

AWS EC2 :

EC2 :

It is a web service that provides resizable compute capacity in the cloud. It reduces the time required to obtain and boot new server instances to minutes, allowing you to scale capacity up and down, as per requirements.

4 different pricing models :

1. **On Demand** : Allows you to pay a fixed rate by the hour with no commitment. Good for apps where compute needs scaling up/down - i.e. usage might increase 10x during certain hours of the day, or certain times of year.

2. **Reserved** : Provides capacity reservation, and offers slight discount on the hourly charge for an instance. Contract terms are 1 year or 3 year terms.

3. **Spot instance** : Enables you to bid whatever price you want for the instance capacity, works like stock market. Provides greater savings if your application has flexible start and end timings.

- Can be extremely cheap
- Can be terminated by you OR AWS at any time
- Best for jobs which can be terminated at any time i.e. certain types of batch processing
- Not charged for partial hour if your instance is terminated by AWS.. charged for the FULL hour if YOU terminate your instance.
- Good for massively parallel computations, or high-compute batch jobs, due to the fact that you can get spot instances for often 50-90% less than on-demand instances, you can massively increase your compute capacity by 2-10x for the same budget.

4. **Dedicated hosts** : *Physical EC2 server dedicated for your use*. They can allow you to reduce costs by allowing you to use your existing server bound software licenses. physical EC2 server available only to you. No shared. i.e. if a regulatory body says that you must not be using multi-tenant computing.

If the spot instance is stopped by AWS, you will not be charged for the partial hour of usage. But, if you terminate the instance yourself, you will be charged for any hour in which the instance ran.

Instance Types :

-- **FIGHT DR McPx zau** (fight Dr Mac pix Zee AU)

F - FPGA

I - IOPS
G - Graphics
H - High throughput
T - General cheap purpose / T2 Micro
D - density
R - RAM
M - Main choice for general purpose apps
C - compute
P - Graphics (Think Pics)
X - Extreme memory
Z - Extreme memory and CPU
A - Arm based workloads
U - Bare metal

Instance startup and termination :

When an EC2 instance is terminated, the root EBS volume is also deleted by default.

[Termination Protection](#) is off by default, and can be used to prevent accidental termination

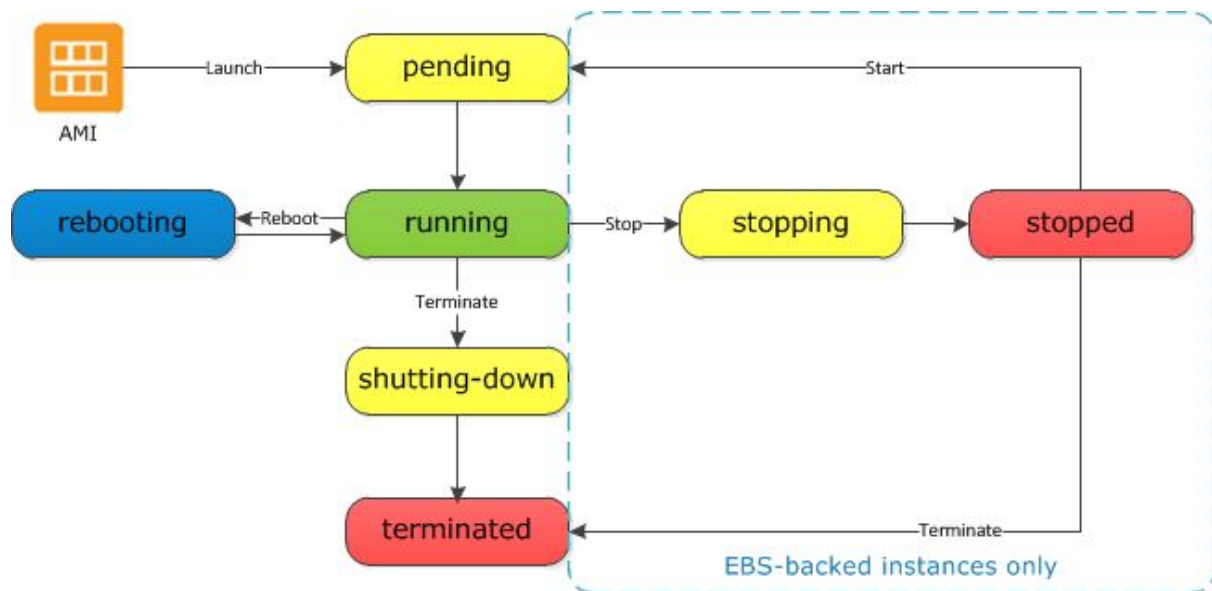
On default AMIs, the EBS root volume can be encrypted via 3rd party software, but not in the AWS console.

