private key to obtain the administrator password that you need to log in to your instance.
- For **Linux AMIs –**
  - Use the private key to use SSH to securely connect to your instance.

An **Amazon Machine Image (AMI)** provides information that is required to launch an **EC2 instance**. You must specify a source AMI when you launch an instance. You can use different AMIs to launch different types of instances. For example, you can choose one AMI to launch an instance that will become a web server and another AMI to deploy an instance that will host an application server. You can also launch multiple instances from a single AMI.

An AMI includes the following components:

- A **template for the root volume** of the instance. A root volume typically contains an operating system (OS) and everything that was installed in that OS (applications, libraries, etc.). Amazon EC2 copies the template to the root volume of a new EC2 instance, and then starts it.
- **Launch permissions** that control which AWS accounts can use the AMI.
- A **block device mapping** that specifies the volumes to attach to the instance (if any) when it is launched.

You can choose many AMIs:

- **Quick Start –** AWS offers a number of pre-built AMIs for launching your instances. These AMIs include many Linux and Windows options.
- **My AMIs –** These AMIs are AMIs that you created.
- **AWS Marketplace –** The AWS Marketplace offers a digital catalog that lists thousands of software solutions. These AMIs can offer specific use cases to help you get started quickly.
- **Community AMIs –** These AMIs are created by people all around the world. These AMIs are not checked by AWS, so use them at your own risk. Community AMIs can offer many different solutions to various problems, but use them with care. Avoid using them in any production or corporate environment.

- Consider your use case
  - How will the EC2 instance you create be used?
- The **instance type** that you choose determines –
  - Memory (RAM)
  - Processing power (CPU)
  - Disk space and disk type (Storage)
  - Network performance

- Instance type categories –
  - General purpose
  - Compute optimized
  - Memory optimized
  - Storage optimized
  - Accelerated computing
- Instance types offer *family*, *generation*, and *size*

When you look at an EC2 instance type, you will see that its name has several parts. For example, consider the T type.
T is the family name, which is then followed by a number.
Here, that number is 3. The number is the generation number of that type. So, a t3 instance is the third generation of the T family. In general, instance types that are of a higher generation are more powerful and provide a better value for the price.
The next part of the name is the size portion of the instance. When you compare sizes, it is important

to look at the coefficient portion of the size category. For example, a t3.2xlarge has twice the vCPU and memory of a t3.xlarge.
The t3.xlarge has, in turn, twice the vCPU and memory of a t3.large.
It is also important to note that network bandwidth is also tied to the size of the Amazon EC2 instance. If you will run jobs that will be very network-intensive, you might be required to increase the instance specifications to meet your needs.

Example instance sizes

Instance Name	vCPU	Memory (GB)
t3.nano	2	0.5
t3.micro	2	1
t3.small	2	2
t3.medium	2	4
t3.large	2	8
t3.xlarge	4	16
t3.2xlarge	8	32

When you launch multiple new EC2 instances, Amazon EC2 attempts to place the instances so that they are spread out across the underlying hardware by default. It does this to minimize correlated failures. However, if you want to specify specific placement criteria, you can use placement groups to influence the placement of a group of interdependent instances to meet the needs of your workload. For example, you might specify that three instances should all be deployed in the same Availability Zone to ensure lower network latency and higher network throughput between instances.

After you have choose an AMI and an instance type, you must specify the network location where the EC2 instance will be deployed. The choice of **Region** must be made before you start the Launch Instance Wizard. Verify that you are in the correct Region page of the Amazon EC2 console before you choose **Launch Instance.**

When you launch an instance in a **default VPC**, AWS will assign it a **public IP address** by default. When you launch an instance into a **nondefault VPC**, the subnet has an attribute that determines whether instances launched into that subnet receive a public IP address from the public IPv4 address pool. By default, AWS will not assign a public IP address to instances that are launched in a nondefault subnet. You can control whether your instance receives a public IP address by either modifying the public IP addressing attribute of your subnet, or by enabling or disabling the public IP addressing feature during launch (which overrides the subnet's public IP addressing attribute).

It is common to use EC2 instances to run an application that must make secure API calls to other AWS services. To support these use cases, AWS enables you to attach an AWS Identity and Access Management (IAM) role to an EC2 instance. Without this feature, you might be tempted to place AWS credentials on an EC2 instance so an application that runs on that instance to use. However, you

should never store AWS credentials on an EC2 instance. It is highly insecure. Instead, attach an IAM role to the EC2 instance.

The IAM role then grants permission to make application programming interface (API) requests to the applications that run on the EC2 instance.

An instance profile is a container for an IAM role. If you use the AWS Management Console to create a role for Amazon EC2, the console automatically creates an instance profile and gives it the same name as the role. When you then use the Amazon EC2 console to launch an instance with an IAM role, you can select a role to associate with the instance. In the console, the list that displays is actually a list of instance profile names.

When you create your EC2 instances, you have the option of passing user data to the instance. User data can automate the completion of installations and configurations at instance launch. For example, a user data script might patch and update the instance's operating system, fetch and install software license keys, or install additional software.

When the EC2 instance is created, the user data script will run with root privileges during the final phases of the boot process. On Linux instances, it is run by the cloud-init service. On Windows instances, it is run by the EC2Configor EC2Launchutility. By default, user data only runs the first time that the instance starts up. However, if you would like your user data script to run every time the instance is booted, you can create a Multipurpose Internet Mail Extensions (MIME) multipart file user data script (this process is not commonly done).

When you launch an EC2 instance, you can configure storage options. For example, you can configure the size of the root volume where the guest operating system is installed. You can also attach additional storage volumes when you launch the instance. Some AMIs are also configured to launch more than one storage volume by default to provide storage that is separate from the root volume. For each volume that your instance will have, you can specify the size of the disks, the volume types, and whether the storage will be retained if the instance is terminated. You can also specify if encryption should be used.

Amazon EC2 storage options

- **Amazon Elastic Block Store (Amazon EBS) –**
  - Durable, block-level storage volumes.
  - You can stop the instance and start it again, and the data will still be there.
- **Amazon EC2 Instance Store –**
  - Ephemeral storage is provided on disks that are attached to the host computer where the EC2 instance is running.
  - If the instance stops, data stored here is deleted.
- Other options for storage (not for the root volume) –
  - Mount an **Amazon Elastic File System (Amazon EFS)** file system.
  - Connect to **Amazon Simple Storage Service (Amazon S3)**.

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value, both of which you define. Tags enable you to categorize AWS resources, such as EC2 instances, in different ways. For example, you might tag instances by purpose, owner, or environment.

Tagging is how you can attach metadata to an EC2 instance. Tag keys and tag values are case-sensitive. For example, a commonly used tag for EC2 instances is a tag key that is called Name and a tag value that describes the instance, such as My Web Server.

The Name tag is exposed by default in the Amazon EC2 console Instances page. However, if you create a key that is called name (with lower-case n), it will not appear in the Name column for the list of instances (though it will still appear in the instance details panel in the Tags tab). It is a best practice to develop tagging strategies. Using a consistent set of tag keys makes it easier for you to manage your resources.

A security group acts as a virtual firewall that controls network traffic for one or more instances. When you launch an instance, you can specify one or more security groups; otherwise, the default security group is used. You can add rules to each security group. Rules allow traffic to or from its associated instances. You can modify the rules for a security group at any time, and the new rules will be automatically applied to all instances that are associated with the security group. When AWS decides whether to allow traffic to reach an instance, all the rules from all the security groups that are associated with the instance are evaluated.

