Reference Guide: Sherlock Innovation Accelerator Platform Amazon Elastic Map Reduce (EMR)
# Table of Contents

Amazon Elastic MapReduce (EMR).................................................................................................................. 4

## Key Benefits........................................................................................................................................... 4

- Turnkey Deployment................................................................................................................................. 4
- Performance............................................................................................................................................... 4
- Availability................................................................................................................................................ 4
- Secure and Compliant............................................................................................................................... 4

## Features.................................................................................................................................................... 4

- Elastic ....................................................................................................................................................... 4
- Flexible Data Stores................................................................................................................................. 5
- Hadoop Tools .......................................................................................................................................... 6
- Additional Features ............................................................................................................................... 8

Sherlock Amazon EMR Platform................................................................................................................... 9

- Fast Performance ..................................................................................................................................... 9
- Develop Applications Quickly.................................................................................................................. 10
- Create Diverse Workflows....................................................................................................................... 10
- Integration with Amazon EMR Feature Set............................................................................................ 10
- Other Distributions with EMR ................................................................................................................ 10
- Build your own EMR Cluster.................................................................................................................. 11

Customer facing workflow ......................................................................................................................... 12

Example Use cases....................................................................................................................................... 13

- Log Analysis ............................................................................................................................................ 13
- Extract Transform Load (ETL) .................................................................................................................. 13
- Predictive Analytics ............................................................................................................................... 13
- Genomics ............................................................................................................................................... 13
- Real-time Analytics............................................................................................................................... 13
AMAZON ELASTIC MAP REDUCE (EMR)

Sherlock’s Innovation Accelerator Platforms provide organizations quick access to on-demand, elastic, secure, big data platforms to tackle large amounts of data. Sherlock’s first offering within this broad capability is the Amazon EMR platform, which is a turnkey HIPAA-compliant EMR platform configured with Apache Spark.

KEY BENEFITS

Turnkey Deployment
The Sherlock team will leverage AWS CloudFormation capability to describe and provision all the infrastructure resources to build a secure, segmented environment within Sherlock Cloud, and deploy the EMR cluster. The goal is to present to the user a platform that is ready for use, with all the security and compliance built in.

Performance
The Sherlock team will configure, size, and tune the EMR platform to operate at peak performance. Compute and storage will be isolated to help ensure a high level of performance for compute-intensive workloads.

Availability
Keep jobs moving and data available at all times. In addition to Hadoop’s innate redundancy model, the Sherlock team will keep the entire system running at peak efficiency.

Secure and Compliant
Sherlock’s AWS EMR platform is built according to NIST 800-53 requirements to comprehensively address the administrative and technical safeguards required by HIPAA.

FEATURES

Elastic

Amazon EMR enables quick and easy provisioning of capacity as needed, and can automatically add and remove capacity. This is a useful feature when dealing with variable or unpredictable processing requirements. For example, if the bulk of data processing occurs at night, users might only need 100 instances during the day but 500 instances at night. Alternatively, users might need significant amount of compute capacity for a short period of time. Amazon EMR can quickly provision hundreds or thousands of instances, automatically scale to match compute requirements, and shut the cluster down when the job is complete (to avoid paying for idle capacity).
There are two main options for adding or removing capacity:

**Deploy multiple clusters**: If more capacity is needed, EMR can easily launch a new cluster and terminate it when it is no longer needed. There is no limit to how many clusters one can have. Users may want to use multiple clusters if the use case supports multiple applications and/or user groups. For example, you can store your input data on Amazon S3 and launch one cluster for each application that needs to process the data. One cluster might be optimized for CPU, and a second cluster might be optimized for storage, etc.

**Resize a running cluster**: With Amazon EMR it is easy to automatically scale or manually resize a running cluster. Users may want to scale out a cluster to temporarily add more processing power, or scale down the cluster to save costs. For example, customers can add hundreds of instances to their clusters to support their data processing needs, and remove the additional instances when processing completes. When adding instances to clusters, EMR can start utilizing provisioned capacity as soon it becomes available. When scaling out, EMR will proactively choose idle nodes to reduce impact on running jobs.

**Flexible Data Stores**

With Amazon EMR, you can leverage multiple data stores, including Amazon S3, the Hadoop Distributed File System (HDFS), etc.

**Amazon S3**: Amazon S3 is a highly durable, scalable, secure, fast, and inexpensive storage service. With the EMR File System (EMRFS), Amazon EMR can efficiently and securely use Amazon S3 as an object store for Hadoop. Amazon EMR has made numerous improvements to Hadoop, allowing you to seamlessly process large amounts of data stored in Amazon S3. Also, EMRFS can enable consistent view to check for list and read-after-write consistency for objects in Amazon S3. EMRFS supports S3 server-side or S3 client-side encryption to process encrypted Amazon S3 objects.

