House Rules

- Welcome! Thank you for joining us for **Taming Big Data with Elastic MapReduce (EMR) on AWS.**
- Please feel free to pop your questions in the Chime Chat and our Engineers will answer them as we go along.
- Our Presenter will also answer some of the questions at different points.
- Please see Github link where you will find some of the information pertaining to our event tonight: https://awsbigdatavls.co.za
- And please see the document for Launching your EMR Cluster, using the Hash Link which Nikki sent on email to you
- If you did not receive a Hash Link, please check your Junk-Mail or Spam

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

About AWS Premium Support

AWS Premium Support is one-on-one, fast-response support from experienced technical support engineers. The service helps customers use AWS products and features. The **Big Data Cloud Support Engineering** team is one of our teams within Premium Support. Our Engineers support our customers around the globe, along with becoming Subject Matter Experts in their respective services, provide and receive training from global partners and have the opportunity to contribute to the development of our services.

Ops Manager - Brian

The purpose of this event is to give you a chance to learn from our Big Data Cloud Support Engineers, learn more about one of our big data services and stand a chance to participate in our recruitment process, should you be successful in the completion of technical challenge on Day 3.

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Cluster Setup

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Getting started with Amazon EMR Day2

(Introduction to Hive and Spark)

Feb 19th August 2021

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Introduction

- Instructors

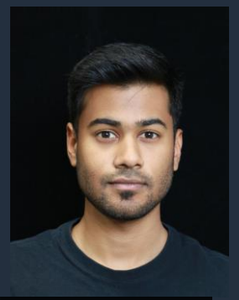
Ridick

Emmanuel

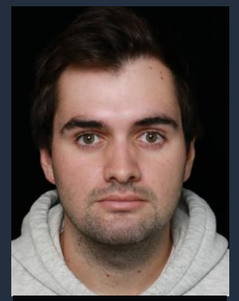
© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Introduction

Team

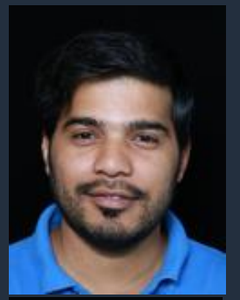

Vidushen


Patrick


Peter


Emmanuel

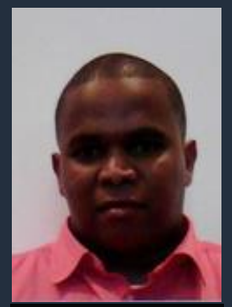

Umesh


Amit


David


Ridick


Andrew


Neil


Riyaad

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Agenda

- Hive Overview
- Hive Architecture
- Benefits of hive
- Data types
- HQL
- Hive Lab
- Spark Overview
- Benefits of spark
- Spark Deployment options
- Monitoring and debugging
- Spark SQL
- Spark Lab
- Questions

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Hive Overview

What Is Hive

- Apache Hive is an open-source, distributed, fault-tolerant system that provides data warehouse-like query capabilities. It runs on top of Hadoop.

- It enables users to read, write, and manage petabytes of data using a SQL-like interface. Hive scripts use an SQL-like language called Hive QL (query language)

- Hive is designed to enable easy data summarization, ad-hoc querying and analysis of large volumes of data.

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Hive Overview

What Hive Is NOT

- Hive is not designed for online transaction processing. It is best used for traditional data warehousing tasks.

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Benefits of Hive

- MR is Java, so to access data on Hadoop you have to know Java, specifically how to write Java MR code.
- Hive provides standard SQL functionality, including many of the later 2003 and 2011 features for analytics.
- Tools to enable easy access to data via SQL, thus enabling data warehousing tasks such as extract/transform/load (ETL), reporting, and data analysis
- A mechanism to impose structure on a variety of data formats

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Question-I

What are different execution engines supported by hive ?

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Benefits of Hive

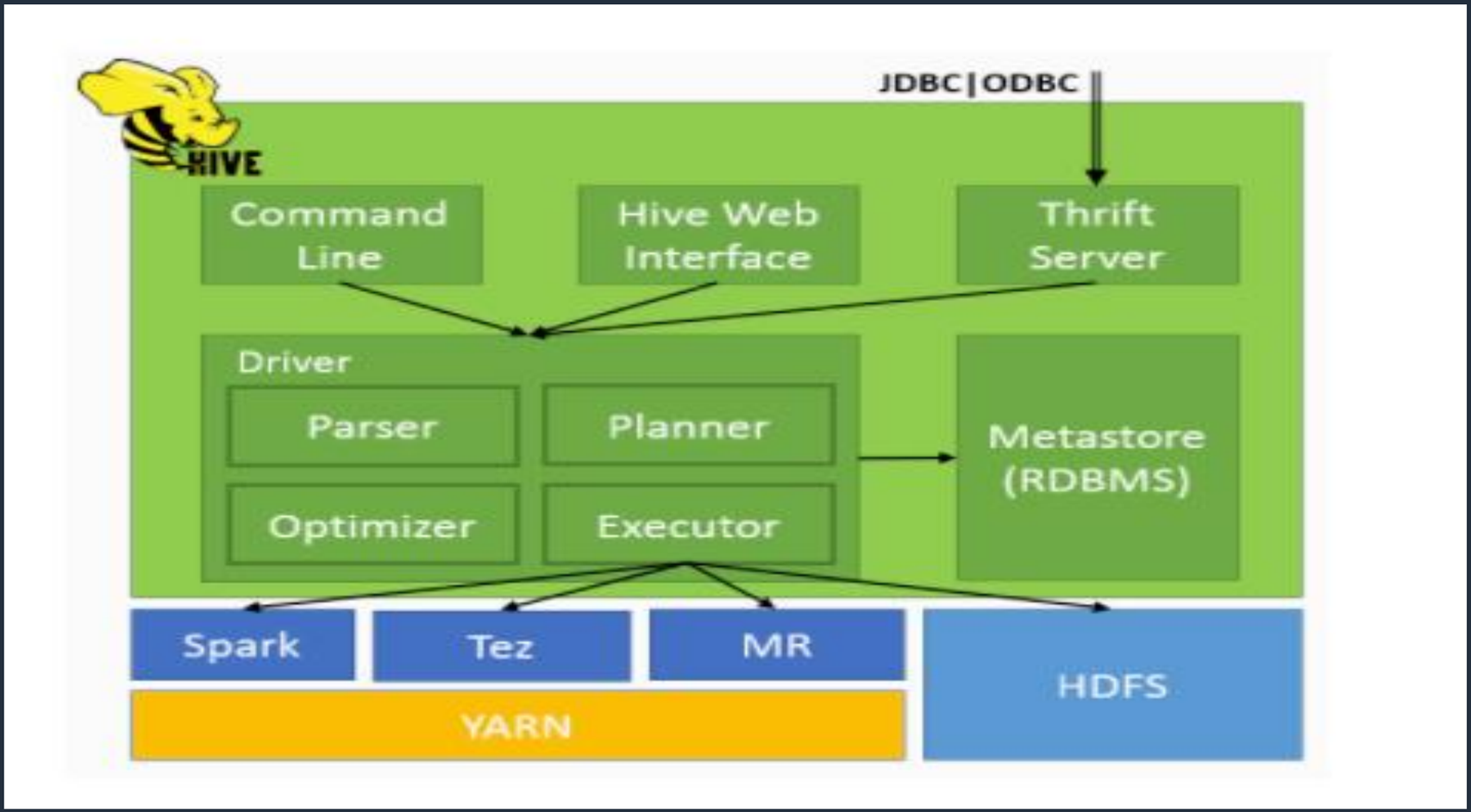
- Hive users have a choice of 3 runtimes when executing SQL queries.
    - Query execution using Apache Hadoop MapReduce, Apache Tez or Apache Spark frameworks.

