What Is EMR ?

Amazon EMR (previously called Amazon Elastic MapReduce) is a managed cluster platform that simplifies running big data frameworks, such as Apache Hadoop and Apache Spark, on AWS to process and analyze vast amounts of data.

Using these frameworks and related open-source projects, you can process data for analytics purposes and business intelligence workloads.

Amazon EMR also lets you transform and move large amounts of data into and out of other AWS data stores and databases, such as Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB.

Amazon EMR is used for various purpose as below:

1. Perform big data analytics
2. Build scalable data
3. Process real-time data
4. Accelerate data science and ML adoption

To create EMR

1. Go to the AWS console home page
2. Then go to the **services** tab
3. Select **Analytics** option
4. Choose **EMR** there
5. In EMR click on the **Create cluster** button
6. In that click on **Go to the advanced** options, because in **General Configuration** we have limited n selected services in that. We require some extra and additional services as per our requirement, hence we have to click on the **Go to advanced options** button

   Further steps In **Advanced Options**

   6.1. **Software and Steps**

   In **Software Configuration** selected the required options as per the requirement

   Reference: EMR Configuration

   6.1.1. Selected the required options as per our project are

   - First Select EMR version - **emr-5.34.0** selected because it is fully tested and stable version available at this condition
   - Select **Hadoop 2.10.1**
   - Select **JupyterHub 1.1.0**
   - Select **Hive 2.3.8**
   - Select **JupyterEnterpriseGateway 2.1.0**
   - Select **Tez 0.9.2**
   - Select **Hue 4.9.0**
   - Select **Spark 2.4.8**

   6.2. **Multiple master nodes (optional)**

   - It is selected if there is any crash in one master node, then the whole cluster will be crashed.
   - So for the backup purpose **Multiple master nodes** option will be selected.

- Here at this condition we are not selecting this option as we don't require it.
  Reference: Master Node Configuration

6.3. **AWS Glue Data Catalog settings (optional)**

6.3.1. **Use for Hive table metadata -** When enabled, specifies the AWS Glue Data Catalog as an external Hive metastore.
Reference**:** emr-hive-metastore-glue

6.3.2. **Use for Spark table metadata -** When enabled, can configure Spark SQL to use the AWS Glue Data Catalog as its metastore.
Reference: emr-spark-glue

It is used when we require a persistent metastore or a metastore shared by different clusters, services, applications, or AWS accounts
We are not using this as we don't require this now.

6.4. **Edit software settings**

- In this we can add any extra settings required in the cluster
  Reference: EMR Configuration

6.5. **Steps (optional)**

- This section describes the methods for submitting work to an Amazon EMR cluster. You can submit work to a cluster by adding steps or by interactively submitting Hadoop jobs to the master node.
  Reference: Steps

Then click on **Next** Button

6.6. **Hardware**

Hardware Configuration

In this we can specify the networking and hardware configuration as per the requirement for the cluster.

6.6.1. **Cluster Composition**

In this we can specify the configuration of the master, core and task nodes as an instance group or instance fleet. This choice applies to all nodes for the lifetime of the cluster.
Reference**:** Cluster Composition

Instance group configuration

- **Uniform instance groups**
  In this type all the selected instances will be from one family type only.
- **Instance fleets**
  In this type we can select instances from different types of families of instances as per our requirement.
  Reference: Configure Instance Fleet

For the selection of **Cluster Composition** we need to know basics of EC2 (i.e. Elastic Compute Cloud) instances.

Amazon EC2 provides a wide selection of instance types optimised to fit different use cases. Instance types comprise varying

combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. Each instance type includes one or more instance sizes, allowing you to scale your resources to the requirements of your target workload.

Reference : Amazon EC2 Instance Types

There are five types of EC2 instance families that are:

1. **General Purpose**

   General purpose instances provide a balance of compute, memory and networking resources, and can be used for a variety of diverse workloads. These instances are ideal for applications that use these resources in equal proportions such as web servers and code repositories.

   It includes Mac, T4g, T3, T3a, T2, M6g, M6i, M6a, M5, M5a, M5n, M5zn, M4, A1 groups in it.

