AWS Module 2 - Compute in the Cloud

What is meant by compute power?

- Compute power refers to the ability of a computer system or infrastructure to perform various computational tasks and operations. It is a measure of the processing capabilities of a computing environment, including its ability to execute instructions, process data, and perform calculations.

Why do we need compute power?

- You're going to need servers to power your business and your applications. You need raw compute capacity to host your applications and provide the compute power that your business needs.
- Hosting- practice of deploying and running software applications on computer servers or cloud infrastructure so that they are accessible and operational for users over a network, typically the internet. When you host an application, you are making it available for users to use, interact with, or access remotely.

What is EC2?

- When you're working with AWS, those servers are virtual. And the service you use to gain access to virtual servers is called EC2.
- With EC2, it's much easier to get started. AWS took care of the hard part for you already. AWS already built and secured the data centers. AWS has already bought the servers, racked and stacked them, and they are already online ready to be used. AWS is constantly operating a massive amount of compute capacity.
- Because with EC2, you only pay for running instances, not stopped or terminated instances.
- When you provision an EC2 instance, you can choose the operating system based on either Windows or Linux. You can provision thousands of EC2 instances on demand. With a blend of operating systems and configurations to power your business' different applications.

Explain Multitenancy

- EC2 runs on top of physical host machines managed by AWS using virtualization technology. When you spin up an EC2 instance, you aren't necessarily taking an entire host to yourself. Instead, you are sharing the host with multiple other instances, otherwise known as virtual machines. And a hypervisor running on the host machine is responsible for sharing the underlying physical resources between the virtual machines. This idea of sharing underlying hardware is called multitenancy.
- The hypervisor is responsible for coordinating this multitenancy and it is managed by AWS. The hypervisor is responsible for isolating the virtual machines from each other as they share resources from the host.

EC2 Instance Types

Explain different Instance Types

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

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

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

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

3.**Memory optimized instances** are designed to deliver fast performance for workloads that process large datasets in memory. In computing, memory is a temporary storage area. It holds all the data and instructions that a central processing unit (CPU) needs to be able to complete actions. Before a computer program or application is able to run, it is loaded from storage into memory. This preloading process gives the CPU direct access to the computer program.

Suppose that you have a workload that requires large amounts of data to be preloaded before running an application. This scenario might be a high-performance database or a workload that involves performing real-time processing of a large amount of unstructured data. In these types of use cases, consider using a memory optimized instance. Memory optimized instances enable you to run workloads with high memory needs and receive great performance.

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

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

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

In computing, the term input/output operations per second (IOPS) is a metric that measures the performance of a storage device. It indicates how many different input or output operations a device can perform in one second. Storage optimized instances are designed to deliver tens of thousands of low-latency, random IOPS to applications.

You can think of input operations as data put into a system, such as records entered into a database. An output operation is data generated by a server. An example of output might be the analytics performed on the records in a database. If you have an application that has a high IOPS requirement, a storage optimized instance can provide better performance over other instance types not optimized for this kind of use case.

Which Amazon EC2 instance type is suitable for data warehousing applications?

- ✗ Memory optimized
- ✓ Storage optimized
- ✗ General purpose
- ✗ Compute optimized

Which Amazon EC2 instance type balances compute, memory, and networking resources?

- ✗ Memory optimized
- ✗ Storage optimized
- ✓ General purpose
- ✗ Compute optimized

Which Amazon EC2 instance type is ideal for high-performance databases?

- ✓ Memory optimized
- ✗ Storage optimized
- ✗ General purpose
- ✗ Compute optimized

Which Amazon EC2 instance type offers high-performance processors?

- ✗ Memory optimized
- ✗ Storage optimized
- ✗ General purpose
- ✓ Compute optimized

Amazon EC2 pricing

1. **On-Demand Instances**

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

- Sample use cases for On-Demand Instances include developing and testing applications and running applications that have unpredictable usage patterns. Some applications experience highly variable or unpredictable usage patterns. For example, an e-commerce website might see a significant increase in traffic during holiday sales but lower traffic during the rest of the year.
- On-Demand Instances are not recommended for workloads that last a year or longer because these workloads can experience greater cost savings using Reserved Instances

2.**Reserved Instances** are a billing discount applied to the use of On-Demand Instances in your account. There are two available types of Reserved Instances:

- Standard Reserved Instances
- Convertible Reserved Instances

- You can purchase Standard Reserved and Convertible Reserved Instances for a 1-year or 3-year term. You realize greater cost savings with the 3-year option.

Standard Reserved Instances:

- This option is a good fit if you know the EC2 instance type and size you need for your steady-state applications and in which AWS Region you plan to run them. Reserved Instances require you to state the following qualifications:
  - Instance type and size: For example, m5.xlarge
  - Platform description (operating system):For example, Microsoft Windows Server or Red Hat Enterprise Linux
  - Tenancy: Default tenancy or dedicated tenancy
- You have the option to specify an Availability Zone for your EC2 Reserved Instances. If you make this specification, you get EC2 capacity reservation. This ensures that your desired amount of EC2 instances will be available when you need them.

Convertible Reserved Instances:

- If you need to run your EC2 instances in different Availability Zones or different instance types, then Convertible Reserved Instances might be right for you. Note: You trade in a deeper discount when you require flexibility to run your EC2 instances.
- At the end of a Reserved Instance term, you can continue using the Amazon EC2 instance without interruption. However, you are charged On-Demand rates until you do one of the following:
  - Terminate the instance.
  - Purchase a new Reserved Instance that matches the instance attributes (instance family and size, Region, platform, and tenancy).

3. **EC2 Instance Savings Plans**

- Helps reduce your EC2 instance costs when you make an hourly spend commitment to an instance family and Region for a 1-year or 3-year term. This term commitment results in savings of up to 72 percent compared to On-Demand rates. Any usage up to the commitment is charged at the discounted Savings Plans rate (for example, $10 per hour). Any usage beyond the commitment is charged at regular On-Demand rates.

