**EC2 Basics**

Amazon **Elastic Compute Cloud** (Amazon EC2):

- Provides **scalable computing capacity** in the Amazon Web Services (AWS) cloud.

- Eliminates your need to invest in hardware up front, so you can **develop and deploy applications faster**.

- Enables you to **scale up or down** to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

Basic Computer   EC2

Components     v Components

---------------+-----------

O/S            : AMIs

CPU            : Instance Type

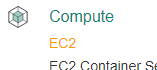
Hard Drive     : EBS

Network Card   : IP Addressing

Firewall       : Security Groups

RAM            : RAM

*Image: Services > Compute > EC2*

[](https://1.bp.blogspot.com/-Sj6ROSgAbUE/WYNCAGfeJpI/AAAAAAAAHSA/W7Ein1fI83Qqjcr1K_0fQWSHYG1FPOzjACLcBGAs/s1600/01+Compute+EC2.png)

Link: [AWS: Instance Purchasing Options](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-purchasing-options.html)

Link: [Amazon EC2 Pricing](https://aws.amazon.com/ec2/pricing/)

*Note:****Free Tier****use is available for EC2.*

***EC2 Instance Purchasing Options****(most common)****:***

- On-Demand

- Reserved

- Spot

**On-Demand**(most expense):

* + - Choose any instance type .. Pay fixed rate by the hour with no commitment

- Only charged when the instance is running (billed per hour).

* For users that want flexibility of EC2 w/out up-front payments or long-term commitment
* Best for burst need servers & unpredictable workloads that cannot be interrupted
* Test/Dev for apps running on EC2 for the 1st time.
* 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.
  + Supplement reserved instance servers (for extra temporary server load)

**Reserved** (significant price discount):

* + Provide capacity reservation and offer significant discount on the hourly charge for an instance (1-3 year terms)

- Pay upfront, partial upfront, no upfront

- Charged regardless of how often you use it

* + Applications have steady state, or predictable usage
  + Apps that require reserved capacity
  + Users able to make upfront payments to reduce their total computing costs even further.
  + the longer your contract, the more you save.

Apps that need reserved capacity, steady state or predictable usage

* Domain Controllers
* 1st web server

**Spot** (substantial price discount):

- “Bid” on an instance type, and only pay for and use when the spot price is equal or below your “bid” price.

* + Bid whatever price you want for instance capacity by the hour

- Allows Amazon to sell the use of **unused instances** for short amounts of time

- Prices fluctuate based on supply and demand (billed per hour)

- Instances automatically terminate when spot price > “bid” price..

* + When your bid price is greater than or equal to the spot price, your instance will boot
* When your bid >= spot price, you get a server
  + When the spot price is greater than your bid price, you lose server with 1 hour warning
  + Applications have flexible start and end times
  + Apps that are only feasible at very low compute prices
  + Users with urgent computing needs for large amounts of additional capacity
* 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.

**How are you charged for using EC2?**

**1) Purchasing Option**

**2) Instance Type** (processing power):

Vary by

* CPU
* Memory
* Storage (size and type)
* Network Performance (low,moderate,high).
* There are 5 families of instances
  + General Purpose - M3 and M4 with fixed performance, and T2 for burstable.
  + Compute Optimized - C3 and C4
  + Memory Optimized - R3
  + GPU Optimized - P2 are Nvidia Tesla parallel processing GPU Compute, and G2 provide high performance 3D graphics (GPU Graphics)
  + Storage Optimized - I2 are SSD based for high IO, D2 are dense (also SSD based) for data intensive operations with sequential IO
* M3 are consistent and cheaper over M1 and also support SSD based instance storage. But if you need a larger disk, use M1.

|  |  |  |
| --- | --- | --- |
| M |  | General Purpose i.e. app server (default) |
| T |  | Micro instances - Low cost, general purpose, web servers |
| C, D, I |  | CPU/IOPS optimised, good for memory intensive compute |
| R, X |  | Memory optimised |
| G, P |  | GPU |
| F |  | FPGA - Field Programmable Gate Array, for hardware acceleration of code |

Useful mnemoic: Dr Mc Gift Px

- General purpose

- Compute optimized

- GPU optimized

- Memory optimized

- Storage optimized

- EBS optimized (option for higher IOPS performance)

**3) AMI Type** (varies on O/S):

- Linux

- Windows

**4) Data Transfer**(in/out of the instance)

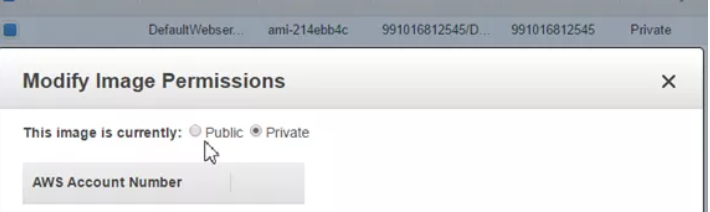
**5) Region**

**5.2) Amazon Machine Images (AMIs)**

**Amazon Machine Image**(AMI) provides the information (template) required to launch an instance (virtual server.)

* AMI = template VM
* AMI's are simply snapshots of a root volume and is stored in S3
* AMI's are regional. You can only launch an AMI from the region in which it was stored.
* You can copy AMI's to other regions using the console, CLI or Amazon EC2 API.

