What is Amazon EC2?

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

Features of Amazon EC2

1. Virtual computing environments, known as instances

2. Preconfigured templates for your instances, known as Amazon Machine Images (AMIs), that package the bits you need for your server (including the operating system and additional software)

3. Various configurations of CPU, memory, storage, and networking capacity for your instances, known as instance types

4. Secure login information for your instances using key pairs (AWS stores the public key, and you store the private key in a secure place)

5. Storage volumes for temporary data that's deleted when you stop, hibernate, or terminate your instance, known as instance store volumes

6. Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as Amazon EBS volumes

7. Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as Regions and Availability Zones

8. A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using security groups

9. Static IPv4 addresses for dynamic cloud computing, known as Elastic IP addresses

10. Metadata, known as tags that you can create and assign to your Amazon EC2 resources

Instance type in EC2

(A) General purpose instance- (balance CPU & memory) in this 3 series available (T, M, A series instances) & MAC1

T series (t2, t3, t3a, t4g)-

These instances provide a baseline level of CPU performance with the ability to burst to a higher level when required by your workload. An Unlimited instance can sustain high CPU performance for any period of time whenever required. They are well-suited for the following:-

Websites and web applications

Code repositories

Development, build, test, and staging environments

Micro services

M series (m4, m5, m5a, m5n, m5zn, m6g, m6i)

M5 and M5a instances-

These instances provide an ideal cloud infrastructure, offering a balance of compute, memory, and networking resources for a broad range of applications that are deployed in the cloud. They are well-suited for the following:

Small and midsize databases Data processing tasks that require additional memory Caching fleet Backend servers for SAP, Microsoft SharePoint, cluster computing, and other enterprise applications

M5zn instances-

These instances are ideal for applications that benefit from extremely high single thread performance, high throughput, and low latency networking. They are well-suited for the following:

Gaming

High performance computing

Simulation modeling

M6g and M6gi instances-

These instances are powered by AWS Graviton2 processors and deliver balanced compute, memory, and networking for a broad range general purpose workloads. They are well suited for the following:

Application server

Micro services

Gaming servers

Midsize data stores

Caching fleets

Mac1 instances-

These instances are powered by Apple Mac mini computers. They provide up to 10 Gbps of network bandwidth and 8 Gbps EBS bandwidth through high-speed Thunderbolt 3 connections. They are well suited to develop, build, test, and sign applications for Apple devices, such as I-Phone, I-Pad, iPod, Mac, Apple Watch, and Apple TV.

Compute optimized instances-

(More CPU than RAM) here C series only

(c4, c5, c5a, c5n, c6i, c6g, c6gn)

Compute optimized instances are ideal for compute-bound applications that benefit from high-performance processors.

C5 and C5n instances-

These instances are well suited for the following:

Batch processing workloads

Media transcoding

High-performance web servers

High-performance computing (HPC)

Scientific modeling

Dedicated gaming servers and ad serving engines

Machine learning inference and other compute-intensive applications

C6g, C6i, and C6gn instances-

These instances are powered by AWS Graviton2 processors and are ideal for running advanced, compute-intensive workloads, such as the following:-

High-performance computing (HPC)

Batch processing

Ad serving

Video encoding

Gaming servers

Scientific modeling

Distributed analytics

CPU-based machine learning inference

Memory optimized instances-

(More RAM than CPU) (r4, r5, r5a, r5b, r5n, r6i, r6g, x1, x1e, x2gd, z1d & high memory)

These instances are well suited for the following:

High-performance, relational (MySQL) and NoSQL (MongoDB, Cassandra) databases.

Distributed web scale cache stores that provide in-memory caching of key-value type data (Memcached and Redis).

In-memory databases using optimized data storage formats and analytics for business intelligence (for example, SAP HANA).

Applications performing real-time processing of big unstructured data (financial services, Hadoop/Spark clusters).

High-performance computing (HPC) and Electronic Design Automation (EDA) applications.