Use 'lsblk' to get a list of mounted disks

To get the reason for an EC2 instance termination from the CLI, you can use the following command: ****aws ec2 describe-instances**** along with the terminated instance id. You will receive a response similar to the following

```
"StateReason" {  
  "Message": "Client.UserInitiatedShutdown: User initiated shutdown",  
  "Code": "Client.UserInitiatedShutdown"  
}
```

Instance Lifecycle :



Difference between Dedicated instance and dedicated hosts:

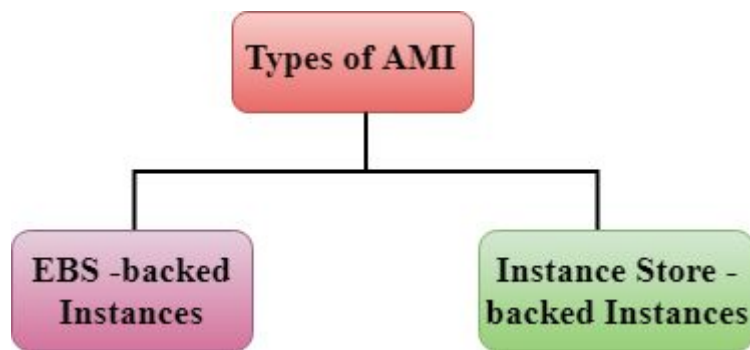
AMI

- An AMI stands for **Amazon Machine Images**.
- An AMI is a virtual image used to create a virtual machine within an EC2 instance.
- You can also create multiple instances using single AMI when you need instances with the same configuration.
- You can also create multiple instances using different AMI when you need instances with a different configuration.
- It also provides a template for the root volume of an instance

AMI Lifecycle

- First, you need to create and register an AMI.
- You can use an AMI to launch EC2 instances.
- You can also copy an AMI to some different region.
- When AMI is no longer required, then you can also deregister it.

AMI Types



AMI is divided into two categories:

- EBS - backed Instances
- Instance Store - backed Instances

There are different types of AMI images

- HVM (Hardware Virtual Machine) EBS-Backed - supported by all instance types (T2, M4, etc)
- HVM Instance Store - supported by M3, C3, X3, R3, I2, D2
- PV (Paravirtual) EBS-Backed - supported by M3, C3
- PV Instance Store - Supported by M3, C3

A Golden Image is an image which you've customised to your liking with all the necessary software, configuration, etc ready to go and saved as a personal AMI, from which you can launch instances.

EBS - backed Instances

- EBS is nothing but a volume that provides you persistent storage.
- When you run an EC2 instance that provides you temporary storage, if you delete an EC2 instance then the data stored in the EC2 instance will also be deleted. To make a data persistent, Amazon provides an EBS Volume. If you launch an EC2 instance and want to make some data persistent, then you need to attach an instance with the EBS Volume so that your data would be available even on deleting an EC2 instance.

- When you launch an EC2 instance, it will always have a root device as an EBS Volume which makes the data persistent. Therefore, we can say that when we delete an EC2 instance, then the data is available in a root device.
- In EBS - backed instances, you will be charged or billed for the storage of static data such as operating systems files, etc.

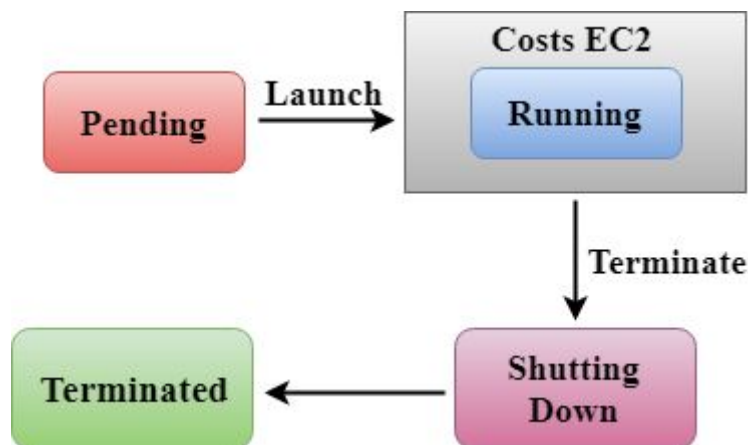
The cost of adding the EBS Volume to an EC2 instance is minimal.

Instance Store - backed Instances

- In Instance-Store, an instance consists of storage approx 1 TB or 2 TB which is temporary storage. As soon as the instance is terminated, all the data will be lost. For example, if you launch an instance, and deploy the database in it. If you delete an instance, then all the data will be lost and this becomes the challenge. In such a scenario, you can add an additional EBS Volume that also stores the data, so even if you delete an instance, your data would not be lost.
- In this case, EBS Volume is not a root volume. It's an additional volume that you attach to your EC2 instance manually.