- Hive's query response time is typically much faster than others on the same volume of big datasets

- Hive supports ad hoc querying data on HDFS or in other data storage systems such as Apache HBase

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Hive Architecture



Source: Oreilly

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Data Units

In the order of granularity - Hive data is organized into:

- Databases:
  Namespaces function to avoid naming conflicts for tables, views, partitions, columns, and so on.

- Tables:
  Homogeneous units of data which have the same schema.

- Partitions:
  Each Table can have one or more partition Keys which determines how the data is stored.

- Buckets:
  Data in each partition may in turn be divided into Buckets based on the value of a hash function of some column of the Table.

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Types of Hive tables

Managed tables

- A managed table is stored under the ‘hive.metastore.warehouse.dir’ path property, by default in a folder path similar to /user/hive/warehouse/databasename.db/tablename/
- If a managed table or partition is dropped, the data and metadata associated with that table or partition are deleted.

External tables

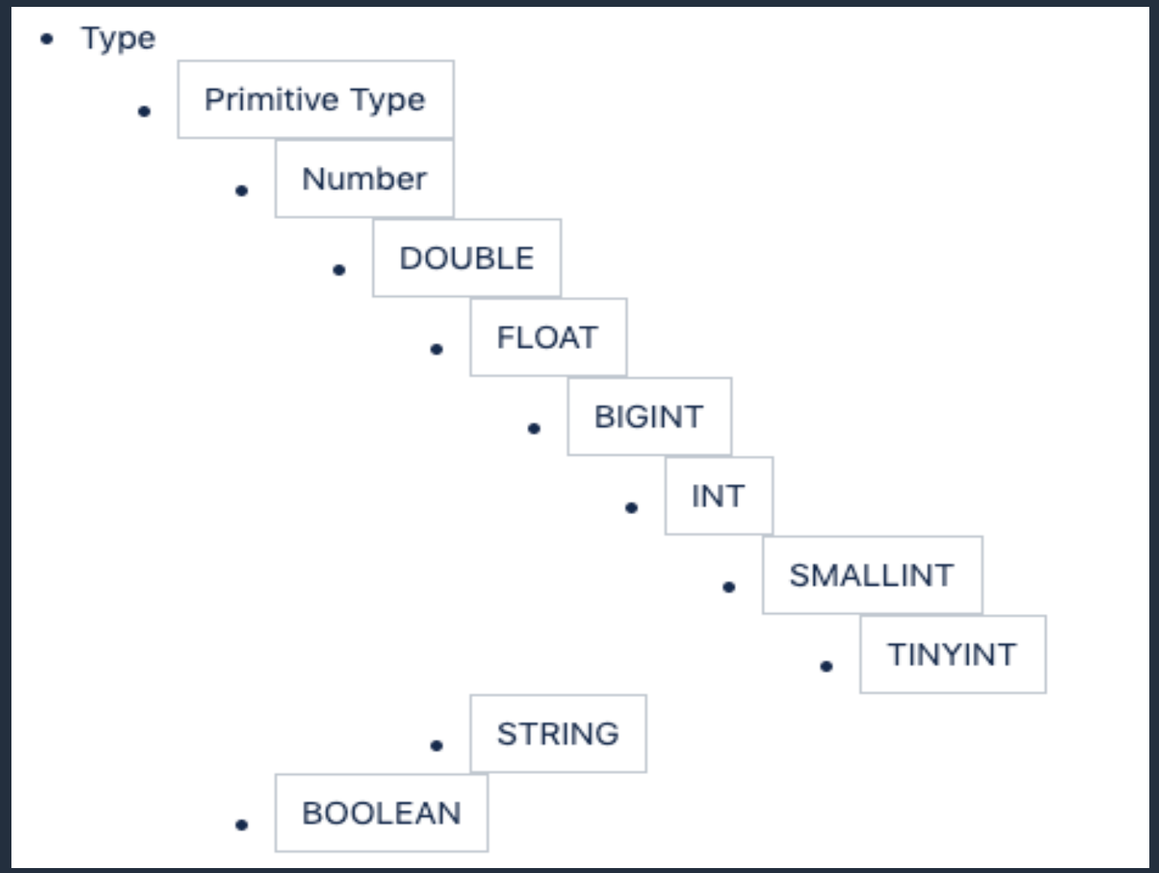
- An external table describes the metadata / schema on external files.
- External tables can access data stored in sources such as s3 or remote HDFS locations
- Use external tables when files are already present or in remote locations, and the files should remain even if the table is dropped.

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Data Types

Hive supports primitive and complex data types



PRIMITIVE TYPES

STRUCTS
EX.
STRUCT {a INT; b INT}

MAPS(key-value tuples)
EX.
'group' -> gid

ARRAYS (indexable lists)
EX.
['a', 'b', 'c']

COMPLEX TYPES

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

HQL – HIVE QUERY LANGUAGE Capabilities

Hive's SQL provides the basic SQL operations. These operations work on tables or partitions. Some of these operations are:

- Ability to filter rows from a table using a WHERE clause.
- Ability to select certain columns from the table using a SELECT clause.
- Ability to store the results of a query into another table.
- Ability to store the results of a query in a hadoop dfs directory.
- Ability to manage tables and partitions (create, drop and alter).