When you launch an instance in a virtual private cloud (VPC), you must either create a new security group or use one that already exists in that VPC. After you launch an instance, you can change its security groups. When you define a rule, you can specify the allowable source of the network communication (inbound rules) or destination (outbound rules). The source can be an IP address, an IP address range, another security group, a gateway VPC endpoint, or anywhere (which means that all sources will be allowed).

By default, a security group includes an outbound rule that allows all outbound traffic. You can remove the rule and add outbound rules that only allow specific outbound traffic. If your security group has no outbound rules, no outbound traffic that originates from your instance is allowed.

After you specify all the required configurations to launch an EC2 instance, and after you customize any optional EC2 launch wizard configuration settings, you are presented with a **Review Instance Launch** window. If you then choose **Launch,** a dialog asks you to choose an existing key pair, proceed without a key pair, or create a new key pair before you can choose **Launch Instances** and create the EC2 instance.

Amazon EC2 uses public–key cryptography to encrypt and decrypt login information. The technology uses a **public key** to encrypt a piece of data, and then the recipient uses the private key to decrypt the data. The public and private keys are known as a **key pair.** Public-key cryptography enables you to securely access your instances by using a private key instead of a password.

When you launch an instance, you specify a key pair. You can specify an existing key pair or a new key pair that you create at launch. If you create a new key pair, download it and save it in a safe location. This opportunity is the only chance you get to save the private key file.

To connect to a **Windows** instance, use the private key to obtain the administrator password, and then log in to the EC2 instance's Windows Desktop by using Remote Desktop Protocol (RDP). To establish an SSH connection from a Windows machine to an Amazon EC2 instance, you can use a tool such as PuTTY, which will require the same private key.

With **Linux** instances, at boot time, the **public key** content is placed on the instance. An entry is created in within `~/.ssh/authorized_keys`. To log in to your Linux instance (for example, by using SSH), you must provide the **private key** when you establish the connection.

Another option: Launch an EC2 instance with the AWS Command Line Interface

- EC2 instances can also be created programmatically.

AWS Command Line Interface (AWS CLI)

- This example shows how simple the command can be.
  - This command assumes that the key pair and security group already exist.
  - More options could be specified. See the AWS CLI Command Reference for details.

Example command:

```
aws ec2 run-instances \
--image-id ami-1a2b3c4d \
--count 1 \
--instance-type c3.large \
--key-name MyKeyPair \
--security-groups MySecurityGroup \
--region us-east-1
```

An instance can be in one of the following states:

•Pending–When an instance is first launched from an AMI, or when you start a stopped instance, it enters the pending state when the instance is booted and deployed to a host computer. The instance type that you specified at launch determines the hardware of the host computer for your instance.

•Running–When the instance is fully booted and ready, it exits the pending state and enters the running state. You can connect over the internet to your running instance.

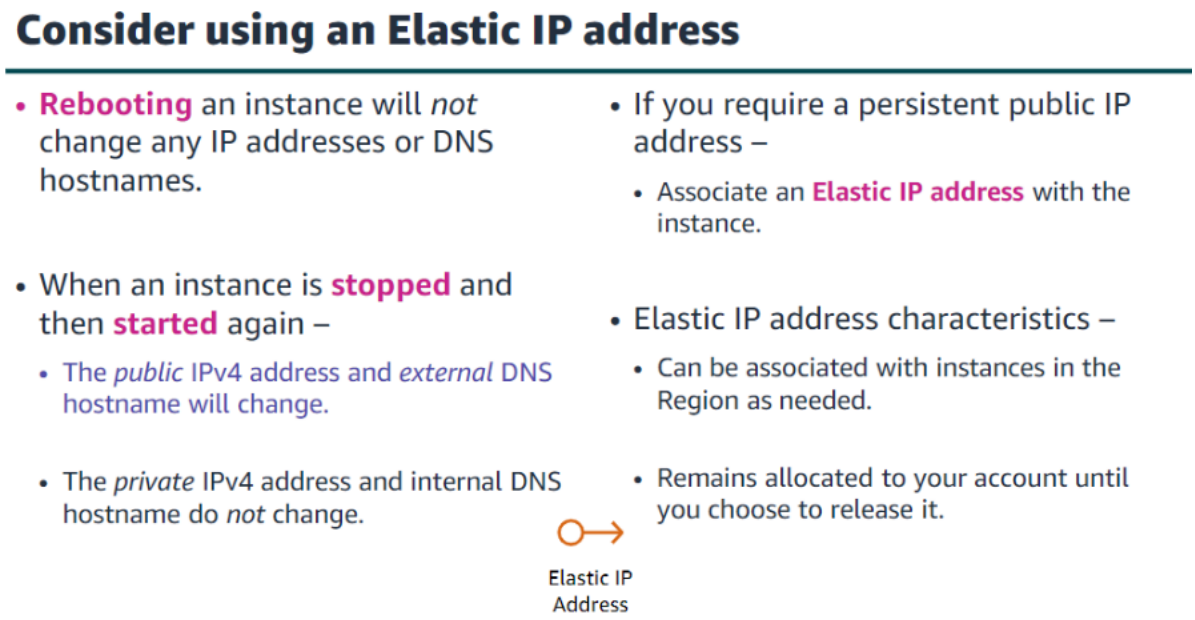
•Rebooting –AWS recommends you reboot an instance by using the Amazon EC2 console, AWS CLI, or AWS SDKs instead of invoking a reboot from within the guest operating system (OS). A rebooted instance stays on the same physical host, maintains the same public DNS name and public IP address, and if it has instance store volumes, it retains the data on those volumes.

•Shutting down –This state is an intermediary state between running and terminated.

•Terminated–A terminated instance remains visible in the Amazon EC2 console for a while before the virtual machine is deleted. However, you can't connect to or recover a terminated instance.

•Stopping–Instances that are backed by Amazon EBS can be stopped. They enter the stopping state before they attain the fully stopped state.

•Stopped–A stopped instance will not incur the same cost as a running instance. Starting a stopped instance puts it back into the pending state, which moves the instance to a new host machine.



You can monitor your instances by using Amazon CloudWatch, which collects and processes raw data from Amazon EC2 into readable, near-real-time metrics. These statistics are recorded for a period of 15 months, so you can access historical information and gain a better perspective on how your web application or service is performing. By default, Amazon EC2 provides basic monitoring, which sends metric data to CloudWatch in 5-minute periods. To send metric data for your instance to CloudWatch in 1-minute periods, you can enable detailed monitoring on the instance.

EC2 instance metadata

- **Instance metadata** is data about your instance.
- While you are connected to the instance, you can view it –
  - In a browser: `http://169.254.169.254/latest/meta-data/`
  - In a terminal window: `curl http://169.254.169.254/latest/meta-data/`
- Example retrievable values –
  - Public IP address, private IP address, public hostname, instance ID, security groups, Region, Availability Zone.
  - Any user data specified at instance launch can also be accessed at: `http://169.254.169.254/latest/user-data/`
- It can be used to configure or manage a running instance.
  - For example, author a configuration script that reads the metadata and uses it to configure applications or OS settings.

