# Amazon Elastic Compute Cloud (Amazon EC2) and

# Amazon Elastic Block Store (Amazon EBS)

# Amazon EC2

## Compute Basics

* ***Compute***refers to the amount of computational power required to full fill your workload.
* Amazon EC2 allows you to acquire compute through the launching of virtual servers called ***instances***.
* When you launch an instance, you can make use of the compute as you wish, just as you would with an on-premises server.
* Because you are paying for the computing power of the instance, you are charged per hour while the instance is running.
* two concepts that are key to launching instances on AWS:
  + the amount of virtual hardware dedicated to the instance.
  + the software loaded on the instance.
* These two dimensions of new instances are controlled, respectively, **by the instance type and the AMI**.

### Instance Types

* The instance type defines the virtual hardware supporting an Amazon EC2 instance.
* There are dozens of instance types available, varying in the following dimensions:
  + Virtual CPUs (vCPUs)
  + Memory
  + Storage (size and type)
  + Network performance
* Instance types are grouped into families based on the ratio of these values to each other.
* Another variable to consider when choosing an instance type is network performance.
* For most instance types, AWS publishes a relative measure of network performance: low, moderate, or high.
* For workloads requiring greater network performance, many instance types support ***enhanced networking***
* Enhanced networking reduces the impact of virtualization on network performance by enabling a capability called Single Root I/O Virtualization (SR-IOV).
* This results in more Packets Per Second (PPS), lower latency, and less jitter.
* Enhanced networking is available only for instances launched in an Amazon Virtual Private Cloud (Amazon VPC).

### Amazon Machine Images (AMIs)

* **The *Amazon Machine Image (AMI)***defines the initial software that will be on an instance when it is launched.
* An AMI defines every aspect of the software state at instance launch, including:
  + The Operating System (OS) and its configuration
  + The initial state of any patches
  + Application or system software
* All AMIs are based on x86 OSs, either Linux or Windows.
* There are four sources of AMIs:
  + ***Published by AWS***—AWS publishes AMIs with versions of many different OSs, both Linux and Windows.
  + Launching an instance based on one of these AMIs will result in the default OS settings
  + As with any OS installation, you should immediately apply all appropriate patches upon launch.
  + ***The AWS Marketplace***—AWS Marketplace is an online store that helps customers find, buy, and immediately start using the software and services that run on Amazon EC2.
  + This provides two benefits: the **customer does not need to install the software**, and the **license agreement is appropriate for the cloud**
  + Instances launched from an AWS Marketplace AMI incur the standard hourly cost of the instance type plus an additional per-hour charge for the additional software.
  + ***Generated from Existing Instances***—An AMI can be created from an existing Amazon EC2 instance.
  + Customers launch an instance from a published AMI, and then the instance is configured to meet all the customer’s corporate standards for updates, management, security, and so on.
  + An AMI is then generated from the configured instance and used to generate all instances of that OS.
  + In this way, all new instances follow the corporate standard and it is more difficult for individual projects to launch non-conforming instances.
  + ***Uploaded Virtual Servers***—Using AWS VM Import/Export service, customers can create images from various virtualization formats, including raw, VHD, VMDK, and OVA.
  + It is incumbent on the customers to remain compliant with the licensing terms of their OS vendor.

## Securely Using an Instance

### Addressing an Instance

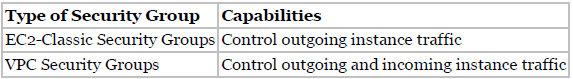
* ***Public Domain Name System (DNS) Name***— When you launch an instance, AWS creates a DNS name that can be used to access the instance.
* This DNS name is generated automatically and cannot be specified by the customer.
* This DNS name persists only while the instance is running and cannot be transferred to another instance.
* ***Public IP***— A launched instance may also have a public IP address assigned.
* This IP address is assigned from the addresses reserved by AWS and cannot be specified.
* This IP address is **unique on the Internet**, persists only while the instance is running, and **cannot be transferred to another instance.**
* ***Elastic IP***—An elastic IP address is an address unique on the Internet that you reserve independently and associate with an Amazon EC2 instance.
* This IP address **persists until the customer releases it** and is not tied to the lifetime or state of an individual instance.
* Because it **can be transferred to a replacement instance in the event of an instance failure**, it is a public address that can be shared externally without coupling clients to a particular instance.
* **Private IP addresses** **and Elastic Network Interfaces (ENIs)** are additional methods of addressing instances that are available in the context of an Amazon VPC.