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Question-II

What is main difference between MR and TEZ execution engine ?

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

MR vs TEZ

Under-the-hood Hive queries still run as MapReduce jobs.

MapReduce is a mature framework that is proven at large scales. However queries using it may experience higher latencies (tens of seconds), even over small datasets.

Apache Tez is designed for interactive query, and has substantially reduced overheads versus MapReduce. MapReduce is still supported for Hive execution but TEZ is now the default engine when running Hive jobs in Hadoop.

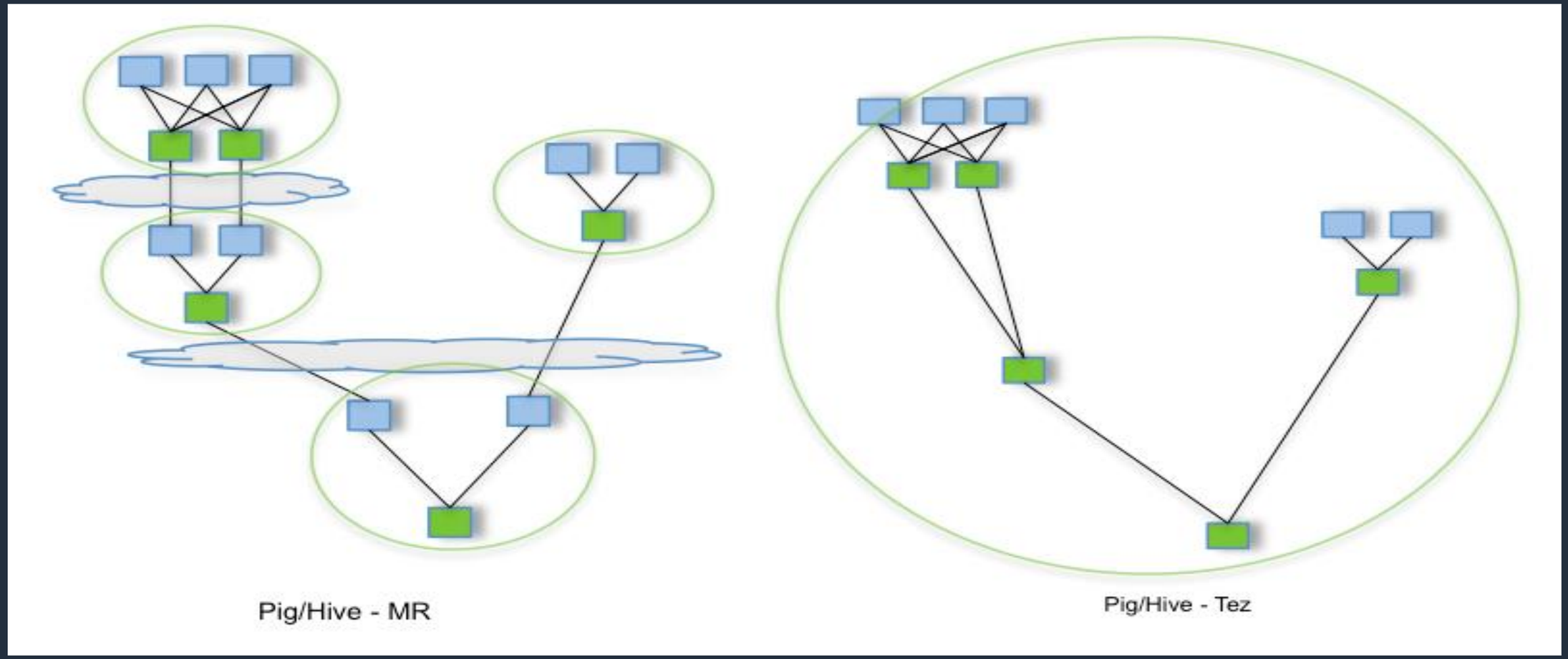
TEZ avoids disk IO by avoiding expensive shuffle and sorts while leveraging more efficient map side joins.

TEZ also utilizes a cost-based optimizer, which helps produce faster execution plans.

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

MR vs TEZ

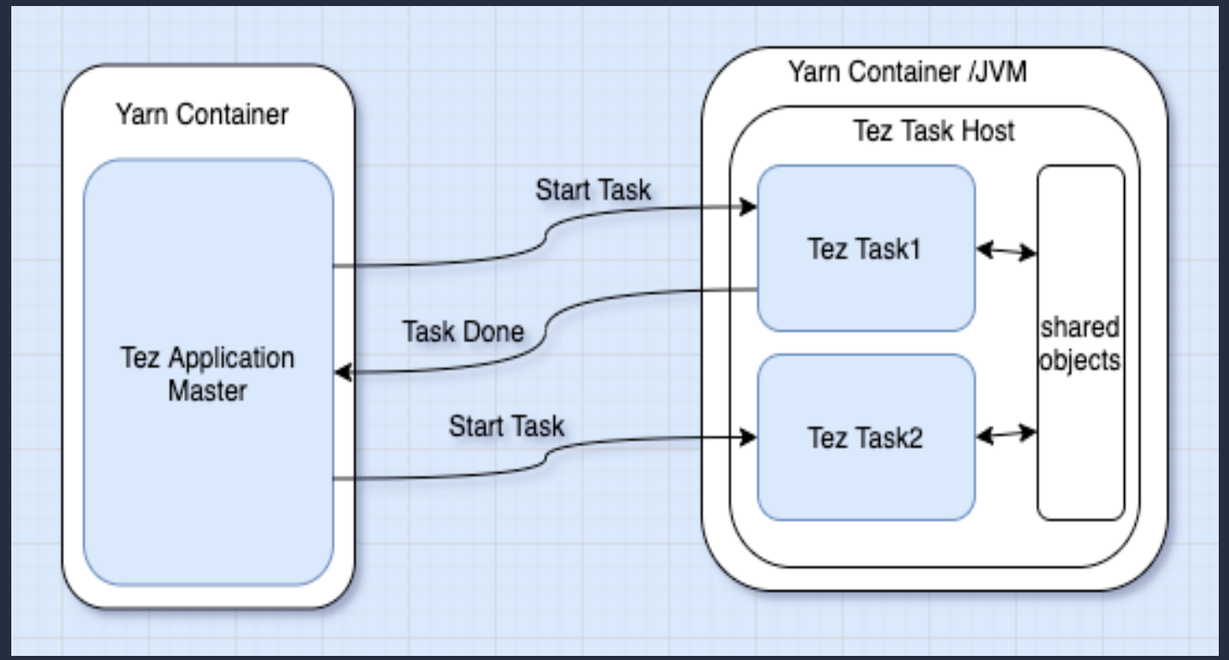
Tez allows Apache Hive to run a complex DAG of tasks, it can be used to process data, that earlier took multiple MR jobs, now in a single Tez job as shown below.



© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

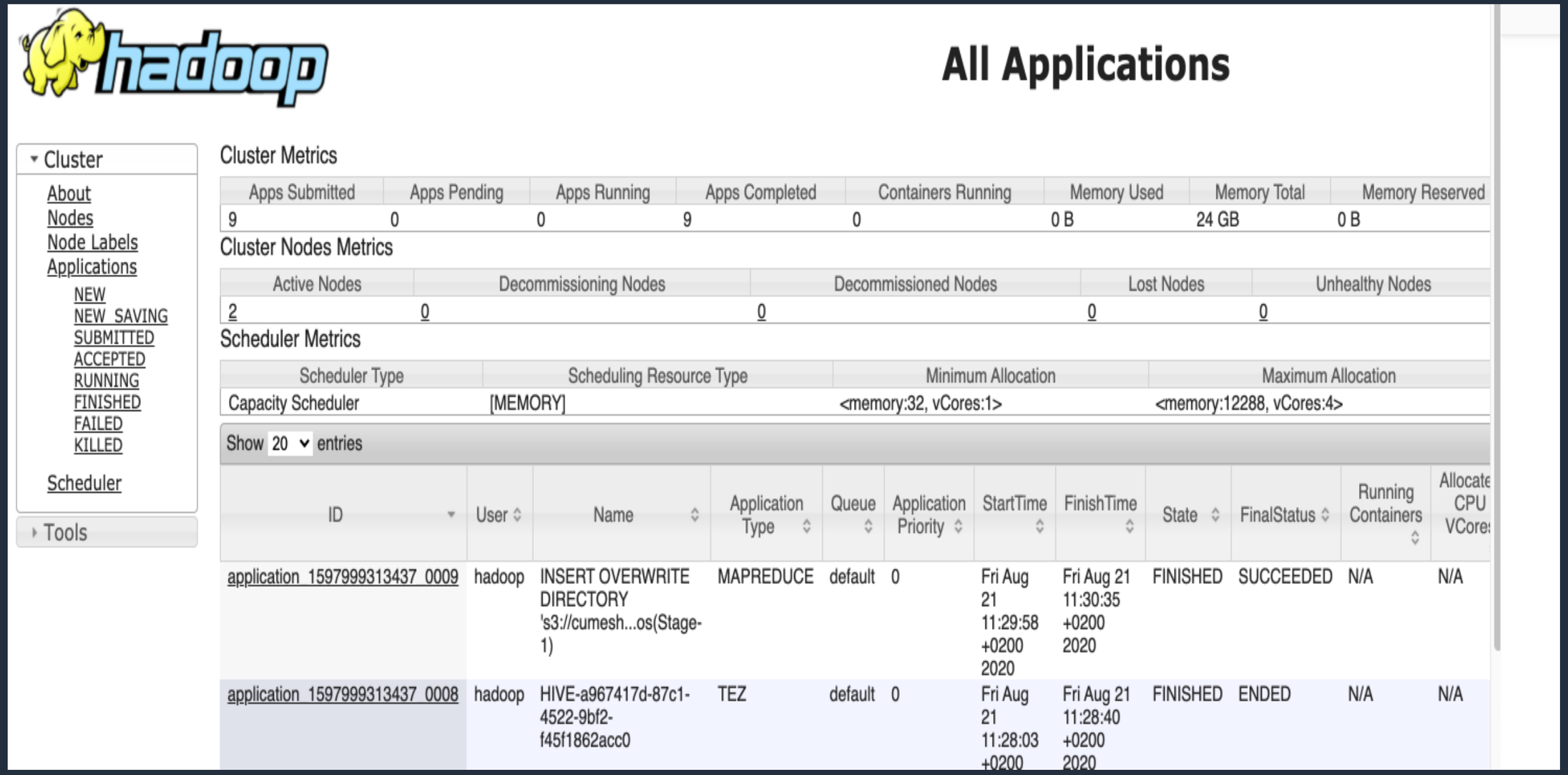
TEZ - Container Reuse

- Reuse YARN containers /JVMs to launch new tasks
- Reduce scheduling and launching delays
- Shared in-memory data across tasks
- JVM JIT friendly execution



© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Hive using MR



hadoop

All Applications

- Cluster
  - About
  - Nodes
  - Node Labels
  - Applications
    - NEW
    - NEW_SAVING
    - SUBMITTED
    - ACCEPTED
    - RUNNING
    - FINISHED
    - FAILED
    - KILLED
  - Scheduler
- Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved
9	0	0	9	0	0 B	24 GB	0 B

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes
2	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:32, vCores:1>	<memory:12288, vCores:4>

Show 20 entries

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocate CPU VCore
application_1597999313437_0009	hadoop	INSERT OVERWRITE DIRECTORY 's3://cumesh...os(Stage-1)	MAPREDUCE	default	0	Fri Aug 21 11:29:58 +0200 2020	Fri Aug 21 11:30:35 +0200 2020	FINISHED	SUCCEEDED	N/A	N/A
application_1597999313437_0008	hadoop	HIVE-a967417d-87c1-4522-9bf2-f45f1862acc0	TEZ	default	0	Fri Aug 21 11:28:03 +0200	Fri Aug 21 11:28:40 +0200 2020	FINISHED	ENDED	N/A	N/A