- The EC2 Instance Savings Plans are a good option if you need flexibility in your Amazon EC2 usage over the duration of the commitment term. You have the benefit of saving costs on running any EC2 instance within an EC2 instance family in a chosen Region (for example, M5 usage in N. Virginia) regardless of Availability Zone, instance size, OS, or tenancy. The savings with EC2 Instance Savings Plans are similar to the savings provided by Standard Reserved Instances.
- Unlike Reserved Instances, however, you don't need to specify up front what EC2 instance type and size (for example, m5.xlarge), OS, and tenancy to get a discount. Further, you don't need to commit to a certain number of EC2 instances over a 1-year or 3-year term. Additionally, the EC2 Instance Savings Plans don't include an EC2 capacity reservation option.

4.**Spot Instances**

- These are ideal for workloads with flexible start and end times, or that can withstand interruptions. Spot Instances use unused Amazon EC2 computing capacity and offer you cost savings at up to 90% off of On-Demand prices.
- Suppose that you have a background processing job that can start and stop as needed (such as the data processing job for a customer survey). You want to start and stop the processing job without affecting the overall operations of your business. If you make a Spot request and Amazon EC2 capacity is available, your Spot Instance launches. However, if you make a Spot request and Amazon EC2 capacity is unavailable, the request is not successful until capacity becomes available. The unavailable capacity might delay the launch of your background processing job.
- After you have launched a Spot Instance, if capacity is no longer available or demand for Spot Instances increases, your instance may be interrupted. This might not pose any issues for your background processing job. However, in the earlier example of developing and testing applications, you would most likely want to avoid unexpected interruptions. Therefore, choose a different EC2 instance type that is ideal for those tasks.

5.**Dedicated Hosts**

- These are physical servers with Amazon EC2 instance capacity that is fully dedicated to your use.
  You can use your existing per-socket, per-core, or per-VM software licenses to help maintain license compliance. You can purchase On-Demand Dedicated Hosts and Dedicated Hosts Reservations.
- Of all the Amazon EC2 options that were covered, Dedicated Hosts are the most expensive

Which Amazon EC2 pricing option provides a discount when you specify a number of EC2 instances to run a specific OS, instance family and size, and tenancy in one Region?

- ○ Convertible Reserved Instances
- ○ EC2 Instance Savings Plans
- ○ Spot Instances
- ◉ Standard Reserved Instances

Which Amazon EC2 pricing option provides a discount when you make an hourly spend commitment to an instance family and Region for a 1-year or 3-year term?

- ⊗ On-demand
- ✓ EC2 Instance Savings Plans
- ⊗ Spot Instances
- ⊗ Reserved Instances

What is Scalability?

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

If you wanted the scaling process to happen automatically, which AWS service would you use?

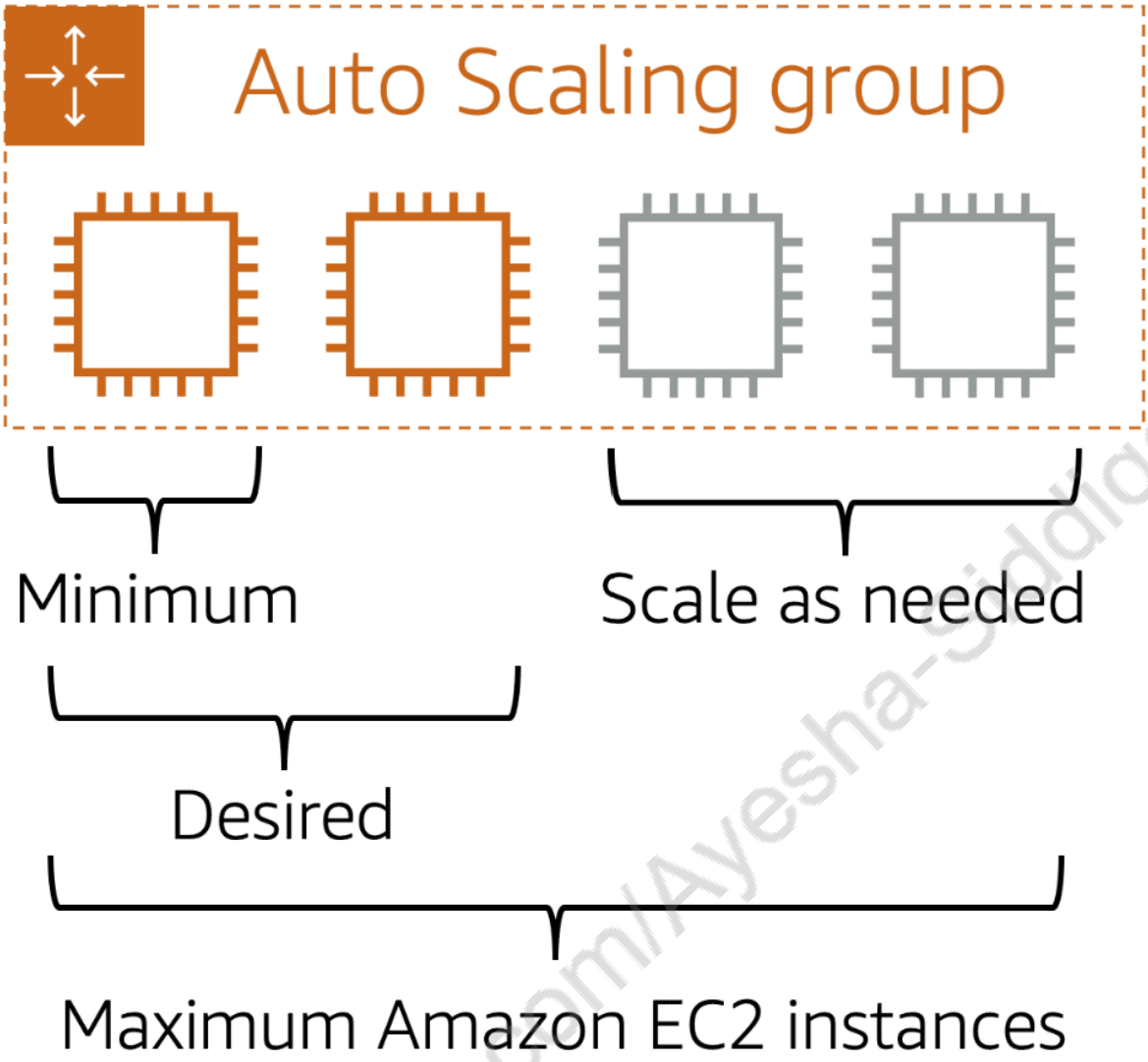
- The AWS service that provides this functionality for Amazon EC2 instances is **Amazon EC2 Auto Scaling**.