2. **Compute Optimized**

   Compute Optimized instances are ideal for compute bound applications that benefit from high performance processors. Instances belonging to this family are well suited for batch processing workloads, media transcoding, high performance web servers, high performance computing (HPC), scientific modeling, dedicated gaming servers and ad server engines, machine learning inference and other compute intensive applications.

   It includes C7g, C6g, C6gn, C6i, C6a, Hpc6a, C5, C5a, C5n, C4 groups in it.

3. **Memory Optimized**

   Memory optimized instances are designed to deliver fast performance for workloads that process large data sets in memory.

   It includes R6g, R6i, R5, R5a, R5b, R5n, R4, x2gd, X2idn, X2iedn, X2iezn, X1e, X1, High Memory, z1d groups in it.

4. **Accelerated Computing**

   Accelerated computing instances use hardware accelerators, or co-processors, to perform functions, such as floating point number calculations, graphics processing, or data pattern matching, more efficiently than is possible in software running on CPUs.

   It includes P4, P3, P2, DL1, Trn1, Inf1, G5G5g, G4dn, G4ad, G3, F1, VT1 groups in it.

5. **Storage Optimized**

   Storage optimized instances are designed for workloads that require high, sequential read and write access to very large data sets on local

storage. They are optimized to deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications.
It includes Im4gn, Is4gen, I4i, I3, I3en, D2, D3, D3en, H1 groups in it

---

As we seen the different families of EC2 instances, now we can select required Cluster Composition as per the requirement.

So as per our requirement we need an instance which can provide a balance of compute, memory and networking resources, and can be used for a variety of diverse workloads.

Hence the instance we are selecting is of **Uniform instance groups.**

6.6.2. Networking

- Most clusters launch into a virtual network using Amazon Virtual Private Cloud (Amazon VPC).
- A VPC is an isolated virtual network within AWS that is logically isolated within your AWS account.
- You can configure aspects such as private IP address ranges, subnets, routing tables, and network gateways.
- Anyone can use a Virtual Private Cloud (VPC) to process sensitive data or connect to a private network.
- Means anyone working on an EMR cluster and has sensitive data for processing then that person can create his private VPC so that no one can threaten the data and has high security. And also can define who has access to the network.
- Also if your data source is located in a private network, it may be impractical or undesirable to upload that data to AWS for import into Amazon EMR, either because of the amount of data to transfer or because of the sensitive nature of the data. Instead, you can launch the cluster into a VPC and connect your data center to your VPC through a VPN connection, enabling the cluster to access resources on your internal network.
- **EC2 Subnet**
  You can launch Amazon EMR clusters in both public and private VPC subnets.
  This means you do not need internet connectivity to run an Amazon EMR cluster; however, you may need to configure network address translation (NAT) and VPN gateways to access services or resources located outside of the VPC, for example in a corporate intranet or public AWS service endpoints like AWS Key Management Service.

6.6.3. Cluster Nodes and Instances

- In this we can select EC2 Instance types as well as number of instances required as per requirements for Master node, core node and task node.

6.6.4. **Cluster scaling**

- When selected, cluster scaling enables EMR-managed scaling. Managed scaling automatically increases and decreases the number of instances in core and task nodes based on workload.
- If we **Enable Cluster Scaling** then we will get more options to set a limit of number of instances for cluster nodes.
  Reference: Cluster Scaling

6.6.5. **Auto-termination**

- When enabled, lets you attach an auto-termination policy to the cluster. In the policy, you specify the amount of idle time after which the cluster automatically shuts down.
  Reference: Auto Termination

6.6.6. **EBS Root Volume**

- When you launch an instance, the *root device volume* contains the image used to boot the instance.
- Reference: Amazon EC2 instance root device volume