Why EBS - backed instance is more popular than Instance Store - backed instance?

Instance Store - backed instances



In **Instance Store - backed instance**, if you launch an instance, it would be in a pending state. After pending state, an instance comes in a running state then it would be in a shutting down state. **Amazon would charge you only when it is in a running state.** When you terminate an instance, Amazon would not charge you any cost. For example, if you want to run an instance for 4 hours a day and it would cost you 10 cents per hour. In instance store, my instance would be running 24 hrs a day as it has no stopped state. Therefore, it would cost 72 dollars a month.

- **EBS - backed Instances**

In EBS - backed instances, an instance can be either in a running state or in a stopped state. In this case, Amazon would cost you only for a running state, not for a stopped state. For example, if you want to run an instance for 4 hours a day and it would cost you 10 cents per hour. In EBS - backed instance, an instance will run for 4 hours as it has stopped state as well. I take a 100 GB volume that would cost you 5 dollars. The running cost of an instance would be 12 dollars in a month. Therefore, the total cost taken by this instance is volume cost plus running cost which is equal to 17 dollars.

On an EBS backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. The additionally attached EBS volumes are not deleted when the instance is terminated, unless the checkbox is checked to delete the additional EBS volumes on termination of instance.

Difference b/w Instance store & EBS - backed instance

Characteristics	EBS-backed instance	Instance Store-backed instance
Lifecycle	It supports stopping as well as restarting of an instance by saving the state to EBS volume.	In this case, an instance cannot be stopped . It can be either in a running or terminated state.
Data Persistence	Data persists in EBS volume. If an instance is terminated, no data would be lost.	Data does not persist so when instance is terminated, data would be lost.
Boot time	It takes less than 1 min.	It usually takes less than 5 min.
Size limit	1 TB	10 - 16 TB
AMI creation	AMI is very easily created by using a single command.	To create an AMI, it requires installation and AMI tools.
Expensive	It is less expensive.	It is more expensive as compared to Instance Store-backed instance.

Instance Metadata

Available from the EC2 instances at the following URL:

<http://169.254.169.254/latest/meta-data/>

Can get the instance IP address via: <http://169.254.169.254/latest/meta-data/public-ipv4/>

EBS :

- EBS stands for **Elastic Block Store**.
- EC2 is a virtual server in a cloud while EBS is a virtual disk in a cloud.
- Amazon EBS allows you to create storage volumes and attach them to the EC2 instances.
- Once the storage volume is created, you can create a file system on the top of these volumes, and then you can run a database, store the files, applications or you can even use them as a block device in some other way.
- Amazon EBS volumes are placed in a specific availability zone, and they are **automatically replicated** to protect you from the failure of a single component. When you create an EBS volume in an AZ, it is automatically replicated within that zone to prevent data loss due to the failure of any single hardware component.
- EBS can't tolerate an entire AZ failure - EBS volumes are **only replicated within the AZ**, so S3 recommends always keeping a snapshot of your EBS volumes in an S3 bucket for high durability.
- EBS volume does not exist on one disk, it spreads across the Availability Zone. EBS volume is a disk which is attached to an EC2 instance.
- EBS volume attached to the EC2 instance where windows or Linux is installed known as **Root device** of volume. If the root volume of an EC2 instance fails, and you need to recover data from it, you can:
 - Detach the volume,
 - Attach it to another instance as a data volume,
 - Fix issues in the files, copy data out if necessary

- Re-attach to the original instance, and restart

EBS volumes appear as native block devices, similar to a hard drive of other physical device.

EBS volumes **can only be scaled up**, not down

All EBS volumes can be changed on the fly, except for Magnetic (standard), however if you do this, [you'll need to wait 6 hours before making any further changes to the volume](#).

For changing a volume, the best practice is to first stop the EC2 instance it's attached to.

-- Termination protection is turned off by default, so if you want to protect your EC2 instance from being accidentally deleted, turn ON the Termination protection. By doing so, no one can terminate the EC2 unless the option is turned off.

- On an EBS backed instance, **the default action is for the root EBS volume to be deleted when the instance is terminated**. The additionally attached EBS volumes are not deleted when the instance is terminated, unless the checkbox is checked to delete the additional EBS volumes on termination of instance.
- EBS **root volumes** of your default AMIs can be **encrypted**. You can also use a third party tool like Bit Locker to encrypt the root volume, or this can be done *while creating AMIs*, in the AWS console or by using the API.
- Instance store volumes is sometimes called 'ephemeral storage'. If the underlying host fails or stops, all data will be lost. No data is lost on reboot.
- Additional volumes can be encrypted.