*What are the two approaches in **Amazon EC2 Auto Scaling**.?*
Within Amazon EC2 Auto Scaling, you can use two approaches: dynamic scaling and predictive scaling.

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

How does Amazon EC2 Auto Scaling work?

- By adding Amazon EC2 Auto Scaling to an application, you can add new instances to the application when necessary and terminate them when no longer needed.
- Suppose that you are preparing to launch an application on Amazon EC2 instances. When configuring the size of your Auto Scaling group, you might set the minimum number of Amazon EC2 instances at one. This means that at all times, there must be at least one Amazon EC2 instance running.
- When you create an Auto Scaling group, you can set the minimum number of Amazon EC2 instances. The **minimum capacity** is the number of Amazon EC2 instances that launch immediately after you have created the Auto Scaling group.
- Next, you can set the **desired capacity** at two Amazon EC2 instances even though your application needs a minimum of a single Amazon EC2 instance to run.
- The third configuration that you can set in an Auto Scaling group is the **maximum capacity**



Terminologies:
Vertical Scaling (Scale Up):

- **Definition:** Vertical scaling, also known as "scaling up," involves increasing the capacity of an existing resource by adding more resources to it. This typically means upgrading the existing resource, such as a server or virtual machine, with more CPU, memory, storage, or other hardware components.
- **How It Works:** In vertical scaling, you make a single resource more powerful by increasing its capacity. For example, you might upgrade a server by adding more CPU cores, increasing RAM, or attaching additional storage devices.

Horizontal Scaling (Scale Out):

- **Definition:** Horizontal scaling, also known as "scaling out," involves adding more instances or copies of resources to a system to distribute the workload. Instead of making a single resource more powerful, you add more identical resources in parallel.
- **How It Works:** In horizontal scaling, you increase capacity by adding more instances of the same resource. For example, you might add additional web servers to handle increased website traffic, each handling a portion of incoming requests.
- **Use Cases:** Horizontal scaling is commonly used to handle increased traffic, concurrency, or demand in distributed systems. It's particularly effective for web applications, microservices,

and cloud-native architectures, where workloads can be distributed across multiple servers or containers.

Scaling In (Horizontal Scaling):

- **Definition:** Scaling in, also referred to as horizontal scaling, involves reducing the number of resources allocated to a system or application to match a decrease in workload or demand.
- **How It Works:** When scaling in, excess resources are removed or deactivated, reducing the number of active instances. This helps conserve resources and lower operational costs during periods of reduced demand.

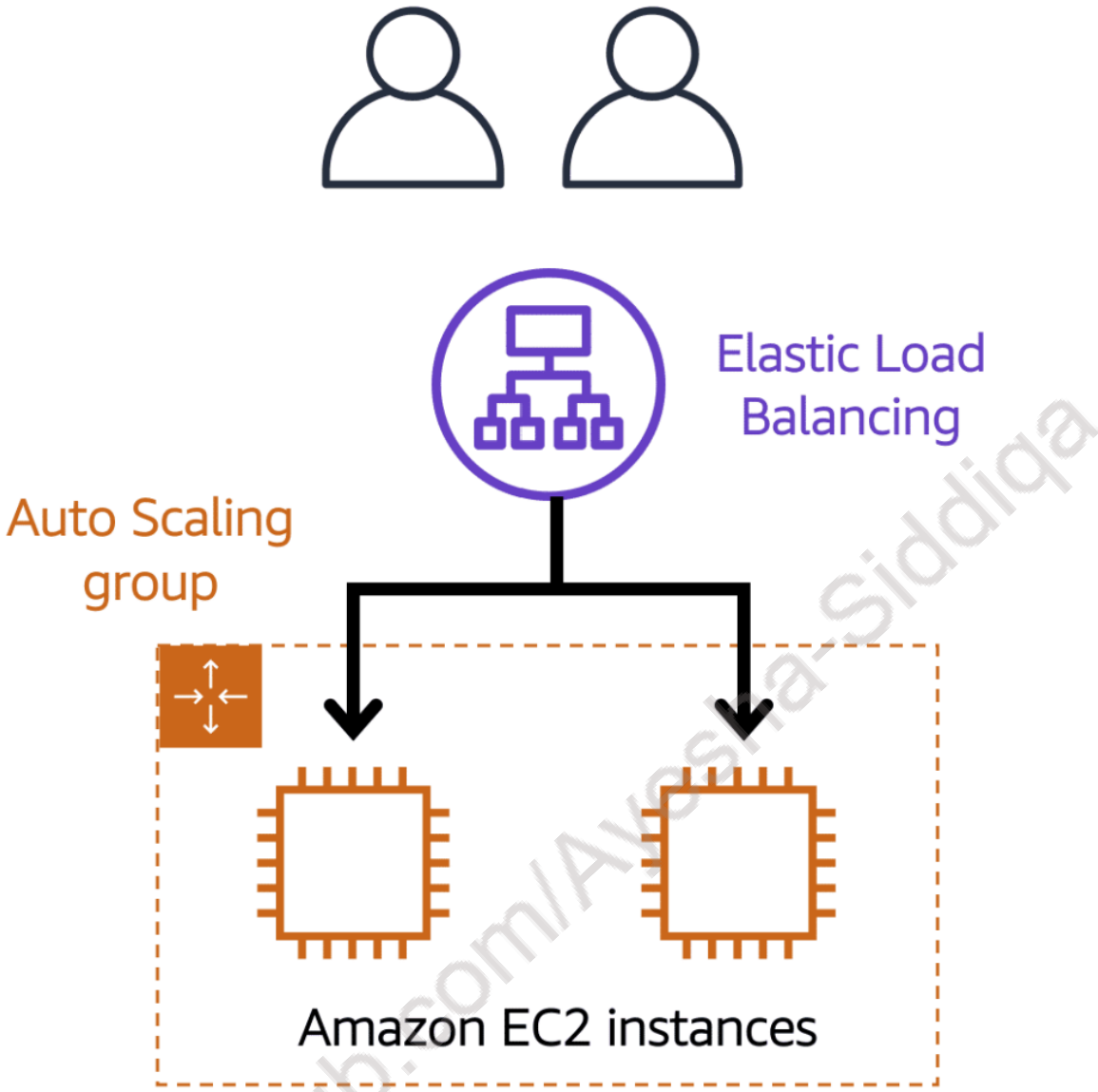
Directing Traffic with Elastic Load Balancing

What is a load balancer?