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

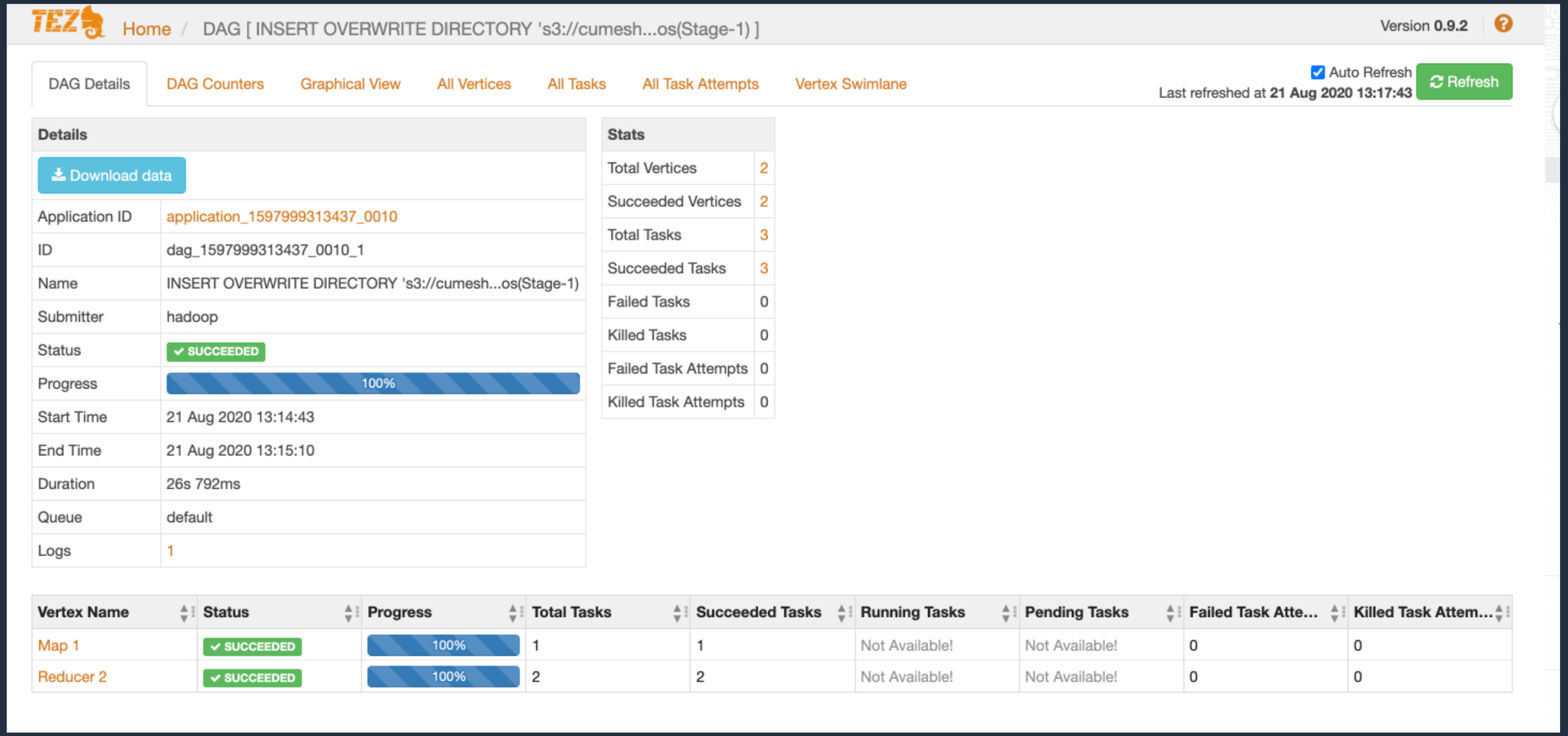
Hive using Tez

```
[hadoop@ip-172-31-16-105 ~]$ hive -f s3://us-east-1.elasticmapreduce.samples/cloudfront/code/Hive_CloudFront.q \
> -d INPUT=s3://us-east-1elasticmapreduce.samples -d OUTPUT=s3://cumesh-hive/tez-test/ \
> -hiveconf hive.execution.engine=tez

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: true
OK
Time taken: 0.918 seconds
Query ID = hadoop_20200821111439_bb4f962c-0e5c-4d01-9761-cc4a4ad54c93
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1597999313437_0010)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 26.19 s
--------------------------------------------------------------------------------
Moving data to directory s3://cumesh-hive/tez-test/os_requests
OK
Time taken: 31.273 seconds
[hadoop@ip-172-31-16-105 ~]$
```

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Hive using Tez



© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Hive Lab

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Lab-I

1) Login to EMR cluster master node using Session Manger

2) Run hive query using MR execution engine

3) open the YARN Web UI and observe the map and reduce tasks

4) Run hive query using Tez execution engine

5) Open the Tez UI and observe the DAG

6) Notice query run time difference between MR and Tez execution engine

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Question-III

Which execution engine is better for running interactive hive queries ?

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Lab-II

1) Login to EMR cluster master node using Session Manager

2) Download the Hive_CloudFront.q on EMR master node

3) Print the content of the file using cat command

4) Try to read the content of file, understand the definition of the table

5) Log in to hive-cli by typing command *hive* on command line on EMR master node

6) create new external table with name test_extrenal_hive_tbl on s3 data location

7) Run the query mentioned at the end of Hive_CloudFront.q file

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Question-IV

What is external hive table? Is it possible to create external hive table on top of data stored in hdfs?

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Break

Will resume in 15 minutes

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Spark Overview

What is Apache Spark ?

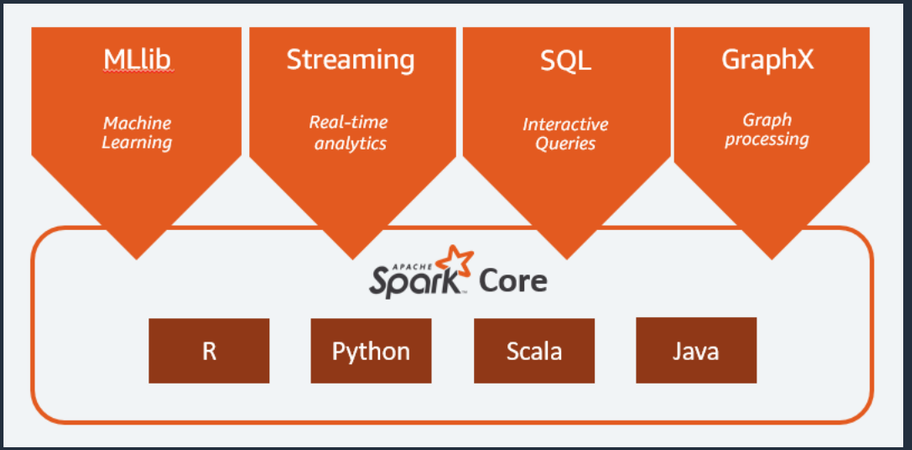
Apache Spark is a unified analytics engine for large-scale data processing.

