AWS EC2

What is EC2?

- Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.
- Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change. Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use.
- Amazon EC2 provides developers the tools to build failure resilient applications and isolate them from common failure scenarios.

Benefits of EC2

- ELASTIC WEB-SCALE COMPUTING : Amazon EC2 enables you to increase or decrease capacity within minutes, not hours or days. You can commission one, hundreds, or even thousands of server instances simultaneously. You can also use Amazon EC2 Auto Scaling to maintain availability of your EC2 fleet and automatically scale your fleet up and down depending on its needs in order to maximize performance and minimize cost.

Benefits of EC2

- COMPLETELY CONTROLLED : You have complete control of your instances including root access and the ability to interact with them as you would any machine. You can stop any instance while retaining the data on the boot partition, and then subsequently restart the same instance using web service APIs.
- FLEXIBLE CLOUD HOSTING SERVICES : You have the choice of multiple instance types, operating systems, and software packages. Amazon EC2 allows you to select a configuration of memory, CPU, instance storage, and the boot partition size that is optimal for your choice of operating system and application.

Benefits of EC2

- INTEGRATED : Amazon EC2 is integrated with most AWS services such as Amazon Simple Storage Service (Amazon S3), Amazon Relational Database Service (Amazon RDS), and Amazon Virtual Private Cloud (Amazon VPC) to provide a complete, secure solution for computing, query processing, and cloud storage across a wide range of applications.
- RELIABLE : Amazon EC2 offers a highly reliable environment where replacement instances can be rapidly and predictably commissioned. The service runs within Amazon's proven network infrastructure and data centers. The Amazon EC2 Service Level Agreement commitment is 99.99% availability for each Amazon EC2 Region.

Benefits of EC2

- SECURE : As an AWS customer, you will benefit from a data center and network architecture built to meet the requirements of the most security-sensitive organizations. Amazon EC2 works in conjunction with Amazon VPC to provide security and robust networking functionality for your compute resources.
- INEXPENSIVE : Amazon EC2 passes on to you the financial benefits of Amazon's scale. You pay a very low rate for the compute capacity you actually consume.

EC2 Pricing Models

- On-Demand Instances: In this model, based on the instances you choose, you pay for compute capacity per hour or per second (only for Linux Instances) and no upfront payments are needed. You can increase or decrease your compute capacity to meet the demands of your application and only pay for the instance you use. This model is suitable for developing/testing application with short-term or unpredictable workloads. On-Demand Instances is recommended for users who prefer low cost and flexible EC2 Instances without upfront payments or long-term commitments.

EC2 Pricing Models

- Reserved Instances : Amazon EC2 Reserved Instances provide you with a discount up to 75% compared to On-Demand Instance pricing. It also provides capacity reservation when used in specific Availability Zone. For applications that have predictable workload, Reserved Instances can provide sufficient savings compared to On-Demand Instances. The predictability of usage ensures compute capacity is available when needed. Customers can commit to using EC2 over a 1- or 3-year term to reduce their total computing costs.

EC2 Pricing Models

- Dedicated Hosts : A Dedicated Host is a physical EC2 server dedicated for your use. Dedicated Hosts can help you reduce costs by allowing you to use your existing server-bound software licenses like Windows server, SQL server etc and also helps you to meet the compliance requirements .Customers who choose Dedicated Hosts have to pay the On-Demand price for every hour the host is active in the account. It supports only per-hour billing and does not support per-second billing scheme.

EC2 Pricing Models

- Spot Instances : Amazon EC2 Spot Instances is unused EC2 capacity in the AWS cloud. Spot Instances are available at up to a 90% discount compared to On-Demand prices. The Spot price of Amazon EC2 spot Instances fluctuates periodically based on supply and demand. It supports both per hour and per second (only for Linux Instances) billing schemes . Applications that have flexible start and end times and users with urgent computing needs for large scale dynamic workload can choose Amazon EC2 spot Instances.