Section 2 key takeaways

- **Amazon EC2** enables you to run Windows and Linux **virtual machines** in the cloud.
- You launch **EC2 instances** from an **AMI** template into a VPC in your account.
- You can choose from many **instance types**. Each instance type offers different combinations of CPU, RAM, storage, and networking capabilities.
- You can configure **security groups** to control access to instances (specify allowed ports and source).
- **User data** enables you to specify a script to run the first time that an instance launches.
- Only **instances that are backed by Amazon EBS can be stopped.**
- You can use **Amazon CloudWatch** to capture and review metrics on EC2 instances.

Section 3: Amazon EC2 cost optimization

Amazon EC2 pricing models

On-Demand Instances

- Pay by the hour
- No long-term commitments.
- Eligible for the AWS Free Tier.

Dedicated Hosts

- A physical server with EC2 instance capacity fully dedicated to your use.

Dedicated Instances

- Instances that run in a VPC on hardware that is dedicated to a single customer.

Reserved Instances

- Full, partial, or no upfront payment for instance you reserve.
- Discount on hourly charge for that instance.
- 1-year or 3-year term.

Scheduled Reserved Instances

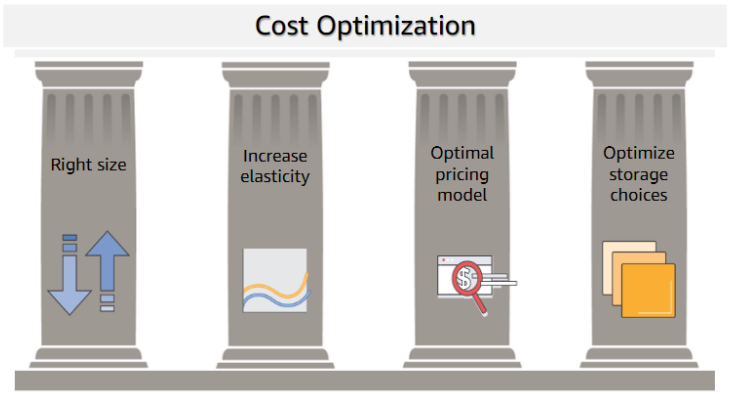
- Purchase a capacity reservation that is always available on a recurring schedule you specify.
- 1-year term.

Spot Instances

- Instances run as long as they are available and your bid is above the Spot Instance price.
- They can be interrupted by AWS with a 2-minute notification.
- Interruption options include terminated, stopped or hibernated.
- Prices can be significantly less expensive compared to On-Demand Instances
- Good choice when you have flexibility in when your applications can run.

Per second billing available for On-Demand Instances, Reserved Instances, and Spot Instances that run Amazon Linux or Ubuntu.

The four pillars of cost optimization



Pillar 1: Right size

Pillars:

1. Right size
2. Increase elasticity
3. Optimal pricing model
4. Optimize storage choices

✓Provision instances to match the need

- CPU, memory, storage, and network throughput
- Select appropriate instance types for your use

✓Use Amazon CloudWatch metrics

- How idle are instances? When?
- Downsize instances

✓Best practice: Right size, then reserve

Pillar 2: Increase elasticity

Pillars:

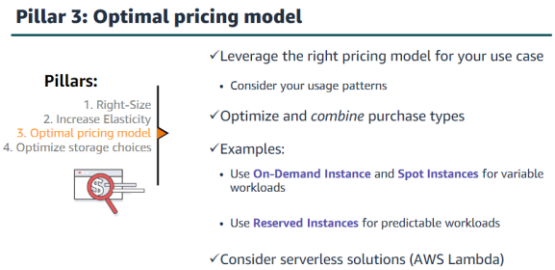
1. Right-Size
2. Increase Elasticity
3. Optimal pricing model
4. Optimize storage choices

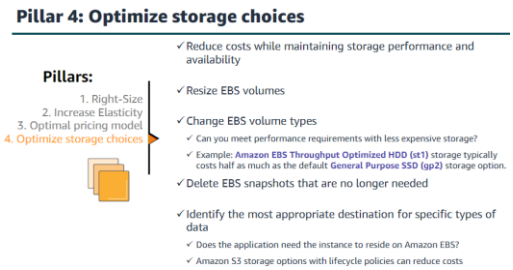
✓**Stop** or **hibernate** Amazon EBS-backed instances that are not actively in use

- Example: non-production development or test instances

✓Use **automatic scaling** to match needs based on usage

- Automated and time-based elasticity

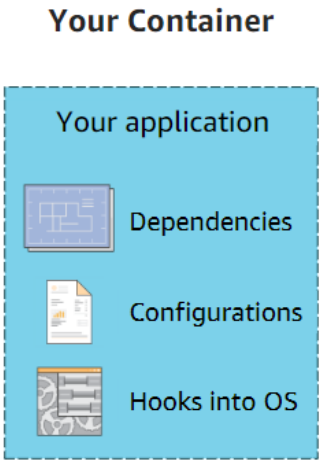




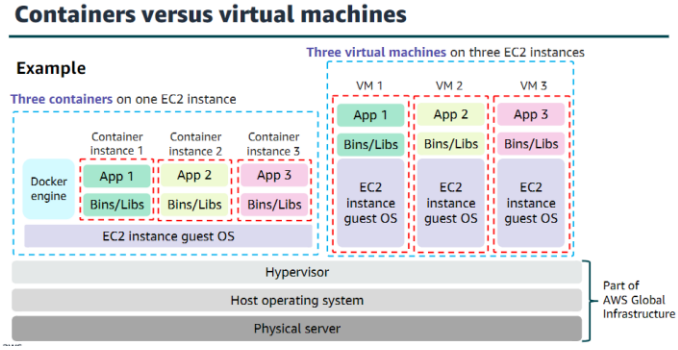
Section 4: Container services

Container basics

- **Containers** are a method of operating system virtualization.
- Benefits –
  - Repeatable.
  - Self-contained environments.
  - Software runs the same in different environments.
    - Developer's laptop, test, production.
  - Faster to launch and stop or terminate than virtual machines



- Docker is a software platform that enables you to build, test, and deploy applications quickly.
- You run containers on Docker.
- Containers are created from a template called an image.
- A container has everything a software application needs to run.



Amazon Elastic Container Service (Amazon ECS) is a highly scalable, high-performance container management service that supports Docker containers. Amazon ECS enables you to easily run applications on a managed cluster of Amazon EC2 instances.

Essential Amazon ECS features include the ability to:

- **Launch** up to tens of thousands of Docker containers in seconds
- **Monitor** container deployment
- **Manage** the state of the cluster that runs the containers
- **Schedule** containers by using a built-in scheduler or a third-party scheduler (for example, Apache Mesos or Blox)

To prepare your application to run on Amazon ECS, you create a task definition which is a text file that describes one or more containers, up to a maximum of ten, that form your application.
A task is the instantiation of a task definition within a cluster. You can specify the number of tasks that will run on your cluster. The Amazon ECS task scheduler is responsible for placing tasks within your cluster.

When Amazon ECS runs the containers that make up your task, it places them on an ECS cluster. The cluster (when you choose the EC2 launch type) consists of a group of EC2 instances each of which is running an Amazon ECS container agent.

Amazon ECS cluster options

- **Key question**: Do ***you*** want to manage the Amazon ECS cluster that runs the containers?
  - If **yes**, create an **Amazon ECS cluster backed by Amazon EC2** (provides more granular control over infrastructure)
  - If no, create an Amazon ECS cluster backed by AWS Fargate (easier to maintain, focus on your applications)

What is Kubernetes?