- When you have multiple EC2 instances all running the same program, to serve the same purpose, and a request comes in, how does that request know which EC2 instance to go to? How can you ensure there's an even distribution of workload across EC2 instances? So not just one is backed up while the others are idle sitting by. You need a way to route requests to instances to process that request. What you need to solve this is called load balancing.
- A load balancer is an application that takes in requests and routes them to the instances to be processed.

What is Elastic Load Balancing?

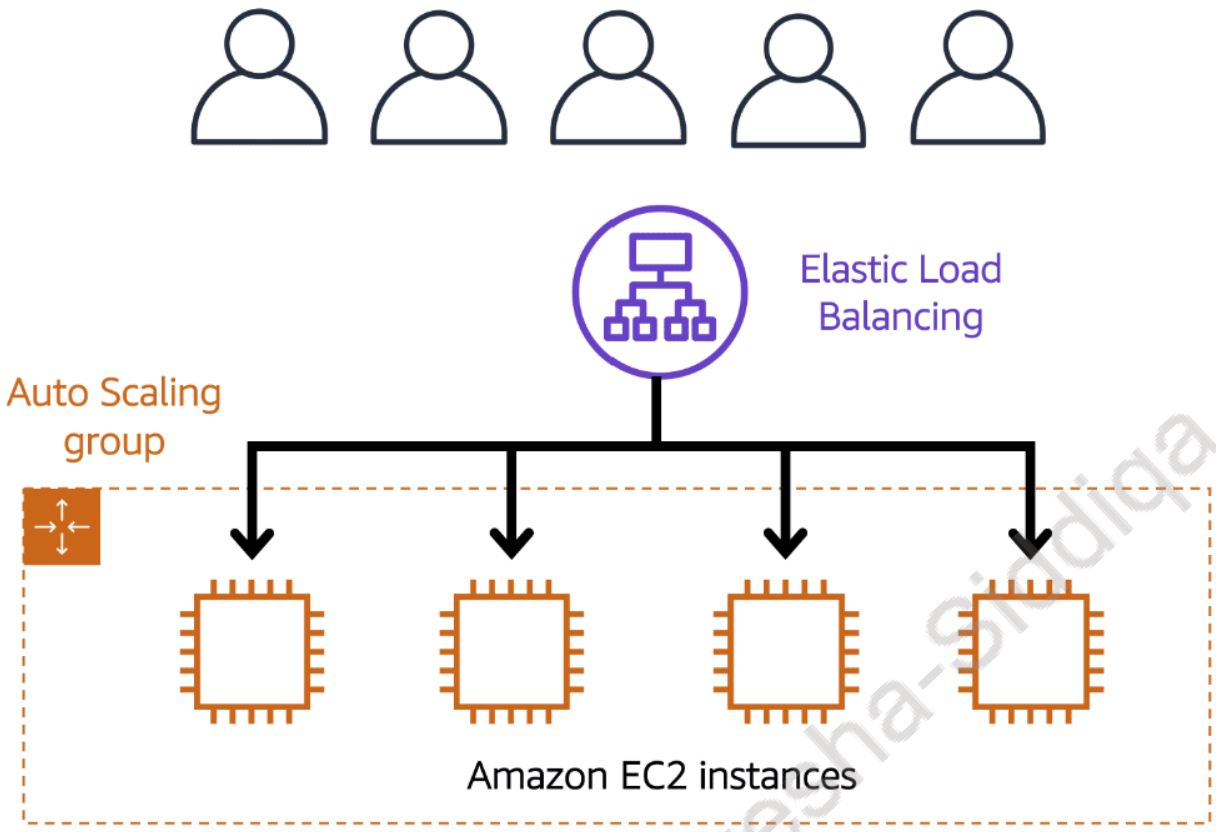
- **Elastic Load Balancing** is the AWS service that automatically distributes incoming application traffic across multiple resources, such as Amazon EC2 instances.
- It is engineered to address the undifferentiated heavy lifting of load balancing.
- Elastic Load Balancing is a regional construct meaning it runs at the Region level rather than on individual EC2 instances, the service is automatically highly available with no additional effort on your part.
- As your traffic grows, ELB is designed to handle the additional throughput with no change to the hourly cost
- A load balancer acts as a single point of contact for all incoming web traffic to your Auto Scaling group. This means that as you add or remove Amazon EC2 instances in response to the amount of incoming traffic, these requests route to the load balancer first. Then, the requests spread across multiple resources that will handle them. For example, if you have multiple Amazon EC2 instances, Elastic Load Balancing distributes the workload across the multiple instances so that no single instance has to carry the bulk of it.
- Although Elastic Load Balancing and Amazon EC2 Auto Scaling are separate services, they work together to help ensure that applications running in Amazon EC2 can provide high performance and availability.



Low Demand Period

Here's an example of how Elastic Load Balancing works. Suppose that a few customers have come to the coffee shop and are ready to place their orders.

If only a few registers are open, this matches the demand of customers who need service. The coffee shop is less likely to have open registers with no customers. In this example, you can think of the registers as Amazon EC2 instances.



High Demand Period

Throughout the day, as the number of customers increases, the coffee shop opens more registers to accommodate them.

Additionally, a coffee shop employee directs customers to the most appropriate register so that the number of requests can evenly distribute across the open registers. You can think of this coffee shop employee as a load balancer.

Messaging and Queuing

What is messaging and queuing?

- This idea of placing messages into a buffer is called messaging and queuing

Explain tightly coupled architecture with an example.

- If we have Application A and it is sending messages directly to Application B, if Application B has a failure and cannot accept those messages, Application A will begin to see errors as well. This is a tightly coupled architecture.

What is loosely coupled architecture and which are the two AWS services that assist in this regard?