It provides high-level APIs in Java, Scala, Python and R, and an optimized engine that supports general execution graphs.

The Spark framework includes:

- Spark Core as the foundation for the platform
- Spark SQL for interactive queries
- Spark Streaming for real-time analytics
- Spark MLlib for machine learning
- Spark GraphX for graph processing

Spark Components



© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

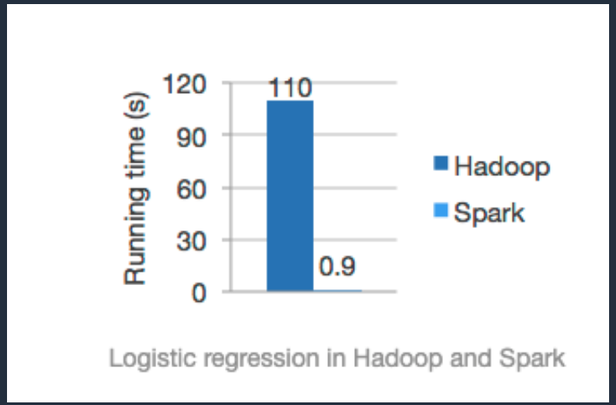
Benefits of Spark

- Speed
  Run workloads 100x faster.

- Ease of Use
  Write applications quickly in Java, Scala, Python, R, and SQL.

- Generality
  Combine SQL, streaming, and complex analytics.

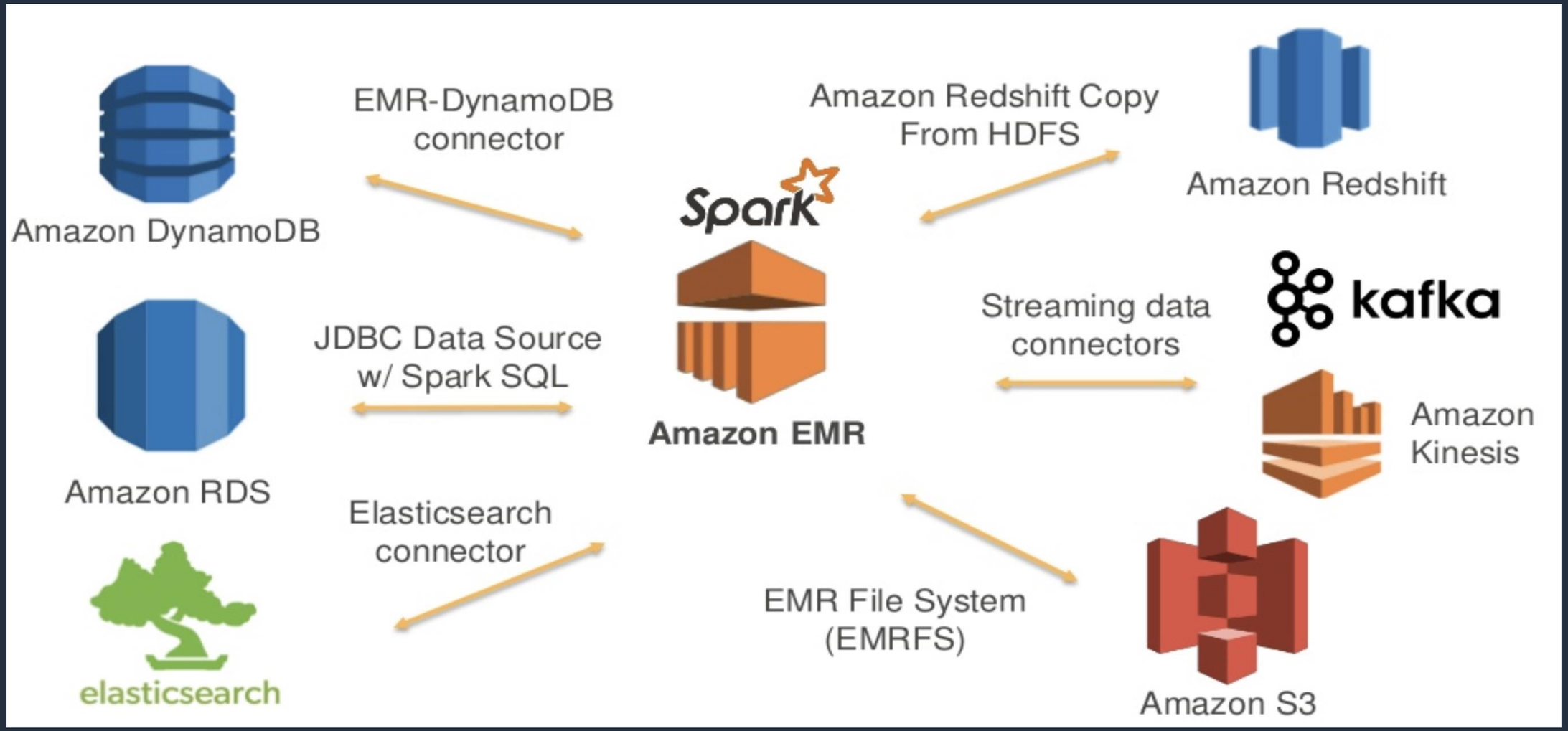
- Runs Everywhere
  Spark runs on Hadoop, Apache Mesos, Kubernetes, standalone, or in the cloud. It can access diverse data sources.



Spark performance

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Many Storage Layers to choose from