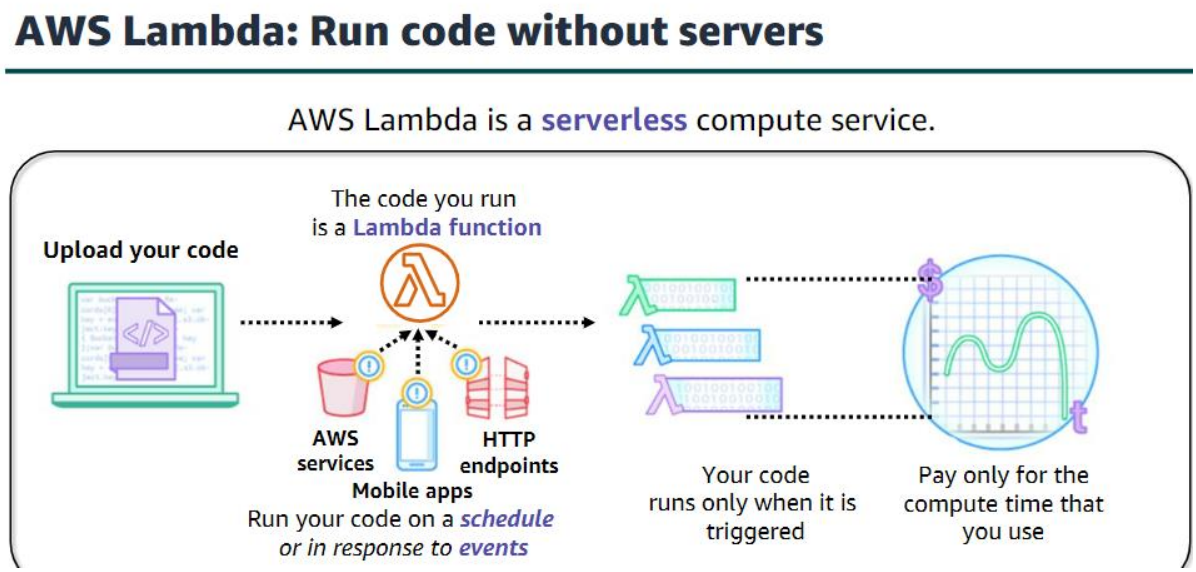
- Kubernetes is open source software for container orchestration.
  - Deploy and **manage containerized applications** *at scale*.
  - The same toolset can be used on premises and in the cloud.
- Complements Docker.
  - Docker enables you to run multiple containers on a single OS host.
  - Kubernetes **orchestrates** multiple Docker hosts (nodes).
- Automates –
  - Container provisioning.
  - Networking.
  - Load distribution.
  - Scaling.

Amazon Elastic Container Registry (Amazon ECR) is a fully managed Docker container registry that makes it easy for developers to store, manage, and deploy Docker container images. It is integrated with Amazon ECS, so you can store, run, and manage container images for applications that run on Amazon ECS. Specify the Amazon ECR repository in your task definition, and Amazon ECS will retrieve the appropriate images for your applications.

Amazon ECR supports Docker Registry HTTP API version 2, which enables you to interact with Amazon ECR by using Docker CLI commands or your preferred Docker tools. Thus, you can maintain your existing development workflow and access Amazon ECR from any Docker environment—whether it is in the cloud, on-premises, or on your local machine.

- **Containers** can hold everything that an application needs to run.
- **Docker** is a software platform that packages software into containers.
  - A single application can span multiple containers.
- Amazon Elastic Container Service (**Amazon ECS**) orchestrates the running of Docker containers.
- **Kubernetes** is open source software for container orchestration.
- Amazon Elastic Kubernetes Service (**Amazon EKS**) enables you to run Kubernetes on AWS
- Amazon Elastic Container Registry (**Amazon ECR**) enables you to store, manage, and deploy your Docker containers.

Section 5: Introduction to AWS Lambda

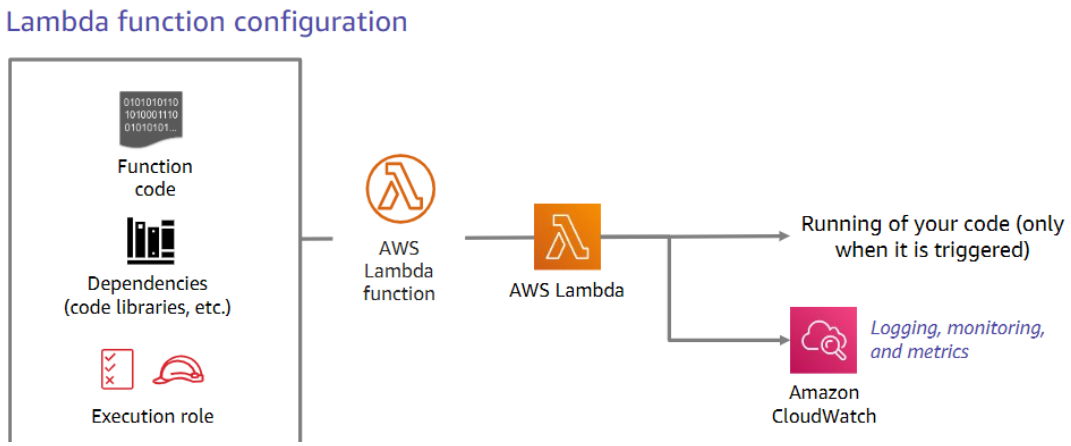


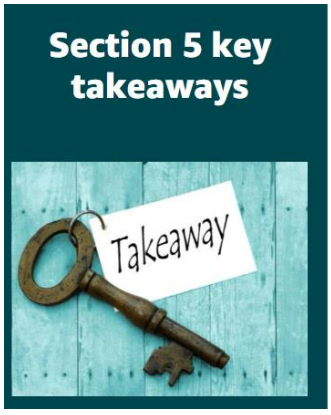
AWS offers many compute options. For example, Amazon EC2 provides virtual machines. As another example, Amazon ECS and Amazon EKS are container-based compute services. However, there is another approach to compute that does not require you to provision or manage servers. This third approach is often referred to as serverless computing
AWS Lambda is an event-driven, serverless compute service. Lambda enables you to run code without provisioning or managing servers.

- It supports multiple programming languages
- Completely automated administration
- Built-in fault tolerance
- It supports the orchestration of multiple functions
- Pay-per-use pricing

An event source is an AWS service or a developer-created application that produces events that trigger an AWS Lambda function to run. Lambda can pull records from an Amazon Simple Queue Service (Amazon SQS) queue and run a Lambda function for each fetched message. Lambda can similarly read events from Amazon DynamoDB. Some services, such as Elastic Load Balancing (Application Load Balancer) and Amazon API Gateway can invoke your Lambda function directly.

You can invoke Lambda functions directly with the Lambda console, the Lambda API, the AWS software development kit (SDK), the AWS CLI, and AWS toolkits.





- **Serverless computing** enables you to build and run applications and services without provisioning or managing servers.
- **AWS Lambda is a serverless compute service** that provides built-in fault tolerance and automatic scaling.
- An **event source** is an AWS service or developer-created application that triggers a Lambda function to run.
- The maximum memory allocation for a single Lambda function is 10,240 MB.
- The maximum run time for a Lambda function is 15 minutes.