EBS – Elastic Block Store

- Amazon Elastic Block Store (EBS) is an easy to use, high performance block storage service designed for use with Amazon Elastic Compute Cloud (EC2) for both throughput and transaction intensive workloads at any scale. 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 are widely deployed on Amazon EBS.
- Designed for mission-critical systems, EBS volumes are replicated within an Availability Zone (AZ) and can easily scale to petabytes of data. Also, you can use EBS Snapshots with automated lifecycle policies to back up your volumes in Amazon S3, while ensuring geographic protection of your data and business continuity.

EBS – Elastic Block Store

	Solid-State Drives (SSD)		Hard Disk Drives (HDD)	
Volume Type	General Purpose SSD (`gp2`)	Provisioned IOPS SSD (`io1`)	Throughput Optimized HDD (`st1`)	Cold HDD (`sc1`)
Description	General purpose SSD volume that balances price and performance for a wide variety of workloads	Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads	Low-cost HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads

Use Cases	• Recommended for most workloads
	• System boot volumes
	• Virtual desktops
	• Low-latency interactive apps
	• Development and test environments
		• Critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/s of throughput per volume
		• Large database workloads, such as:
		  ◦ MongoDB
		  ◦ Cassandra
		  ◦ Microsoft SQL Server
		  ◦ MySQL
		  ◦ PostgreSQL
		  ◦ Oracle
			• Streaming workloads requiring consistent, fast throughput at a low price
			• Big data
			• Data warehouses
			• Log processing
			• Cannot be a boot volume
				• Throughput-oriented storage for large volumes of data that is infrequently accessed
				• Scenarios where the lowest storage cost is important
				• Cannot be a boot volume

API Name	`gp2`	`io1`	`st1`	`sc1`
Volume Size	1 GiB - 16 TiB	4 GiB - 16 TiB	500 GiB - 16 TiB	500 GiB - 16 TiB
Max IOPS per Volume	16,000 (16 KiB I/O) *	64,000 (16 KiB I/O) †	500 (1 MiB I/O)	250 (1 MiB I/O)
Max Throughput per Volume	250 MiB/s *	1,000 MiB/s †	500 MiB/s	250 MiB/s
Max IOPS per Instance ††	80,000	80,000	80,000	80,000
Max Throughput per Instance ††	1,750 MiB/s	1,750 MiB/s	1,750 MiB/s	1,750 MiB/s
Dominant Performance Attribute	IOPS	IOPS	MiB/s	MiB/s

EBS vs Instance Store

- Some Amazon Elastic Compute Cloud (Amazon EC2) instance types come with a form of directly attached, block-device storage known as the instance store. The instance store is ideal for temporary storage, because the data stored in instance store volumes is not persistent through instance stops, terminations, or hardware failures.
- For data you want to retain longer, or if you want to encrypt the data, use Amazon Elastic Block Store (Amazon EBS) volumes instead. EBS volumes preserve their data through instance stops and terminations, can be easily backed up with EBS snapshots, can be removed from one instance and reattached to another, and support full-volume encryption.

Snapshots

- You can back up the data on your Amazon EBS volumes to Amazon S3 by taking point-in-time snapshots. Snapshots are *incremental* backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved. This minimizes the time required to create the snapshot and saves on storage costs by not duplicating data. When you delete a snapshot, only the data unique to that snapshot is removed. Each snapshot contains all of the information that is needed to restore your data (from the moment when the snapshot was taken) to a new EBS volume.

Volumes and Snapshots

- When you create an EBS volume based on a snapshot, the new volume begins as an exact replica of the original volume that was used to create the snapshot. The replicated volume loads data in the background so that you can begin using it immediately. If you access data that hasn't been loaded yet, the volume immediately downloads the requested data from Amazon S3, and then continues loading the rest of the volume's data in the background.