- In this case, messages are sent into the queue by Application A and they are processed by Application B. If Application B fails, Application A doesn't experience any disruption. Messages being sent can still be sent to the queue and will remain there until they are eventually processed.
- Amazon Simple Queue Service or SQS and Amazon Simple Notification Service or SNS are the two services.

Explain SNS and SQS
Amazon SQS (Simple Queue Service):

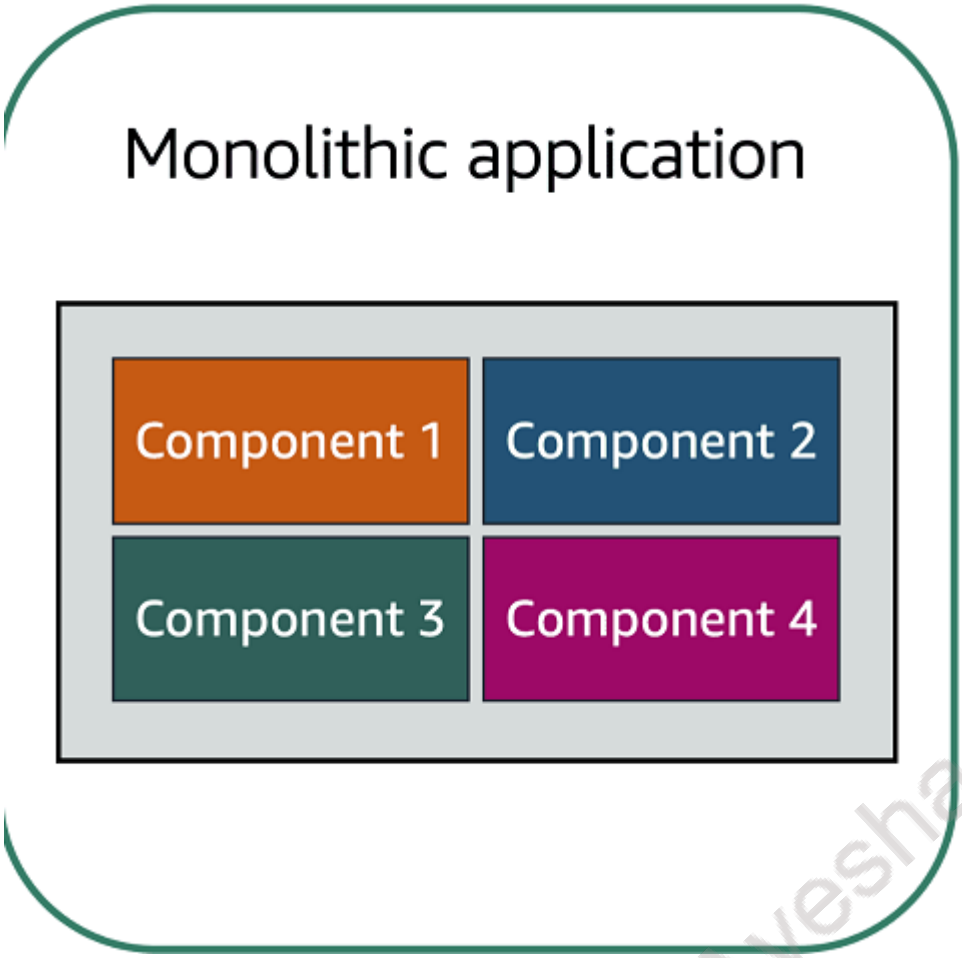
- SQS allows sending, storing, and receiving messages between software components.
- Messages are like coffee orders, with a person's name, coffee order, and order time, and this data is called the payload.
- SQS queues store messages until they are processed.
- AWS manages the underlying infrastructure for hosting SQS queues.
- SQS queues automatically scale, are reliable, and easy to configure and use.

Amazon SNS (Simple Notification Service):

- SNS is used for sending messages to services and notifications to end users.
- SNS operates using a publish/subscribe (pub/sub) model.
- SNS topics act as channels for message delivery.
- Subscribers are configured to SNS topics to receive messages.
- Messages published to an SNS topic are distributed to all subscribers at once (fan-out).
- Subscribers can be endpoints like SQS queues, AWS Lambda functions, HTTPS/HTTP webhooks, and more.

Explain Monolithic Applications.

- Monolithic applications consist of tightly coupled components.
- Components include databases, servers, user interfaces, business logic, etc.
- Tight coupling means these components are highly interdependent.
- If one component fails, it can lead to the failure of other components.
- A single component failure might result in the entire application failing.
- Changes or updates to one part of the application often require rebuilding or redeploying the entire application.
- Scaling monolithic applications can be challenging, as all components scale together.



Explain Microservices.

- Microservices involve loosely coupled application components.
- Loosely coupled components are not highly dependent on each other.
- When one component fails, others can continue functioning independently.
- This prevents the entire application from failing due to a single component failure.
- Microservices allow for modular updates and scalability of individual components.
- On AWS, Amazon Simple Notification Service (SNS) and Amazon Simple Queue Service (SQS) help facilitate microservices-based application integration.
- SNS and SQS enable communication and coordination between microservices components.

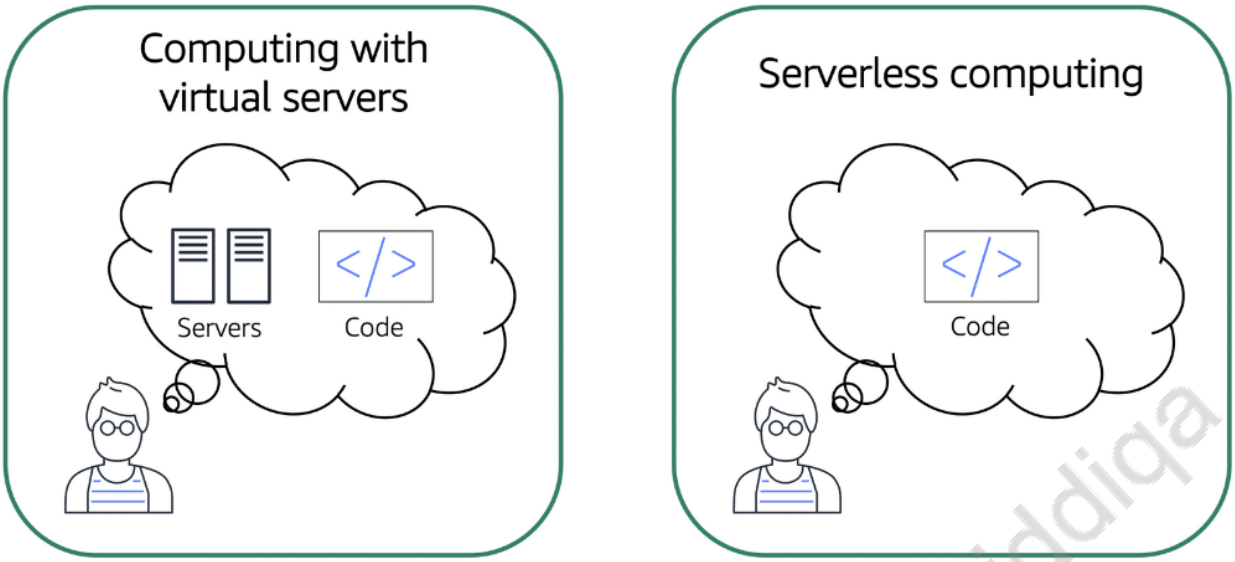
Which AWS service is the best choice for publishing messages to subscribers?