After deploying a cluster, Amazon EMR streams the data from Amazon S3 to each instance in the cluster and begins processing it immediately. One advantage of storing data in Amazon S3
and processing it with Amazon EMR is data can use multiple clusters to process. For example, a customer might have a development cluster that is optimized for memory and another cluster that is optimized for CPU, both using the same input data set.

**Hadoop Distributed File System (HDFS):** HDFS is the Hadoop file system. Amazon EMR’s current topology groups its instances into 3 logical instance groups: Master Group, which runs the YARN Resource Manager and the HDFS Name Node Service; Core Group, which runs the HDFS DataNode Daemon and the YARN Node Manager service; and Task Group, which runs the YARN Node Manager service. Amazon EMR installs HDFS on the storage associated with the instances in the Core Group.

Each EC2 instance comes with a fixed amount of storage, referenced as "instance store", attached with the instance. You can also customize the storage on an instance by adding Amazon EBS volumes to an instance. Amazon EMR allows you to add General Purpose (SSD), Provisioned (SSD) and Magnetic volumes types. The EBS volumes added to an EMR cluster do not persist data after the cluster is shutdown. EMR will automatically clean up the volumes, once you terminate your cluster.

You can also enable complete encryption for HDFS using an Amazon EMR security configuration, or manually create HDFS encryption zones with the Hadoop Key Management Server.

**Hadoop Tools**

Amazon EMR supports powerful and proven Hadoop tools such as Hive, Pig, HBase, and Impala. Additionally, it can run distributed computing frameworks besides Hadoop MapReduce such as Spark or Presto using bootstrap actions. You can also use Hue and Zeppelin as GUIs for interacting with applications on your cluster.

**Spark** is an engine in the Hadoop ecosystem for fast processing for large data sets. It uses in-memory, fault-tolerant resilient distributed datasets (RDDs) and directed, acyclic graphs (DAGs) to define data transformations. Spark also includes Spark SQL, Spark Streaming, MLlib, and GraphX. Learn about Spark, and more about Spark on Amazon EMR.

**Hive** is an open source data warehouse and analytics package that runs on top of Hadoop. HiveQL, an SQL-based language, which allows users to structure, summarize, and query data, operates hive. HiveQL goes beyond standard SQL, adding first-class support for map/reduce functions and complex extensible user-defined data types like JSON and Thrift. This capability allows processing of complex and unstructured data sources such as text documents and log files. Hive allows user extensions via user-defined functions written in Java. Amazon EMR has made numerous improvements to Hive, including direct integration with Amazon DynamoDB and Amazon S3. For example, with Amazon EMR you can load table partitions automatically from Amazon S3, you can write data to tables in Amazon S3 without using temporary files, and you can access resources in Amazon S3 such as scripts for custom map/reduce operations and additional libraries. Learn more about Hive and Amazon EMR.
**Pig** is an open source analytics package that runs on top of Hadoop. Pig is operated by Pig Latin, an SQL-like language that allows users to structure, summarize, and query data. As well as SQL-like operations, Pig Latin also adds first-class support for map/reduce functions and complex extensible user defined data types. This capability allows processing of complex and unstructured data sources such as text documents and log files. Pig allows user extensions via user-defined functions written in Java. Amazon EMR has made numerous improvements to Pig, including the ability to use multiple file systems (normally Pig can only access one remote file system), the ability to load customer JARs and scripts from Amazon S3 (e.g. “REGISTER s3:///my-bucket/piggybank.jar”), and additional functionality for String and DateTime processing. Learn more about Pig and Amazon EMR.

**HBase** is an open source, non-relational, distributed database modeled after Google's BigTable. It was developed as part of Apache Software Foundation's Hadoop project and runs on top of Hadoop Distributed File System (HDFS) to provide BigTable-like capabilities for Hadoop. HBase provides you a fault-tolerant, efficient way of storing large quantities of sparse data using column-based compression and storage. In addition, HBase provides fast lookup of data because it caches data in-memory. HBase is optimized for sequential write operations, and it is highly efficient for batch inserts, updates, and deletes. HBase works seamlessly with Hadoop, sharing its file system and serving as a direct input and output to Hadoop jobs. HBase also integrates with Apache Hive, enabling SQL-like queries over HBase tables, joins with Hive-based tables, and support for Java Database Connectivity (JDBC). With Amazon EMR, you can use Amazon S3 as a data store for Apache HBase, enabling you to lower costs and reduce operational complexity. If you use HDFS as a data store, you can back up HBase to Amazon S3 and you can restore from a previously created backup. Learn more about HBase and Amazon EMR.

