Why AWS EC2?

Why not buy our own stack of servers and work independently? Suppose we are a developer and want to work independently. We buy some servers, we estimated the capacity and the computing power may be sufficient. Now, we have to look after the updating of security patches every day, we have to troubleshoot any problem which might occur at a back-end level in the servers and so on. These are all extra chores that we will be doing or maybe we will hire someone else to do these things for us.

But if we buy an EC2 instance, we do not have to worry about any of these things as it will all be managed by Amazon; we just have to focus on our application. That too, at a fraction of a cost that we were incurring earlier. Isn't that interesting?

It is difficult to predict how much computing power one might require for an application which we might have just launched. There can be two scenarios: we may over-estimate the requirement and buy stack of servers which may not be of any use or we may under-estimate the usage, which will lead to the crashing of our application. In this chapter, we will understand all the key concepts and perform hands-on to launch an Ubuntu instance.

What is AWS EC2?

Amazon Elastic Compute Cloud, EC2 is a web service from Amazon that provides re-sizable compute services in the cloud.

How are they re-sizable?

They are re-sizable because you can quickly scale up or scale down the number of server instances you are using if your computing requirements change.

What is an Instance?

An instance is a virtual machine server for running applications on Amazon's EC2. It can also be treated like a tiny part of a larger computer, a tiny part which has its own hard drive, network connection, OS but is actually all virtual. We can have multiple "tiny" computers on a single physical machine, and all these tiny machines are called Instances.

Difference between a service and an Instance?

EC2 is a service along with other Amazon Web Services like S3 etc.

When we use EC2 or any other service, we use it through an instance, e.g. t2.micro instance, in EC2 etc.

How to run systems in EC2?

- Login to your AWS account and click on AWS EC2.
- Under create instance, click on launch instance.

Now you have to select an Amazon Machine Image (AMI), AMIs are templates of OS and they provide the information needed to launch an instance.

When we want to launch an instance, we have to specify which AMI we want to use. It could be Ubuntu, windows server etc.

The AMIs could be preconfigured or you can configure it on your own according to your requirements.

- For preconfigured AMIs you have to select it from AWS marketplace.
- For setting up your own, go to quick-start and select one.

While configuring, we will reach a point where we have to select an EBS storage option.

Let's understand Cost Savings using an example.

Suppose instead of taking AWS EC2, we consider taking a dedicated set of servers, so, what all we might have to face:

- Now for using these servers we have to hire an IT team which can handle them.
- Also, having a fault in the system is unavoidable, therefore we have to bear the cost of getting it fixed, and if you don't want to compromise on your up-time you have to keep your systems redundant to other servers, which might become more expensive.
- Your own purchased assets will depreciate over the period of time, however, as a matter of fact the cost of an instance have dropped more than 50% over a 3-year period, while improving processor type and speed. So eventually, moving to Cloud is all more suggested.
- For scaling up we have to add more servers, and if your application is new and you experience a sudden traffic, scaling up that quickly might become a problem.

These are just a few problems and there are many others scenarios which make the case for EC2 stronger!

Let us understand the types of EC2 Computing Instances:

Computing is a very broad term; the nature of our task decides what kind of computing we need.

Therefore, AWS EC2 offers 5 types of instances which are as follows:

- *General Instances*
  - For applications that require a balance of performance and cost.
    - E.g. email responding systems, where we need a prompt response as well as that it should be cost effective, since it doesn't require much processing.
- *Compute Instances*
  - For applications that require a lot of processing from the CPU.
    - E.g. analysis of data from a stream of data, like Twitter stream
- *Memory Instances*
  - For applications that are heavy in nature, therefore, require a lot of RAM.
    - E.g. when your system needs a lot of applications running in the background i.e. multitasking.
- *Storage Instances*
  - For applications that are huge in size or have a data set that occupies a lot of space.
    - E.g. when your application is of huge size.
- *GPU Instances*
  - For applications that require some heavy graphics rendering.
    - E.g. 3D modelling etc.

Now, every instance type has a set of instances which are optimized for different workloads:

- General Instances
  - t2
  - m4
  - m3
- Compute Instances
  - c4
  - c3
- Memory Instances
  - r3
  - x1
- Storage Instances
  - i2
  - d2
- GPU Instances
  - g2

Now let's understand the kind of work that each instance is optimized for.

Burstable Performance Instances

- *T2 instances* are burstable instances, meaning the CPU performs at a baseline, say 20% of its capability. When our application needs more than 20% of the performance of the CPU, the CPU enters into a burst mode giving higher performance for a limited amount of time, therefore work happens faster.
    - We get these credits when our CPU is idle.
    - Each CPU credit gives a burst of 1 minute to the CPU.
    - If our CPU credits are not used, they are credited to our account and they stay there for 24 hours.
    - Based on our credit balance, we can decide whether the t2 instance, should be scaled up or down.
    - These bursts happen at a cost, every time a burst happens in a CPU, CPU credits are used.

EBS-optimized Instance

- *C4, M4, and D2 instances*, are EBS optimized by default, EBS means Elastic Block Storage, which is a storage option provided by AWS in which the IOPS* rate is quite high. Therefore, when an EBS volume is attached to an optimized instance, single digit millisecond latencies can be achieved.

*IOPS (Input/output Operations per Second, pronounced eye-ops) is a performance measurement used to characterize computer storage devices.

Cluster Networking Instances

- *X1, M4, C4, C3, I2, G2 and D2 instances* support cluster networking. Instances launched into a common placement group are put in a logical group that provides high-bandwidth, low latency between all the instances in the group.
    - A placement group is basically a logical cluster where some select EC2 instances which are a part of that group can utilize up to 10Gbps for single flow and 20Gbps for multi flow traffic in each direction.
    - Instances which are not a part of that group are limited to 5 Gbps speed in multi flow traffic. Cluster Networking is ideal for high performance analytic system.

Dedicated Instances

- They are the instances that run on single-tenant hardware dedicated to a single customer.
- They are perfect for workloads where a corporate policy or industry regulation requires that your instance should be isolated from any other customer's instance, therefore they go for their own separate machines, and their instances are isolated at the hardware level.

Let's understand this through an example. Suppose in our organization we have the following tasks:

- *Analysis of customer's data*

  Customer's website activity, etc. should all be monitored in real-time. There will be times when the traffic on the website will be minimum, therefore using a very powerful processor should not be considered, since it will become expensive for the company because it will not be used for every hour of the day. Hence, for this task, we might take **t2 instances** because they give **Burstable CPU performance** i.e. when the traffic will be more the CPU performance will be increased accordingly to meet the requirements.
- *Our auto-response emailing system*
  - It should be quick, therefore we would require systems, where the response time is as short as possible. This could be achieved by using **EBS optimized instances**, as they offer high IOPS and hence, low latencies.
- *The search engine on our website*
  - It should be able to sort the keywords and return relevant results, therefore we might have 2 servers for this. One is the database and the other server for processing the keywords. Therefore, the communication between these servers should be at the maximum possible rate. To achieve this, we can put them in a placement group and for that we have to use **Cluster Networking Instances.**
- *Some processes in every organization are highly confidential*
  - Because these processes give us an edge over other companies, no matter how secure the servers, maybe, some policies are still made to be sure. Therefore, we might use **Dedicated Instances** for these kinds of processes.