Section 6: Introduction to AWS Elastic Beanstalk

AWS Elastic Beanstalk

- An easy way to get **web applications** up and running
- A **managed service** that automatically handles –
  - Infrastructure provisioning and configuration
  - Deployment
  - Load balancing
  - Automatic scaling
  - Health monitoring
  - Analysis and debugging
  - Logging
- No additional charge for Elastic Beanstalk
  - Pay only for the underlying resources that are used

AWS Elastic Beanstalk deploys your code on Apache Tomcat for Java applications; Apache HTTP Server for PHP and Python applications; NGINX or Apache HTTP Server for Node.js applications; Passenger or Puma for Ruby applications; and Microsoft Internet Information Services (IIS) for .NET applications, Java SE, Docker, and Go.

Module summary

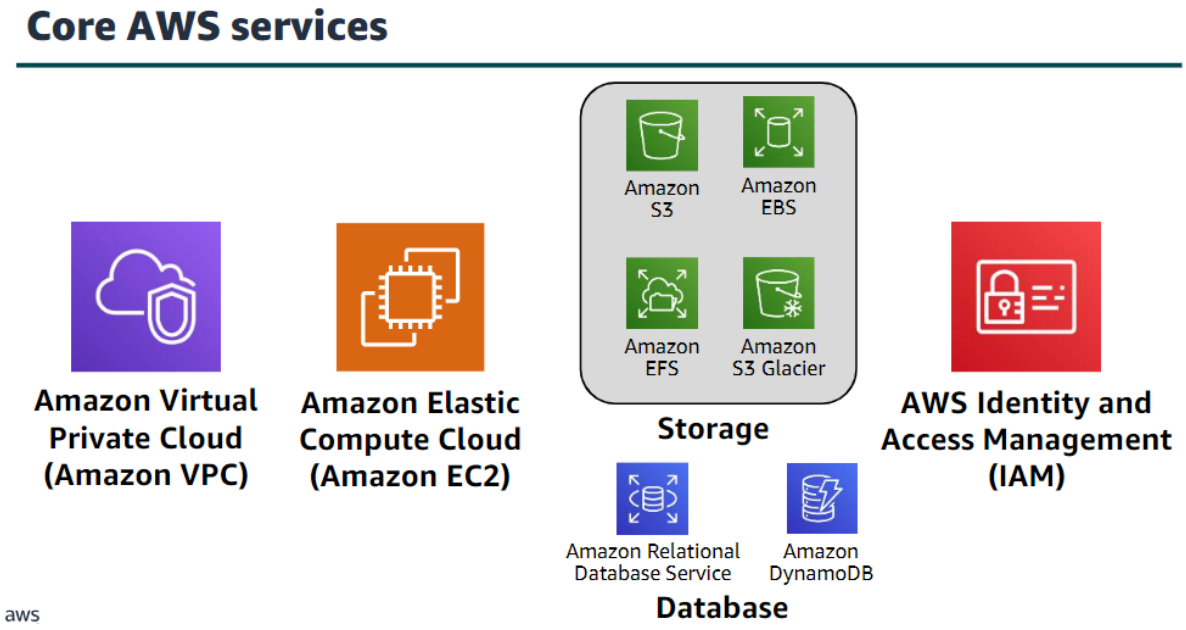
In summary, in this module, you learned how to:

- Provide an overview of different AWS compute services in the cloud
- Demonstrate why to use Amazon Elastic Compute Cloud (Amazon EC2)
- Identify the functionality in the Amazon EC2 console
- Perform basic functions in Amazon EC2 to build a virtual computing environment
- Identify Amazon EC2 cost optimization elements
- Demonstrate when to use AWS Elastic Beanstalk
- Demonstrate when to use AWS Lambda
- Identify how to run containerized applications in a cluster of managed servers

Module 7: Storage

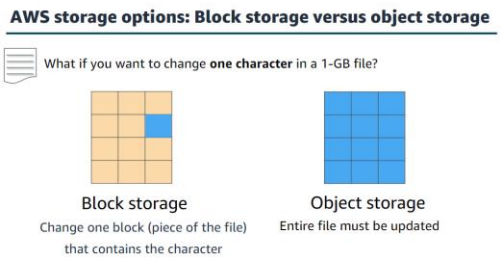
This module addresses the following topics:

- Amazon Elastic Block Store (Amazon EBS)
- Amazon Simple Storage Service (Amazon S3)
- Amazon Elastic File System (Amazon EFS)
- Amazon Simple Storage Service Glacier



Section 1: Amazon Elastic Block Store (Amazon EBS)

Amazon EBS provides persistent block storage volumes for use with Amazon EC2 instances. Persistent storage is any data storage device that retains data after power to that device is shut off.
It is also sometimes called non-volatile storage. Each Amazon EBS volume is automatically replicated within its Availability Zone to protect you from component failure. It is designed for high availability and durability.
Amazon EBS volumes provide the consistent and low-latency performance that is needed to run your workloads. With Amazon EBS, you can scale your usage up or down within minutes, while paying a low price for only what you provision.



Amazon EBS

Amazon EBS enables you to **create individual storage volumes** and **attach them** to an Amazon EC2 instance:

- Amazon EBS offers block-level storage.
- Volumes are automatically replicated within its Availability Zone.
- It can be backed up automatically to Amazon S3 through snapshots.
- Uses include –
  - Boot volumes and storage for Amazon Elastic Compute Cloud (Amazon EC2) instances
  - Data storage with a file system
  - Database hosts
  - Enterprise applications

Solid State Drives (SSD)		Hard Disk Drives (HDD)	
General Purpose	Provisioned IOPS	Throughput-Optimized	Cold
• This type is recommended for most workloads	• Critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/second of throughput per volume	• Streaming workloads that require consistent, fast throughput at a low price	• Throughput-oriented storage for large volumes of data that is infrequently accessed
• System boot volumes	• Large database workloads	• Big data	• Scenarios where the lowest storage cost is important
• Virtual desktops		• Data warehouses	• It cannot be a boot volume
• Low-latency interactive applications		• Log processing	
• Development and test environments		• It cannot be a boot volume	

Amazon EBS features

- Snapshots –
  - Point-in-time snapshots
  - Recreate a new volume at any time
- Encryption –
  - Encrypted Amazon EBS volumes
  - No additional cost
- Elasticity –
  - Increase capacity
  - Change to different types

Amazon EBS: Volumes, IOPS, and pricing

1. **Volumes –**
   - Amazon EBS volumes persist independently from the instance.
   - All volume types are charged by the amount that is provisioned per month.
2. **IOPS –**
   - General Purpose SSD:
     - Charged by the amount that you provision in GB per month until storage is released.
   - Magnetic:
     - Charged by the number of requests to the volume.
   - Provisioned IOPS SSD:
     - Charged by the amount that you provision in IOPS (multiplied by the percentage of days that you provision for the month).

Amazon EBS: Snapshots and data transfer

3. **Snapshots –**
   - Added cost of Amazon EBS snapshots to Amazon S3 is per GB-month of data stored.

