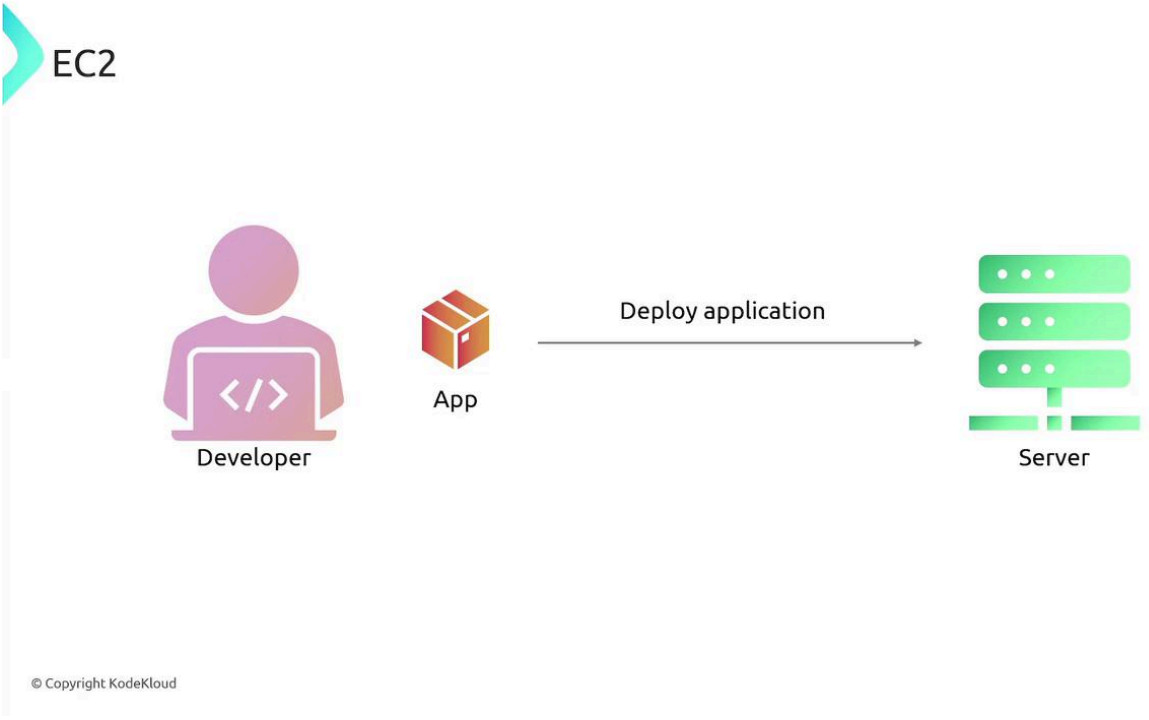
EC2 101

EC2

In this lesson, we explore Amazon EC2 and how it enables you to run your application code on virtual servers in the cloud. EC2 simplifies the process of deploying applications by allowing you to focus on your code rather than managing the underlying physical hardware.

When you create an application, you eventually need to deploy it to a server—a computer that handles client requests, processes your application code, and returns responses. The term "server" originates from the client-server architecture, where a computer provides resources (such as a web application) to the client. For example, when you access a website, your browser sends a request to the server's IP address or domain name, and the server responds by serving the corresponding HTML file.

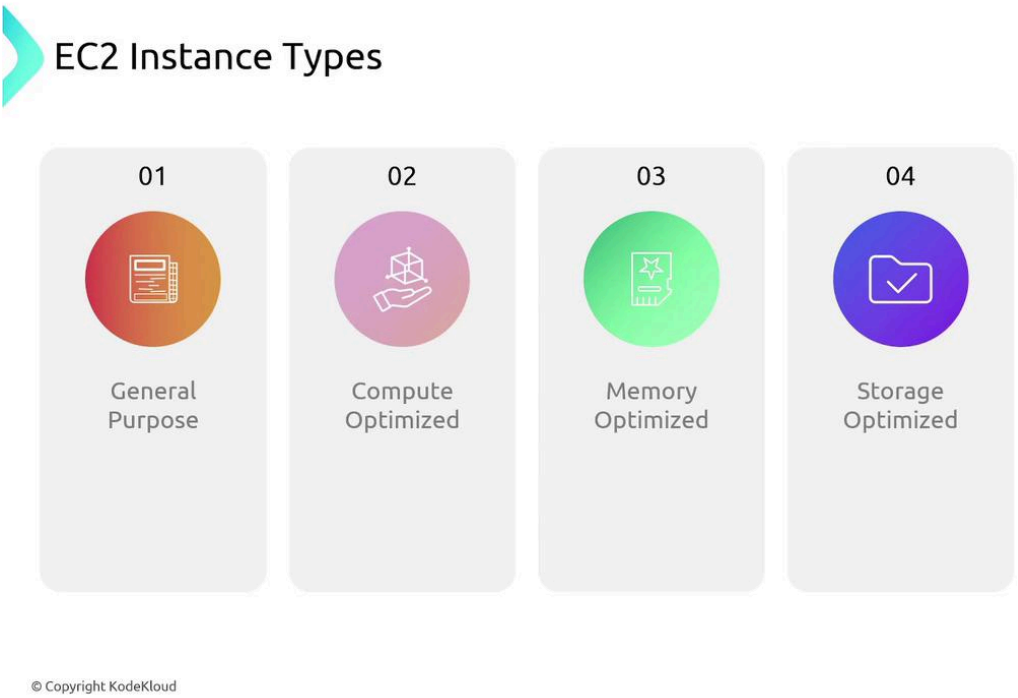


Before the rise of cloud computing, deploying an application involved renting space in a data center, purchasing physical servers (e.g., Dell or HP), installing and securing an operating system like Ubuntu, and manually managing hardware failures. AWS eliminates these complexities by offering EC2 instances —virtual servers for deploying and running your applications without worry about the infrastructure.

An EC2 instance is a virtual server that offers computing power equivalent to a physical host. AWS provides various instance types to match your application's CPU, memory, and storage needs. These include:

- **General Purpose:** Offers a balance of compute, memory, and networking resources for diverse workloads, such as web servers.
- **Compute Optimized:** Designed for compute-bound applications requiring high-performance processors.
- **Memory Optimized:** Ideal for workloads that need to process large data sets in memory.