**Impala** is an open source tool in the Hadoop ecosystem for interactive, ad hoc querying using SQL syntax. Instead of using MapReduce, it leverages a massively parallel processing (MPP) engine similar to that found in traditional relational database management systems (RDBMS). Learn more about Impala and Amazon EMR.

**Hue** is an open source user interface for Hadoop that makes it easier to run and develop Hive queries, manage files in HDFS, run and develop Pig scripts, and manage tables. Hue on Amazon EMR also integrates with Amazon S3, so you can query directly against Amazon S3 and easily transfer files between HDFS and Amazon S3. Learn more about Hue and Amazon EMR.

**Presto** is 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. Learn what is Presto, and more about Presto on Amazon EMR.

**Zeppelin** is an open source GUI that creates interactive and collaborative notebooks for data exploration using Spark. You can use Scala, Python, SQL (using Spark SQL), or HiveQL to manipulate data and quickly visualize results. Zeppelin notebooks can be shared among several users, and visualizations can be published to external dashboards. Learn more about Zeppelin on Amazon EMR.
Oozie is a workflow scheduler for Hadoop, where you can create Directed Acyclic Graphs (DAGs) of actions. Also, you can easily trigger your Hadoop workflows by actions or time. Learn more about Oozie on Amazon EMR.

Tez is an execution framework on Hadoop YARN that offers fast performance from optimized query plans and enhanced resource management. You can use Tez with Apache Hive and Apache Pig instead of Hadoop MapReduce, and you can visualize execution plans using the Tez UI. Learn more about Tez on Amazon EMR.

Flink is a streaming dataflow engine that makes it easy to run real-time stream processing on high-throughput data sources. It supports event time semantics for out of order events, exactly-once semantics, backpressure control, and APIs optimized for writing both streaming and batch applications. Learn more about Flink on EMR.

Other: Amazon EMR also supports a variety of other popular applications and tools, such as R, Apache MXNet (deep learning), Mahout (machine learning), Ganglia (monitoring), Accumulo (secure NoSQL database), Sqoop (relational database connector), HCatalog (table and storage management), and more. The Amazon EMR team maintains an open source repository of bootstrap actions that can be used to install additional software, configure your cluster, or serve as examples for writing your own bootstrap actions.

Additional Features

Hadoop Streaming: Hadoop Streaming is a utility that comes with Hadoop that enables you to develop MapReduce executables in languages other than Java. Streaming is implemented in the form of a JAR file. Learn more about Hadoop Streaming with Amazon EMR.

Custom Jar: Write a Java program, compile against the version of Hadoop you want to use, and upload to Amazon S3. You can then submit Hadoop jobs to the cluster using the Hadoop JobClient interface. Learn more about Custom Jar processing with Amazon EMR.
SHERLOCK AMAZON EMR PLATFORM

Sherlock’s current service offering includes the latest version of EMR configured with Spark. This EMR cluster will be a short-term, managed cluster resource (available for 48 hours or less) that will be controlled and executed by the Sherlock team. It will not provide any monitoring features and will typically have a single user job/process running during that time. Individual customers will have their own S3 bucket for each EMR cluster. Each EMR cluster will have its own KMS Key for encryption and each S3 bucket will have its own AWS Access Key and AWS Secret Key to access the bucket.

The following diagram provides a high level architectural view of the various components deployed.

The following tasks are executed as part of the turnkey deployment:

- Creation of a segmented project environment within Sherlock Cloud
- S3 bucket for customer data and application
- Creation of managed file transfer capability for user to move data and files onto the S3 bucket
- KMS key for encryption at rest on S3 and the EMR cluster volumes
- Remote desktop server for the customer access
- Configuration of the EMR cluster with Apache Spark

Some of the key benefits of this Spark/EMR platform include:

**Fast Performance**

By using a directed acyclic graph (DAG) execution engine, Apache Spark can create efficient query plans for data transformations. Apache Spark also stores input, output, and intermediate
Develop Applications Quickly
Apache Spark natively supports Java, Scala, and Python, giving users a variety of languages for building their applications. Also, users can submit SQL or HiveQL queries to Apache Spark using the Spark SQL module. In addition to running applications, users can use the Apache Spark API interactively with Python or Scala directly in the Apache Spark shell on their cluster. They can also leverage Zeppelin to create interactive and collaborative notebooks for data exploration and visualization. Additionally, they can tune and debug your workloads using Spark application history in the Amazon EMR console or the native Spark UI and history server on their cluster.