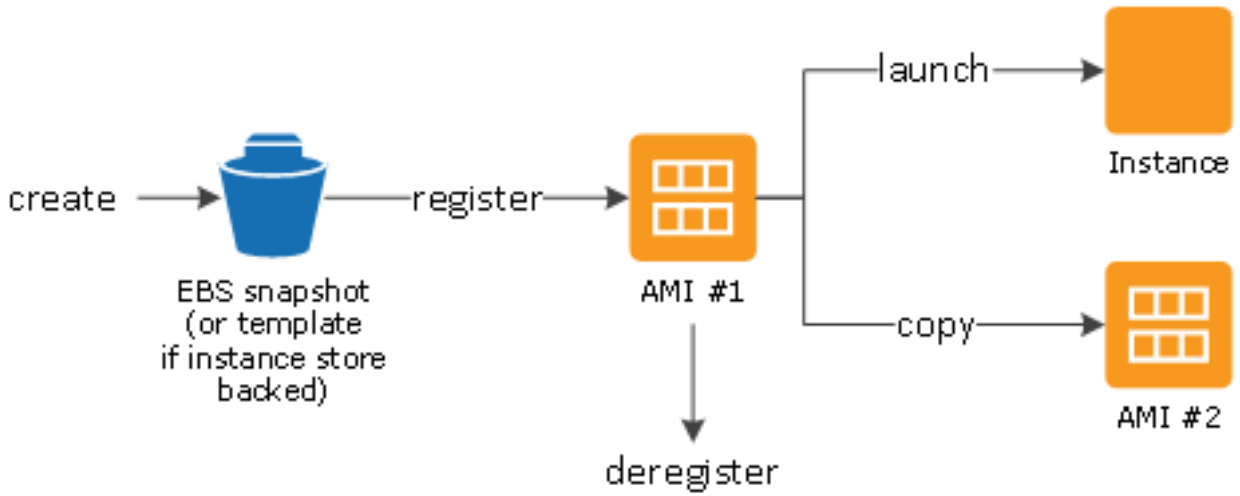
AMI – Amazon Machine Image

- An Amazon Machine Image (AMI) provides the information required to launch an instance. You must specify an AMI when you launch an instance. You can launch multiple instances from a single AMI when you need multiple instances with the same configuration. You can use different AMIs to launch instances when you need instances with different configurations.

An AMI includes the following:

- One or more EBS snapshots, or, for instance-store-backed AMIs, a template for the root volume of the instance (for example, an operating system, an application server, and applications).
- Launch permissions that control which AWS accounts can use the AMI to launch instances.
- A block device mapping that specifies the volumes to attach to the instance when it's launched.

AMI – Amazon Machine Image



EC2 Instance Metadata

- Instance metadata is data about your instance that you can use to configure or manage the running instance.
- Because your instance metadata is available from your running instance, you do not need to use the Amazon EC2 console or the AWS CLI. This can be helpful when you're writing scripts to run from your instance. For example, you can access the local IP address of your instance from instance metadata to manage a connection to an external application.
- http://169.254.169.254/latest/meta-data/
- The IP address 169.254.169.254 is a link-local address and is valid only from the instance

EC2 with IAM Role

- An EC2 instance can access any other service in AWS through an IAM role attached to the instance that gives the instance temporary credentials.
- For Eg : The website content is present in an S3 bucket in the Northern Virginia region. The EC2 instance is present in the same region and wants to access the data residing in the S3 bucket. Therefore an IAM role will be attached to the instance while creating the instance or after creating the instance to access the respective S3 bucket.

ELB – Elastic Load Balancer

- Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, and Lambda functions. It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones. Elastic Load Balancing offers three types of load balancers that all feature the high availability, automatic scaling, and robust security necessary to make your applications fault tolerant.

ELB – Elastic Load Balancer

- Application Load Balancer : Application Load Balancer is best suited for load balancing of HTTP and HTTPS traffic and provides advanced request routing targeted at the delivery of modern application architectures, including microservices and containers. Operating at the individual request level (Layer 7), Application Load Balancer routes traffic to targets within Amazon Virtual Private Cloud (Amazon VPC) based on the content of the request.