**Contains:**

* Template for root volume for the instance (OS, application servers, apps, etc)
* Launch permissions that control with AWS accounts can use the AMI to launch instances
* Block device mapping that specifies which volumes to attach when launching instance
* When you create an AMI, by default its marked private. You have to manually change the permissions to make the image public or share images with individual accounts
* Hardware Virtual Machines (HVM) AMI's Available
* Paravirtual (PV) AMI's Available
* You can select an AMI based on:
  + Region
  + OS
  + Architecture (32 vs. 64 bit)
  + Launch Permissions
  + Storage for the root device (Instance Store Vs. EBS).

### **There are different types of AMI images**

* HVM (Hardware Virutal 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

AMIs come in 3 main categories:

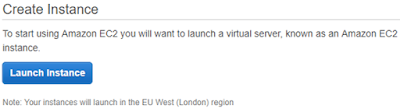
**1) Community AMIs:** Free to use, essentially just an OS

**2) AWS Marketplace AMIs:** Pay to use, generally package with additional licensed software

**3) My AMIs**

* + 1. published by AWs

*Image: Amazon EC2 > Launch Instance*

[](https://4.bp.blogspot.com/-kR7jgtNx_xU/WYNCHObkOkI/AAAAAAAAHSE/zJvW1-F1au894_afoT8SE43X8AQrMOIygCLcBGAs/s1600/02+Create+Instance.png)

*Image: AMI categories*

[](https://3.bp.blogspot.com/-JsGUraI0HWY/WYNCLDg9LcI/AAAAAAAAHSI/D4syXj4Y260ZpzqCx3uebpM9MIN5mCcxACLcBGAs/s1600/03+AMI+categories.png)

**5.3) Instance Types**

*When you launch and instance...*

The **instance type** determines the (virtual) hardware used for your instance.

*Instance Type Components:*

**(1) Family***(General Purpose, Compute Optimized, Memory Optimized, Storage Optimized)*

**(2) Type**

**(3) vCPUs**

**(4) Memory (GiB)**

**(5) Instance Storage (GB)**

**(6) EBS-Optimized Available**

**(7) Network Performance**

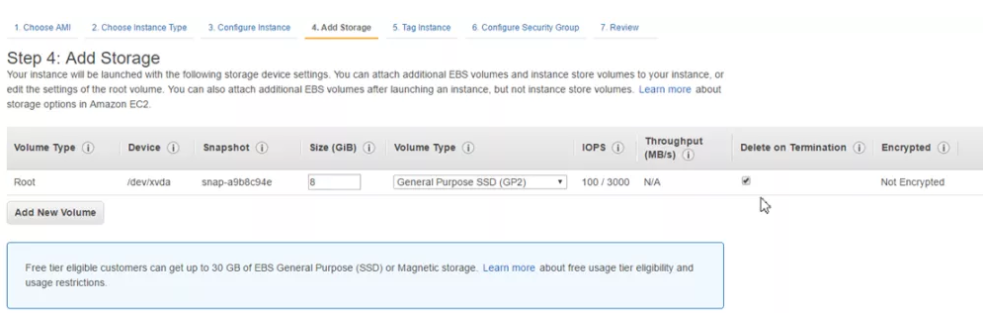
|  |  |  |
| --- | --- | --- |
| Characteristic | Amazon EBS-Backed | Amazon Instance Store-Backed |
| Boot time | Usually less than 1 minute | Usually less than 5 minutes |
| Size limit | 16 TiB | 10 GiB |
| Root device volume | Amazon EBS volume | Instance store volume |
| Data persistence | By default, the root volume is deleted when the instance terminates.\* Data on any other Amazon EBS volumes persists after instance termination by default. Data on any instance store volumes persists only during the life of the instance. | Data on any instance store volumes persists only during the life of the instance. Data on any Amazon EBS volumes persists after instance termination by default. |
| Upgrading | The instance type, kernel, RAM disk, and user data can be changed while the instance is stopped. | Instance attributes are fixed for the life of an instance. |
| Charges | You’re charged for instance usage, Amazon EBS volume usage, and storing your AMI as an Amazon EBS snapshot. | You’re charged for instance usage and storing your AMI in Amazon S3. |
| AMI creation/bundling | Uses a single command/call | Requires installation and use of AMI tools |
| Stopped state | Can be placed in stopped state where instance is not running, but the root volume is persisted in Amazon EBS | Cannot be in stopped state; instances are running or terminated |

**5.4) Elastic Block Store (EBS)**

* Billing ends when the instance terminates or stops. If it is shut down, the instance hours aren’t billed but you pay for the EBS storage that has the root device data on it
* If using instance store backed instance, for durable data, use S3. Use EBS backed if you need the data to be independent of the instance’s life.
* All instances are billed in hourly usage units. Even if the instance is up for a minute, you get billed for a full hour.
* Data transferred between AWS services in different regions is charged on both ends.
* Billing starts when the boot sequence of the instance is initiated.