Create Diverse Workflows
Apache Spark includes several libraries to help build applications for machine learning (MLlib), stream processing (Spark Streaming), and graph processing (GraphX). These libraries are tightly integrated in the Apache Spark ecosystem, and they can be leveraged out of the box to address a variety of use cases. Additionally, users can use deep learning frameworks like Apache MXNet with your Spark applications.

Integration with Amazon EMR Feature Set
Submit Apache Spark jobs with the Amazon EMR Step API, use Apache Spark with EMRFS to directly access data in Amazon S3, save costs using Amazon EC2 Spot capacity, use Auto Scaling to dynamically add and remove capacity, and launch long-running or ephemeral clusters to match your workload. Users can also easily configure Spark encryption and authentication with Kerberos using an Amazon EMR security configuration. Additionally, users can use the AWS Glue Data Catalog to store Spark SQL table metadata, or use Amazon SageMaker with your Spark machine learning pipelines. Amazon EMR installs and manages Apache Spark on Hadoop YARN, and you can also add other Hadoop ecosystem applications on your cluster.

Other Distributions with EMR
Users will have the option to select other packages as part of the EMR deployment as shown in the diagram below.
Presto on Amazon EMR comes with Presto 0.206 with Hadoop 2.8.4 HDFS and Hive 2.3.3 Metastore. Presto uses a custom query execution engine with operators designed to support SQL semantics. Different from Hive/MapReduce, Presto executes queries in memory, pipelined across the network between stages, thus avoiding unnecessary I/O. Presto supports the ANSI SQL standard, which makes it easy for data analysts and developers to query both structured and unstructured data at scale. Currently, Presto supports a wide variety of SQL functionality, including complex queries, aggregations, joins, and window functions.

Hbase on Amazon EMR comes with Hbase 1.4.6 with Ganglia 3.7.2, Hadoop 2.84, Hive 2.3.3, Hue 4.2.0, Phoenix 4.14.0 and ZooKeeper 3.4.12. Apache HBase is a massively scalable, distributed big data store in the Apache Hadoop ecosystem. It is an open-source, non-relational, versioned database, and it is built for random, strictly consistent realtime access for tables with billions of rows and millions of columns. Apache Phoenix integrates with Apache HBase for low-latency SQL access over Apache HBase tables and secondary indexing for increased performance.

Core Hadoop on Amazon EMR comes with Hadoop 2.8.4 with Ganglia 3.7.2, Hive 2.3.3, Hue 4.2.0 Mahout 0.13.0, Pig 0.17.0 and Tez 0.8.4. Apache™ Hadoop® is an open source software project that can be used to efficiently process large datasets. Instead of using one large computer to process and store the data, Hadoop allows clustering commodity hardware together to analyze massive data sets in parallel. There are many applications and execution engines in the Hadoop ecosystem, providing a variety of tools to match the needs of your analytics workloads.

**Build your own EMR Cluster**

Users will also have the option of customizing their EMR cluster with a number of software libraries.
1. The Sherlock team will provide a list of questions to the customers, which will help determine the numbers of nodes required for the cluster, and the kind of instances needed for the cluster.

2. The customer will identify what packages they would like included in the deployment of their EMR cluster.

3. Sherlock team will provide customer a way to transfer customer data and the processes/jobs/executables to the S3 bucket.

4. Sherlock team will create an encrypted S3 bucket where user data and code will be stored.

5. User will be able to test their code and update/modify as needed within the environment.

6. Sherlock team will create a step execution cluster. With step execution, EMR will create the cluster, execute added steps (process) and terminate when it’s done.

7. The customer won’t have any access on the cluster. The EMR cluster will typically use only one S3 bucket, but is not limited to one bucket. S3 bucket will store data, process, logs and process result/outcome.

8. Processed data will be stored on the S3 bucket for user to access.
Here are some example use cases that can run on the EMR platform.

**Log Analysis**
Amazon EMR can be used to process logs generated by web and mobile applications. Amazon EMR helps customers turn petabytes of un-structured or semi-structured data into useful insights about their applications or users.

**Extract Transform Load (ETL)**
Amazon EMR can be used to quickly and cost-effectively perform data transformation workloads (ETL) such as - sort, aggregate, and join - on large datasets.

**Predictive Analytics**
Apache Spark on Amazon EMR includes MLlib for scalable machine learning algorithms. By storing datasets in-memory, Spark can provide great performance for common machine learning workloads.

**Genomics**
Amazon EMR can be used to process vast amounts of genomic data and other large scientific data sets quickly and efficiently.

**Real-time Analytics**
Consume and process real-time data Spark Streaming on Amazon EMR. Perform streaming analytics in a fault-tolerant way and write results to Amazon S3.