# EC2 – The Backbone of AWS

## EC2 101

* Elastic Compute Cloud
* Provides resizable compute capacity in cloud (scale up and down)
* Helps developers’ build failure resilient systems and isolate them

### EC2 Pricing

* On demand Instance
  + Pay per second (linux)/ hour (windows) of usage with no commitment
  + Applications with short term, spiky usage patterns or unpredictable workloads that cannot be interrupted
  + New apps on AWS being tested
* Reserved Instance
  + Reserve capacity over significant period of time (1-3 years). Significant discount.
  + Applications with steady or predictable usage over time. Reserved capacity required.
  + Further discount if upfront payment
  + Massive discount for reserved instances over a long period of time – upto 70% for 3 years.
  + Standard Reserved Instance – 75% off on demand
  + Convertible RI ( Upto 54% off on demand) Can convert type of instance
  + Scheduled RI: When required for a brief time but predictable schedule
* Spot Instance
  + Bid your price for compute. When bid price is higher than Spot price, then you can provision it. When it goes lower, instance is terminated. Useful for applications who have flexible start / stop times
  + [Exam Tip] If AWS terminates instance, you are not charged for partial hour. If you terminate, you will be charged for the hour.
  + Applications that are feasible only at very low compute prices. E.g. pharma simulations, big data processing etc.
  + Applications with urgent computing capacity
* Dedicated physical machines – pay by hour.
  + Useful for regulatory requirements
  + Certain licensing agreements prevent usage on virtual machine / multi-tenancy deployments.

### EC2 Instance Types

| **Sr. No** | **Family** | **Specialty** | **Use Case** | **Type** |
| --- | --- | --- | --- | --- |
| 1 | D2 | Dense Storage | File Servers / DWH / Hadoop | Storage Optimized |
| 2 | R4. R3 | Memory Optimized | Memory Intensive / DBs | Memory Optimized |
| 3 | M4. M3 | General Purpose | Application Servers | General Purpose |
| 4 | C4, C3 | Compute Optimized | CPU Intensive Apps, DBs | Compute O |
| 5 | G2 | Graphics Intensive | Video Encoding / 3D Application Streaming |  |
| 6 | I2 | High speed storage (IOPS) | NoSQL DBs, DWH |  |
| 7 | F1 | Field Programmable Gate Array | Hardware acceleration of Code |  |
| 8 | T2 | Lowest Cost, General Purpose | Web Servers/ Small DBs | General Purpose |
| 9 | P2 | Graphics / General Purpose GPU[Parallel Processing] | Machine Learning / Bit Coin Mining. |  |
| 10 | X1 | Memory Optimized | SAP HANA / Apache Spark | - |

Acronym – \*DIRT MCG FPX -

D – Density , I - IOPS , R – RAM , T – cheap T2, M – Main Choice ( default) – Apps, C – Compute, G – Graphics, F – FPGA , P – Graphics – Pics – Parallel Processing , X – Extreme Memory - \*

Use M3 for general purpose instances – balanced compute, memory and network resources

[Exam Tip] You will be asked to provide which instance type to use for a given scenario. Usually 3 options are fictitious.

EC2 Key Pairs are region specific

## EBS (Elastic Block Storage)

* Block based storage
* You can install OS, Database on it, unlike S3
* Placed in specific AZ. Automatically replicated within the AZ to protect from failure.
* [Exam Tips] \*\* - EBS Volume Types\*\*

SSD Drives

* (root volume) **General Purpose SSD** – up to 10,000 IOPS. 3 IOPS per GB. Balances price and performance. You can burst upto 3000 IOPS for 1GB
* (root volume) **Provisioned SSD** – when you need more than 10,000 IOPS. Large RDBMS DBs and NoSQL DBs. Up to 20000 IOPS now

Magnetic Drives

* **HDD, Throughput Optimized– ST1** – Required for data written in sequence. Big Data, DWH, Log processing. Cannot be used as boot volumes
* **HDD, Cold– SC1** – Data that isn’t frequently accessed. E.g. File Server. Cannot be used as boot volume
* (root volume) **HDD, Magnetic (Standard)** – Cheapest bootable EBS volume type. Used for apps where data is less frequently accessed and low cost is important.
* You cannot mount 1 EBS volume to multiple EC2 Instances. Use EFS instead.
* EBS Root Volumes can be encrypted on custom AMIs only. Not on the default available AMIs. To encrypt root volumes, create a new AMI and encrypt root volume. You can also encrypt using 3rd party software like Bit Locker. Additional volumes attached to EC2 instance can be encrypted.
* EC2 – 1 subnet equals 1 Availability Zone.
* Default VPC & Security group is created in when you create your account.
* Default CloudWatch monitoring – every 5 mins. Can enabled advanced monitoring to check at interval of each minute.
* Volume – Virtual Hard Disk
* Tag everything on AWS
* Default Linux EC2 username is ec2-user
* Default Windows EC2 username is Administrator
* Termination protection is turned off by default. You need to turn it on.
* When instance is terminated, root volume is deleted. You can turn if off.
* System Status Check – Overall health of hosting infrastructure. If they arise, Terminate instance and recreate
* Instance Status Check – Health of instance. If they arise, reboot the instance.