\*EBS (Elastic Block Storage) – **Storage volumes that are attached to EC2 instances (think VMDKs).**

* When creating an AMI, on Step 4(Add storage) "Delete on Termination" is checked and not encrypted by default (i.e. Termination protection is turned off by default):
* On an EBS-backed instance, the default action is for the root EBS vol to be deleted when the instance is terminated.



* On an EBS-backed instance, the default action is for the root EBS vol to be deleted when the instance is terminated.
* Root volumes cannot be encrypted by default, you'll need a 3rd party tool (bit locker, etc) to encrypt root vols.
* When an EC2 instance is terminated, the root EBS volume is also deleted by default.
* [Termination Protection](https://aws.amazon.com/premiumsupport/knowledge-center/accidental-termination/) is off by default, and can be used to prevent accidental termination
* Use ‘lsblk’ to get a list of mounted disks.
* 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.
* When you create an EBS volume in an AZ, it it automatically replicated within that zone to prevent data loss due to the failure or 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.

If the root volume of an EC2 instance fails, and you need to recover data from it, you can:

* Deta 0 ch 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

Storage Types:

* Instance Store (Ephemeral):
  + Instances using instance store storage can not be stopped. If they are, data loss would result

(When using an instance store volume, you can not stop the instance (the option to do so will not be available, as the instance moves to another host and and would cause complete data loss)

* + If there is an issue with the underlying host and your instance needs to be moved, or is lost, Data is also lost.
* You can reboot without losing data, if you stop the instance, the data will be wiped.
* Placed in specific AZs & automatically replicated
  + Instance store volumes cannot be detached and reattached to other instances; They exist only for the life of that instance

(Once an Instance has been launched with instance store storage, you can not attach additional instance store volumes after the instance is launched, only EBS volumes.)

* + Best used for scratch storage, storage that can be lost at any time with no bad ramifications, such as a **cache store**
* **The root device for an instance launched from the AMI is an instance store volume created from a template stored in S3**
* EBS (Elastic Block Storage):
* **The root device for an instance launched from the AMI is an EBS volume created from an EBS snapshot**
* Can be stopped, you will not lose data if the underlying host fails.
* Can be detached and reattached to other EC2 instances
  + Elastic Block Storage is persistent storage that can be used to procure storage to EC2 instances.
  + You can NOT mount 1 EBS volume to multiple EC2 instances instead you **must use EFS**
  + Default action for EBS volumes is for the root EBS volume to be deleted when the instance is terminated
* **Termination protection is turned off by default, you must turn it on**
  + By default, ROOT volumes will be deleted on termination, however with EBS volumes only, you can tell AWS to keep the root device volume
  + EBS backed instances can be stopped, you will NOT lose any data.
  + Encryption:
    - Root Volumes cannot be encrypted by default, you need a 3rd party utility
* Other volumes added to an instance can be encrypted and attached to an EC2 instance.
* You can poll an instances meta-data by using curl http://169.254.169.254/latest/meta-data/
* You can get an instance's IP address by using curl http://169.254.169.254/latest/meta-data/public-ipv4
* No such thing as user-data, remember its always meta-data not user-data
* When configuring a launch configuration for an auto-scaling group, the Health Check Grace Period is the period of time to ignore health checks while instances or auto-scaled instances are added and booting.
* AWS does not recommend ever putting RAID 5's on EBS

EBS volumes can be detached and reattached to other EC2 instances 3 Types of available EBS volumes can be provisioned and attached to an EC2 instance: **EBS 3 Volume Types:**

* 3 Types of EBS volumes - General Purpose SSD, PIOPS SSD, Magnetic.
* GP SSD EBS volumes provide ratio of 3 IOPS per GB, and can burst up to 3000 IOPS.
* IOPS is measured as 256kBps for SSD, 1024kBps for Magnetic.
* PIOPS EBS is good for large NoSQL and RDS databases
* HDD or Magnetic is cheapest and fit for infrequently accessed data.
  + - General Purpose SSD (GP2):
* 1GB – 16TB
  + - * General Purpose up to 10K IOPS
      * 99.999% availability
      * Ratio of 3 IOPS per GB with up to 10K IOPS and ability to burst
      * Up to 3K IOPS for short periods for volumes under 1GB.
* SSD under 1 TB can burst to 3000 IOPS
  + stores credits when under 1500 IOPS.
  + uses credits and can  burst to 3000 IOPS until credit runs  out
* dev and test environments
* billed on  space provisioned
  + - Provisioned IOPS SSD (I01)
      * Designed for I/O intensive applications such as large relational or No-SQL DBs.
      * Use if need more than 10K IOPS
    - Magnetic (Standard)
* HDD or Magnetic is cheapest and fit for infrequently accessed data.
* 1GB – 1TB
  + - * Ideal for workloads where data is accessed infrequently and apps where the lowest cost storage is important.
* billed on space provisioned
* Cheapest/GB.
* Good for infrequently accessed data (fileservers).