ELB – Elastic Load Balancer

- Network Load Balancer : Network Load Balancer is best suited for load balancing of Transmission Control Protocol (TCP), User Datagram Protocol (UDP) and Transport Layer Security (TLS) traffic where extreme performance is required. Operating at the connection level (Layer 4), Network Load Balancer routes traffic to targets within Amazon Virtual Private Cloud (Amazon VPC) and is capable of handling millions of requests per second while maintaining ultra-low latencies. Network Load Balancer is also optimized to handle sudden and volatile traffic patterns.

ELB – Elastic Load Balancer

- Classic Load Balancer : Classic Load Balancer provides basic load balancing across multiple Amazon EC2 instances and operates at both the request level and connection level. Classic Load Balancer is intended for applications that were built within the EC2-Classic network.

ELB – Target Groups

- Each *target group* is used to route requests to one or more registered targets. When you create each listener rule, you specify a target group and conditions. When a rule condition is met, traffic is forwarded to the corresponding target group. You can create different target groups for different types of requests. For example, create one target group for general requests and other target groups for requests to the microservices for your application.

ELB – Health Checks

Setting	Description
HealthCheckProtocol	The protocol the load balancer uses when performing health checks on targets. The possible protocols are HTTP and HTTPS. The default is the HTTP protocol.
HealthCheckPort	The port the load balancer uses when performing health checks on targets. The default is to use the port on which each target receives traffic from the load balancer.
HealthCheckPath	The ping path that is the destination on the targets for health checks. Specify a valid URI (*/path*?*query*). The default is /.
HealthCheckTimeoutSeconds	The amount of time, in seconds, during which no response from a target means a failed health check. The range is 2–120 seconds. The default is 5 seconds if the target type is `instance` or `ip` and 30 seconds if the target type is `lambda`.
HealthCheckIntervalSeconds	The approximate amount of time, in seconds, between health checks of an individual target. The range is 5–300 seconds. The default is 30 seconds if the target type is `instance` or `ip` and 35 seconds if the target type is `lambda`.
HealthyThresholdCount	The number of consecutive successful health checks required before considering an unhealthy target healthy. The range is 2–10. The default is 5.
UnhealthyThresholdCount	The number of consecutive failed health checks required before considering a target unhealthy. The range is 2–10. The default is 2.
Matcher	The HTTP codes to use when checking for a successful response from a target. You can specify values or ranges of values between 200 and 499. The default value is 200.

Launch Configuration & ASG

- A *launch configuration* is an instance configuration template that an Auto Scaling group uses to launch EC2 instances. When you create a launch configuration, you specify information for the instances. Include the ID of the Amazon Machine Image (AMI), the instance type, a key pair, one or more security groups, and a block device mapping. If you've launched an EC2 instance before, you specified the same information in order to launch the instance.
- You can only specify one launch configuration for an Auto Scaling group at a time, and you can't modify a launch configuration after you've created it. To change the launch configuration for an Auto Scaling group, you must create a launch configuration and then update your Auto Scaling group with it.

Launch Configuration & ASG

- An *Auto Scaling group* contains a collection of Amazon EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management.
- The size of an Auto Scaling group depends on the number of instances you set as the desired capacity. You can adjust its size to meet demand, either manually or by using automatic scaling.
- An Auto Scaling group starts by launching enough instances to meet its desired capacity. It maintains this number of instances by performing periodic health checks on the instances in the group. The Auto Scaling group continues to maintain a fixed number of instances even if an instance becomes unhealthy. If an instance becomes unhealthy, the group terminates the unhealthy instance and launches another instance to replace it.