EBS types :

General purpose SSD : general purpose that balances price and performance for most of the transactional workloads

-- 1 GB to 16 Tib

-- gp2

-- max iops per volume 16000

- General Purpose SSD is also sometimes referred to as a GP2.
- It is a General purpose SSD volume that **balances both price and performance**.

- You can get a ratio of 3 IOPS per GB with up to 10,000 IOPS and the ability to burst up to 3000 IOPS for an extended period of time for volumes at 3334 GiB and above. For example, if you get less than 10,000 IOPS, then GP2 is preferable as it gives you the best performance and price

IOPS - Number of read write operations mostly useful for OLTP transactions used in AWS for DBs like Cassandra.

Throughput - Is the number of bit transferred per sec. i.e.data transferred per sec. Mainly a unit for high data transfer applications like big data hadoop,kafka streaming

The Disk IOPS Describes the count of input/output operations on the disk per seconds, regardless block size.

The disk throughput describes how many data may be transferred per second, so the block size play a huge role upon calculating the throughput required by app

Provisioned IOPS SSD

: highest performance SSD designed for mission critical apps

- Databases
- io1
- 4Gib - 16 Tib
- 64000 iops per volume

- It is also referred to as IO1.
- It is mainly used for high-performance applications such as intense applications, relational databases.
- It is designed for *I/O intensive applications* such as **large relational or NOSQL databases.**
- It is used when you require more than 10,000 IOPS.

Throughput Optimized HDD

: Low cost HDD volume designed for frequently accessed, throughput intensive workloads.

- : *Big data and warehouses*
- : st1

: 500 Gib - 16 Tib, 500 iops

- It is also referred to as ST1.
- Throughput Optimized HDD is a low-cost HDD designed for those applications that require *higher throughput* up to 500 MB/s.
- It is useful for those applications that require the *data to be frequently accessed*.
- It is used for Big data, Data warehouses, Log processing, etc.
- It cannot be a boot volume, so it contains some additional volume. For example, if we have Windows server installed in a C: drive, then C drive cannot be a Throughput Optimized Hard disk, D: drive or some other drive could be a Throughput Optimized Hard disk.
- The size of the Throughput Hard disk can be 500 GiB to 16 TiB.
- It supports up to 500 IOPS.

Cold HDD

: Lowest cost HDD volume for ***less frequently accessed workloads***

-- File servers

- sc1

-- 500 Gib to 16Tib

- 250

- It is also known as SC1.
- It is the lowest cost storage designed for the applications where the workloads are infrequently accessed.
- It is useful when data is rarely accessed.
- It is *mainly used for a File server*.
- It cannot be a boot volume.
- The size of the Cold Hard disk can be 500 GiB to 16 TiB.
- It supports up to 250 IOPS.

Magnetic EBS

: *previous generation*

: workloads where data is infrequently accessed

: standard, 1Gib to 1Tib

- It is the lowest cost storage per gigabyte of all EBS volume types.
- It is ideal for the applications where the data is accessed infrequently
- It is useful for applications where the lowest storage cost is important.
- Magnetic volume is the only hard disk which is bootable. Therefore, we can say that it can be used as a boot volume.

RAID Arrays

To increase performance, it's possible to configure EBS volumes as a RAID array.

RAID #	
RAID 0	Striped, no redundancy.
RAID 1	Mirrored, has redundancy.
RAID 5	Good for reads, bad for writes. Can rebuild RAID array if necessary. Note: AWS strongly discourages use of RAID 5. Do not fall for it in the exam.
RAID 10	Striped and mirrored. Good redundancy, and good performance.

Generally use RAID 0 (no redundancy) or RAID 10 (good redundancy)

Once you provision your EBS volumes for the EC2 instance, for Windows instances, RAID is configured via Disk Management in the instance itself.

Due to caching, and to prevent any I/O while the snapshot is being created, before taking a snapshot of a RAID array, you'll need to:

1. Freeze the file system
2. Unmount the RAID array
3. Shut down the EC2 instance

If you need to minimize the downtime while backing up RAIDed EBS volumes, you can:

1. Suspend disk I/O
2. Start the EBS snapshot of volumes

3. Wait for snapshots to complete
4. Resume disk I/O

Snapshots :

Volumes exist on EBS, think of EBs as a virtual hard disk.

Snapshot exist on S3, think of Snapshot as a photograph of the disk.

Snapshots are point in time copies of volumes.

Snapshots are incremental - this means that blocks which have changed since the last snapshot are moved to S3.

If it is the first snapshot it may take time to create.