#### **General Purpose SSD (gp2)**

The performance of GP2 volumes are tied to volume size - larger volumes perform better. 3 IOPS per GB, up to 1000 IOPS. Can burst up to 3000 IOPS for extended periods of time.

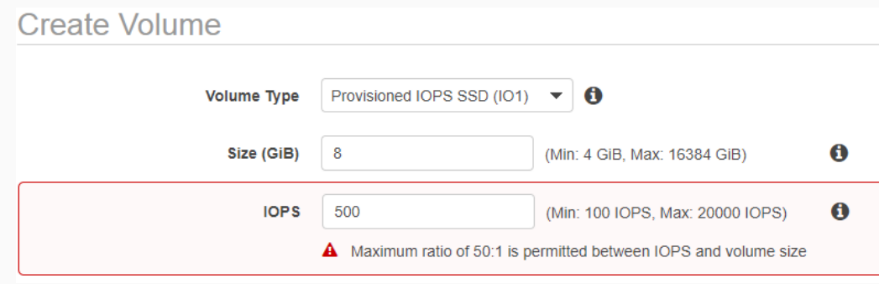
Bursting and I/O credits are only relevant for volumes under 1TB, where burst performance exceeds baseline performance. i.e. Burst is 3000 IOPS per second, baseline performance at 1TB is 3000 IOPS per second - you are always getting burst performance.

#### **Provisioned IOPS SSD (io1)**

Highest performance SSD. Intended for apps which require more than 10000 IOPS per volume or 160 MiB/s of throughput per volume.

IOPS must be between 100 and 20000.

**The maximum ratio of between IOPS and Volume Size is 50:1**. This means that for an 8GiB volume, you can have a max IOPS of 50\*8 = 400 IOPS.



#### **Throughput optimised magnetic (st1)**

Cannot be a boot volume

Useful for:

* Frequently accessed data
* Big data
* Data warehousing
* Log processing

#### **Cold HDD magnetic (sc1)**

Cannot be a boot volume

Cheaper than st1, designed for infrequently accessed, large, sequential workloads

#### **Magnetic (standard)**

Magentic is one of the previous generation volumes, it’s recommended you use one of the newer volume types i.e. st1 or sc1, but may still be in the exam

Amazon Elastic Block Store (EBS) provides **block level storage volumes** for use with EC2 instances.

- EBS volumes are **highly available and reliable storage volumes** that can be **attached to any running instance** that is in the **same Availability Zone**.

- EBS volumes that are attached to an EC2 instance are exposed as **storage volumes that persist independently from the life of the instance**.

How to “grow” an EBS volume:

* Detach the original Amazon EBS volume.
* Create a snapshot of the original Amazon EBS volume’s data in Amazon S3.
* Create a new Amazon EBS volume from the snapshot, but specify a larger size than the original volume.
* Attach the new, larger volume to your Amazon EC2 instance in place of the original. (In many cases, an OS-level utility must also be used to expand the file system.)
* Delete the original Amazon EBS 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

*AWS Definition of IOPs:*

*“IOPs are a unit measure representing****input/output operations per second****. The operations are measured in KiB, and the underlying drive technology determines the maximum amount of data that a volume type counts as a single I/O. I/O size is capped at****256 KiB for SSD****volumes and****1024 KiB for HDD****volumes.”*

EBS volume size determines the amount of IOPS. **Larger size = more IOPS**.

**“Root” vs Additional EBS Volumes**

(1) Every EC2 instance MUST have a “root” volume, which may or may not be EBS

(2) By default, EBS “root” volumes are set to be deleted when the instance is terminated (you can choose to have EBS volume persist)

(3) During the creation of an EC2 instance (and afterwards) you can add additional EBS Volumes to the instance.

(4) Any Additional EBS volume can be attached or detached from the instance at any time, and is not deleted (by default) when the instance is terminated.

**Volumes vs Snapshots**

* Volume
  + A volume is a virtual hard disk (think VMDK)
  + Volumes exist on EBS
  + If you take a snapshot of a volume, this will store that volume on S3
* Snapshot
  + Point in time copy of a volume
  + Exists on S3
  + Are incremental, only the blocks that have changed since the last snap are moved to S3
  + 1st snap takes some time to create
  + Can use snap to create a new volume & change the disk type (magnetic -> GP2 or IO1 or any other combination)
  + If you want to snap a root volume, you should stop the instance before taking snap
    - If you don’t, AWS will stop it prior to taking snap.