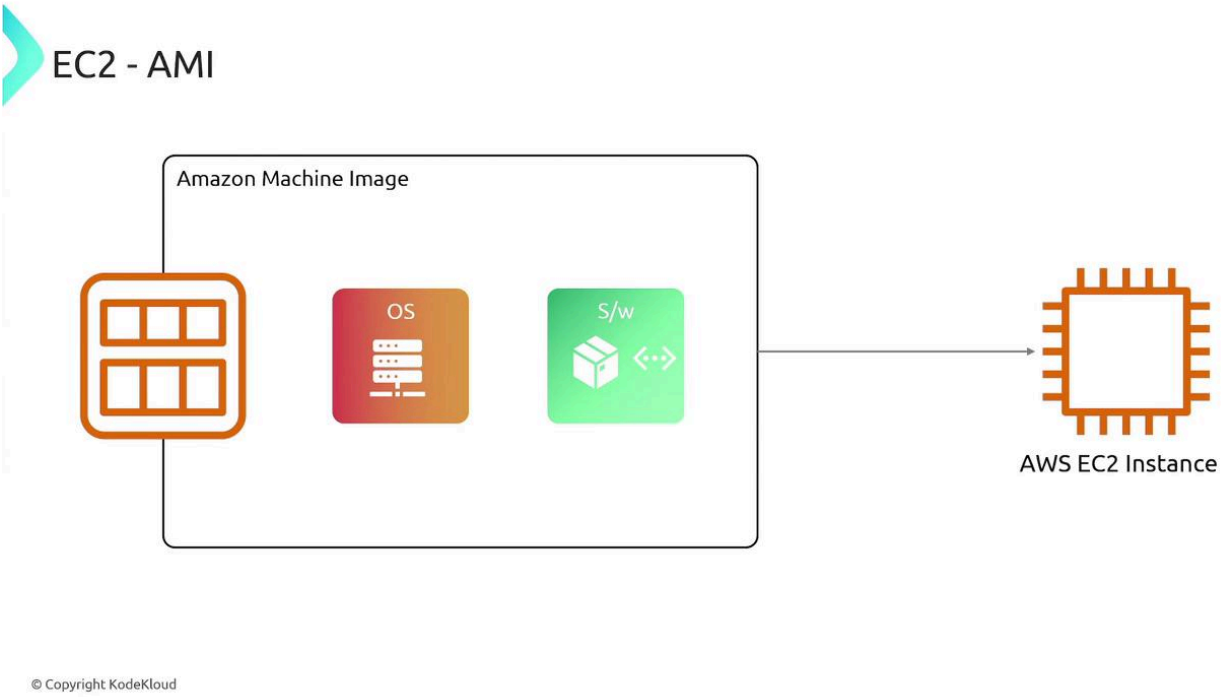
- **Storage Optimized:** Suited for applications with high I/O demands that frequently access disk storage.
- **GPU Instances:** Perfect for high-performance tasks like machine learning and deep learning that require powerful GPUs.



Amazon Machine Images (AMIs)

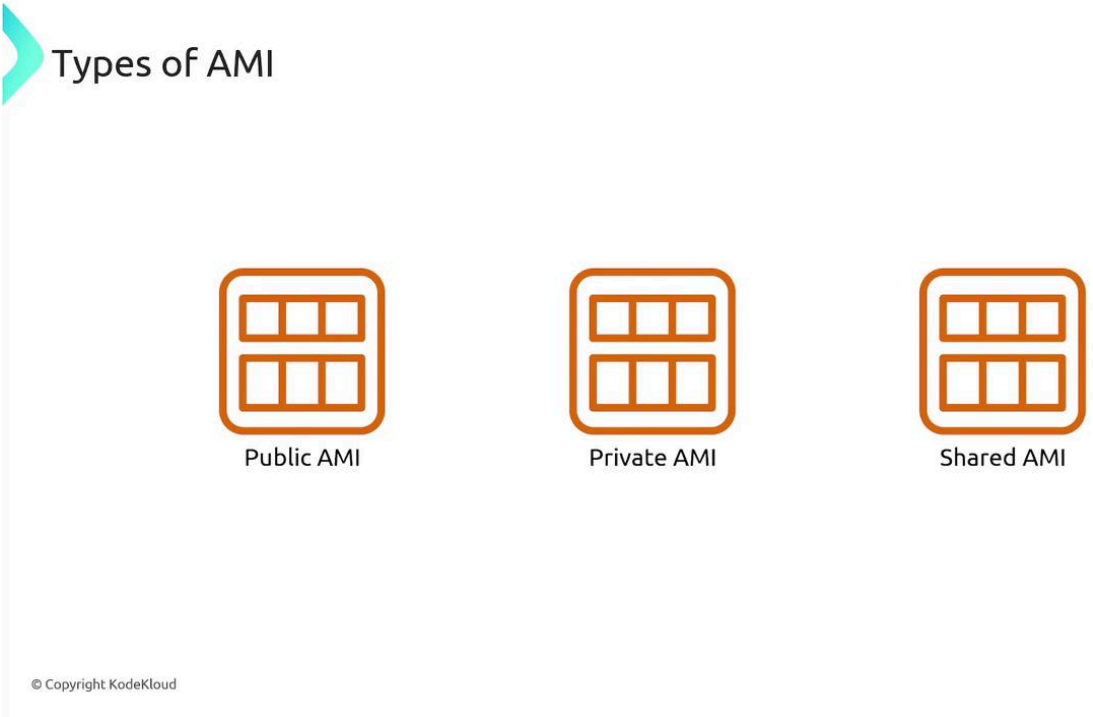
When launching an EC2 instance, you must select an Amazon Machine Image (AMI), which serves as the blueprint for your instance. The AMI includes the operating system and may come pre-configured with additional software or settings you require. Available for various operating systems—Linux distributions (like Ubuntu, Red Hat, or CentOS), macOS, or Windows—AMIs enable rapid deployment of identical servers at scale.

Think of an AMI as a recipe for creating your server. If you modify an instance (for example, by installing new software or configuring security settings), you can create a new AMI from that instance and use it for future deployments.