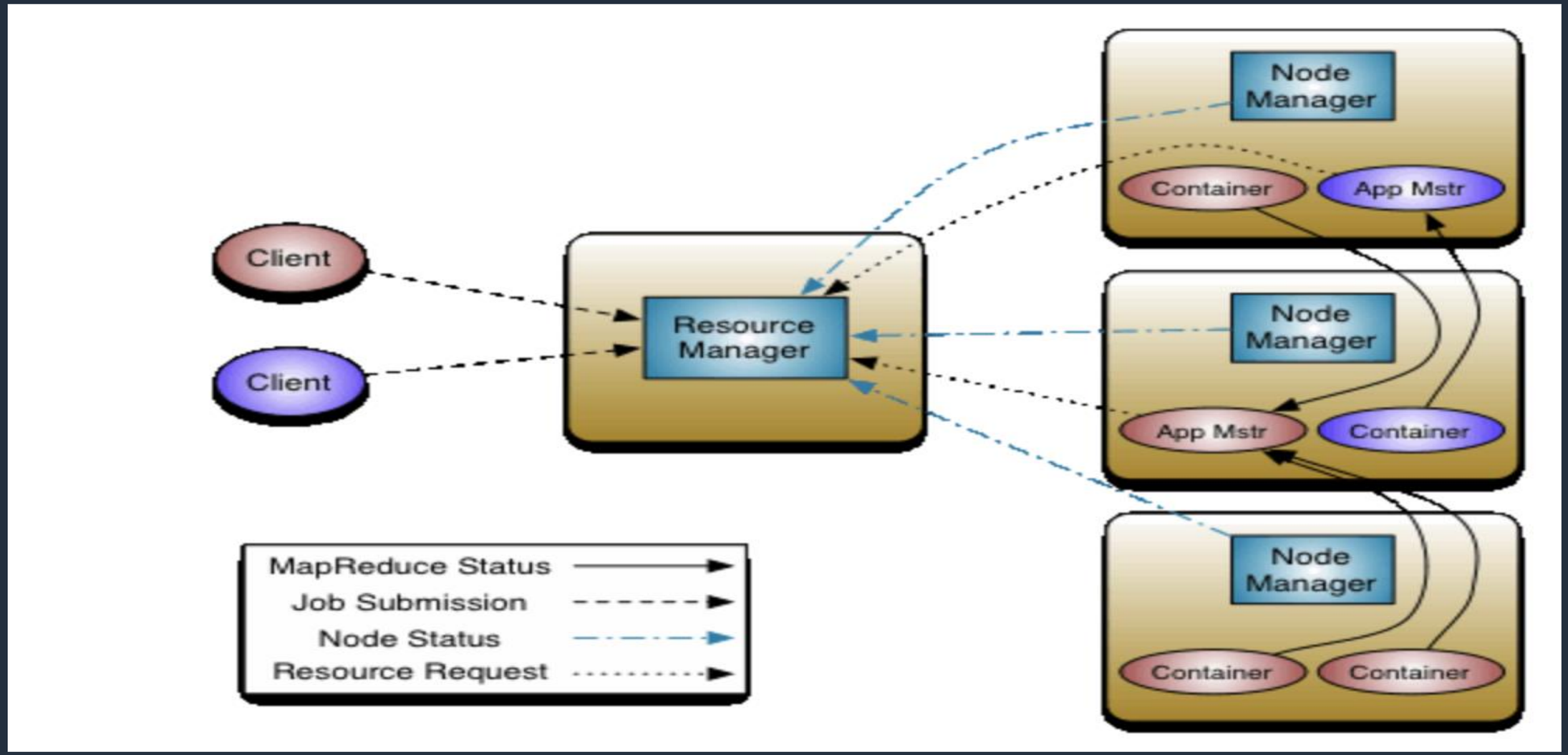
© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Spark Deployment options

