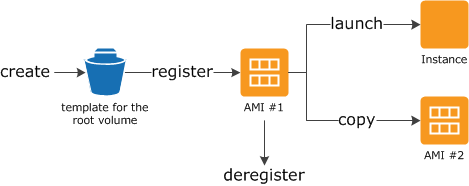
An Amazon Machine Image (AMI) provides the information required to launch an instance, which is a virtual server in the cloud. You must specify a source AMI when you launch an instance. You can launch multiple instances from a single AMI when you need multiple instances with the same configuration. You can use different AMIs to launch instances when you need instances with different configurations.

An AMI includes the following:

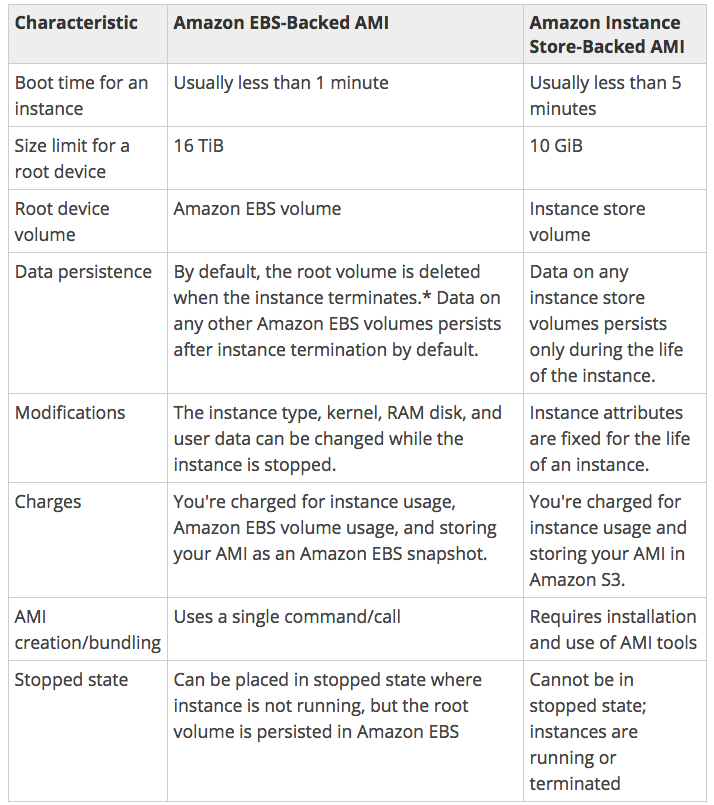
* A template for the root volume for the instance (for example, an operating system, an application server, and applications)
* Launch permissions that control which AWS accounts can use the AMI to launch instances
* A block device mapping that specifies the volumes to attach to the instance when it's launched



Amazon Linux 2 and the Amazon Linux AMI are supported and maintained Linux images provided by AWS. The following are some of the features of Amazon Linux 2 and Amazon Linux AMI:

* A stable, secure, and high-performance execution environment for applications running on Amazon EC2.
* Provided at no additional charge to Amazon EC2 users.

All AMIs are categorized as either *backed by Amazon EBS* or *backed by instance store*. The former means that the root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot. The latter means that the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3.



An instance store provides temporary block-level storage for your instance. This storage is located on disks that are physically attached to the host computer. Instance store is ideal for temporary storage of information that changes frequently, such as buffers, caches, scratch data, and other temporary content, or for data that is replicated across a fleet of instances, such as a load-balanced pool of web servers.

An instance store consists of one or more instance store volumes exposed as block devices. The size of an instance store as well as the number of devices available varies by instance type. While an instance store is dedicated to a particular instance, the disk subsystem is shared among instances on a host computer.

Instance metadata is data about your instance that you can use to configure or manage the running instance. Instance metadata is divided into categories. For more information, see [Instance Metadata Categories](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html#instancedata-data-categories).

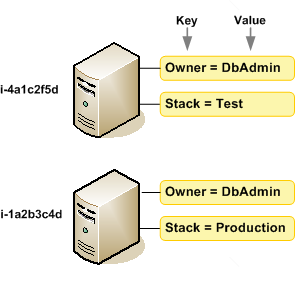
http://169.254.169.254/latest/meta-data/

**Important**

Although you can only access instance metadata and user data from within the instance itself, the data is not protected by cryptographic methods. Anyone who can access the instance can view its metadata.

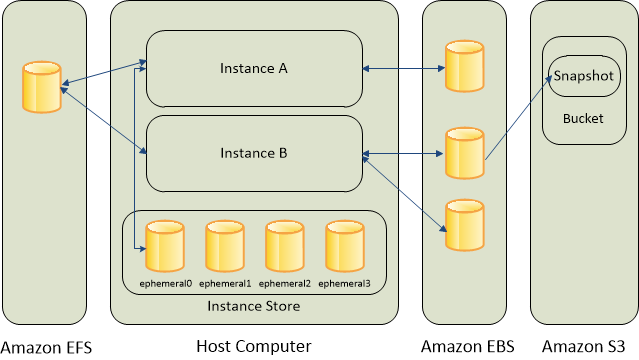
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value, both of which you define.

Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment.



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.

Amazon EC2 instances support Intel Hyper-Threading Technology, which enables multiple threads to run concurrently on a single Intel Xeon CPU core. Each thread is represented as a virtual CPU (vCPU) on the instance. An instance has a default number of CPU cores, which varies according to instance type. For example, an m5.xlarge instance type has two CPU cores and two threads per core by default—four vCPUs in total.