- ✗ Amazon Simple Queue Service (Amazon SQS)
- ✗ Amazon EC2 Auto Scaling
- ✓ Amazon Simple Notification Service (Amazon SNS)
- ✗ Elastic Load Balancing

While running applications on EC2 what do you need to take care?

- **Provision Instances:**
  - Launch and configure Amazon EC2 instances with desired specifications.
- **Upload Your Code:**
  - How do you make your application available on EC2 instances?
  - Upload your application code and any necessary data to the provisioned instances.
- **Continue to Manage the Instances While Your Application is Running:**
  - What's required to maintain EC2 instances during application operation?
  - Tasks include monitoring, security updates, and scaling to meet demand.
  - Utilize AWS services like CloudWatch, Elastic Load Balancing, and Auto Scaling for efficient management.

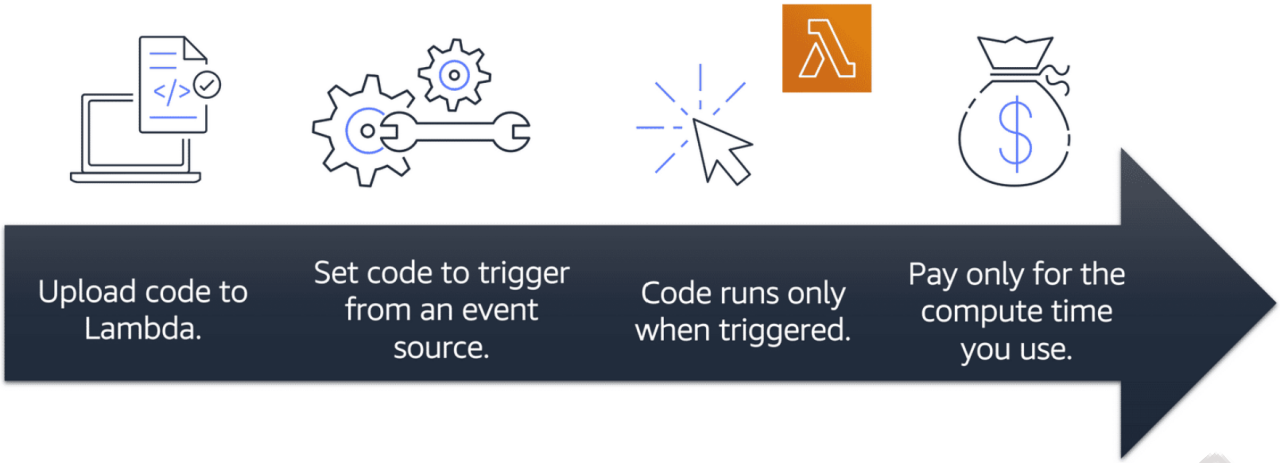
Explain Serverless computing.



Comparison between computing with virtual servers (thinking about servers and code) and serverless computing (thinking only about code).

- **Serverless means code runs on servers but without the need to provision or manage servers.**
  - Code execution occurs on infrastructure managed by the cloud provider, abstracting server management.
  - Developers can focus on innovation rather than server maintenance.
- **Serverless computing enables automatic scaling of applications by adjusting units of consumption:**
  - Applications' capacity can scale automatically based on factors like throughput and memory requirements.
  - Scalability is handled by the serverless platform, reducing the need for manual intervention.
- **AWS Lambda is an AWS service for serverless computing:**
  - AWS Lambda allows you to execute code in response to various triggers without the need to manage underlying infrastructure.
  - It's a key component for building serverless applications on AWS