Then click on **Next** Button

6.7. **General Cluster Settings**

6.7.1. **General Options**

- In this we can name our cluster
  Reference: Logging
- **Logging**
  - When logging is enabled, Amazon EMR writes detailed log data to the specified Amazon S3 folder. Logging can only be enabled at cluster creation. You can't change the setting later. Quick Options chooses a default Amazon S3 bucket. You can optionally specify your own bucket.
- **Log Encryption**
  - Amazon EMR supports encrypting log files using Customer-managed Customer master keys (CMKs) stored in AWS Key Management Service(KMS).
  - Amazon EMR automatically upload log files to Amazon S3 when logging and debugging is enabled With this new feature, you can associate Customer managed CMKs in AWS KMS when launching a cluster.
  - Amazon EMR will automatically encrypt logs using the Customer managed CMKs in AWS KMS.
    Reference: Log Encryption

- **Debugging**
  - The debugging tool allows you to more easily browse log files from the EMR console
  - When you enable debugging on a cluster, Amazon EMR archives the log files to Amazon S3 and then indexes those files.
  - You can then use the console to browse the step, job, task, and task-attempt logs for the cluster in an intuitive way.
    Reference: Debugging
- **Termination Protection**
  - If a cluster terminates because of a failure or by an accident, any data stored on the cluster is deleted, and the cluster state is set to TERMINATED_WITH_ERRORS.
  - If you enable termination protection, you can retrieve data from your cluster, and then remove termination protection and terminate the cluster.
    Reference: Termination Protection

6.7.2. **Tags**

- It can be convenient to categorize your AWS resources in different ways; for example, by purpose, owner, or environment.
- You can achieve this in Amazon EMR by assigning custom metadata to your Amazon EMR clusters using tags.
- A tag consists of a key and a value, both of which you define. For Amazon EMR, the cluster is the resource-level that you can tag. For example, you could define a set of tags for your account's clusters that helps you track each cluster's owner or identify a production cluster versus a testing cluster.
  Reference: Cluster Tags

6.7.3. **Additional Options**

- EMRFS consistent view allows EMR clusters to check for list and read-after-write consistency for Amazon S3 objects written by or synced with EMRFS,
  Reference: Consistent View
- **Custom AMI ID**
  - An Amazon Machine Image (AMI) defines the programs and settings that will be applied when you launch an EC2 instance. Once you have finished configuring the data, services, and applications on your ArcGIS Server instance, you can save your work as a custom AMI stored in Amazon EC2.
  - Custom AMI is useful in following conditions:

        - Pre-install applications and perform other customizations instead of using bootstrap actions. This can improve cluster start time and streamline the startup workflow.
        - Implement more sophisticated cluster and node configurations than bootstrap actions allow.

    Reference: Custom AMI

- **Bootstrap Actions**
    - Bootstrap actions are scripts that are executed during setup before Hadoop starts on every cluster node.
    - You can use them to install additional software and customize your applications.

    Reference: Bootstrap Actions

6.8. Security Options

6.8.1. EC2 key pair

- A key pair, consisting of a public key and a private key, is a set of security credentials that you use to prove your identity when connecting to an Amazon EC2 instance.
- Amazon EMR cluster nodes run on Amazon EC2 instances. You can connect to cluster nodes in the same way that you can connect to Amazon EC2 instances.
- When you create a cluster, you can specify the Amazon EC2 key pair that will be used for SSH connections to all cluster instances.

  Reference: EC2 key pair

6.8.2. Cluster visible to all IAM users in account

- When we select this checkbox it can allow all the IAM users connected with this account to perform actions.
- With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied.

  Reference: EMR work with IAM

6.8.3. Permissions

- Amazon EMR uses IAM service roles to perform actions on your behalf when provisioning cluster resources, running applications, dynamically scaling resources, and creating and running EMR Notebooks.
- IAM roles give the EMR service and applications running on an EMR cluster requisite permissions to call other AWS services.

  Reference: IAM service roles

6.8.4. Security Configuration

- In this we can choose a security configuration to specify the authentication and encryption options for your cluster. You can create one on the Security configuration page in the console.

-

6.8.5. **EC2 security groups**

- An EC2 security group acts as a virtual firewall for your cluster nodes to control inbound and outbound traffic.
- There are two types of security groups you can configure, EMR managed security groups and additional security groups.
  Reference: EMR Managed Security Groups, Additional Security Groups

Finally all the steps are completed for creating the cluster

Click on **Create Cluster** button

And the cluster will be created after all the processes and configurations.