### **Snapshots**

Before taking a snapshot of an EBS volume that serves as a root device, you first need to stop the EC2 instance which it’s attached to.

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.

To change the type of a volume, first create a snapshot, then use the snapshot to create the new volume.

EBS volumes must be in the same AZ as the instance they’re attached to. To transfer between instances, snapshot the volume, and use it to create a new volume in the desired AZ.

Additional EBS volumes can be encrypted via the console.

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

Snapshot backups

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.

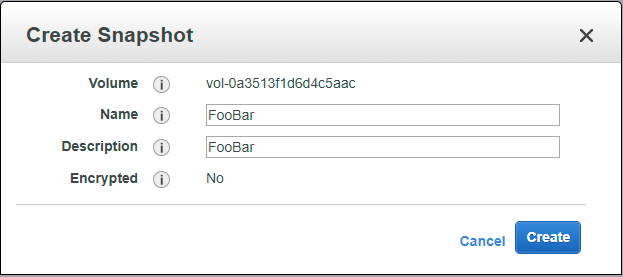
**Backup/Recovery (snapshots)**

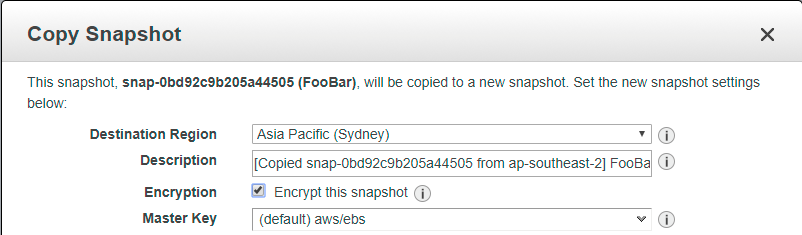
* point in time
* incremental
* stored in S3 pay only for storage (cannot access through client S3)
* created immediately
* Creating a Volume  from a snapshot
  + to use a snapshot you have to create a volume from it
  + when creating, place in Availability zone where the volume is going to be used
  + creates it immediately
* Recovering volumes
  + detach and reattach to another instance
* Encryption Options
  + uses AWS Key Management Services
  + a new master key will be created
  + snapshots of encrypted volumes are encrypted
  + volumes created from encrypted snapshots are encrypted

### **Snapshot encryption**

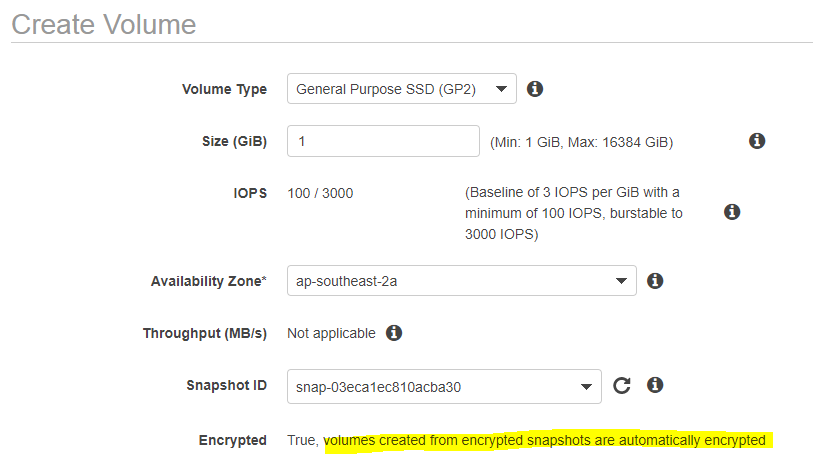
Can encypt snapshots via the ‘copy’ option; if you have an EC2 instance that has an unencrypted volume, and you want to create an encrypted volume from it:

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





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



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

**Snapshots:**

Go into EC2 -> Volumes -> create volume (make sure it's in the same AZ as your server!) -> Actions -> attach to server.

Use \_lsblk \_to view disks to confirm new volume attached.

Use \_file –s /dev/xvdf \_to make sure it's clean

Use \_mkfs –t ext4 /dev/xvdf \_to make file system, then \_mkdir /fileserver *to create directory*, & mount /dev/xvdf/fileserver \_to mount

**Volumes vs Snapshots – Security**

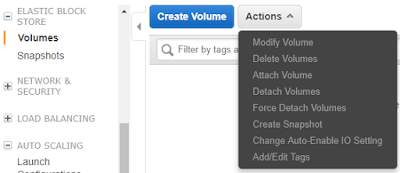
* Once instance can have many volumes but one volume can only be tied to one instance.
* Snapshots of encrypted volumes are encrypted automatically
* Volumes restored from encrypted snapshots are encrypted automatically
* You can share snapshots but only if they are not encrypted
* Snapshots can be shared with other AWS accounts or made public in the market place again as long as they are NOT encrypted

- A snapshot is an “image” of an EBS volume that can be **stored as a backup** of the volume OR used to **create a duplicate**. It is **NOT an active EBS volume** (cannot attach or detach to an EC2 instance.)