To create a snapshot for the volumes that serve as root devices, you should stop the instance before taking the snapshot.

However, you can also take the snapshot while the instance is running. While a snapshot is pending, it's safe to use the EBS volume. An in-progress snapshot is not affected by reads and writes to the volume.

You can create AMIs from both volumes and snapshots.

You can change EBS volume sizes on the fly, including changing the size and storage type. volumes will always be in the same AZ as the EC2 instance.

To transfer between instances, snapshot the volume, and use it to create a new volume in the desired AZ.

Move a volume

Move a volume to a new availability zone :

like from US east 1 to US east 2 :

To replicate EBS into a different AZ,

create a snapshot of the volume, then create its image and launch the EC2 with this image.

1. Go to actions, create a snapshot.. Snapshot is a photograph of the volume at that instance.
2. now, click on the snapshot , actions, and create an image. We will use this image to be deployed to other availability zones.
3. If we want to create multiple deployments of the image, use the option 'Hardware assisted virtualization' while launching the image, in the option 'Virtualization type'.
4. Now, once the image is created, we can use this to provision new EC2 instances.
5. Images created will be found under AMIs.
6. Go to AMIs. select the image and click Launch from above.
7. We see all EC2 instances available for launching, so we can configure the instance details.

8. HVM will give you many more EC2 options, if very less EC2 options are visible for selection, then it is possible that you selected Paravirtual type virtualization.
9. Now complete and launch the EC2, select a different subnet in a different AZ, so that this instance is launched in a new AZ, but with the copy of the same EBS.

Copy a AMI from one region to another :

We can also copy a volume from one region to another.

Go to AMIs, select Actions and copy AMI.

Select destination Region.

Then complete the copy process.

Once it is copied to the new region, we can launch it from there and then the volume would be launching a new instance in a new region altogether, instead of a new AZ.

Also, you can copy the AMI into a different AZ or a different region.

When you terminate an EC2, the root device volume is terminated but the attached volumes are persisted, not deleted.

Snapshots can be shared with other accounts, or shared in the AWS marketplace.

There are two types of snapshots supported:

- Point in time - single copy of entire volume
- Incremental
 - The first snapshot can take some time to create, and will be large, as it's backing up the entire volume
 - Subsequent snapshots are smaller as only new or changed data is snapshotted

Snapshot encryption

Can encrypt snapshots via the 'copy' option; if you have an EC2 instance that has an unencrypted volume, and you want to create an encrypted volume from it:

1. Create a snapshot of your unencrypted EBS volume. This snapshot will also be unencrypted.
2. Copy the snapshot, ensuring that the Encrypt this snapshot checkbox is checked
3. Restore the encrypted snapshot to a new volume, which will also be encrypted

Create Snapshot

Volume ⓘ

vol-0a3513f1d6d4c5aac

Name ⓘ

FooBar

Description ⓘ

FooBar

Encrypted ⓘ

No

Cancel

Create

Copy Snapshot

This snapshot, **snap-0bd92c9b205a44505 (FooBar)**, will be copied to a new snapshot. Set the new snapshot settings below:

Destination Region

Asia Pacific (Sydney) ⓘ

Description

[Copied snap-0bd92c9b205a44505 from ap-southeast-2] FooBa ⓘ

Encryption

☒ Encrypt this snapshot ⓘ

Master Key

(default) aws/ebs ⓘ

Snapshots of already encrypted volumes are encrypted automatically. Similarly, volumes restored from encrypted snapshots are encrypted automatically.

Create Volume

Volume Type

General Purpose SSD (GP2) ⓘ

Size (GiB)

1

(Min: 1 GiB, Max: 16384 GiB) ⓘ

IOPS

100 / 3000

(Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS) ⓘ

Availability Zone*

ap-southeast-2a ⓘ

Throughput (MB/s)

Not applicable ⓘ

Snapshot ID

snap-03eca1ec810acba30 ⓘ

Encrypted

True, volumes created from encrypted snapshots are automatically encrypted

Encrypted snapshots cannot be shared - the encryption key is tied to the AWS root account.

Logging in to an instance

EC2 Key Pairs

Amazon EC2 uses public key cryptography to encrypt and decrypt login information.

To log in to your EC2 instance, you must create a key pair, and provide the private key when connecting to the instance.

Linux instances have no password and you must use a key pair when logging in via SSH.

For Windows instances, you use the key pair to obtain the administrator password, then log in using RDP.

SSH uses port 22, and RDP uses port 3389. If you can't connect to your instance, check your Security Group and NACL configurations.