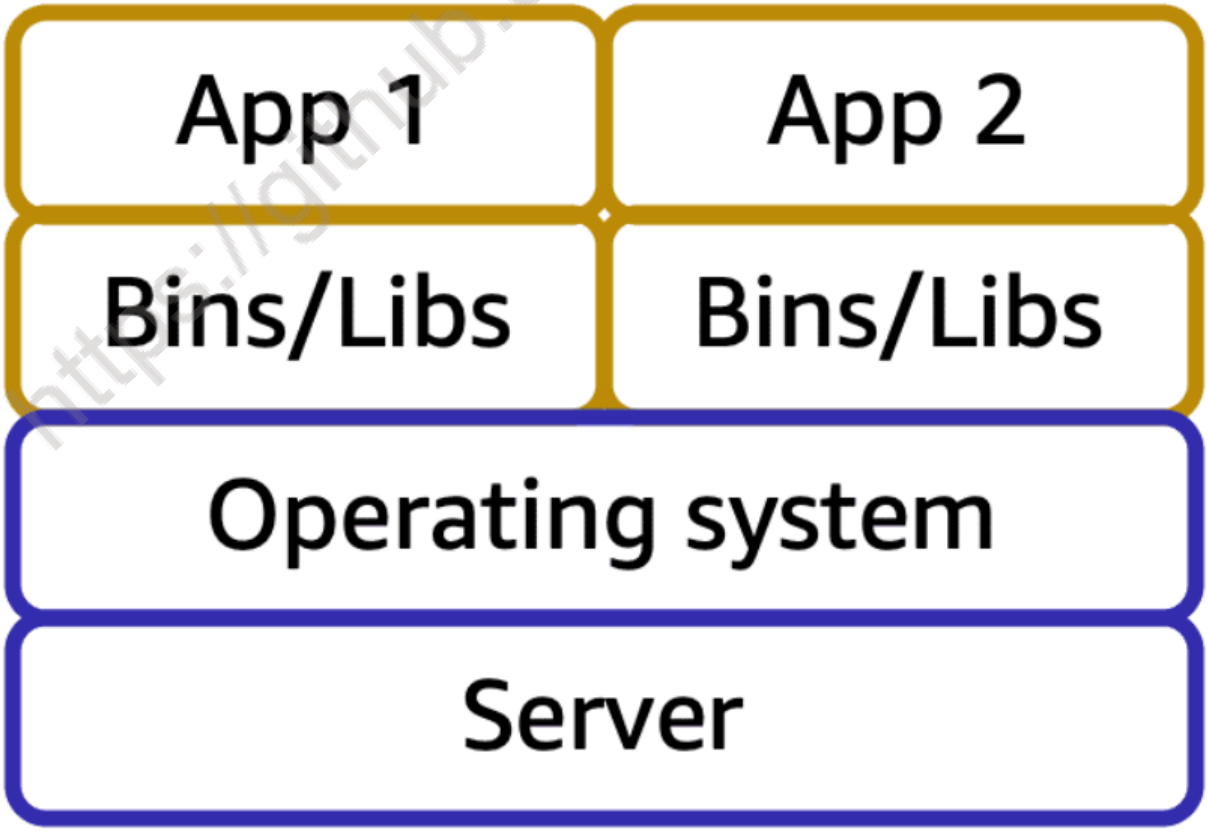
Explain AWS Lambda



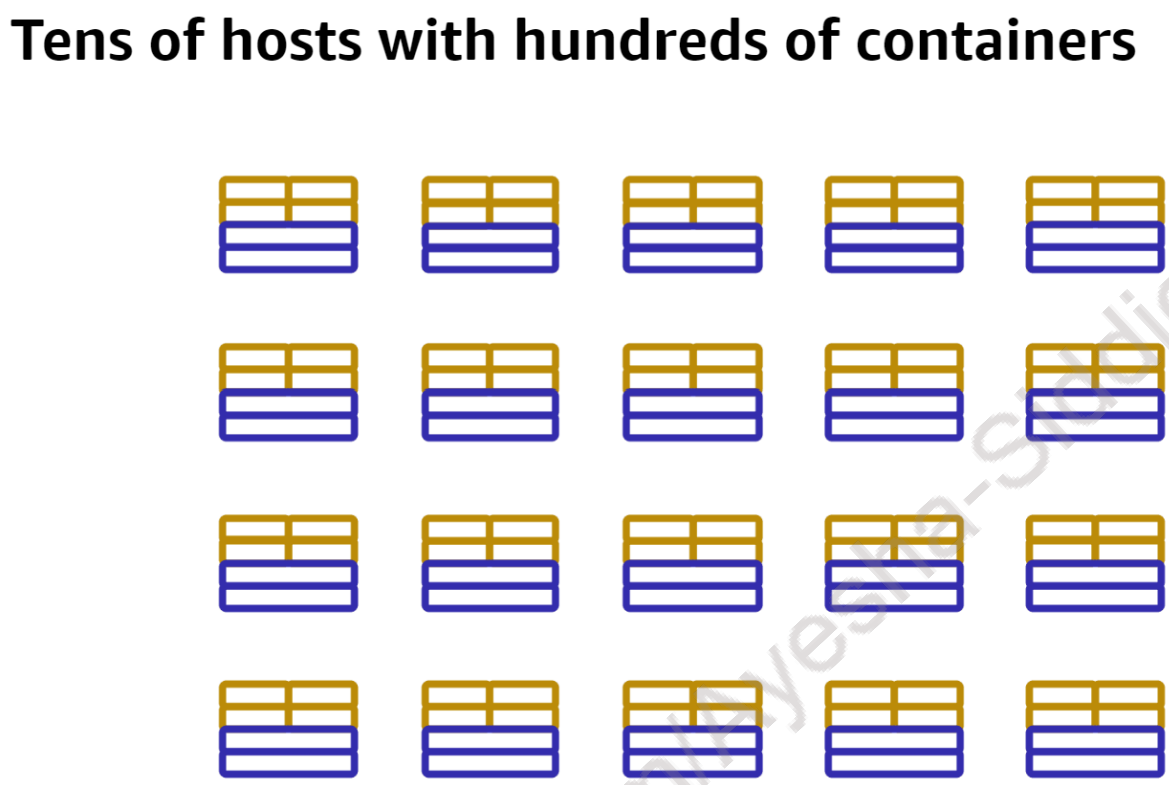
- You upload your code to Lambda.
- You set your code to trigger from an event source, such as AWS services, mobile applications, or HTTP endpoints.
- Lambda runs your code only when triggered.
- You pay only for the compute time that you use. In the previous example of resizing images, you would pay only for the compute time that you use when uploading new images. Uploading the images triggers Lambda to run code for the image resizing function.

Explain Containers, Containerized approach and Container Orchestration Services.

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



Suppose that a company's application developer has an environment on their computer that is different from the environment on the computers used by the IT operations staff. The developer wants to ensure that the application's environment remains consistent regardless of deployment, so they use a containerized approach. This helps to reduce time spent debugging applications and diagnosing differences in computing environments.



When running containerized applications, it's important to consider scalability. Suppose that instead of a single host with multiple containers, you have to manage tens of hosts with hundreds of containers. Alternatively, you have to manage possibly hundreds of hosts with thousands of containers. At a large scale, imagine how much time it might take for you to monitor memory usage, security, logging, and so on. Container orchestration services help you to deploy, manage, and scale your containerized applications

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

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

Explain Amazon EKS

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

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

Explain AWS Fargate

AWS Fargate(opens in a new tab) is a serverless compute engine for containers. It works with both Amazon ECS and Amazon EKS.

When using AWS Fargate, you do not need to provision or manage servers. AWS Fargate manages your server infrastructure for you. You can focus more on innovating and developing your applications, and you pay only for the resources that are required to run your containers.