High memory (u-\*) instances- 6 TB, 9 TB, and 12 TB instances are powered by 2.1 GHz (with Turbo Boost to 3.80 GHz) Intel® Xeon® Platinum 8176M (Skylake) processors

18 TB and 24 TB instances are powered by 2nd Generation 2.7 GHz (with Turbo Boost to 4.0 GHz) Intel® Xeon® Scalable (Cascade Lake) processors

Use Case - Ideal for running large enterprise databases, including production installations of SAP HANA in-memory database in the cloud.

Storage optimized instances-

Storage optimized instances are designed for workloads that require high, sequential read and write access to very large data sets on local storage. They are optimized to deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications.

D2 instances-

These instances are well suited for the following:

Massive parallel processing (MPP) data warehouse

Map Reduce and Hadoop distributed computing

Log or data processing applications

D3 and D3en instances-

These instances are well suited for the following:

Distributed file systems for Hadoop workloads

File storage workloads such as GPFC and BeeFS

Large data lakes for HPC workloads

I3 and I3en instances-

These instances are well suited for the following:

High frequency online transaction processing (OLTP) systems

Relational databases

NoSQL databases

Cache for in-memory databases (for example, Redis)

Data warehousing applications

Distributed file systems

H1 instances-

These instances are well suited for the following:

Data-intensive workloads such as Map Reduce and distributed file systems

Applications requiring sequential access to large amounts of data on direct-attached instance storage

Applications that require high-throughput access to large quantities of data

Accelerated Computing-

Accelerated computing instances use hardware accelerators, or co-processors, to perform functions, such as floating point number calculations, graphics processing, or data pattern matching, more efficiently than is possible in software running on CPUs.

G5, G4ad, G4dn & G3 instances-

GPU instances are designed to accelerate graphics-intensive applications and machine learning inference. They can also be used to train simple to moderately complex machine learning models.

P2, P3 & P4 instances- are the GPU-based instances and provide highest performance for machine learning training and high performance computing in the cloud.

use case-

Machine/Deep learning, high performance computing, computational fluid dynamics, computational finance, seismic analysis, speech recognition, autonomous vehicles, drug discovery.

Video transcoding instances (VT1)-

These instances are designed to accelerate video transcoding workloads, such as live broadcast, video conferencing, and just-in-time transcoding.

DL1 instances-

Amazon EC2 DL1 instances are powered by Gaudi accelerators from Habana Labs (an Intel company). They deliver up to 40% better price performance for training deep learning models compared to current generation GPU-based EC2 instances.

Use case- Deep learning training, object detection, image recognition, natural language processing, and recommendation engines.

INF1 instances-

Amazon EC2 Inf1 instances are built from the ground up to support machine learning inference applications.

Use case- Recommendation engines, forecasting, image and video analysis, advanced text analytics, document analysis, voice, conversational agents, translation, transcription, and fraud detection.

Purchasing option for EC2-

1. On-demand instance -
2. Spot instance-
3. Reserved instance -
4. Dedicated (host/instance)-

Placement Group-

When you launch a new EC2 instance, the EC2 service attempts to place the instance in such a way that all of your instances are spread out across underlying hardware to minimize correlated failures. You can use placement groups to influence the placement of a group of interdependent instances to meet the needs of your workload. Depending on the type of workload, you can create a placement group using one of the following placement group types-

1. Cluster-A cluster placement group is a logical grouping of instances within a single Availability Zone. Cluster placement groups are recommended for applications that benefit from low network latency, high network throughput, or both. They are also recommended when the majority of the network traffic is between the instances in the group. To provide the lowest latency and the highest packet-per-second network performance for your placement group, choose an instance type that supports enhanced networking.

You launch your instances in the following way:

1. Use a single launch request to launch the number of instances that you need in the placement group.

Use the same instance type for all instances in the placement group.

1. If you try to add more instances to the placement group later, or if you try to launch more than one instance type in the placement group, you increase your chances of getting an insufficient capacity error.
2. If you stop an instance in a placement group and then start it again, it still runs in the placement group. However, the start fails if there isn't enough capacity for the instance.
3. If you receive a capacity error when launching an instance in a placement group that already has running instances, stop and start all of the instances in the placement group, and try the launch again. Starting the instances may migrate them to hardware that has capacity for all of the requested instances.
4. A cluster placement group cannot span multiple availability zones.