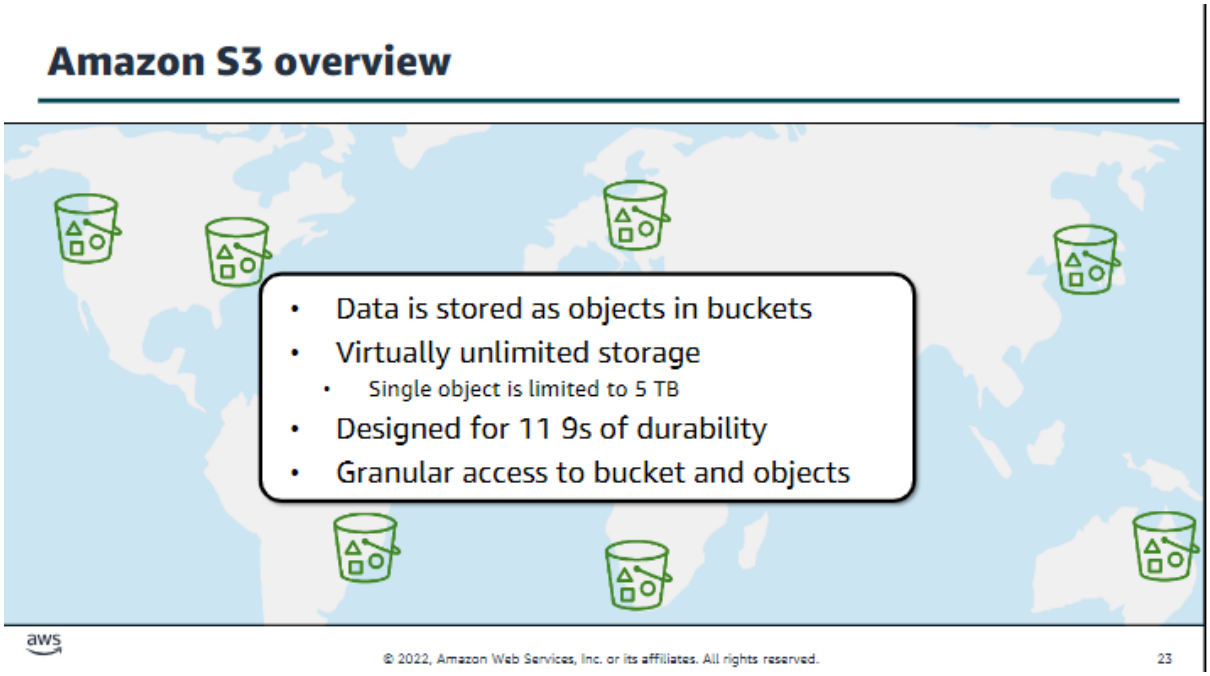
4. **Data transfer –**
   - Inbound data transfer is free.
   - Outbound data transfer across Regions incurs charges.

Section 1 key takeaways

Amazon EBS features:

- Persistent and customizable block storage for Amazon EC2
- HDD and SSD types
- Replicated in the same Availability Zone
- Easy and transparent encryption
- Elastic volumes
- Back up by using snapshots

Section 2: Amazon Simple Storage Service (Amazon S3)



Amazon S3 offers a range of object-level storage classes that are designed for different use cases.
These classes include:

•Amazon S3 Standard –Amazon S3 Standard is designed for high durability, availability, and performance object storage for frequently accessed data. Because it delivers low latency and high throughput, Amazon S3 Standard is appropriate for a variety of use cases, including cloud applications, dynamic websites, content distribution, mobile and gaming applications, and big data analytics.

•Amazon S3 Intelligent-Tiering–The Amazon S3 Intelligent-Tiering storage class is designed to optimize costs by automatically moving data to the most cost-effective access tier, without performance impact or operational overhead. For a small monthly monitoring and automation fee per object, Amazon S3 monitors access patterns of the objects in Amazon S3 Intelligent-Tiering, and moves the objects that have not been accessed for 30 consecutive days to the infrequent access tier. If an object in the infrequent access tier is

accessed, it is automatically moved back to the frequent access tier.

•Amazon S3 Standard-Infrequent Access (Amazon S3 Standard-IA) –The Amazon S3 Standard-IA storage class is used for data that is accessed less frequently, but requires rapid access when needed. Amazon S3 Standard-IA is designed to provide the high durability, high throughput, and low latency of Amazon S3 Standard, with a low per-GB storage price and per-GB retrieval fee.

•Amazon S3 Glacier –Amazon S3 Glacier is a secure, durable, and low-cost storage class for data archiving. You can reliably store any amount of data at costs that are competitive with—or cheaper than—on-premises solutions. To keep costs low yet suitable for varying needs, Amazon S3 Glacier provides three retrieval options that range from a few minutes to hours.

•Amazon S3 Glacier Deep Archive –Amazon S3 Glacier Deep Archive is the lowest-cost storage class for Amazon S3. It supports long-term retention and digital preservation for data that might be accessed once or twice in a year. It is designed for customers—particularly customers in highly regulated industries, such as financial services, healthcare, and public sectors—that retain datasets for 7–10 years (or more) to meet regulatory compliance requirements.

Section 3: Amazon Elastic File System (Amazon EFS)

Amazon Elastic File System (Amazon EFS) provides simple, scalable, elastic file storage for use with AWS services and on-premises resources. It offers a simple interface that enables you to create and configure file systems quickly and easily. Amazon EFS is built to dynamically scale on demand without disrupting applications—it will grow and shrink automatically as you add and remove files. It is designed so that your applications have the storage they need, when they need it.

- Amazon EFS provides file storage over a network.
- Perfect for big data and analytics, media processing workflows, content management, web serving, and home directories.
- Fully managed service that eliminates storage administration tasks.
- Accessible from the console, an API, or the CLI.
- Scales up or down as files are added or removed and you pay for what you use.

Section 4: Amazon S3 Glacier

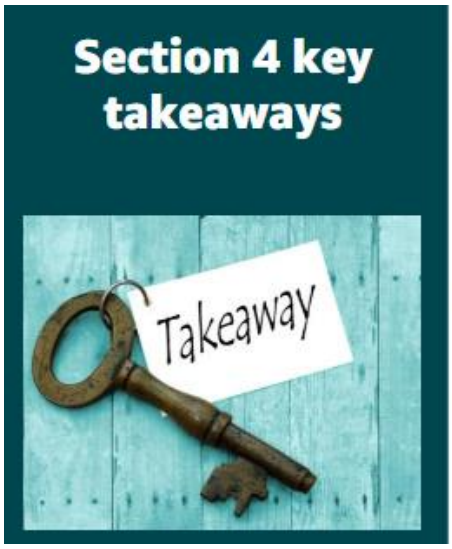
Amazon S3 Glacier review

Amazon S3 Glacier is a data archiving service that is designed for security, durability, and an extremely low cost.

- Amazon S3 Glacier is designed to provide 11 9s of durability for objects.
- It supports the encryption of data in transit and at rest through Secure Sockets Layer (SSL) or Transport Layer Security (TLS).
- The Vault Lock feature enforces compliance through a policy.
- Extremely low-cost design works well for long-term archiving.
  - Provides three options for access to archives—expedited, standard, and bulk—retrieval times range from a few minutes to several hours.

Three options are available for retrieving data with varying access times and cost: expedited, standard, and bulk retrievals. They are listed as follows:

- **Expedited** retrievals are typically made available within 1 – 5 minutes (highest cost).
- **Standard** retrievals typically complete within 3 – 5 hours (less than expedited, more than bulk).
- **Bulk** retrievals typically complete within 5 – 12 hours (lowest cost).



- Amazon S3 Glacier is a data archiving service that is designed for security, durability, and an extremely low cost.
- Amazon S3 Glacier pricing is based on Region.
- Its extremely low-cost design works well for long-term archiving.
- The service is designed to provide 11 9s of durability for objects.