## EC2 Security Groups

* A security group is a virtual firewall.
* First line of defense. Network ACLs are second line.
* 1 instance can have multiple security groups. As each security group only "allows" inbound traffic, there will never be a conflict on security group rules.
* Security group changes are applied immediately.
* Security groups are "stateful". Rules added as inbound rules – automatic outbound rules are added. Response back on the same channel. NACLs are stateless.
* All inbound traffic is blocked by default. You have to allow specific inbound rules for protocols
* All outbound traffic is allowed by default.
* Only allow rules, no deny rules exist. Use NACLs to deny specific IPs
* Any number of EC2 instances in a security group.
* EC2 instances in the default security group can communicate with each other.
* Multiple security groups can be attached to an instance.

## Volumes and Snapshots

* Volumes are virtual hard disks.
* You can attach volume to EC2 instance belonging to same AZ
* To Detach a volume from EC2 instance, you have to umount it first
* Snapshots are point in time copies of volumes – stored in S3. Taking first snapshot takes a while.
* Subsequent snapshots will only store the delta in S3. Only changed blocks are stored in S3.
* You can create volumes from Snapshots. During this you can also change Volume Storage Type
* Volume is just block data. You need to format it create specific file system e.g. ext4
* Root Volume is one where OS is installed / booted. It is not encrypted by default on AWS AMIs

### RAID, Volumes & Snapshots.

* RAID 0 – Striped, No Redundancy , Good Performance – No Backup/Failover
* RAID 1 – mirrored, Redundancy
* RAID 5 – Good for reads, bad for writes. AWS doesn’t recommend using RAID 5 on EBS
* RAID 10 – Raid 0 + Raid 1
* Use RAID Arrays when a single volume IOPs are not sufficient for your need. E.g. Database. Then you create RAID Array to meet IOPs requirements.
* To take snapshot of RAID Array –
  1. Stop the application from writing to cache and flush all cache to Disk
  2. Freeze the file system
  3. Umount the RAID Array
  4. Shutdown the EC2 instance
* Snapshots of encrypted volumes are encrypted automatically.
* You can copy snapshot to another region while encrypting it.
* Create Image from snapshot.
* The EC2 instance thus created will have root volume encrypted.
* You can’t share encrypted snapshots as the encryption key is tied to your account.

## EBS backed v/s Instance store

* You can reboot or terminate instance store backed EC2 VMs
* You can start , stop , reboot or terminate EBS backed EC2 VMs
* EC2 instance on instance store is lost if host hypervisor fails. Not so with EBS backed instances.
* EBS volumes can be attached / detached to EC2 instances. One at a time
* EBS backed AMI is from EBS snapshot
* Instance store back volume is from template in S3. Hence slower to provision
* You will not lose data is you reboot for both.
* With EBS, you can ask AWS not to delete the volume upon instance termination.

## EC2 Status Checks

There are two types of status checks: system status checks and instance status checks.

### System Status Checks

Monitor the AWS systems required to use your instance to ensure they are working properly. These checks detect problems with your instance that require AWS involvement to repair. When a system status check fails, you can choose to wait for AWS to fix the issue, or you can resolve it yourself (for example, by stopping and starting an instance, or by terminating and replacing an instance).

The following are examples of problems that can cause system status checks to fail:

* Loss of network connectivity
* Loss of system power
* Software issues on the physical host
* Hardware issues on the physical host that impact network reachability

## Instance Status Checks

Monitor the software and network configuration of your individual instance. These checks detect problems that require your involvement to repair. When an instance status check fails, typically you will need to address the problem yourself (for example, by rebooting the instance or by making instance configuration changes).

The following are examples of problems that can cause instance status checks to fail:

* Failed system status checks
* Incorrect networking or startup configuration
* Exhausted memory
* Corrupted file system
* Incompatible kernel

## CloudWatch

* Default Metrics – Network, Disk , CPU and Status check ( Instance and System)
* Memory – RAM is a custom metric
* You can create custom dashboards all CloudWatch metrics.
* CloudWatch alarms – set notifications when particular thresholds are hit.
* CloudWatch events help you respond to state changes. E.g. run Lambda function in response to.
* CloudWatch logs helps you monitor EC2 instance/application/system logs. Logs send data to CloudWatch
* Standard monitoring 5 mins. Detailed monitoring 1 minute.
* CloudWatch is for logging. CloudTrail is for auditing your calls.

## AWS CLI Usage

* Users can login with Access Key ID and Secret Access Key. If anything is compromised, you can regenerate the secret access key.
* Also you can delete the user and recreate.

## IAM Roles for EC2

* Avoid using user credentials on servers
* IAM roles can be assigned/replaced to existing EC2 instances using AWS CLI. Not through the console.
* A trick is to assign policies to the existing role. This will avoid the need to create new instances.
* Role assigned to instance is stuck to the lifetime of the instance – until you delete the role. Easier to modify existing role by adding / removing policies.
* Roles are universal. Applicable to all regions.

## Bootstrap scripts.

* Scripts can be passed on to the EC2 instance at first boot time as part of user-data.

## EC2 Instance Meta-Data

* curl <http://169.254.169.254/latest/meta-data/>
* Instance information is available in Meta-Data. Not in User-Data

## Auto Scaling 101

* Before you can create Auto scaling group you need to create a launch configuration
* Launch Configuration – Select AMI, Instance Type , Bootstrap script
* No actual instances are created just with launch configuration
* Auto scaling group – Set minimum size, spread it over subnets (AZs)- select all available AZs
* Run health checks from ELB
* Configure Auto scaling policy – Based on Alarm take action – trigger a new instance creation when CPU Utilization is greater than 90% for 5 minutes. You can also delete instance based on alarms
* When Auto scaling group is launched it creates the instances based on definition.

## EC2 Placement groups

* Logical grouping of instances within a single AZ
* Instances can participate in low latency, 10 GBPs network.