1. Spread- A spread placement group is a group of instances that are each placed on distinct racks, with each rack having its own network and power source.
2. A spread placement group can span multiple Availability Zones in the same Region. You can have a maximum of seven running instances per Availability Zone per group.
3. If you start or launch an instance in a spread placement group and there is insufficient unique hardware to fulfill the request, the request fails. Amazon EC2 makes more distinct hardware available over time, so you can try your request again later.
4. Spread placement groups are recommended for applications that have a small number of critical instances that should be kept separate from each other.
5. Launching instances in a spread placement group reduces the risk of simultaneous failures that might occur when instances share the same racks. Spread placement groups provide access to distinct racks, and are therefore suitable for mixing instance types or launching instances over time.
6. Partition- Partition placements groups help reduce the likelihood of correlated hardware failures for your application. When using partition placement groups, Amazon EC2 divides each group into logical segments called partitions. Amazon EC2 ensures that each partition within a placement group has its own set of racks. Each rack has its own network and power source. No two partitions within a placement group share the same racks, allowing you to isolate the impact of hardware failure within your application.
7. Partition placement groups can be used to deploy large distributed and replicated workloads, such as HDFS, HBase, and Cassandra, across distinct racks.
8. When you launch instances into a partition placement group, Amazon EC2 tries to distribute the instances evenly across the number of partitions that you specify. You can also launch instances into a specific partition to have more control over where the instances are placed.
9. A partition placement group supports a maximum of seven partitions per Availability Zone. The number of instances that you can launch in a partition placement group is limited only by your account limits.
10. When instances are launched into a partition placement group, Amazon EC2 tries to evenly distribute the instances across all partitions. Amazon EC2 doesn’t guarantee an even distribution of instances across all partitions.
11. A partition placement group with Dedicated Instances can have a maximum of two partitions.


                    A partition placement group with three partitions
                

EBS Volume

An Amazon EBS volume is a durable, block-level storage device that you can attach to your instances. After you attach a volume to an instance, you can use it as you would use a physical hard drive.

EBS volumes are flexible. For current-generation volumes attached to current-generation instance types, you can dynamically increase size, modify the provisioned IOPS capacity, and change volume type on live production volumes.

You can use EBS volumes as primary storage for data that requires frequent updates, such as the system drive for an instance or storage for a database application. You can also use them for throughput-intensive applications that perform continuous disk scans.

You can attach multiple EBS volumes to a single instance. The volume and instance must be in the same Availability Zone.

Depending on the volume and instance types, you can use Multi-Attach to mount a volume to multiple instances at the same time.

Multi-Attach volume

Amazon EBS Multi-Attach enables you to attach a single Provisioned IOPS SSD (io1 or io2) volume to multiple instances that are in the same Availability Zone. You can attach multiple Multi-Attach enabled volumes to an instance or set of instances. Each instance to which the volume is attached has full read and write permission to the shared volume.

Limitations-

Multi-Attach enabled volumes can be attached to up to 16 Linux instances built on the Nitro System that are in the same Availability Zone.

Multi-Attach is supported exclusively on Provisioned IOPS SSD volumes.

Multi-Attach for io1 volumes are available in the following Regions only: us-east-1, us-west-2, eu-west-1, and ap-northeast-2.

Multi-Attach for io2 and io2 Block Express volumes are available in all Regions that support those volumes types’ volumes.

Multi-Attach enabled volumes can't be created as boot volumes.

Multi-Attach can't be enabled during instance launch using either the Amazon EC2 console or Run Instances API.

Benefits of using EBS volumes

Data availability

### Data persistence

### Data encryption

### Snapshots

### Flexibility

EBS volume types

Solid state drives (SSD)

Hard disk drives (HDD)

Previous generation (Magnetic)

1. Solid state drives (SSD)- Optimized for transactional workloads involving frequent read/write operations with small I/O size, where the dominant performance attribute is IOPS.

SSD storage divided into four categories- (GP2, GP3, IO1, IO2)

GP2-

General Purpose SSD (gp2) volumes offer cost-effective storage that is ideal for a broad range of workloads.