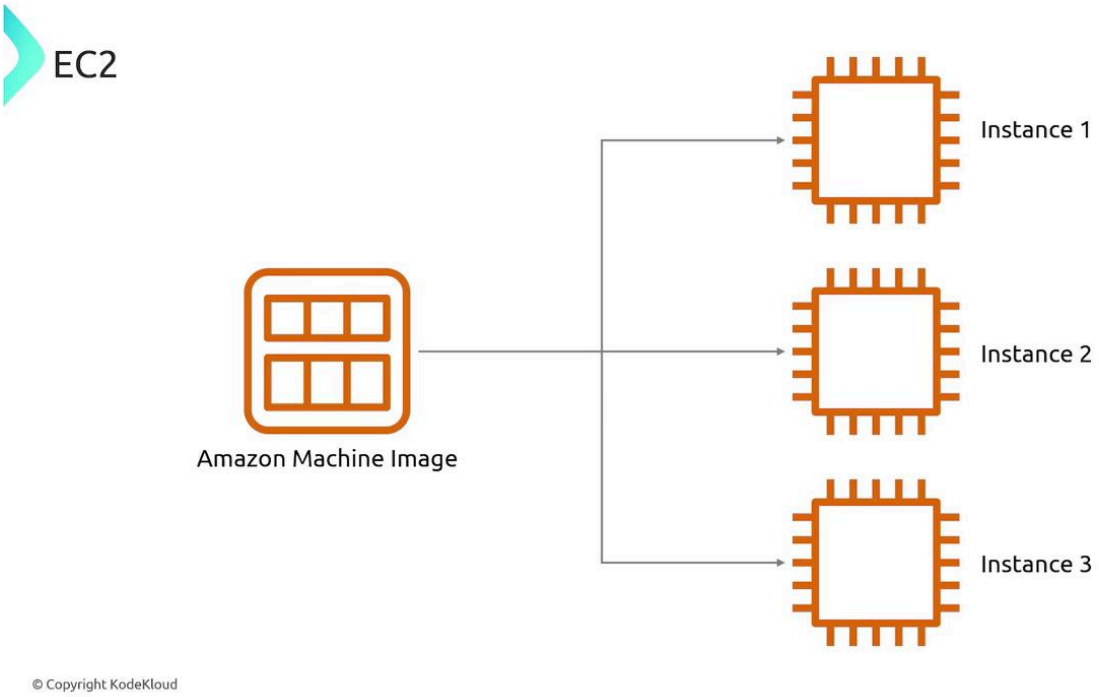


There are three types of AMIs:

- **Public AMIs:** Shared within the AWS community and available to anyone at no cost.
- **Private AMIs:** Customized AMIs that remain accessible only to the owner or specified AWS accounts.
- **Shared AMIs:** Private AMIs explicitly shared with select AWS accounts for controlled collaboration.

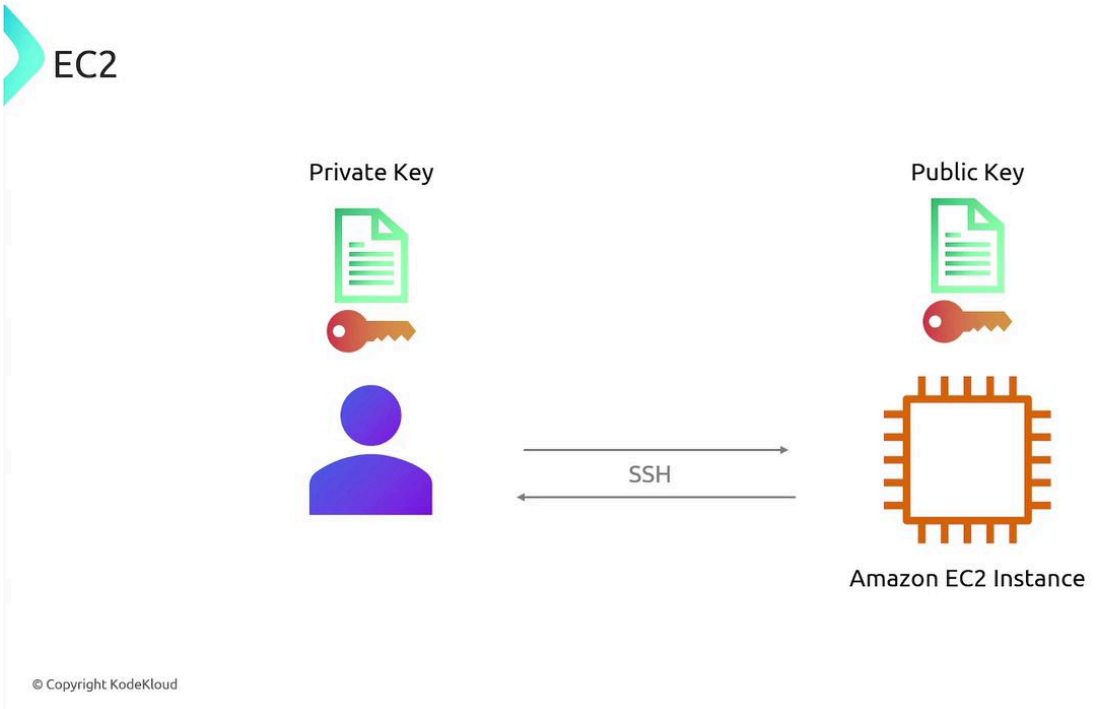


Once an AMI is defined, you can create multiple, identically configured EC2 instances.



Securely Connecting to an EC2 Instance

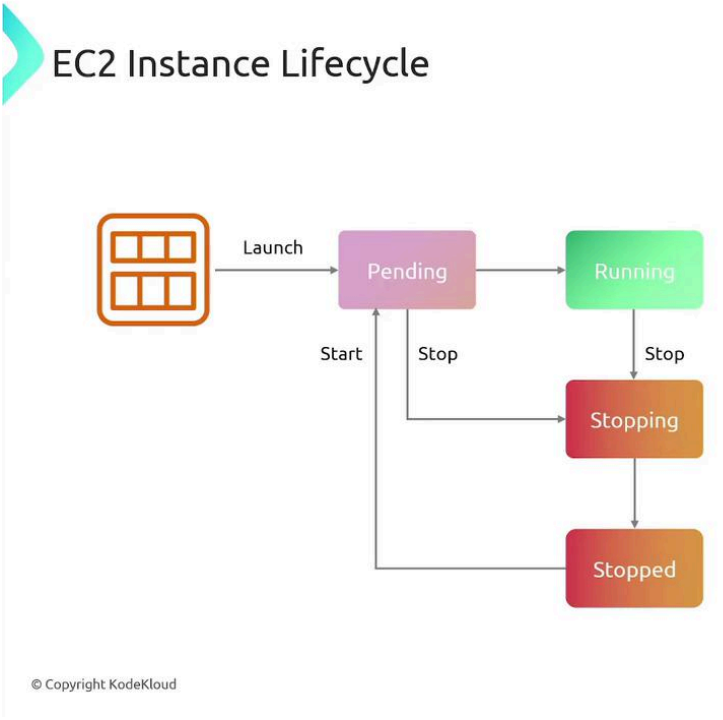
After your EC2 instance is deployed, connecting via SSH is the typical method to manage and configure it. For a secure connection, you use a key pair consisting of a public key and a private key. The public key is embedded into the instance during launch, while you use the corresponding private key to connect via SSH. AWS allows you to manage multiple key pairs, enabling different access configurations per instance.