### Initial Access

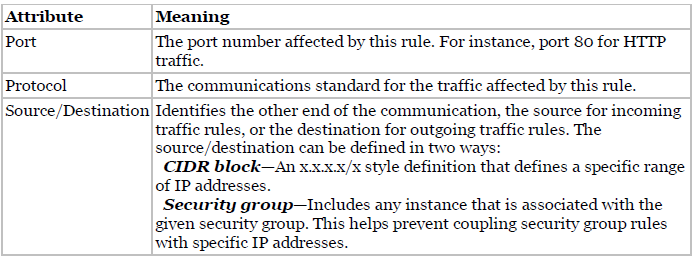
* Amazon EC2 uses public-key cryptography to encrypt and decrypt login information.
* Publickey cryptography uses a public key to encrypt a piece of data and an associated private key to decrypt the data.
* These two keys together are called a ***key pair***.
* AWS stores the public key, and the private key is kept by the customer.
* The private key is essential to acquiring secure access to an instance for the first time.

### Virtual Firewall Protection

* AWS allows you to control traffic in and out of your instances through virtual firewalls called ***security groups***.
* Security groups allow you to control traffic based on port, protocol, and source/destination.
* Security groups have different capabilities depending on whether they are associated with an Amazon VPC or Amazon EC2-Classic.



* Security groups are associated with instances when they are launched.
* **Every instance must have at least one security group** but can have more.
* A security group is **default deny**; that is, it does not allow any traffic that is not explicitly allowed by a security group rule.
* When an instance is associated with multiple security groups, the rules are aggregated and all traffic allowed by each of the individual groups is allowed.



* A security group is a **stateful firewall**; that is, an outgoing message is remembered so that the response is allowed through the security group without an explicit inbound rule being required.
* Security groups are **applied at the instance level**, as opposed to a traditional on-premises firewall that protects at the perimeter. The effect of this is that instead of having to breach a single perimeter to access all the instances in your security group, an attacker would have to breach the security group repeatedly for each individual instance.

## The Lifecycle of Instances

**Launching**

* **Bootstrapping** A great benefit of the cloud is the ability to script virtual hardware management in a manner that is not possible with on-premises hardware.
* There has to be some way to configure instances and install applications programmatically when an instance is launched. The process of providing code to be run on an instance at launch is called ***bootstrapping***.
* On Linux instances this can be shell script, and on Windows instances this can be a batch style script or a PowerShell script.
* The script can perform tasks such as:
  + Applying patches and updates to the OS
  + Enrolling in a directory service
  + Installing application software
  + Copying a longer script or program from storage to be run on the instance
  + Installing Chef or Puppet and assigning the instance a role so the configuration management software can configure the instance
* UserData is stored with the instance and is not encrypted, so it is important to not include any secrets such as passwords or keys in the UserData.
* **VM Import/Export** In addition to importing virtual instances as AMIs, VM Import/Export enables you to easily import Virtual Machines (VMs) from your existing environment as an Amazon EC2 instance and export them back to your on-premises environment.
* You can only export previously imported Amazon EC2 instances. Instances launched within AWS from AMIs cannot be exported.
* **Instance Metadata** Instance metadata is data about your instance that you can use to configure or manage the running instance.
* This is unique in that it is a mechanism to obtain AWS properties of the instance from within the OS without making a call to the AWS API.
* An HTTP call to *http://169.254.169.254/latest/meta-data/* will return the top node of the instance metadata tree.