Security Groups :

- All inbound traffic is blocked by default.
- All outbound traffic is allowed
- Changes to security groups take effect immediately.
- You can have any number of EC2 instances within a security group. You can have multiple security groups attached to EC2 instances.
- Security Groups are stateful. Means that when you open up a port, e.g. 80, it would be opened for both inbound and outbound traffic.
- You cannot block specific IP addresses using security groups, instead use Network ACLs.
- You can specify allow rules, but not deny rules.
- All instances in a security group can communicate with all other instances in that same security group by default.
- To change the security group of an instance, right click on the instances, select 'networking', and select 'change security group'
- It's possible to have multiple security groups associated with an instance.

Monitoring

There are two types of EBS monitoring:

1. **Basic Monitoring** - available at no charge, and is enabled by default, with samples taken every 5 minutes
2. **Detailed Monitoring** - can be enabled for a fee, reducing the sample time to 1 minute.

System status checks make sure that packets can reach the instance (checking hypervisor is up)

Instance status checks make sure that the operating system can accept traffic

AWS CloudWatch :

It is a monitoring service to **monitor the AWS resources** and applications that run on AWS. **Monitors performance.**

It can monitor things like :

- Compute :
 - EC2 instance
 - Autoscaling groups
 - Elastic Load balancers
 - Route 53 health checks
- Storage and content delivery :
 - EBS volumes
 - Storage Gateways
 - Cloud Front

Host Level / Default metrics consist of :

- CPU
- Network
- Disk
- Status check

Everything else is a custom metric. i.e. Memory is a custom CloudWatch metric.

What can you do with cloudwatch :

1. Dashboards : ***create dashboards*** to see what is happening with your AWS environment.
2. Alarms : Allows you to ***set Alarms*** that notify you when particular thresholds are reached.
3. Events : Help you to ***respond to state changes*** i your AWS resources.
4. Logs : ***Logs*** help you aggregate, monitor and store logs.

Cloudwatch supports the following alarm states:

- OK - the metric is within the threshold
- ALARM - The metric is outside the threshold
- INSUFFICIENT_DATA - The alarm has just started, but the metric is not available, or not enough data is available for the metric to determine the alarm state

AWS CloudTrail :

It increases visibility into your user and resource activity by recording AWS Management console actions and API calls.

You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred.

-- Cloudwatch monitors performance, whereas Cloudtrail monitors API calls in the AWS platform.

Remember :

Note that CloudWatch and CloudTrail are distinct products:

- CloudWatch - for performance monitoring and logging
- CloudTrail - for auditing i.e. when a new AWS role, user, etc is created. [Stores all of it's data in S3.](#) - when enabling CloudTrail, you need to provide a S3 bucket where all logs can be written to.

-- Cloudwatch is used for monitoring performance

-- Cloudwatch can monitor most of AWS and the applications that run on AWS.

-- Cloudwatch with EC2 will monitor events every 5 minutes by default.

-- You can have 1 minute intervals by turning on detailed monitoring.

-- You can also create cloudwatch alarms which trigger notifications.

-- cloudwatch is all about performance and cloudtrail is all about Auditing.

CLI :

you can access AWS from anywhere in the world just by using the CLI .
You need to setup access in IAM.

Roles :

- Roles are more secure than storing your access key and secret access key on an individual EC2 instances.
 - Roles are easier to manage.
 - Roles can be assigned to an EC2 instance after it is created using both the console and command line.
 - Roles are universal, you can use them in any region.
 - It's possible to set role in instance creation and add/remove roles while the instance is running.
 - Using IAM roles means that you don't need to store credentials (such as AWS Secret Key and Access Key) in the EC2 instance itself.
i.e. if you want to give your EC2 instance full access to S3, you can use the AmazonS3FullAccess IAM role. You can then run s3 commands such as 'aws s3 ls' within the EC2 instance.
 - IAM roles are created globally
-

Bootstrap scripts :

They run when a instance first boots.
Can be a powerful way of automating software installs and updates.

Instance metadata and user data :

Used to get information about an instance such as public IP.

```
curl http://169.254.169.254/latest/meta-data/
```

```
curl http://169.254.169.254/latest/user-data/ // this is the bootstrap script loaded when creating instance.
```

Elastic File System :

- Supports the Network file system version 4 NFSV4 protocol.
 - You can pay only for the storage you use.
 - Can scale up to petabytes.
 - Can support thousands of concurrent NFS connections.
 - An EBS cannot be shared between multiple EC2 instances, but an EFS can be shared between multiple EC2 instances.
 - Data is stored across multiple AZs within a region.
 - Read after write consistency.
-

Placement Groups :

The way of placing your EC2 instances :

-- Clustered Placement Groups :

: All instances in the same AZ and very close to each other

: Used when requirements are low network latency / high network throughput

-- Spread Placement Group :

: Individual critical EC2 instances.