EC2 Instance Lifecycle

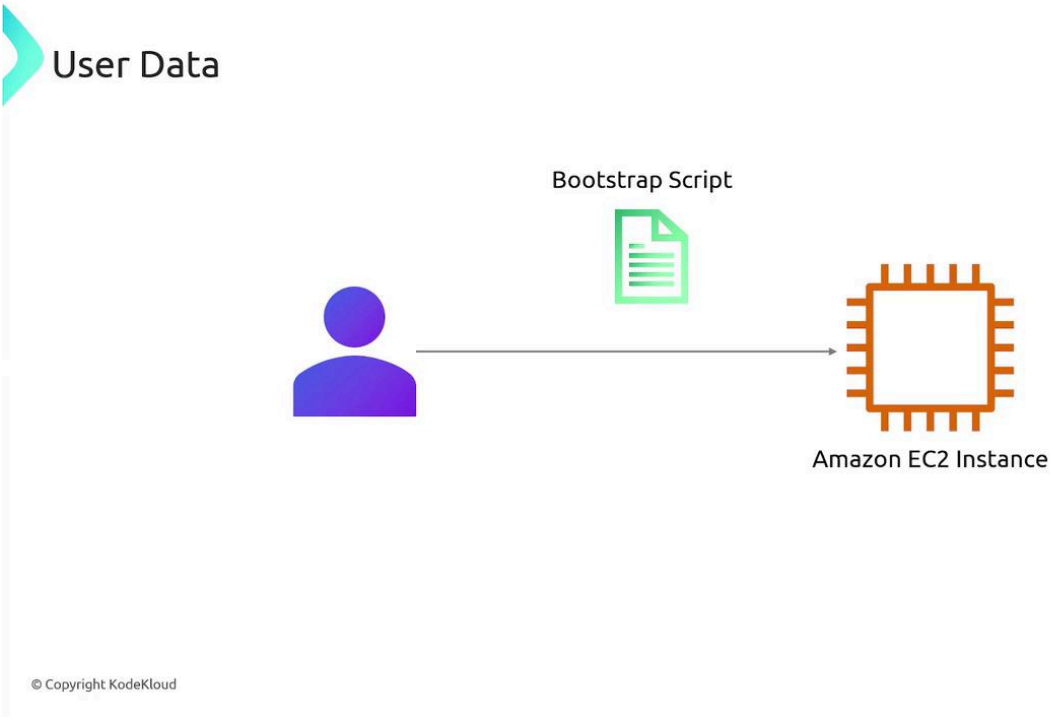
Understanding the lifecycle of an EC2 instance is crucial for effective management. The lifecycle includes several states:

- **Pending:** The instance is launching and transitioning to the running state.
- **Running:** The instance is active and available for connection.
- **Stopping:** The instance is in the process of shutting down.
- **Stopped:** The instance is powered off but can be re-started later.
- **Shutting Down:** The instance is preparing to be terminated.
- **Terminated:** The instance has been permanently deleted and cannot be recovered.



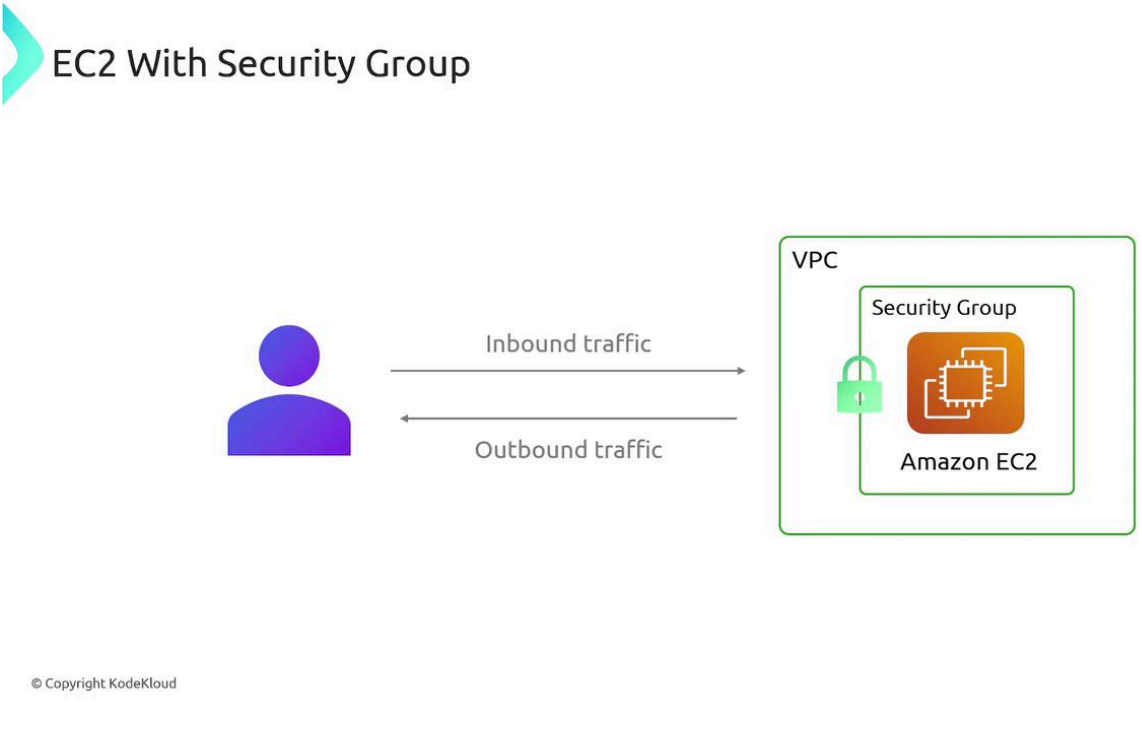
Bootstrapping with User Data

When launching an EC2 instance, you can provide user data—usually a shell script or cloud-init directives—which executes during startup. This automation allows you to install software, configure settings, or download necessary files immediately as the instance launches. Keep in mind that user data scripts have a size limit of 16 kilobytes.