- To **restore**create a **new EBS volume** using the **snapshot as template**.

* You can take a snapshot of a volume, this will store that volumes snapshot on S3
* Snapshots are point in time copies of volumes
* The first snapshot will be a full snapshot of the volume and can take a little time to create
* Snapshots are incremental, which means that only the blocks that have changes since your last snapshot are moved to S3
* **If you are making a snapshot of a root volume, you should stop the instance before taking the snapshot**

*Image: AWS Console > Services > EC2 > Elastic Block Store: (EBS) Volume and Snapshots*

[](https://4.bp.blogspot.com/-LUZhoA-XBGQ/WYNCQvPC9QI/AAAAAAAAHSM/zPWXxsv5BDglwblEWwoUyFy0XeGUngICgCLcBGAs/s1600/04+EBS+Volumes+and+Snapshots.png)

**RAID Volumes:**

* If you take a snapshot, the snapshot excludes data held in the cache by applications or OS. This tends to not be an issue on a single volume, however multiple volumes in a RAID array, can cause a problem due to interdependencies of the array
* Take an application consistent snapshot
  + Stop the application from writing to disk
  + Flush all caches to the disk
* Snapshot of RAID array --> 3 Methods:
  + Freeze the file system
  + Unmount the RAID Array
  + Shutdown the EC2 instance --> Take Snapshot --> Turn it back on.

**RAID, Volumes & Snapshots**

* RAID = Redundant Array of Independent Disks
  + RAID 0 – Striped, no redundancy, good performance
  + RAID 1 – mirrored, redundancy
  + RAID 5 – good for reads, bad for writes, **AWS does not recommend ever putting RAID 5’s on EBS**
  + RAID 10 – Striped & Mirrored, good redundancy, good performance
* Why create a RAID in AWS?
  + Not getting Disk I/O that you require from GP2 or IO1 on a single volume.
* Why create a RAID in AWS?
  + Not getting Disk I/O that you require from GP2 or IO1 on a single volume.

Generally use RAID 0 (no reduncancy) or RAID 10 (good redunancy)

Once you provision your EBS volumes for the EC2 instance, for Windows instances, RAID is convered 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

**Placement Groups:**

* A logical group of instance in a single AZ.
* Cannot span multiple AZs
* Are good for grid computing or any time you need **high throughput and very low network latency** between your instances.
* Must have a unique name in your AWS account
* Using placement groups enables applications to participate in a low latency, 10Gbps network
* Placement groups are recommended for applications that benefit from low network latency, high network throughput or both
* A placement group can't span multiple AZ's so it is a SPoF.
* Then name you specify for a placement group must be unique within your AWS account
* Only certain types of instances can be launched in a placement group. Computer Optimized, GPU, Memory Optimized, and Storage Optimized.
* AWS recommends that you use the same instance family and same instance size within the instance group.
* You can't merge placement groups
* You can't 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.
* May only contain certain types of instances:
  + Compute optimised
  + GPU
  + Memory optimised
  + Storage optimised
* Work best when instance types within are homogenous - i.e. all compute optimised or all memory optimised
* Cannot be merged, however it is possible to move existing instances into placement groups

**Addressing an instance**

* dns name (based on ip – public or private)
* Public ip
* EIP – Elastic IP

## LIFECYCLE

* launching
  + bootstrapping
    - providing code to be run at launch
    - specified in UserData
  + VM Import/export
    - can import vm as an instance
    - instances launched within AWS from AMIs cannot be exported
  + Instance metadata
    - data about the instance
    - http://169.254.169.254/latest/meta-data/

Managing Instances

* Tagging
  + name-value pairs
* Monitoring Instances
* Cloudwatch – monitoring and alerting

Modifying an Instance

* instance type
* security group

Termination Protections

* can be enabled
* termination fails until disabled

### **Logging in to an instance**

* initial access is setup by public key – key pairs
* Windows  – AWS generates a random password and the client needs the private key to decrypt it

[EC2 Key Pairs](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html)

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.

### **5.5) Security Groups and NACLs**

A **security group** acts as a **virtual firewall** that controls the traffic for **one or more instances**.

* Block all traffic by default; no traffic rules exist in a freshly created security group.
* You can have any number of EC2 instances associated with a security group

- You associate one or more security groups with an instance.

- Can **modify** the rules for a security group **at any time**.

* If you edit a security group, it takes effect immediately.
* There is an implicit deny any any at the end of the security group rules.
* Can’t add a rule to deny a specific protocol inbound or outbound
* You don't need outbound rules for any inbound request. Rules are stateful meaning that any request allowed in, is automatically allowed out.