Amazon EBS provides durable, block-level storage volumes that you can attach to a running instance. You can use Amazon EBS as a primary storage device for data that requires frequent and granular updates. For example, Amazon EBS is the recommended storage option when you run a database on an instance.

To keep a backup copy of your data, you can create a *snapshot* of an EBS volume, which is stored in Amazon S3. You can create an EBS volume from a snapshot, and attach it to another instance.

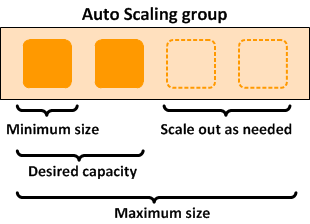
An EBS volume behaves like a raw, unformatted, external block device that you can attach to a single instance. The volume persists independently from the running life of an instance. After an EBS volume is attached to an instance, you can use it like any other physical hard drive. As illustrated in the previous figure, multiple volumes can be attached to an instance. You can also detach an EBS volume from one instance and attach it to another instance. You can dynamically change the configuration of a volume attached to an instance. EBS volumes can also be created as encrypted volumes using the Amazon EBS encryption feature.

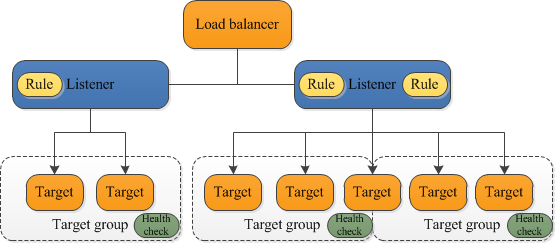
Many instances can access storage from disks that are physically attached to the host computer. This disk storage is referred to as *instance store*. Instance store provides temporary block-level storage for instances. The data on an instance store volume persists only during the life of the associated instance; if you stop or terminate an instance, any data on instance store volumes is lost.

Amazon EFS provides scalable file storage for use with Amazon EC2. You can create an EFS file system and configure your instances to mount the file system. You can use an EFS file system as a common data source for workloads and applications running on multiple instances.

Amazon EC2 Auto Scaling is designed to automatically launch or terminate EC2 instances based on user-defined policies, schedules, and health checks.

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.





An Application Load Balancer functions at the application layer, the seventh layer of the Open Systems Interconnection (OSI) model. After the load balancer receives a request, it evaluates the listener rules in priority order to determine which rule to apply, and then selects a target from the target group for the rule action.

A Network Load Balancer functions at the fourth layer of the Open Systems Interconnection (OSI) model. It can handle millions of requests per second. After the load balancer receives a connection request, it selects a target from the target group for the default rule. It attempts to open a TCP connection to the selected target on the port specified in the listener configuration.

When you enable an Availability Zone for the load balancer, Elastic Load Balancing creates a load balancer node in the Availability Zone. By default, each load balancer node distributes traffic across the registered targets in its Availability Zone only. If you enable cross-zone load balancing, each load balancer node distributes traffic across the registered targets in all enabled Availability Zones.

AWS Lambda is a compute service that lets you run code without provisioning or managing servers. AWS Lambda executes your code only when needed and scales automatically, from a few requests per day to thousands per second. You pay only for the compute time you consume - there is no charge when your code is not running.

**EFS is:**

* [Generally Available](https://aws.amazon.com/about-aws/whats-new/2016/06/amazon-elastic-file-system-efs-is-now-generally-available/)(out of preview), but may not yet be available in your region
* Network filesystem (that means it may have bigger latency but it can be shared across several instances; even between regions)
* It is expensive compared to EBS (~10x more) but it gives extra features.
* It's a highly available service.
* It's a managed service
* You can attach the EFS storage to an EC2 Instance
* Can be accessed by multiple EC2 instances simultaneously
* Since 2016.dec.20 it's possible to attach your EFS storage directly to [on-premise servers via Direct Connect.](https://aws.amazon.com/blogs/aws/amazon-efs-update-on-premises-access-via-direct-connect-vpc/) ()

**EBS is:**

* A block storage (so you need to format it). This means you are able to choose which type of file system you want.
* As it's a block storage, you can use Raid 1 (or 0 or 10) with multiple block storages
* It is really fast
* It is relatively cheap
* With the new announcements from Amazon, you can store up to 16TB data per storage on SSD-s.
* You can snapshot an EBS (while it's still running) for backup reasons
* But it only exists in a particular region. Although you can migrate it to another region, you cannot just access it across regions (only if you share it via the EC2; but that means you have a file server)
* You need an EC2 instance to attach it to
* [New feature](https://aws.amazon.com/blogs/aws/amazon-ebs-update-new-elastic-volumes-change-everything/) (2017.Feb.15): You can now increase volume size, adjust performance, or change the volume type while the volume is in use. You can continue to use your application while the change takes effect.

You can launch or start instances in a placement group, which determines how instances are placed on underlying hardware. When you create a placement group, you specify one of the following strategies for the group:

* Cluster—clusters instances into a low-latency group in a single Availability Zone
* Spread—spreads instances across underlying hardware

There is no charge for creating a placement group.

You can control whether an EBS root volume is deleted when its associated instance is terminated. The default delete-on-termination behaviour depends on whether the volume is a root volume, or an additional volume. By default, the DeleteOnTermination attribute for root volumes is set to 'true.' However, this attribute may be changed at launch by using either the AWS Console or the command line. For an instance that is already running, the DeleteOnTermination attribute must be changed using the CLI.