Security Groups

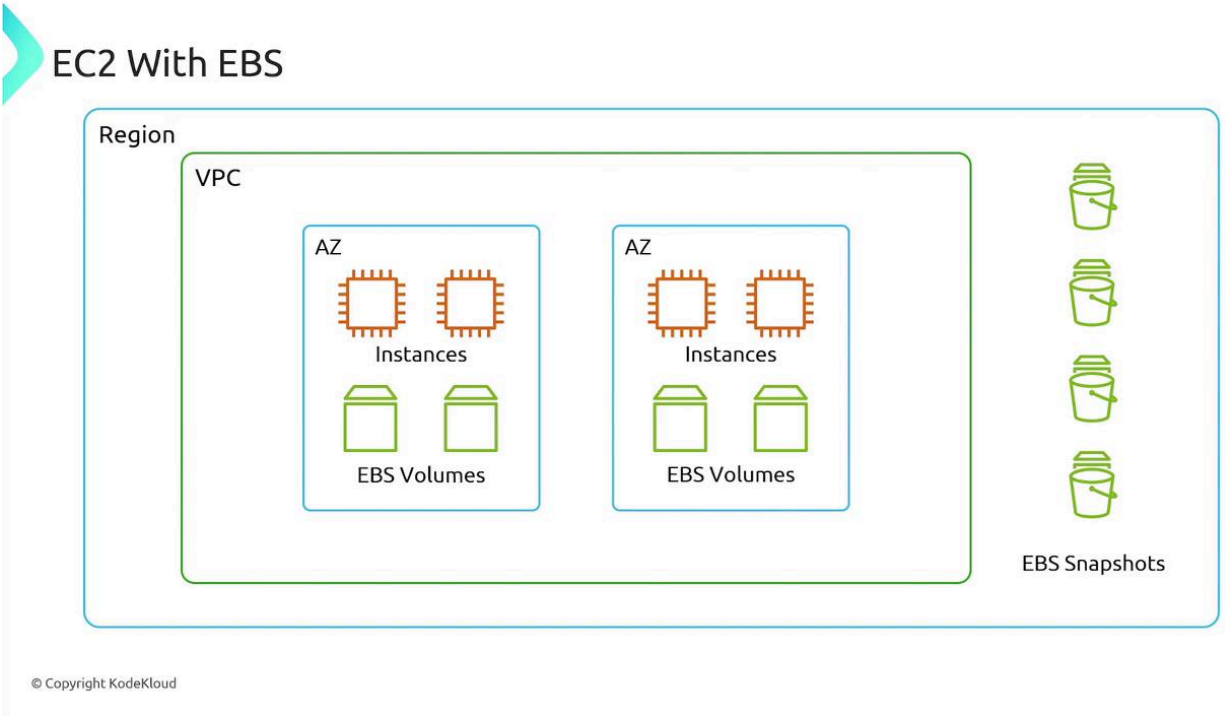
Security groups in AWS act as virtual firewalls for your EC2 instances, controlling inbound and outbound traffic. For example, if you deploy a web server, you can configure your security group to allow inbound HTTP (port 80) and HTTPS (port 443) traffic. This ensures that only the necessary traffic reaches your instance while maintaining robust security.



Persistent Storage with Elastic Block Store (EBS)

EC2 instances provide computing power, but many applications require persistent data storage. AWS Elastic Block Store (EBS) is a scalable block storage service that can be attached to your instances. EBS is commonly used for boot volumes, databases, file storage, and backup solutions. It also supports

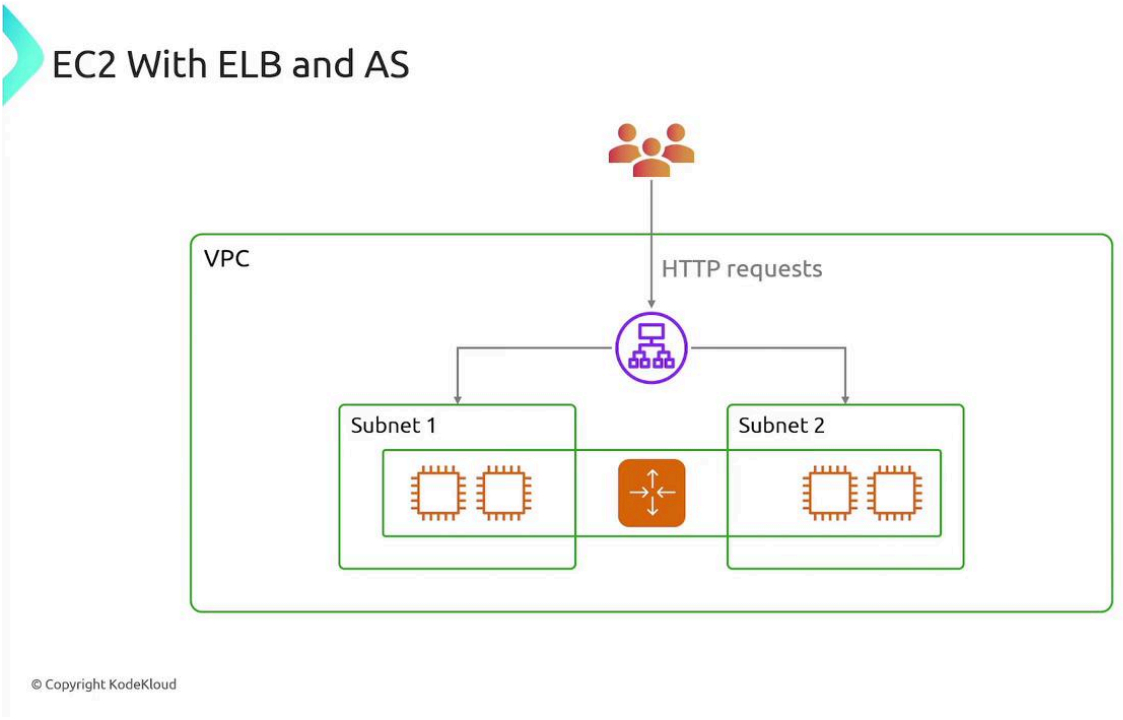
snapshots—point-in-time copies of your volumes stored in S3. These snapshots are incremental, saving only changes since the previous snapshot, which helps reduce both storage costs and backup time.