These volumes deliver single-digit millisecond latencies and the ability to burst to 3,000 IOPS for extended periods of time. Between a minimum of 100 IOPS (at 33.33 GiB and below) and a maximum of 16,000 IOPS (at 5,334 GiB and above), baseline performance scales linearly at 3 IOPS per GiB of volume size.

AWS designs gp2 volumes to deliver their provisioned performance 99% of the time.

gp2 volume can range in size from 1 GiB to 16 TiB.

GP3 –

General Purpose SSD (gp3) volumes offer cost-effective storage that is ideal for a broad range of workloads.

These volumes deliver a consistent baseline rate of 3,000 IOPS and 125 MiB/s, included with the price of storage. You can provision additional IOPS (up to 16,000) and throughput (up to 1,000 MiB/s) for an additional cost.

Gp3 volume can range in size from 1 GiB to 16 TiB.

Provisioned IOPS SSD volumes (IO1)-

IO1 volume can range in size from 4 GiB to 16 TiB.

io1 volumes are designed to provide 99.8 to 99.9 percent volume durability with an annual failure rate (AFR) no higher than 0.2 percent, which translates to a maximum of two volume failures per 1,000 running volumes over a one-year period.

The maximum ratio of provisioned IOPS to requested volume size (in GiB) is 50:1 for io1 volumes

For example, a 100 GiB io1 volume can be provisioned with up to 5,000 IOPS

Provisioned IOPS SSD volumes (IO2)-

IO2 volume can range in size from 4 GiB to 16 TiB.

Io2 volumes are designed to provide 99.999 percent volume durability with an AFR no higher than 0.001 percent, which translates to a single volume failure per 100,000 running volumes over a one-year period.

The maximum ratio of provisioned IOPS to requested volume size (in GiB) is 500:1 for io2 volumes

For example, a 100 GiB io2 volume can be provisioned with up to 5 0000 IOPS

Io2 Block Express volumes

io2 Block Express volumes are supported with R5b instances only.

Sub-millisecond average latency

Storage capacity up to 64 TiB (65,536 GiB)

Provisioned IOPS up to 256,000, with an IOPS:GiB ratio of 1,000:1. Maximum IOPS can be provisioned with volumes 256 GiB in size and larger (1,000 IOPS × 256 GiB = 256,000 IOPS).

Cold HDD volumes (SC1)

Volume size – 125GB-64TB

Cold HDD (sc1) volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. With a lower throughput limit than st1, sc1 is a good fit for large, sequential cold-data workloads.

If you require infrequent access to your data and are looking to save costs, sc1 provides inexpensive block storage.

Bootable sc1 volumes are not supported.

Throughput Optimized HDD volumes (ST1)-

Volume size- 125GB-64TB

Throughput Optimized HDD (st1) volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS.

This volume type is a good fit for large, sequential workloads such as Amazon EMR, ETL, data warehouses, and log processing.

Bootable st1 volumes are not supported.

Magnetic volumes

Volume size – 1GB-1TB

Magnetic volumes are backed by magnetic drives and are suited for workloads where data is accessed infrequently and scenarios where low-cost storage for small volume sizes is important.

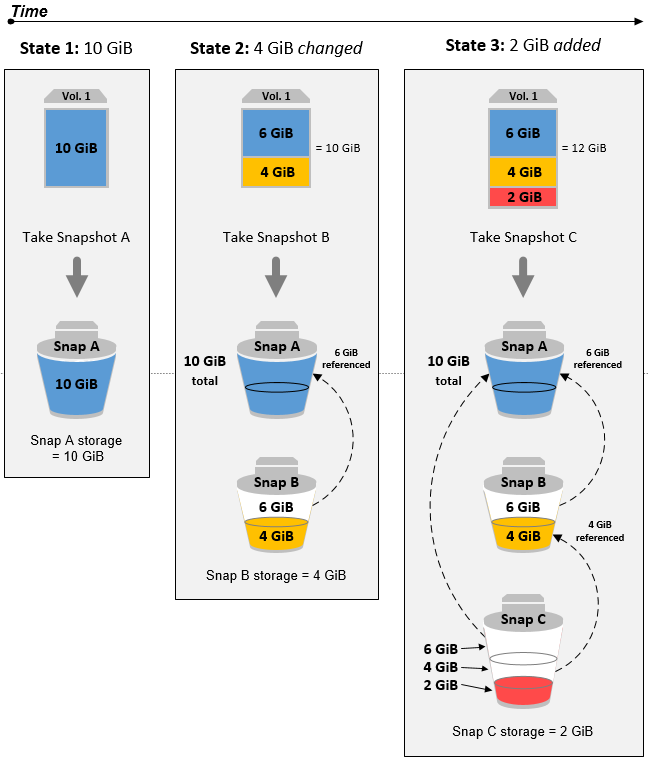
These volumes deliver approximately 100 IOPS on average, with burst capability of up to hundreds of IOPS

