

# What is Hadoop?

Apache Hadoop is an open source framework that is used to efficiently store and process large datasets ranging in size from gigabytes to petabytes of data. Instead of using one large computer to store and process the data, Hadoop allows clustering multiple computers to analyze massive datasets in parallel more quickly.

## What are the four main modules of Hadoop?

Hadoop consists of four main modules:

- Hadoop Distributed File System (HDFS) – A distributed file system that runs on standard or low-end hardware. HDFS provides better data throughput than traditional file systems, in addition to high fault tolerance and native support of large datasets.
- Yet Another Resource Negotiator (YARN) – Manages and monitors cluster nodes and resource usage. It schedules jobs and tasks.
- MapReduce – A framework that helps programs do the parallel computation on data. The map task takes input data and converts it into a dataset that can be computed in key value pairs. The output of the map task is consumed by reduce tasks to aggregate output and provide the desired result.
- Hadoop Common – Provides common Java libraries that can be used across all modules.

## How does Hadoop work?

Hadoop makes it easier to use all the storage and processing capacity in cluster servers, and to execute distributed processes against huge amounts of data. Hadoop provides the building blocks on which other services and applications can be built.

Applications that collect data in various formats can place data into the Hadoop

cluster by using an API operation to connect to the NameNode. The NameNode tracks the file directory structure and placement of “chunks” for each file, replicated across DataNodes. To run a job to query the data, provide a [MapReduce](#) job made up of many map and reduce tasks that run against the data in HDFS spread across the DataNodes. Map tasks run on each node against the input files supplied, and reducers run to aggregate and organize the final output.

## How has the Hadoop ecosystem evolved?

The Hadoop ecosystem has grown significantly over the years due to its extensibility. Today, the Hadoop ecosystem includes many tools and applications to help collect, store, process, analyze, and manage big data. Some of the most popular applications are:

- [Spark](#) – An open source, distributed processing system commonly used for big data workloads. Apache Spark uses in-memory caching and optimized execution for fast performance, and it supports general batch processing, streaming analytics, machine learning, graph databases, and ad hoc queries.
- [Presto](#) – An open source, distributed SQL query engine optimized for low-latency, ad-hoc analysis of data. It supports the ANSI SQL standard, including complex queries, aggregations, joins, and window functions. Presto can process data from multiple data sources including the Hadoop Distributed File System (HDFS) and Amazon S3.
- [Hive](#) – Allows users to leverage Hadoop MapReduce using a SQL interface, enabling analytics at a massive scale, in addition to distributed and fault-tolerant data warehousing.
- [HBase](#) – An open source, non-relational, versioned database that runs on top of Amazon S3 (using EMRFS) or the Hadoop Distributed File System (HDFS). HBase is a massively scalable, distributed big data store built for random, strictly consistent, real-time access for tables with billions of rows and millions of columns.
- Zeppelin – An interactive notebook that enables interactive data exploration.

# How can AWS support your Hadoop requirements?

[Amazon EMR](#) is a managed service that lets you process and analyze large datasets using the latest versions of [big data](#) processing frameworks such as Apache Hadoop, Spark, HBase, and Presto on fully customizable clusters.

- **Easy to use:** You can launch an Amazon EMR cluster in minutes. You don't need to worry about node provisioning, cluster setup, Hadoop configuration, or cluster tuning.
- **Low cost:** Amazon EMR pricing is simple and predictable: You pay an hourly rate for every instance hour you use and you can leverage Spot Instances for greater savings.
- **Elastic:** With Amazon EMR, you can provision one, hundreds, or thousands of compute instances to process data at any scale.
- **Transient:** You can use EMRFS to run clusters on-demand based on HDFS data stored persistently in Amazon S3. As jobs finish, you can shut down a cluster and have the data saved in [Amazon S3](#). You pay only for the compute time that the cluster is running.
- **Secure:** Amazon EMR uses all common security characteristics of AWS services:
  - Identity and Access Management (IAM) roles and policies to manage permissions.
  - Encryption in-transit and at-rest to help you protect your data and meet compliance standards, such as HIPAA.
  - Security groups to control inbound and outbound network traffic to your cluster nodes.
  - [AWS CloudTrail](#): Audit all Amazon EMR API calls made in your account to provide security analysis, resource change tracking, and compliance auditing.