Integration with Elastic Load Balancing and Auto Scaling

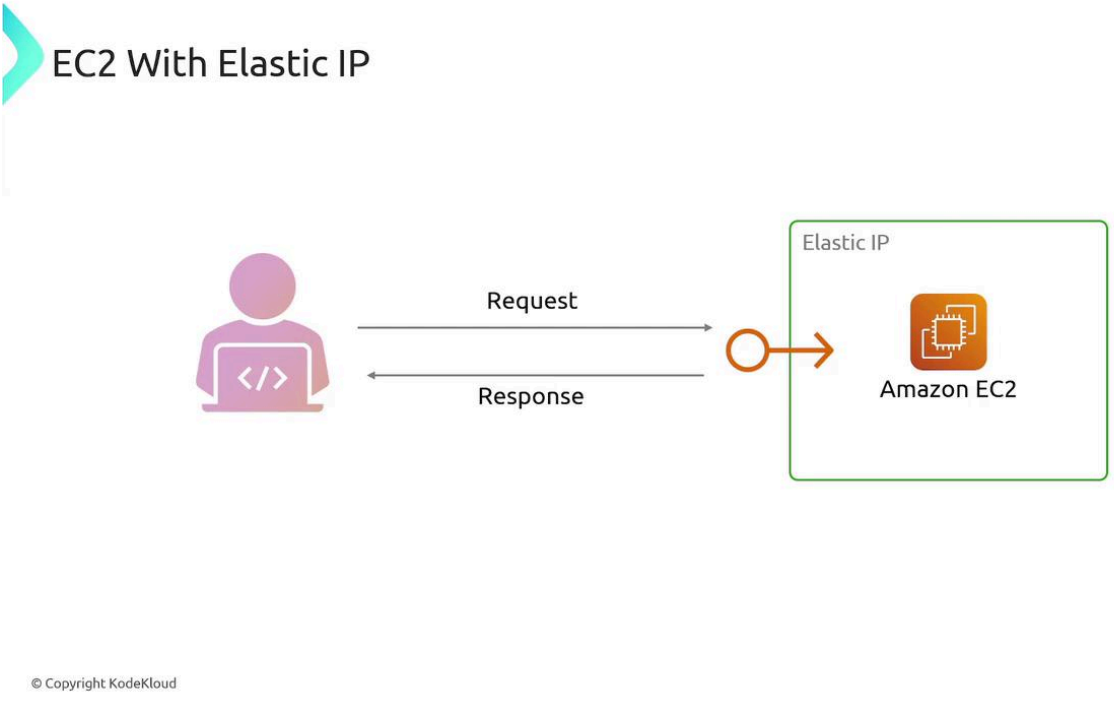
EC2 instances can be integrated with other AWS services to enhance performance and scalability:

- **Elastic Load Balancer (ELB):** Automatically distributes incoming traffic across multiple targets such as EC2 instances, containers, or IP addresses, ensuring seamless user experiences.
- **Auto Scaling Groups:** Automatically adjusts the number of EC2 instances based on traffic demand, scaling out during high-demand periods and scaling in to reduce costs during lower demand.



Elastic IP Addresses

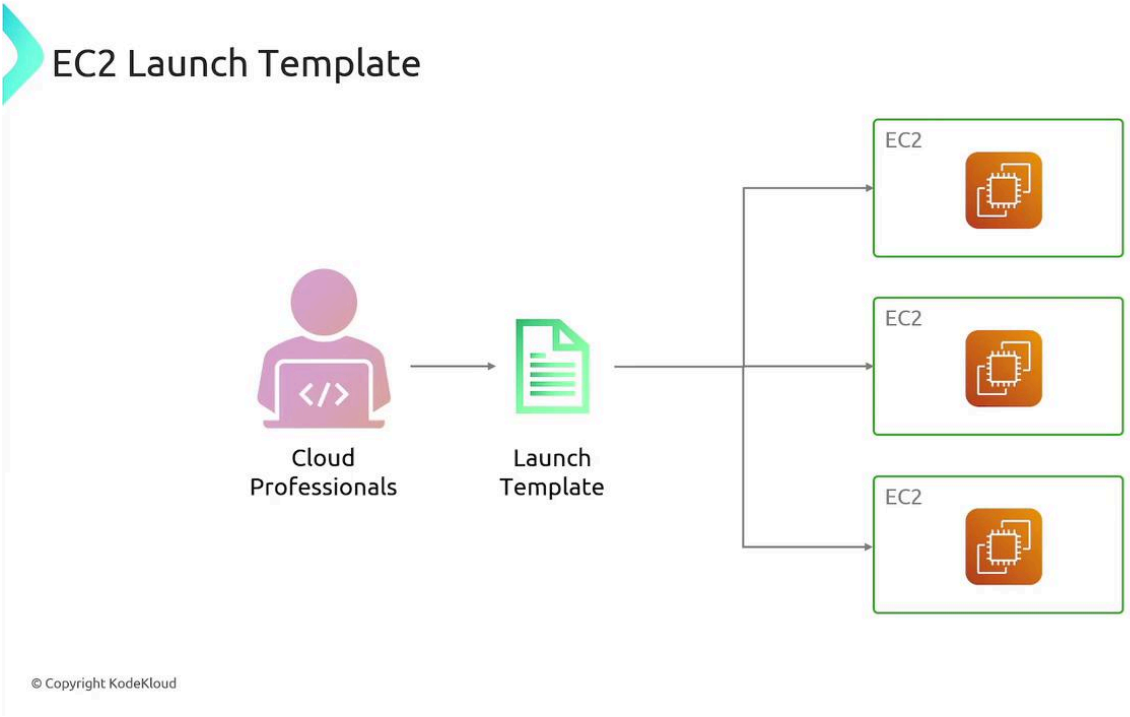
By default, public IP addresses assigned to EC2 instances may change when the instance stops and restarts. To maintain a consistent IP, you can use an Elastic IP—a static, reserved IP address associated with your account. Elastic IPs remain constant even if you move the underlying instance between physical hosts, and can be re-assigned to different instances as needed.



