Amazon WEB SERVICE ES

zzak00

What is Amazon EC2 ?

Amazon Elastic Compute Cloud (EC2) is a part of Amazon.com's cloud-computing platform, Amazon Web Services (AWS), that allows users to rent virtual computers on which to run their own computer applications. EC2 encourages scalable deployment of applications by providing a web service through which a user can boot an Amazon Machine Image (AMI) to configure a virtual machine, which Amazon calls an "instance", containing any software desired. A user can create, launch, and terminate server-instances as needed, paying by the second for active servers – hence the term "elastic". EC2 provides users with control over the geographical location of instances that allows for latency optimization and high levels of redundancy.

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 server for running applications on Amazon's EC2. It can also be understood like a tiny part of a larger computer, a tiny part which has its own Hard drive, network connection, OS etc. But it is actually all virtual. You 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?

1. EC2 is a service along with other Amazon Web Services like S3 etc.
2. When we use EC2 or any other service, we use it through an instance, e.g. t2.micro instance, in EC2 etc.

Types of EC2 computing Instances :

Computing is a very broad term, the nature of your task decides what kind of computing you need.
Therefore, AWS EC2 offers 5 types of instances which are as follows:

1. *General Instances:* ***t2, m4, m3***
   For applications that require a balance of performance and cost.

   E.g : email responding systems, where you need a prompt response as well as it should be cost-effective since it doesn't require much processing.
2. *Compute Instances:* ***c4, c3***
   For applications that require a lot of processing from the CPU.

   E.g : analysis of data from a stream of data, like a Twitter stream
3. *Memory Instances:* ***r3, x1***
   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.
4. *Storage Instances:* ***i2, d2*** 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.
5. *GPU Instances:* ***g2***
   For applications that require some heavy graphics rendering.

   E.g : 3D modeling etc.

What is Amazon EMR ?

Amazon EMR (previously known as Amazon Elastic MapReduce) is an Amazon Web Services (AWS) tool for big data processing and analysis. Amazon markets EMR as an expandable, low-configuration service that provides an alternative to running on-premises cluster computing. Amazon EMR processes big data across a Hadoop cluster of virtual servers on Amazon Elastic Compute Cloud (EC2) and Amazon Simple Storage Service (S3). The Elastic in EMR's name refers to its dynamic resizing ability, which enables administrators to increase or reduce resources, depending on their current needs.

Amazon EMR use cases:

1. *Machine learning:*
   EMR's built-in ML tools use the Hadoop framework to create a variety of algorithms to support decision-making, including decision trees, random forests, support-vector machines and logistic regression.
2. *Extract, transform and load:*
   ETL is the process of moving data from one or more data stores to another. Data transformations – such as sorting, aggregating and joining – can be done using EMR.
3. *Clickstream analysis:*
   Clickstream data from Amazon S3 can be analyzed with Apache Spark and Apache Hive. Apache Spark is an open source data processing tool that can help make data easy to manage and analyze. Spark uses a framework that enables jobs to run across large clusters of computers and can process data in parallel. Apache Hive is a data warehouse infrastructure built on top of Hadoop that provides tools for working with data that Spark can analyze. Clickstream analysis can help organizations understand customer behaviors, find ways to improve a website layout, discover which keywords people are using in search engines and see which word combinations lead to sales.
4. *Real-time streaming:*
   Users can analyze events using streaming data sources in real time with Apache Spark Streaming and Apache Flink. This enables streaming data pipelines to be created on EMR.
5. *Interactive analytics:*
   EMR Notebooks are a managed service that provide a secure, scalable and reliable environment for data analytics. Using Jupyter Notebook – an open source web application data scientists can use to create and share live code and equations – data can be prepared and visualized to perform interactive analytics.
6. *Genomics:*
   Organizations can use EMR to process genomic data to make data processing and analysis scalable for industries including medicine and telecommunications.

What is Amazon Kinesis ?

Amazon Kinesis makes it easy to collect, process, and analyze real-time, streaming data to get timely insights and react quickly to new information. Collection of services for processing streams of various data. In kinesis, the data is processed in *"shards"*.

Kinesis Data Streams is used for rapid and continuous data intake and aggregation. The type of data used can include IT infrastructure log data, application logs, social media, market data feeds, and web clickstream data.

Shreds in Kinesis:

A shard is a uniquely identified sequence of data records in a stream. A stream is composed of one or more shards, each of which provides a fixed unit of capacity. Each shard can support up to 5 transactions per second for reads, up to a maximum total data read rate of 2 MB per second and up to 1,000 records per second for writes, up to a maximum total data write rate of 1 MB per second (including partition keys). The data capacity of your stream is a function of the number of shards that you specify for the stream. The total capacity of the stream is the sum of the capacities of its shards.

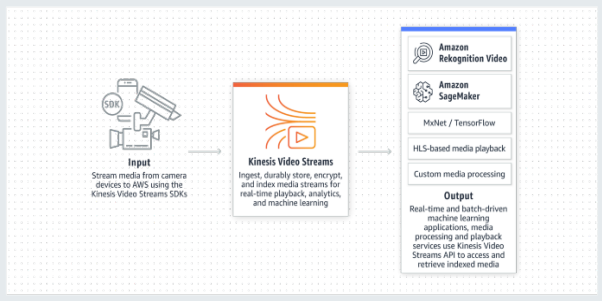
Mainly the shreds are used to send a certain amount of capacity in an ordered way.

If the application must process all messages in order, then you can only use one shard. *Think of it as a line at a bank — if there is one line, then everybody gets served in order.*

If messages only need to be ordered for a certain subset of messages, they can be sent to separate shards. *For example, multiple lines in a bank, where each line gets served in order.*

Kinesis Video Streams

Kinesis Video Streams makes it easy to securely stream video from connected devices to AWS for analytics, machine learning (ML), and other processing.