EBS 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 mean 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.

Incremental snapshots-

Volume has 10 GiB of data. Because Snap A is the first snapshot taken of the volume, the entire 10 GiB of data must be copied.

Volume still contains 10 GiB of data, but 4 GiB have changed. Snap B needs to copy and store only the 4 GiB that changed after Snap A was taken. The other 6 GiB of unchanged data, which are already copied and stored in Snap A

2 GiB of data have been added to the volume, for a total of 12 GiB. Snap C needs to copy the 2 GiB that were added after Snap B was taken. Snap C also references 4 GiB of data stored in Snap B, and 6 GiB of data stored in Snap A.

Amazon EC2 Auto Scaling-

Amazon EC2 Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application.

You create collections of EC2 instances, called Auto Scaling groups.

You can specify the minimum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes below this size.

You can specify the maximum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes above this size.

If you specify the desired capacity, either when you create the group or at any time thereafter, Amazon EC2 Auto Scaling ensures that your group has this many instances.

Auto scaling components

Auto scaling group

Launch template

Scaling policy

Launch templates- A launch template is similar to a launch configuration, in that it specifies instance configuration information.

It includes the ID of the Amazon Machine Image (AMI), the instance type, a key pair, security groups, and other parameters used to launch EC2 instances.

However, defining a launch template instead of a launch configuration allows you to have multiple versions of a launch template.

For example, you can create a launch template that defines a base configuration without an AMI or user data script. After you create your launch template, you can create a new version and add the AMI and user data that has the latest version of your application for testing. This results in two versions of the launch template.

We recommend that you use launch templates to ensure that you're accessing the latest features and improvements. Not all Amazon EC2 Auto Scaling features are available when you use launch configurations.

Amazon EC2 Auto Scaling Capacity Rebalancing-

1. Amazon EC2 Auto Scaling is aware of EC2 instance rebalance recommendation notifications. The Amazon EC2 Spot service emits these notifications when Spot Instances are at elevated risk of interruption.
2. When Capacity Rebalancing is enabled for an Auto Scaling group, Amazon EC2 Auto Scaling attempts to proactively replace Spot Instances in the group that have received a rebalance recommendation, providing the opportunity to rebalance your workload to new Spot Instances that are not at elevated risk of interruption.
3. This means that your workload can continue to process the work while Amazon EC2 Auto Scaling launches a new Spot Instance before an existing instance is interrupted.
4. Scaling behavior-

Instances distribution total no. of ec2 running

Example1 10 20 30 40

On demand base 10

On-Demand percentage above base: 50%

Spot percentage: 50%

Example2 10 20 30 40

On-Demand base: 0

On-Demand percentage above base: 0%

Spot percentage: 100%

Example3 10 20 30 40

On-Demand base: 0

On-Demand percentage above base: 60%

Spot percentage: 40%

Example4 10 20 30 40

On-Demand base: 0

On-Demand percentage above base: 100%

Spot percentage: 0%

Example5 10 20 30 40

On-Demand base: 12

On-Demand percentage above base: 0%

Spot percentage: 100%

Cool down period- when using auto scaling group and using any metric type based on load increase & two new ec2 created. But again alarm coming because still load increasing requires more ec2 instance. But ASG ignore all alarm till 300 sec. after 300 sec alarm coming then create new ec2.

Warm up period- when we launching new ec2 instance till 300 sec we wait to start ec2 instance that time called warm up period.

Scaling policy- Scaling is the ability to increase or decrease the compute capacity of your application. Scaling starts with an event, or scaling action, which instructs an Auto Scaling group to either launch or terminate Amazon EC2 instances.