Launch Templates

Launch templates let you define a set of configuration parameters (such as the AMI, security groups, subnet, and instance type) that can be reused when launching EC2 instances. They are particularly

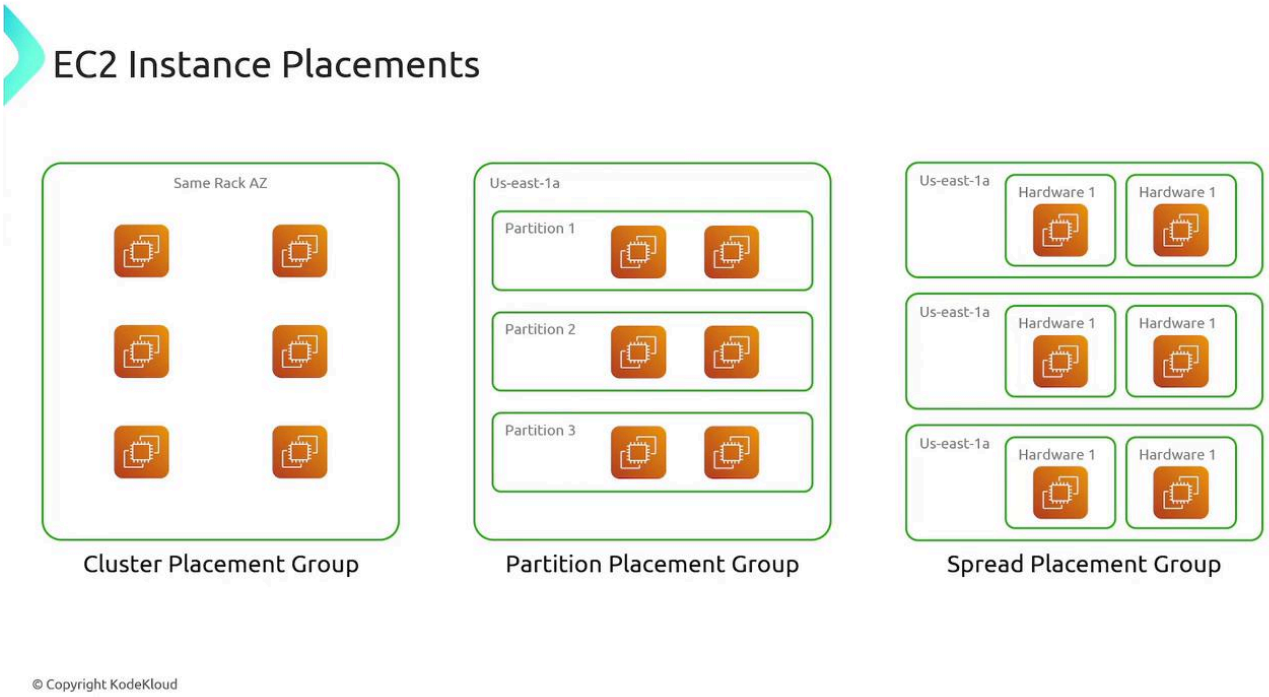
useful with Auto Scaling groups, as AWS uses these templates to automatically launch new instances during scale-out events.



EC2 Instance Placement

AWS automatically selects a suitable physical host in an availability zone when you deploy an EC2 instance. However, you can influence instance placement to best meet your application's requirements:

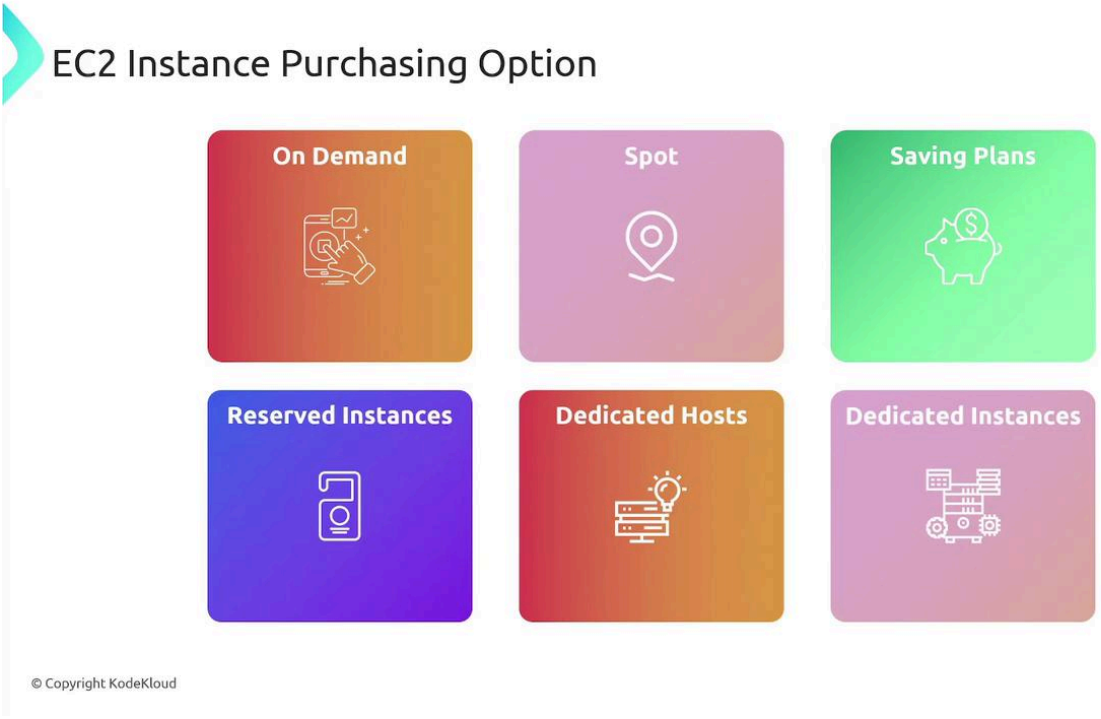
- **Cluster Placement Group:** Places instances as close together as possible to achieve low-latency, high-throughput networking. Ideal for high-performance computing and big data analytics.
- **Partition Placement Group:** Distributes instances across logical partitions to reduce the risk of simultaneous hardware failures. This is particularly useful for distributed workloads such as Hadoop.
- **Spread Placement Group:** Distributes instances across distinct physical hardware to minimize the risk of correlated failures, perfect for a few critical instances that must remain isolated.