Module 8: Databases

This module will address the following topics:
•Amazon Relational Database Service (Amazon RDS)
•Amazon DynamoDB
•Amazon Redshift
•Amazon Aurora

When to Use Amazon RDS

Use Amazon RDS when your application requires:

- Complex transactions or complex queries
- A medium to high query or write rate – Up to 30,000 IOPS (15,000 reads + 15,000 writes)
- No more than a single worker node or shard
- High durability

- **Do not use Amazon RDS when your application requires:**
  - Massive read/write rates (for example, 150,000 write/second)
  - Sharding due to high data size or throughput demands
  - Simple GET or PUT requests and queries that a NoSQL database can handle
  - Relational database management system (RDBMS) customization

Section 1 key takeaways

- With Amazon RDS, you can set up, operate, and scale relational databases in the cloud.
- Features –
  - Managed service
  - Accessible via the console, AWS Command Line Interface (AWS CLI), or application programming interface (API) calls
  - Scalable (compute and storage)
  - Automated redundancy and backup are available
  - Supported database engines:
    - Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle, Microsoft SQL Server

Section 2: Amazon DynamoDB

What is Amazon DynamoDB?

Fast and flexible NoSQL database service for any scale

Amazon DynamoDB

- NoSQL database tables
- Virtually unlimited storage
- Items can have differing attributes
- Low-latency queries
- Scalable read/write throughput

Section 2 key takeaways

Amazon DynamoDB:

- Runs exclusively on SSDs.
- Supports document and key-value store models.
- Replicates your tables automatically across your choice of AWS Regions.
- Works well for mobile, web, gaming, adtech, and Internet of Things (IoT) applications.
- Is accessible via the console, the AWS CLI, and API calls.
- Provides consistent, single-digit millisecond latency at any scale.
- Has no limits on table size or throughput.

aws

Section 3: Amazon Redshift

Amazon Redshift is a fast, fully managed data warehouse that makes it simple and cost-effective to analyze all your data by using standard SQL and your existing business intelligence (BI) tools.

Section 3 key takeaways

Amazon Redshift features:

- Fast, fully managed data warehouse service
- Easily scale with no downtime
- Columnar storage and parallel processing architectures
- Automatically and continuously monitors cluster
- Encryption is built in

Section 4: Amazon Aurora

Amazon Aurora

Amazon Aurora

- Enterprise-class relational database
- Compatible with MySQL or PostgreSQL
- Automate time-consuming tasks (such as provisioning, patching, backup, recovery, failure detection, and repair).

Section 4 key takeaways

Amazon Aurora features:

- High performance and scalability
- High availability and durability
- Multiple levels of security
- Compatible with MySQL and PostgreSQL
- Fully managed

The right tool for the right job

What are my requirements?	
Enterprise-class relational database	Amazon RDS
Fast and flexible NoSQL database service for any scale	Amazon DynamoDB
Operating system access or application features that are not supported by AWS database services	Databases on Amazon EC2
Specific case-driven requirements (machine learning, data warehouse, graphs)	AWS purpose-built database services

aws

Module 9: Cloud Architecture

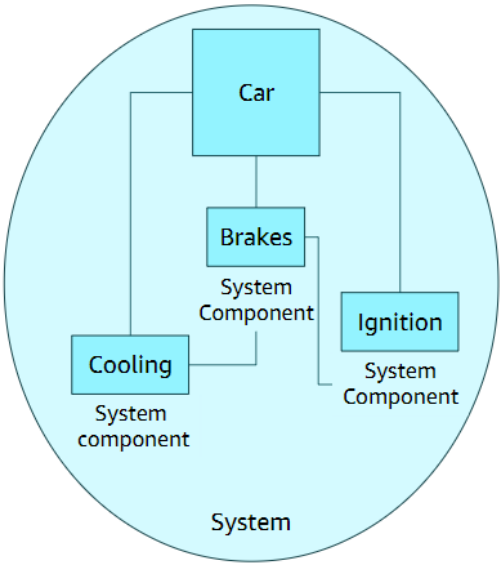
Section 1: AWS Well-Architected Framework

- The AWS Well-Architected Framework provides a consistent approach to evaluate cloud architectures and guidance to help implement designs.
- The AWS Well-Architected Framework documents a set of design principles and best practices that enable you to understand if a specific architecture aligns well with cloud best practices.
- The AWS Well-Architected Framework is organized into six pillars.
- Each pillar includes its own set of design principles and best practices.

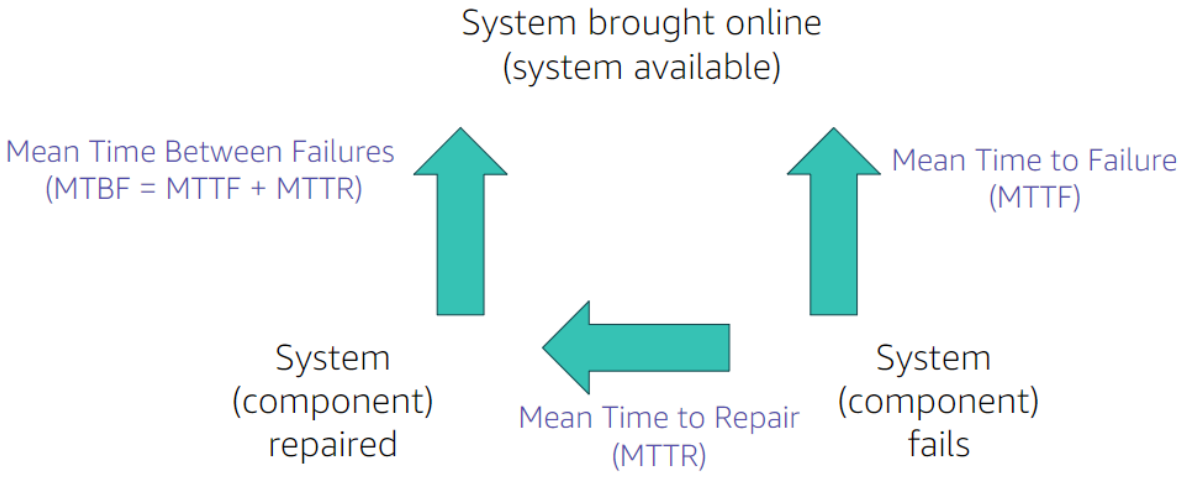
Section 2: Reliability and availability

Reliability

- A measure of your system's ability to provide functionality when desired by the user.
- System includes all system components: hardware, firmware, and software.
- Probability that your entire system will function as intended for a specified period.
- Mean time between failures (MTBF) = total time in service/number of failures



Understanding reliability metrics



High availability

- System can withstand some measure of degradation while still remaining available.
- Downtime is minimized.
- Minimal human intervention is required.

Availability tiers

Availability	Max Disruption (per year)	Application Category
99%	3 days 15 hours	Batch processing, data extraction, transfer, and load jobs
99.9%	8 hours 45 minutes	Internal tools like knowledge management, project tracking
99.95%	4 hours 22 minutes	Online commerce, point of sale
99.99%	52 minutes	Video delivery, broadcast systems
99.999%	5 minutes	ATM transactions, telecommunications systems

Section 2 key takeaways

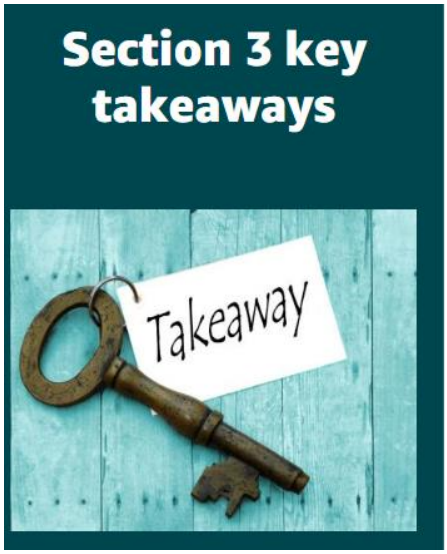
- Reliability is a measure of your system's ability to provide functionality when desired by the user, and it can be measured in terms of MTBF.
- Availability is the percentage of time that a system is operating normally or correctly performing the operations expected of it (or normal operation time over total time).
- Three factors that influence the availability of your applications are fault tolerance, scalability, and recoverability.
- You can design your workloads and applications to be highly available, but there is a cost tradeoff to consider.