You want to use an Amazon EC2 instance for a batch processing workload. What would be the best Amazon EC2 instance type to use?

- ✗ General purpose
- ✗ Memory optimized
- ✓ Compute optimized
- ✗ Storage optimized

The correct response option is **Compute optimized**.

The other response options are incorrect because:

- General purpose instances provide a balance of compute, memory, and networking resources. This instance family would not be the best choice for the application in this scenario. Compute optimized instances are more well suited for batch processing workloads than general purpose instances.
- Memory optimized instances are more ideal for workloads that process large datasets in memory, such as high-performance databases.
- Storage optimized instances are designed for workloads that require high, sequential read and write access to large datasets on local storage. The question does not specify the size of data that will be processed. Batch processing involves processing data in groups. A compute optimized instance is ideal for this type of workload, which would benefit from a high-performance processor.

What are the contract length options for Amazon EC2 Reserved Instances? (Select TWO.)

- [x] 1 year
- [ ] 2 years
- [x] 3 years
- [ ] 4 years
- [ ] 5 years

You have a workload that will run for a total of 6 months and can withstand interruptions. What would be the most cost-efficient Amazon EC2 purchasing option?

- [ ] Reserved Instance
- [x] Spot Instance
- [ ] Dedicated Instance
- [ ] On-Demand Instance

Which process is an example of Elastic Load Balancing?

- ✓ Ensuring that no single Amazon EC2 instance has to carry the full workload on its own
- ✗ Removing unneeded Amazon EC2 instances when demand is low
- ✗ Adding a second Amazon EC2 instance during an online store's popular sale
- ✗ Automatically adjusting the number of Amazon EC2 instances to meet demand

You want to deploy and manage containerized applications. Which service should you use?

- ✗ AWS Lambda
- ✗ Amazon Simple Notification Service (Amazon SNS)
- ✗ Amazon Simple Queue Service (Amazon SQS)
- ✓ Amazon Elastic Kubernetes Service (Amazon EKS)

Some other services:

1. **Amazon EC2 Image Builder:**
   - Simplifies building, testing, and deploying virtual machine and container images for AWS or on-premises use.
   - Reduces the effort of keeping images up-to-date and secure.

2. **Amazon Lightsail:**
    - Designed for launching and managing virtual private servers with AWS.
    - Includes a virtual machine, SSD-based storage, data transfer, DNS management, and a static IP address at a predictable price.
3. **AWS App Runner:**
    - A fully managed service for quickly deploying containerized web applications and APIs without prior infrastructure experience.
    - Automatically builds, deploys, and scales web applications with load balancing.
4. **AWS Batch:**
    - Enables efficient execution of batch computing jobs on AWS by dynamically provisioning compute resources based on job requirements.
    - Eliminates the need to manage batch computing software or server clusters.
5. **AWS Elastic Beanstalk:**
    - An easy-to-use service for deploying and scaling web applications and services developed with various programming languages.
    - Automatically handles deployment, capacity provisioning, load balancing, and more.
6. **AWS Fargate:**
    - A compute engine for running containers without managing servers or clusters.
    - Removes the need for server management, offering simplicity in container deployment.
7. **AWS Lambda:**
    - Allows running code without server provisioning or management.
    - Pay only for compute time consumed, making it suitable for various applications and services.
8. **AWS Serverless Application Repository:**
    - Enables quick deployment of code samples, components, and complete applications for common use cases.
    - Allows sharing and publishing applications with AWS Serverless Application Model (SAM) templates.
9. **AWS Outposts:**
    - Extends AWS services, infrastructure, and operating models to on-premises data centers or co-location spaces.
    - Offers a consistent hybrid experience with two variants: VMware Cloud on AWS Outposts and AWS native variant.
10. **AWS Wavelength:**
    - Optimized AWS infrastructure for mobile edge computing applications.
    - Wavelength Zones are AWS infrastructure deployments with embedded AWS compute and storage for low-latency processing.
11. **VMware Cloud on AWS:**
    - VMware Cloud on AWS enables easy migration of VMware workloads to AWS while integrating AWS services, simplifying operations, and ensuring workload portability.

AWS compute services

Category	Service description	AWS service
Instances (virtual machines)	Secure and resizable compute capacity (virtual servers) in the cloud	Amazon Elastic Compute Cloud (EC2)
	Run fault-tolerant workloads for up to 90% off	Amazon EC2 Spot
	Automatically add or remove compute capacity to meet changes in demand	Amazon EC2 Autoscaling
	Easy-to-use cloud platform that offers you everything you need to build an application or website	Amazon Lightsail
	Fully managed batch processing at any scale	AWS Batch
Containers	Highly secure, reliable, and scalable way to run containers	Amazon Elastic Container Service (ECS)
	Run containers on customer-managed infrastructure	Amazon ECS Anywhere
	Easily store, manage, and deploy container images	Amazon Elastic Container Registry (ECR)
	Fully managed Kubernetes service	Amazon Elastic Kubernetes Service (EKS)
	Create and operate Kubernetes clusters on your own infrastructure	Amazon EKS Anywhere
	Serverless compute for containers	AWS Fargate
	Build and run containerized applications on a fully managed service	AWS App Runner
Serverless	Run code without thinking about servers. Pay only for the compute time you consume	AWS Lambda
Edge and hybrid	Run AWS infrastructure and services on premises for a truly consistent hybrid experience	AWS Outposts
	Collect and process data in rugged or disconnected edge environments	AWS Snow Family
	Deliver ultra-low latency application for 5G devices	AWS Wavelength
	Preferred service for all vSphere workloads to rapidly extend and migrate to the cloud	VMware Cloud on AWS
	Run latency sensitive applications closer to end-users	AWS Local Zones
Cost and capacity management	Flexible pricing model that provides savings of up to 72% on AWS compute usage	AWS Savings Plan
	Recommends optimal AWS compute resources for your workloads to reduce costs and improve performance	AWS Compute Optimizer
	Easy-to-use service for deploying and scaling web applications and services	AWS Elastic Beanstalk
	Build and maintain secure Linux or Windows Server images	EC2 Image Builder
	Automatically distribute incoming application traffic across multiple targets	Elastic Load Balancing (ELB)