EC2 Instance Pricing Options

AWS offers a range of pricing models designed to suit various workloads and budgets:

- **On-Demand:** Pay per hour or second with no long-term commitment, ideal for short-term or unpredictable workloads.
- **Spot Instances:** Bid on unused AWS capacity at discounts of up to 90% compared to on-demand prices. Because spot instances can be interrupted, they are best suited for flexible applications like batch processing.
- **Savings Plans:** Commit to a consistent hourly cost over a one- or three-year term in exchange for lower prices, offering predictability and cost savings for steady-state usage.
- **Reserved Instances:** Reserve a specific amount of compute capacity for one or three years, providing cost savings for predictable workloads.
- **Dedicated Hosts:** Rent an entire physical server exclusively for your use—ideal for licensing requirements and compliance, ensuring your instances run on the same physical server.
- **Dedicated Instances:** Similar to Dedicated Hosts, these instances run on hardware isolated for your use, although the specific physical host may change over time.

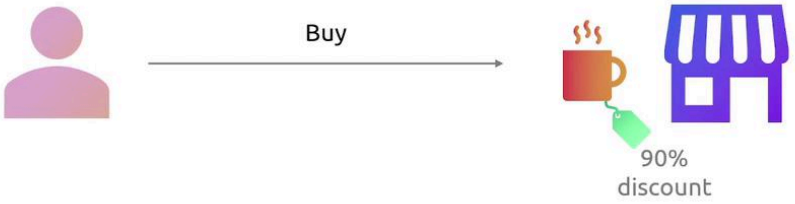


Pricing Analogy

To better understand these pricing models, consider this analogy involving coffee purchases:

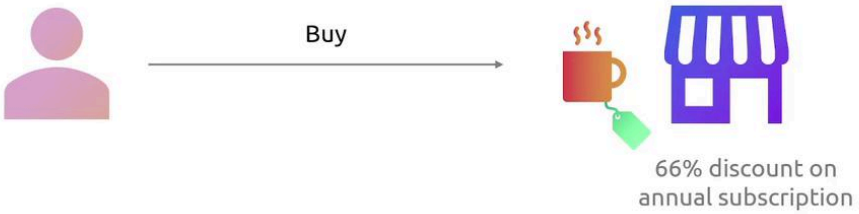
- **On-Demand:** Like buying a cup of coffee at full price whenever you want without any commitment.
- **Spot Instances:** Similar to catching a flash sale at your favorite coffee shop where leftover coffee is sold at a significant discount, albeit with the risk of unavailability.
- **Savings Plans:** Comparable to committing to spend a specific amount on coffee each month in exchange for a lower price.
- **Reserved Instances:** Like subscribing to a daily coffee plan at a discounted rate because you know you will be drinking coffee every day.
- **Dedicated Hosts:** Resembles renting an entire coffee machine exclusively for your use.
- **Dedicated Instances:** Similar to reserving your own seat in a coffee shop where the machine may be shared, but your seat remains exclusively yours.

EC2 Instance Purchasing Option – Spot Instance

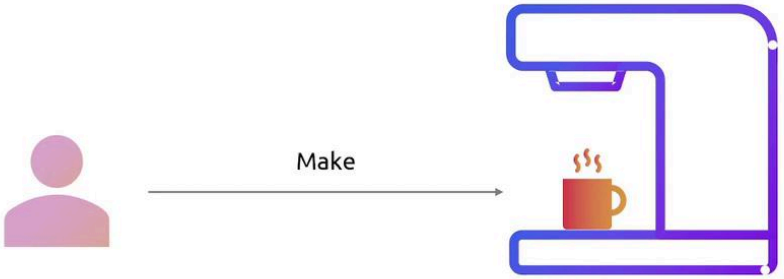


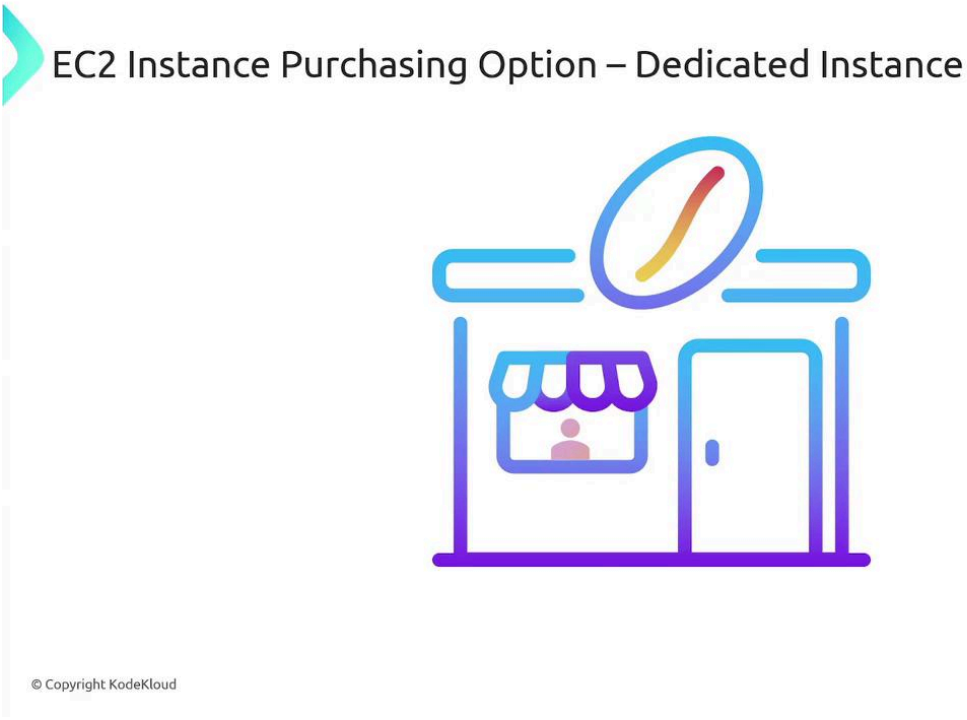
EC2 Instance Purchasing Option – Savings Plan

EC2 Instance Purchasing Option – Reserved Instance



EC2 Instance Purchasing Option – Dedicated Host





Tip

By understanding these pricing options, you can choose the most cost-effective strategy to deploy and manage your applications on AWS EC2.

This comprehensive overview of AWS EC2 covers the key concepts—from virtual server provisioning with AMIs and secure connections using SSH to integrating with other AWS services like Elastic Load Balancing and Auto Scaling. Whether you're launching your first instance or managing a complex architecture, knowing these fundamentals helps ensure your cloud deployment is efficient, secure, and scalable.

For further details, consider exploring:

- AWS EC2 Documentation
- AWS Pricing Models
- AWS Auto Scaling

Happy cloud computing!