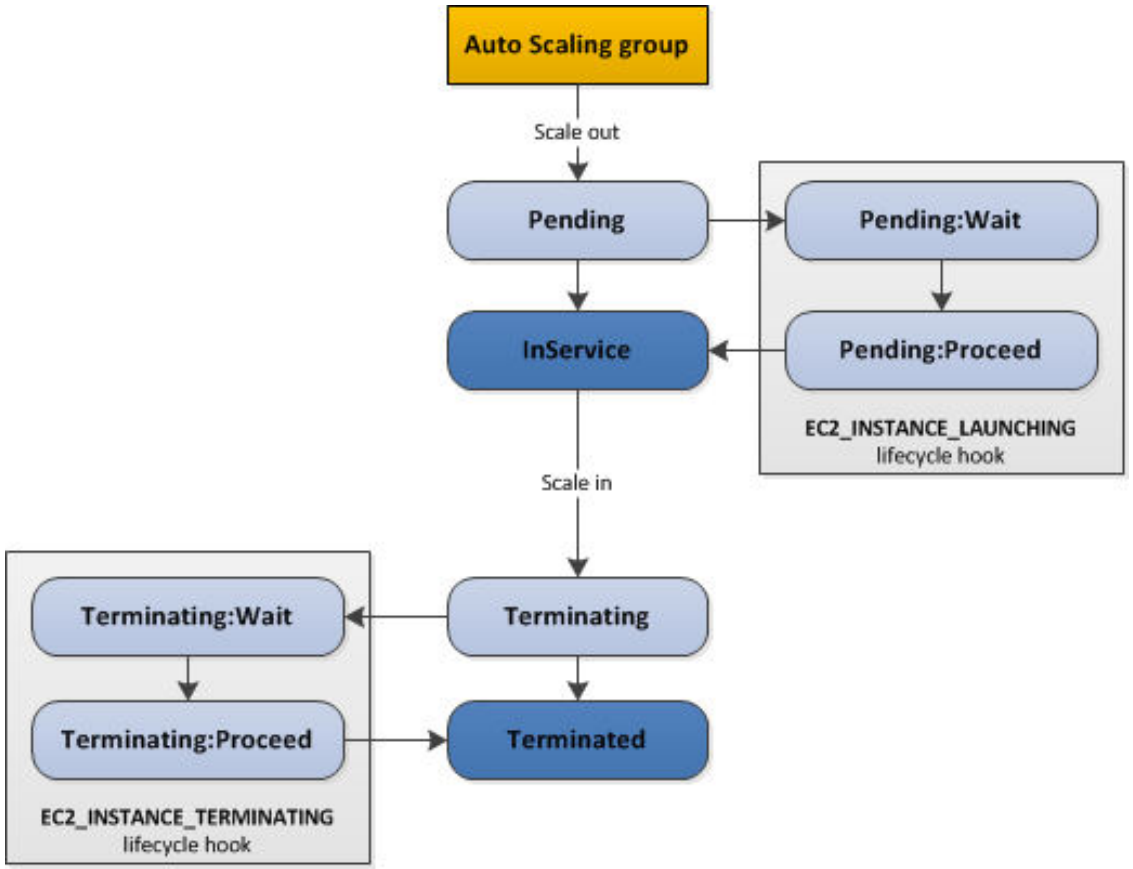
Launch Configuration & ASG

- You can use scaling policies to increase or decrease the number of instances in your group dynamically to meet changing conditions. When the scaling policy is in effect, the Auto Scaling group adjusts the desired capacity of the group, between the minimum and maximum capacity values that you specify, and launches or terminates the instances as needed. You can also scale on a schedule.
- An Auto Scaling group can launch On-Demand Instances, Spot Instances, or both. You can specify multiple purchase options for your Auto Scaling group only when you configure the group to use a launch template.

Launch Configuration & ASG

- When instances are launched, if you specified multiple Availability Zones, the desired capacity is distributed across these Availability Zones. If a scaling action occurs, Amazon EC2 Auto Scaling automatically maintains balance across all of the Availability Zones that you specify.

Auto Scaling Life Cycle Hooks



Auto Scaling Notification

- It is useful to know when Amazon EC2 Auto Scaling is launching or terminating the EC2 instances in your Auto Scaling group. Amazon SNS coordinates and manages the delivery or sending of notifications to subscribing clients or endpoints. You can configure Amazon EC2 Auto Scaling to send an SNS notification whenever your Auto Scaling group scales.
- For example, if you configure your Auto Scaling group to use the autoscaling: EC2_INSTANCE_TERMINATE notification type, and your Auto Scaling group terminates an instance, it sends an email notification.

HA Architecture using ASG and ELB