Security groups are stateful:

* If you allow a protocol inbound, automatically it’s added to outbound

- We **evaluate all the rules** from **all the security groups** that are associated with the instance.

**Inbound and Outbound Rules**

- When you create a new Security Group, **ALL inbound traffic is DENIED**  (except for ssh for listros and rdp for windows) and **ALL outbound traffic is ALLOWED** by default.

- All traffic is DENIED unless there is an EXPLICIT ALLOW rule for it (there are only ALLOW rules.)

**Best Practice: Allow ONLY traffic that is required.**

* **Do not allow blocking of specific ip addresses (use NACs for this instead)**

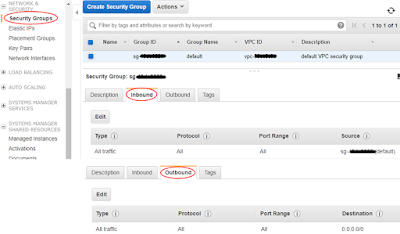
**All instances in a security group can communicate with all other instances in that same security group by default.**

To change the security group of an instance, right click on the instances, select ‘networking’, and select ‘change security group’

NACLs

* Are stateless. If you have inbound traffic - in order for responses to inbound traffic to be received, any inbound traffic rule will require a corresponding outbound rule.

*Image: AWS Console > Services > EC2 > Network & Security > Security Groups*

[](https://2.bp.blogspot.com/-Ndwpmz0w2sA/WYNCW_ue2SI/AAAAAAAAHSQ/4Tj-I6041kotOBehSVwpDa6Mg2OMf70QACLcBGAs/s1600/05+Security+Groups.png)

*Note: In the image above, inbound just allows traffic from whatever is in the security group.*

**5.6) IP Addressing**

* There are 5 Elastic IPs per region in any account by default, because ipv4 is a scare resource
* If the EIP isn’t associated with any running instance, you get charged an hourly penalty for hogging a scarce resource
* By default, every instance comes with a Public IP and a Private IP.
* When the instance stops or terminates, the private IP is released. So you may see a diff private IP when you boot it back up. The public IP behaves the same way but it can also be replaced with an EIP.

- (By default) ALL EC2 instances have a **private IP address**. *Private IP addresses allow for instances to communicate with each other as long as they are located in the same VPC or broader private network.*

- EC2 instances can be launched with or without a **public IP address**, depending on VPC/subnet settings. *Public IP addresses are REQUIRED for the instance to communicate with the Internet.*

*Note: The “default” VPC and subnets are configured so that any new instance that is provisioned has a public IP address.*

**5.7) Provisioning EC2 (example)**

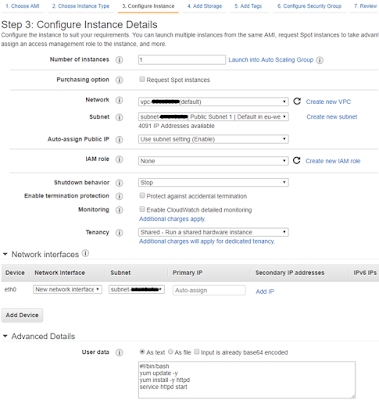
**AWS Console > Services > EC2 > Launch Instance**

Step 1) Quick Start: **Amazon Linux** **(Free tier eligible)** >**Select**

Step 2) Instance Type: **General Purpose t2.micro** **(Free tier eligible)**> **Next: Configure Instance Details**

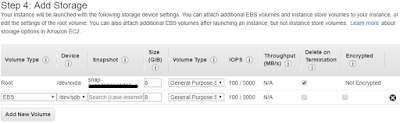
Step 3) Configure Instance Details: **Configure** > **Next: Add Storage**

*Image: Step 3) Configure Instance Details (example)*

[](https://2.bp.blogspot.com/-j0xZSj0L0jI/WYNCc3PvTpI/AAAAAAAAHSU/jafDSvMQMIcI41KDU87y-sdVCCMS-MnrgCLcBGAs/s1600/06+Configure+Instance+Details.png)

Step 4) Add Storage: **Configure > Next: Add Tags**

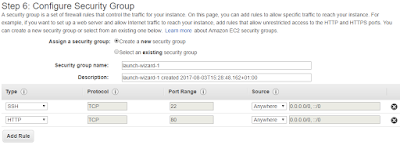
*Image: Step 4) Add Storage (example)*

[](https://2.bp.blogspot.com/-OTo6yByv9Es/WYNClTcnQlI/AAAAAAAAHSY/YAmiuYePpOsxvT29hVPfVGnqOG01nbccgCLcBGAs/s1600/07+Add+Storage.png)

Step 5) Add Tag: **Give the ‘Name’ key a value** (optional) > **Next: Configure Security Group**