Kinesis Data Streams

Kinesis Data Streams enables you to build custom applications that process or analyze streaming data for specialized needs. Kinesis Data Streams enables real-time processing of streaming big data. Also, it is useful for rapidly moving data off data producers and then continue processing the data.

Kinesis Data Streams stores data for later processing by applications (key difference with Firehose which delivers data directly to AWS services).

Kinesis Data Firehose

Kinesis Data Firehose is the easiest way to load streaming data into data stores and analytics tools. It captures, transforms, and loads streaming data. Kinesis Data Firehose enables near real-time analytics with existing business intelligence tools and dashboards.

Another important point about Kinesis Data Firehose is Kinesis Data Streams can be used as the source(s) to Kinesis Data Firehose.

You can configure Kinesis Data Firehose to transform your data before delivering it. It can batch, compress, and encrypt data before loading it. With Kinesis Data Firehose you don't need to write an application or manage resources.

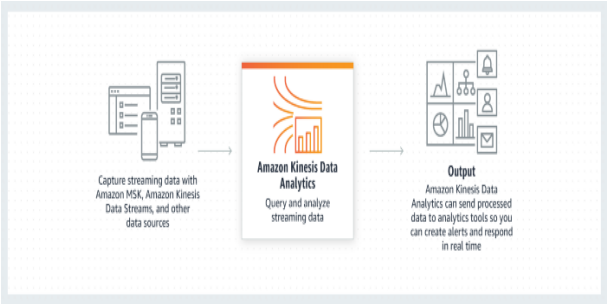
Kinesis Data Analytics

Amazon Kinesis Data Analytics is the easiest way to process and analyze real-time, streaming data. You can use standard SQL queries to process Kinesis data streams. Also Provides real-time analysis.

Use cases:

1. Generate time-series analytics.
2. Feed real-time dashboards.
3. Create real-time alerts and notifications.

You can quickly author and run powerful SQL code against streaming sources. It accepts data from Kinesis Streams and Kinesis Firehose. The Kinesis Data Analytics able to write the output to S3, RedShift, Elasticsearch and Kinesis Data Streams. You can write kinesis queries over Kinesis Data Streams and Kinesis Data Firehose.



What is KIBANA ?

Kibana is a data visualization and exploration tool used for log and time-series analytics, application monitoring, and operational intelligence use cases. It offers powerful and easy-to-use features such as histograms, line graphs, pie charts, heat maps, and built-in geospatial support. Also, it provides tight integration with Elasticsearch, a popular analytics and search engine, which makes Kibana the default choice for visualizing data stored in Elasticsearch.

ElasticSearch

Elasticsearch is a highly scalable open-source full-text search and analytics engine. It allows you to store, search, and analyze big volumes of data quickly and in near real time. It is generally used as the underlying engine/technology that powers applications that have complex search features and requirements. Elasticsearch provides a distributed system on top of Lucene StandardAnalyzer for indexing and automatic type guessing and utilizes a JSON based REST API to refer to Lucene features.
It is easy to set up out of the box since it ships with sensible defaults and hides complexity from beginners. It has a short learning curve to grasp the basics so anyone with a bit of efforts can become productive very quickly. It is schema-less, using some defaults to index the data.

Backend Components:

To better understand Elasticsearch and its usage is good to have a general understanding of the main backend components.

Node

A node is a single server that is part of a cluster, stores our data, and participates in the cluster's indexing and search capabilities. Just like a cluster, a node is identified by a name which by default is a random Universally Unique Identifier (UUID) that is assigned to the node at startup. We can edit the default node names in case we want to.

Cluster

A cluster is a collection of one or more nodes that together holds your entire data and provides federated indexing and search capabilities. There can be N nodes with the same cluster name. Elasticsearch operates in a distributed environment: with cross-cluster replication, a secondary cluster can spring into action as a hot backup.

Index

The index is a collection of documents that have similar characteristics. For example, we can have an index for a specific customer, another for a product information, and another for a different typology of data. An index is identified by a unique name that refers to the index when performing indexing search, update, and delete operations. In a single cluster, we can define as many indexes as we want. Index is similiar to database in an RDBMS.

Document

A document is a basic unit of information that can be indexed. For example, you can have an index about your product and then a document for a single customer. This document is expressed in JSON (JavaScript Object Notation) which is a ubiquitous internet data interchange format. Analogy to a single raw in a DB. Within an index, you can store as many documents as you want, so that in the same index you can have a document for a single product, and yet another for a single order.

Shard and Replicas

Elasticsearch provides the ability to subdivide your index into multiple pieces called shards. When you create an index, you can simply define the number of shards that you want. Each shard is in itself a fully-functional and independent "index"

that can be hosted on any node in the cluster. Shards is important cause it allows to horizontally split your data volume, potentially also in multiple nodes paralelizing operations thus increasing performance. Shards can also be used by making multiple copies of your index into replicas shards, which in cloud environments could be useful to provide high availability.

The Elastic stack

Although search engine at its core, users started using Elasticsearch for logs and wanted to easily ingest and visualize them. Elasticsearch, Logstash, Kibana are the main components of the elastic stack and are know as ELK.

Elasticsearch use cases

Elasticsearch can be used in so various ways that is difficult for me to capture all the most interesting use cases.

1. **Main data store:** Create searchable catalog, document store and logging system.
2. **Complementary Technology:** add visualization capabilities to SQL, mongoDB, cast indexing and search to Hadoop, or add processing and storage to kafka.
3. **Additive technology:** In case you have already logs in Elasticsearch, you may want to add metrics, monitoring, and analytics capabilities.