Dynamic scaling-

A dynamic scaling policy instructs Amazon EC2 Auto Scaling to track a specific Cloud Watch metric, and it defines what action to take when the associated Cloud Watch alarm is in ALARM.

The metrics that are used to trigger an alarm are an aggregation of metrics coming from all of the instances in the Auto Scaling group.

(For example, let's say you have an Auto Scaling group with two instances where one instance is at 60 percent CPU and the other is at 40 percent CPU. On average, they are at 50 percent CPU.) When the policy is in effect, Amazon EC2 Auto Scaling adjusts the group's desired capacity up or down when the alarm is triggered.

Dynamic scaling policy types-

Target tracking scaling- Increase or decrease the current capacity of the group based on a target value for a specific metric. This is similar to the way that your thermostat maintains the temperature of your home—you select a temperature and the thermostat does the rest.

Step scaling-Increase or decrease the current capacity of the group based on a set of scaling adjustments, known as step adjustments, that vary based on the size of the alarm breach.

Simple scaling-Increase or decrease the current capacity of the group based on a single scaling adjustment.

Predictive scaling-

Use predictive scaling to increase the number of EC2 instances in your Auto Scaling group in advance of daily and weekly patterns in traffic flows.

Predictive scaling is well suited for situations where you have:

Cyclical traffic, such as high use of resources during regular business hours and low use of resources during evenings and weekends

Recurring on-and-off workload patterns, such as batch processing, testing, or periodic data analysis

Applications that takes a long time to initialize, causing a noticeable latency impact on application performance during scale-out events

Scheduled scaling-

Scheduled scaling helps you to set up your own scaling schedule according to predictable load changes.

For example, let's say that every week the traffic to your web application starts to increase on Wednesday, remains high on Thursday, and starts to decrease on Friday. You can configure a schedule for Amazon EC2 Auto Scaling to increase capacity on Wednesday and decrease capacity on Friday.

ELB (Elastic Load Balancer)

Load balancer distributes the web traffic to the available server.

ELB maintain load balancing between different servers. If any server not working ELB not send traffic to that server.

ELB types

1. Classic Load Balancer- the Classic Load Balancer supports load balancing of applications using HTTP, HTTPS, SSL and TCP protocols.

Protocols port supported are (1-65535) for EC2-VPC

Protocols port supported are (25, 80, 443, 465, 587, 1024-65535) for EC2 classic.

It supports ipv4, ipv6 & dual stack.

Classic load balancer work 4th & 7th OSI model layer.

By default cross zone load balancing disable here. But you can enable it if require.

AWS will be retiring the EC2-Classic network on August 15, 2022.

1. Application Load Balancer- Application Load Balancer supports load balancing of applications using HTTP and HTTPS protocols.

Protocols port supported are (1-65535)

Application load balancer work 7th OSI model layer.

It supports ipv4, ipv6 & dual stack

By default cross zone load balancing is enable here & no option to disable it.

Limit –

Load balancers per Region: 50

Target groups per Region: 3000

Listeners per load balancer: 50

Targets per load balancer: 1000

Target groups per load balancer: 100

Subnets per Availability Zone per load balancer: 1

Security groups per load balancer: 5

1. Network Load Balancer- Network load balancer work 4th OSI model layer

It supports ipv4, ipv6 & dual stack

Network load balancer supports TCP, UDP & TLS protocols.

Cross zone load balancing disabled by default. But you can enable it & charges apply for transferring data.

Limit-

Network Load Balancers per Region: 50

Network Load Balancer ENIs per Amazon VPC: 300

Target groups per Region: 3,000

Listeners per load balancer: 50

Targets per load balancer: 3,000

Subnets per Availability Zone per load balancer: 1

1. Gateway Load Balancer- Gateway load balancer work 3rd OSI model layer.

Gateway load balancer support GENEVE protocols.

Gateway Load Balancer runs within one AZ.

By default cross zone load balancing disabled. You can enable it after creation of load balancer.

Limit-

Gateway Load Balancers per Region: 20

Gateway Load Balancers per VPC: 10

Target groups with GENEVE protocol: 100