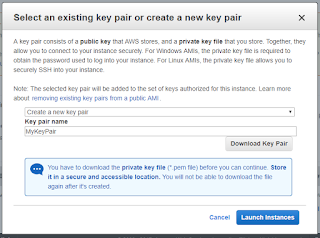
Step 6) Configure Security Group: **Configure** > **Review and Launch**

*Image: Step 6) Configure Security Group (example)*

[](https://2.bp.blogspot.com/-Vro6LLhCZxU/WYNCqItcokI/AAAAAAAAHSc/X0EhEKWpmJ82OpQ8DDWof3mlsrzxIc52ACLcBGAs/s1600/08+Configure+Security+Group.png)

Step 7) Review Instance Launch: **Launch** > **Create a new key pair** > **Download Key Pair** >**Launch Instance**

*Image: Step 7) Downloading Key Pair prior to Launch Instance(s)*

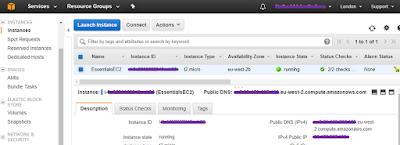
[](https://1.bp.blogspot.com/-HNq5AaGfhOI/WYNCuZn3TrI/AAAAAAAAHSg/auyuNuNujws1EOYZ9HgTYpzVGp-6KJh8ACLcBGAs/s1600/09+Download+Key+Pair.png)

Give it a few minutes and your instance is up and running!

[AWS Docs: Connect to Your Linux Instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstances.html)

[AWS Docs: Connecting to Your Linux Instance from Windows Using PuTTY](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html)

*Image: AWS Console > Services > EC2 > Instances: Instances*

[](https://1.bp.blogspot.com/-BbThNX89MKo/WYNC1NehA2I/AAAAAAAAHSk/Ce3WacmGFfkvYWmiwUnKnFs3BzQRYITXACLcBGAs/s1600/10+Running+EC2+Instance.png)

*Note: If you’re using the Free Tier, remember to stop your instances when you’re not using them.*

Roles:

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..

* You can only assign an EC2 role to an instance on create. You can not assign a role after the instance has been created and/or is running
* You can change the permissions on a role post creation, but can NOT assign a new role to an existing instance
* Role permissions can be changed, but not swapped
* Roles are more secure then storing your access key and secret key on individual EC2 instances
* Roles are easier to manager, You can assign a role, and change permissions on that role at any time which take effect immediately
* Roles can only be assigned when that EC2 instance is being provisioned
* Roles are universal, you can use them in any region.

### **Bootstrap scripts**

Bootstrap scripts are set up in the EC2 user data, can be used to install PHP, etc on new EC2 instances.

i.e. a bootstrap script could look like:

#!/bin/bash

yum update -y

hum install yttpd -y

service httpd start

chkconfig httpd on

aws s3 cp s3://...

## **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

* You can have up to 10 tags per EC2 instance.
* **Resource or Operation Default Limit**
* **Elastic IP addresses for EC2-Classic: 5**
* **Security groups for EC2-Classic per instance: 500**
* **Rules per security group for EC2-Classic: 100**
* **Key pairs: 5000**
* **On-Demand instances: Varies based on instance type**
* **Spot Instances: Varies based on instance type**
* **Reserved Instances: 20 instance reservations per Availability Zone, per month**
* **Dedicated Hosts: Up to 2 Dedicated Hosts per instance family, per region can be allocated**
* **AMI Copies: Destination regions are limited to 50 concurrent AMI copies at a time, with no more than 25 of those coming from a single source region.**
* **Throttle on the emails that can be sent : Throttle applied**
* **Tags per EC2 instance: 10**
* **ELB (Elastic Block Storage Limits)**
* **Resource or Operation Default Limit**
* **Number of EBS volumes: 5000**
* **Number of EBS snapshots: 10,000**
* **Total volume storage of General Purpose SSD (gp2) volumes: 20 TiB**
* **Total volume storage of Provisioned IOPS SSD (io1) volumes: 20 TiB**
* **Total volume storage of Throughput Optimized HDD (st1): 20 TiB**
* **Total volume storage of Cold HDD (sc1): 20 TiB**
* **Total volume storage of Magnetic volumes: 20 TiB**
* **Total provisioned IOPS: 40,000**
* **For additional information about EC2 Limits, see Limits in Amazon EC2**
* **ELB (Elastic Block Storage Limits)**
* **Resource or Operation Default Limit**
* **Number of EBS volumes: 5000**
* **Number of EBS snapshots: 10,000**
* **Total volume storage of General Purpose SSD (gp2) volumes: 20 TiB**
* **Total volume storage of Provisioned IOPS SSD (io1) volumes: 20 TiB**
* **Total volume storage of Throughput Optimized HDD (st1): 20 TiB**
* **Total volume storage of Cold HDD (sc1): 20 TiB**
* **Total volume storage of Magnetic volumes: 20 TiB**
* **Total provisioned IOPS: 40,000**