**Managing Instances**

* **Tags** can help you manage not just your Amazon EC2 instances, but also many of your AWS Cloud services.
* Tags are key/value pairs you can associate with your instance or other service. Tags can be used to identify attributes of an instance like project, environment (dev, test, and so on), billable department, and so forth.
* You can apply up to **10 tags per instance**.

**Monitoring Instances**

* AWS offers a service called Amazon **CloudWatch** that provides monitoring and alerting.

**Modifying an Instance**

* **Instance Type** Instances can be resized using the AWS Management Console, CLI, or API.
* Choose the “Change Instance Type” function in the tool of your choice and select the desired instance type.
* Restart the instance and the process is complete.
* **Security Groups** If an instance is running in an Amazon VPC you can change which security groups are associated with an instance while the instance is running.
* For instances outside of an Amazon VPC (called EC2-Classic), the association of the security groups cannot be changed after launch.

**Termination Protection**

* While enabled, calls to terminate the instance will fail until termination protection is disabled.
* It does not prevent termination triggered by an OS shutdown command, termination from an Auto Scaling group, or termination of a Spot Instance due to Spot price changes
* **Shared tenancy** is the default tenancy model for all Amazon EC2 instances, regardless of instance type, pricing model, and so forth. shared tenancy means that a single host machine may house instances from different customers.
* **Dedicated Instances** run on hardware that’s dedicated to a single customer. As a customer runs more Dedicated Instances, more underlying hardware may be dedicated to their account.
* An Amazon EC2 **Dedicated Host** is a physical server with Amazon EC2 instance capacity fully dedicated to a single customer’s use. Dedicated Hosts can help you address licensing requirements and reduce costs by allowing you to use your existing serverbound software licenses.
* A **placement group** is a logical grouping of instances within a single Availability Zone.
* Placement groups enable applications to participate in a low-latency, 10 Gbps network.
* Placement groups are recommended for applications that benefit from low network latency, high network throughput, or both.

**Instance Stores**

* An **instance store** (sometimes referred to as **ephemeral storage**) provides temporary block level storage for your instance.
* This storage is **located on disks that are physically attached to the host computer**.
* An 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.
* The instance type also determines the type of hardware for the instance store volumes. While some provide Hard Disk Drive (HDD) instance stores, other instance types use Solid State Drives (SSDs) to deliver very high random I/O performance.

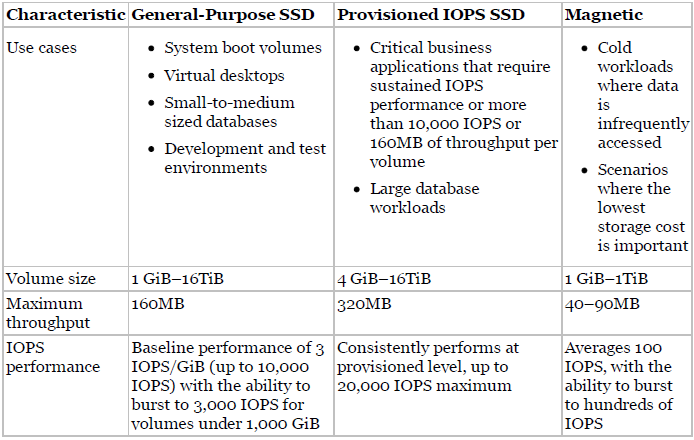
# Amazon Elastic Block Store (Amazon EBS)

## Elastic Block Store Basics

* Amazon EBS provides **persistent block-level storage volumes** for use with Amazon EC2 instances.
* Each Amazon EBS volume **is automatically replicated within its Availability Zone** to protect you from component failure, offering high availability and durability.
* Multiple Amazon EBS volumes can be attached to a single Amazon EC2 instance, although a volume can only be attached to a single instance at a time.