: Independent instances which are not dependent on each other, and also in different AZs, so that if one instance fails, then the other ones are not impacted.

-- Partitioned placement group

: multiple EC2 instances, like for HDFS, HBase and Cassandra.

: Multiple instances in a partition and each partition is in a separate rack / separate hardware from the other ones.

Clustered Placement group cannot span multiple AZs whereas the other ones can.

The name you specify for your placement group must be unique within your AWS account.

Only certain types of instances can be launched in a placement group : Compute optimized, memory optimized, GPU, storage optimized.

AWS recommend homogenous instances within clustered placement groups.



You cannot merge placement groups.

You cannot move an existing instance into a placement group. you can create an AMI from your existing instance, and then launch a new instance from the AMI into a placement group.

ELB

ELB supports Perfect Forward Secrecy.

Types of ELB:

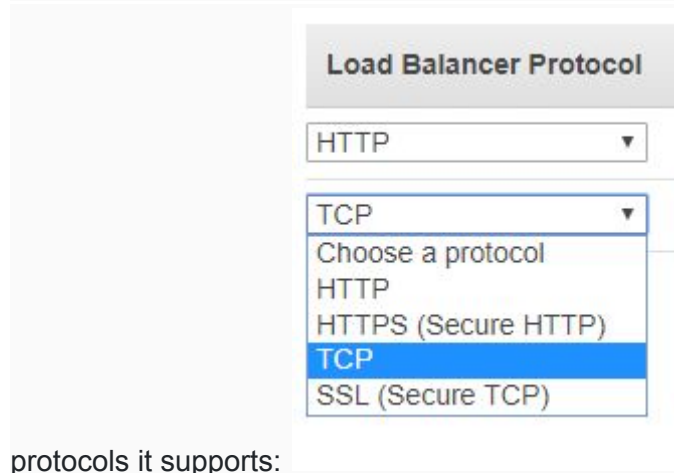
Application Load Balancer	Network Load Balancer	Classic Load Balancer
 Create	 Create	PREVIOUS GENERATION for HTTP, HTTPS, and TCP Create
Choose an Application Load Balancer when you need a flexible feature set for your web applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing, TLS termination and visibility features targeted at application architectures, including microservices and containers.	Choose a Network Load Balancer when you need ultra-high performance and static IP addresses for your application. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second while maintaining ultra-low latencies.	Choose a Classic Load Balancer when you have an existing application running in the EC2-Classic network.

Classic Load Balancer

It is the **previous generation** Load balancer for HTTP, HTTPS, and TCP traffic.

- Can load balance *HTTP/HTTPS* applications
- Can use *layer 7-specific features* such as X-Forwarded and sticky sessions
- Can also use strict layer 4 load balancing for applications that rely purely on the TCP protocol.

- When using the classic load balancer, you have the option of selecting which



Load Balancer Protocol

HTTP

TCP

Choose a protocol

HTTP

HTTPS (Secure HTTP)

TCP

SSL (Secure TCP)

protocols it supports:

Application Load Balancer

It is the current generation load balancer for HTTP, and HTTPS traffic.

- Can be used instead of the Classic Load Balancer when **using exclusively HTTP/HTTPS** traffic. **Do not use if the application depends on the TCP protocol.**
- Operates at the request level
- Made available half way through 2016.

Network Load Balancer

It is the current generation load balancer when **using exclusively TCP traffic.**

- Can be used instead of the Classic Load Balancer when using TCP traffic.
- Operates at the connection level
- The Network Load Balancer is suitable when you need **ultra-high performance, and have static IP addresses** for your application.

Cross-zone load balancing

By default, the Classic Load Balancer distributes traffic across all EC2 instances regardless of AZ. If you want to balance evenly across AZ, make sure you enable Cross-Zone load balancing:

Availability Zone Distribution

- ☒ Enable Cross-Zone Load Balancing ⓘ
- ☒ Enable Connection Draining ⓘ 300 seconds

Cross-Zone Load Balancing distributes traffic evenly across all your back-end instances in all Availability Zones.

Make sure you remember to shut down your ELBs if you're not using them

The major reason that people exceed the free tier is because they forgot to terminate their ELBs. It's a good idea to tag your ELBs so that they can be tracked via resource groups.

ELBs initially have a DNS name, but no public IP address.

When an ELB is monitoring instances, the instance can have one of two status:

- In service
- Out of service

Set Evaluate Target Health to true, and enable Latency Based Routing for HA (High Availability)

Auto scaling

Scaling Policy is a set of rules i.e. Increase if average CPU > 80% for a consecutive period of 5 minutes.