Section 3: AWS Trusted Advisor

AWS Trusted Advisor

AWS Trusted Advisor

- Online tool that provides real-time guidance to help you provision your resources following AWS best practices.
- Looks at your entire AWS environment and gives you real-time recommendations in five categories.



- AWS Trusted Advisor is an online tool that provides real-time guidance to help you provision your resources by following AWS best practices.
- AWS Trusted Advisor looks at your entire AWS environment and gives you real-time recommendations in five categories.
- You can use AWS Trusted Advisor to help you optimize your AWS environment as soon as you start implementing your architecture designs.

Module 10: Automatic Scaling and Monitoring

Section 1: Elastic Load Balancing

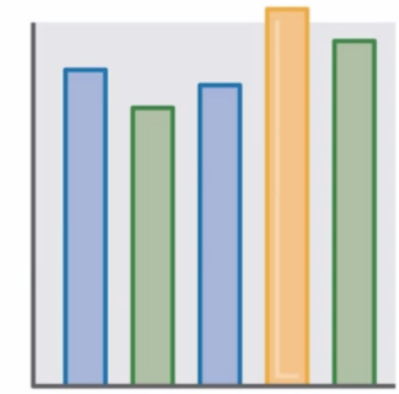
Types of load balancers

Application Load Balancer	Network Load Balancer	Classic Load Balancer (Previous Generation)
• Load balancing of HTTP and HTTPS traffic	• Load balancing of TCP, UDP, and TLS traffic where extreme performance is required	• Load balancing of HTTP, HTTPS, TCP, and SSL traffic
• Routes traffic to targets based on content of request • Provides advanced request routing targeted at the delivery of modern application architectures, including microservices and containers	• Routes traffic to targets based on IP protocol data • Can handle millions of requests per second while maintaining ultra-low latencies • Is optimized to handle sudden and volatile traffic patterns	• Load balancing across multiple EC2 instances
• Operates at the application layer (OSI model layer 7)	• Operates at the transport layer (OSI model layer 4)	• Operates at both the application and transport layers.

Activity: Elastic Load Balancing Answers

You must support traffic to a containerized application.	Application Load Balancer
You have extremely spiky and unpredictable TCP traffic.	Network Load Balancer
You need simple load balancing with multiple protocols.	Classic Load Balancer
You need to support a static or Elastic IP address, or an IP target outside a VPC.	Network Load Balancer
You need a load balancer that can handle millions of requests per second while maintaining low latencies.	Network Load Balancer
You must support HTTPS requests.	Application Load Balancer

Load balancer monitoring



- **Amazon CloudWatch metrics** – Used to verify that the system is performing as expected and creates an alarm to initiate an action if a metric goes outside an acceptable range.
- **Access logs** – Capture detailed information about requests sent to your load balancer.
- **AWS CloudTrail logs** – Capture the who, what, when, and where of API interactions in AWS services.

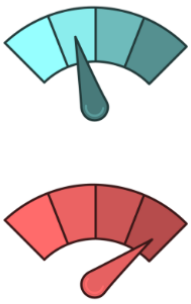
Section 1 key takeaways

- Elastic Load Balancing distributes incoming application or network traffic across multiple targets in one or more Availability Zones.
- Elastic Load Balancing supports three types of load balancers:
  - Application Load Balancer
  - Network Load Balancer
  - Classic Load Balancer
- ELB offers instance health checks, security, and monitoring.

Section 2: Amazon CloudWatch

Amazon CloudWatch

Amazon CloudWatch



- Monitors –
  - AWS resources
  - Applications that run on AWS
- Collects and tracks –
  - Standard metrics
  - Custom metrics
- Alarms –
  - Send notifications to an Amazon SNS topic
  - Perform Amazon EC2 Auto Scaling or Amazon EC2 actions
- Events –
  - Define rules to match changes in AWS environment and route these events to one or more target functions or streams for processing

Activity: Amazon CloudWatch Answers

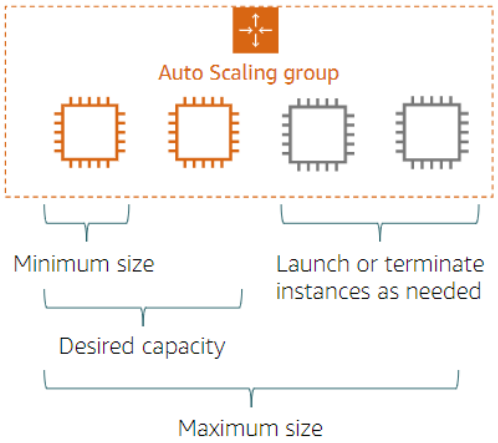
Amazon EC2	If average CPU utilization is > 60% for 5 minutes…	Correct!
Amazon RDS	If the number of simultaneous connections is > 10 for 1 minute…	Correct!
Amazon S3	If the maximum bucket size in bytes is around 3 for 1 day…	Incorrect. *Around* is not a threshold option. You must specify a threshold of >, >=, <=, or <.
lastic Load Balancing	If the number of healthy hosts is < 5 for 10 minutes…	Correct!
Amazon Elastic Block Store	If the volume of read operations is > 1,000 for 10 seconds…	Incorrect. You must specify a statistic (for example, *average volume*).

Section 2 key takeaways

- Amazon CloudWatch helps you monitor your AWS resources—and the applications that you run on AWS—in real time.
- CloudWatch enables you to –
  - Collect and track standard and custom metrics.
  - Set alarms to automatically send notifications to SNS topics, or perform Amazon EC2 Auto Scaling or Amazon EC2 actions.
  - Define rules that match changes in your AWS environment and route these events to targets for processing.

Section 3: Amazon EC2 Auto Scaling

Auto Scaling groups



An **Auto Scaling group** is a collection of EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management.

With Amazon EC2 Auto Scaling, launching instances is referred to as scaling out, and terminating instances is referred to as scaling in.

AWS Auto Scaling

AWS Auto Scaling

- Monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost
- Provides a simple, powerful user interface that enables you to build scaling plans for resources, including –
  - Amazon EC2 instances and Spot Fleets
  - Amazon Elastic Container Service (Amazon ECS) Tasks
  - Amazon DynamoDB tables and indexes
  - Amazon Aurora Replicas

Section 3 key takeaways

- Scaling enables you to respond quickly to changes in resource needs.
- Amazon EC2 Auto Scaling maintains application availability by automatically adding or removing EC2 instances.
- An Auto Scaling group is a collection of EC2 instances.
- A launch configuration is an instance configuration template.
- Dynamic scaling uses Amazon EC2 Auto Scaling, CloudWatch, and Elastic Load Balancing.
- AWS Auto Scaling is a separate service from Amazon EC2 Auto Scaling.