## Types of Amazon EBS Volumes

* **Magnetic volumes** have the lowest performance characteristics of all Amazon EBS volume types.
* A magnetic Amazon EBS volume can **range in size from 1 GB to 1 TB** and will average 100 IOPS, but has the ability to burst to hundreds of IOPS.
* They are best suited for:
  + Workloads where data is accessed infrequently
  + Sequential reads
  + Situations where low-cost storage is a requirement
* Magnetic volumes are billed based on the amount of data space provisioned, regardless of how much data you actually store on the volume.
* A **general-purpose SSD** volume **can range in size from 1 GB to 16 TB** and provides a baseline performance of three IOPS per gigabyte provisioned, capping at 10,000 IOPS.
* General-purpose SSD volumes under 1 TB also feature the ability to burst to up to 3,000 IOPS for extended periods of time.
* For instance, if you have a 500 GB volume you can expect a baseline of 1,500 IOPS. Whenever you are not using these IOPS, they are accumulated as I/O credits. When your volume then has heavy traffic, it will use the I/O credits at a rate of up to 3,000 IOPS until they are depleted. At that point, your performance reverts to 1,500 IOPS. At 1 TB, the baseline performance of the volume is already at 3,000 IOPS, so bursting behaviour does not apply.
* They are suited for a wide range of workloads where the very highest disk performance is not critical, such as:
  + System boot volumes
  + Small- to medium-sized databases
  + Development and test environments
* **Provisioned IOPS SSD volumes** are designed to meet the needs of I/O-intensive workloads, particularly database workloads that are sensitive to storage performance and consistency in random access I/O throughput.
* A Provisioned IOPS SSD volume can **range in size from 4 GB to 16 TB.**
* When you provision a Provisioned IOPS SSD volume, you specify not just the size, but also the desired number of IOPS, up to the lower of the maximum of 30 times the number of GB of the volume, or 20,000 IOPS.
* Pricing is based on the size of the volume and the amount of IOPS reserved.
* An **additional monthly fee** is applied based on the number of IOPS provisioned, whether they are consumed or not.
* Are well suited for:
  + Critical business applications that require sustained IOPS performance
  + Large database workloads



* **Throughput-Optimized HDD volumes** are low-cost HDD **volumes designed for frequent access, throughput-intensive workloads** such as big data, data warehouses, and log processing. Volumes can be up to 16 TB with a maximum IOPS of 500 and maximum throughput of 500 MB/s. These volumes are significantly less expensive than general purpose SSD volumes.
* **Cold HDD volumes** are **designed for less frequently accessed workloads**, such as colder data requiring fewer scans per day. Volumes can be up to 16 TB with a maximum IOPS of 250 and maximum throughput of 250 MB/s. These volumes are significantly less expensive than Throughput-Optimized HDD volumes.

## Protecting Data

### Backup/Recovery (Snapshots)

* Taking Snapshots You can take snapshots in many ways:
  + Through the AWS Management Console
  + Through the CLI
  + Through the API
  + By setting up a schedule of regular snapshots
* When you request a snapshot, the point-in-time snapshot is created immediately and the volume may continue to be used, but the snapshot may remain in pending status until all the modified blocks have been transferred to Amazon S3.
* It’s important to know that while snapshots are stored using Amazon S3 technology, they are **stored in AWS-controlled storage** and **not in your account’s Amazon S3 buckets**. This means **you cannot manipulate them like other Amazon S3 objects**.
* Rather, you **must use the Amazon EBS snapshot features** to manage them.
* Snapshots are **constrained to the region in which they are created**, meaning you can use them to create new volumes only in the same region**. If you need to restore a snapshot in a different region, you can copy a snapshot to another region.**
* **Creating a Volume from a Snapshot** To use a snapshot, you create a new Amazon EBS volume from the snapshot. When you do this, **the volume is created immediately** **but the data is loaded lazily.** This means that the **volume can be accessed upon creation**, **and if the data being requested has not yet been restored, it will be restored upon first request.** Because of this, **it is a best practice to initialize a volume created from a snapshot by accessing all the blocks in the volume.**
* Because **Amazon EBS volumes persist beyond the lifetime of an instance**, it is possible to recover data if an instance fails.
* Unless the DeleteOnTermination flag for the volume has been set to false**, the volume should be detached before the instance is terminated.**