Desired Instances is adjusted based on the scaling policy, and won't go below the minimum or above the maximum size of the group.

Note that there is a default maximum of 20 running on-demand EC2 instances regardless of the min/max you set in your ASG scaling policies. You can request a limit increase by getting in touch with AWS... if your auto scaling triggers are firing, but you are not getting any more instances, check that you haven't reached the default maximum.

Deleting an ASG will automatically delete any instances that it created.

Launch configurations cannot be modified after creation. If you need to make a change, create a new launch configuration and update your auto scaling group to use it.

Launch configurations can belong to multiple Auto Scaling groups, however you can only specify one launch configuration at a time for an Auto Scaling group.

AMIs can be used with Auto Scaling groups.

[More info on launch configurations](#)

The following scale out options are available:

- Scheduled scaling - adjusting the size of a group at a specific time
- Dynamic scaling - via creating a scaling policy to automatically adjust the size of the group based on a specified increase in demand
- Manual scaling - via manually increasing the size of the group

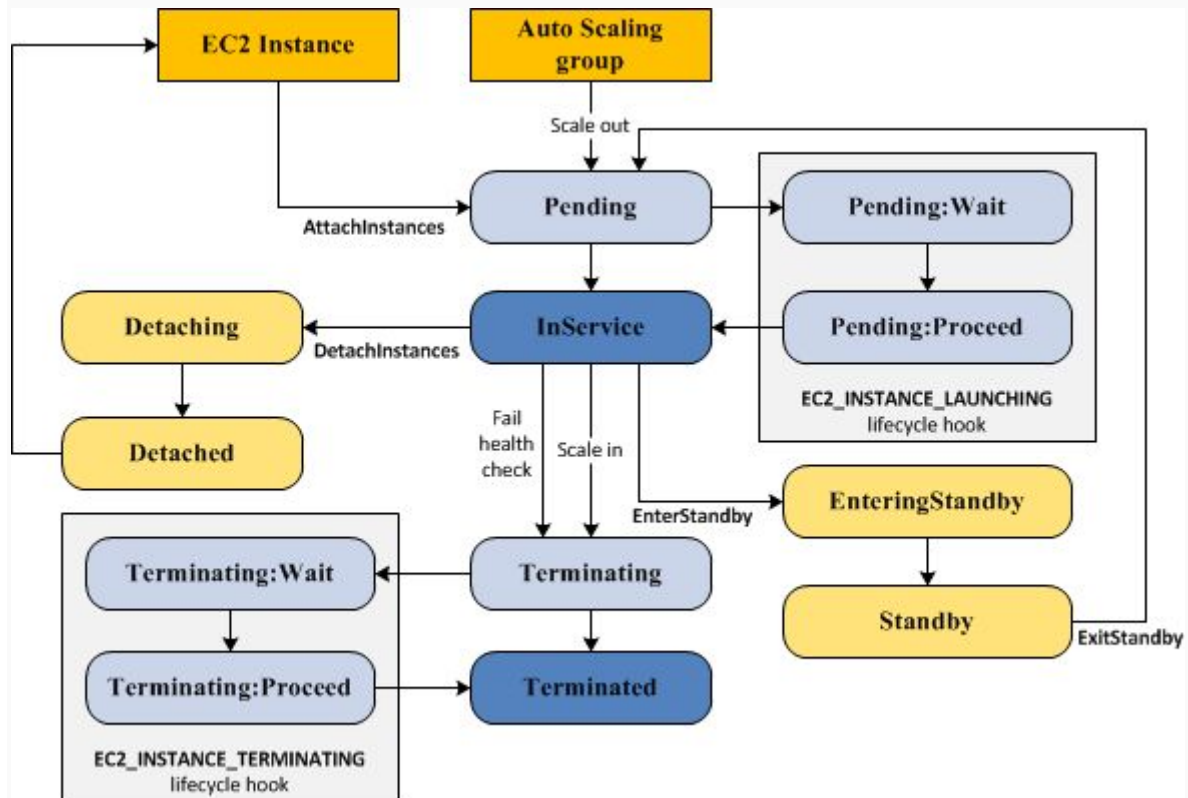
[More info on auto scaling lifecycles](#)

To attach EC2 instances to an Auto Scaling group, ensure that:

- The instance is in the running state
- The AMIs used to launch the instance still exist
- The instance is not a member of another Auto Scaling group
- The instance is in the same AZ as the Auto Scaling group

[More info on attaching instances to an Auto Scaling group](#)

Auto Scaling Lifecycle and Lifecycle Hooks



In the Pending:Wait state, no lifecycle policies take effect.

The cooldown period is the number of seconds after a scaling activity completes before another can start.

Auto Scaling Group Termination Policy