- Standalone
- Apache Mesos
- Hadoop YARN
- Kubernetes

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

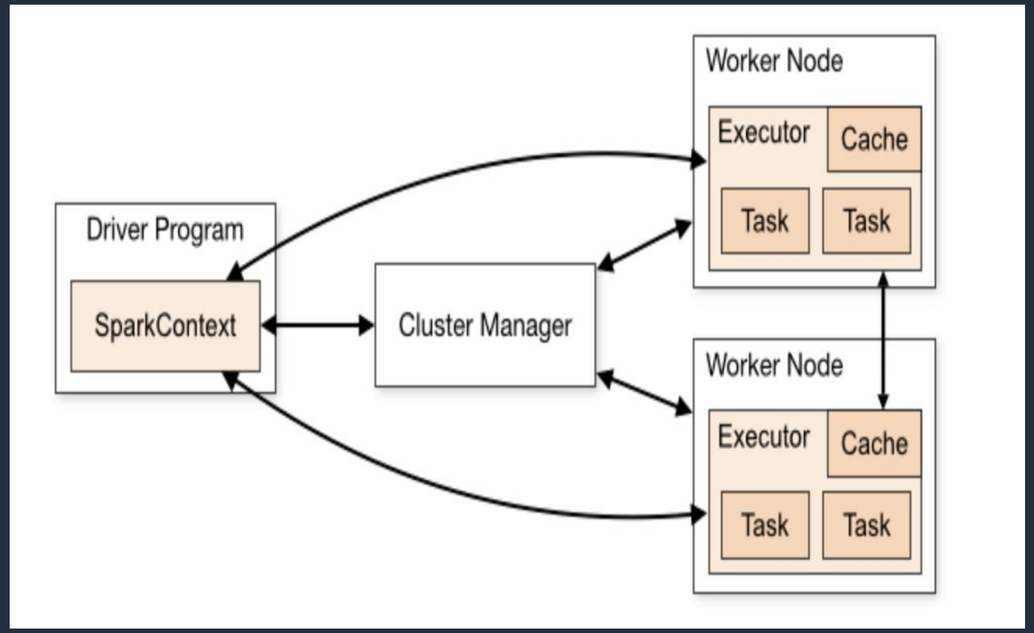
Quick Recap-YARN Architecture



© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Spark Terminology

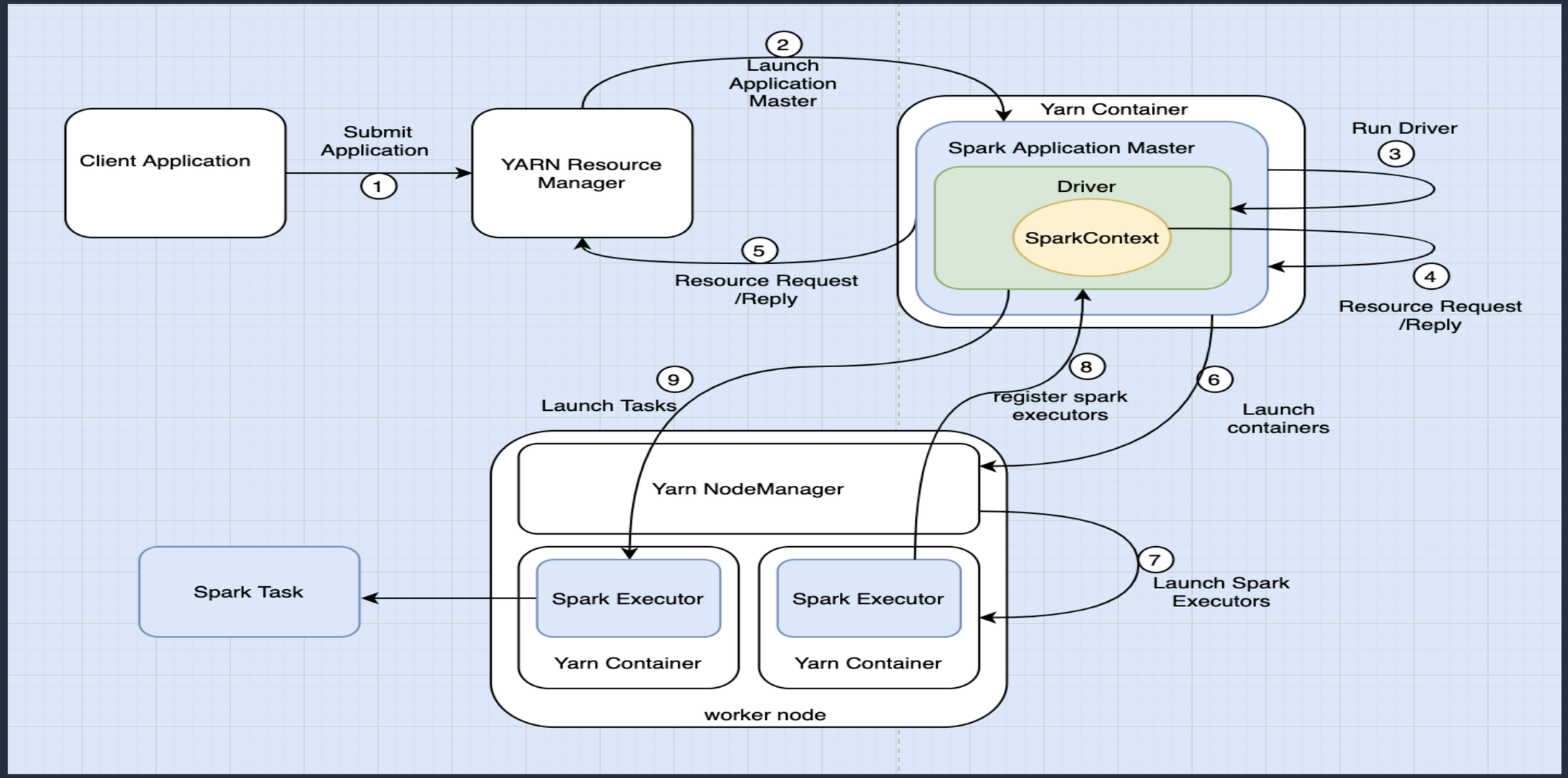
- Driver program - The process running the main() function of the application and creating the SparkContext
- Cluster manager - An external service for acquiring resources on the cluster (e.g. standalone manager, Mesos, YARN)
- Worker node - Any node that can run application code in the cluster
- Executor - A process launched for an application on a worker node, that runs tasks and keeps data in memory or disk storage across them.



- Task - A unit of work that will be sent to one executor
- Job - A parallel computation consisting of multiple tasks that gets spawned in response to a Spark action (e.g. save, collect)
- Stage - Each job gets divided into smaller sets of tasks called stages that depend on each other (similar to the map and reduce stages in MapReduce)

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Spark Execution Model



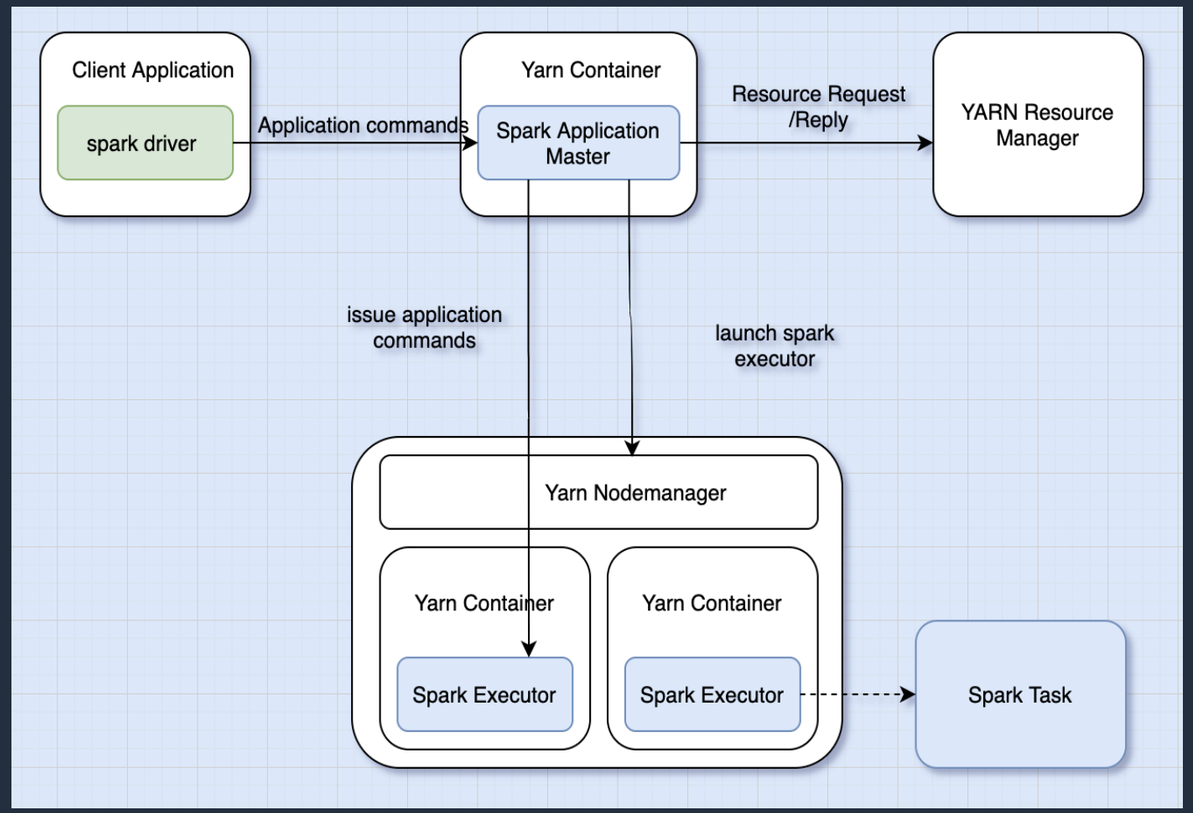
© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Question-II

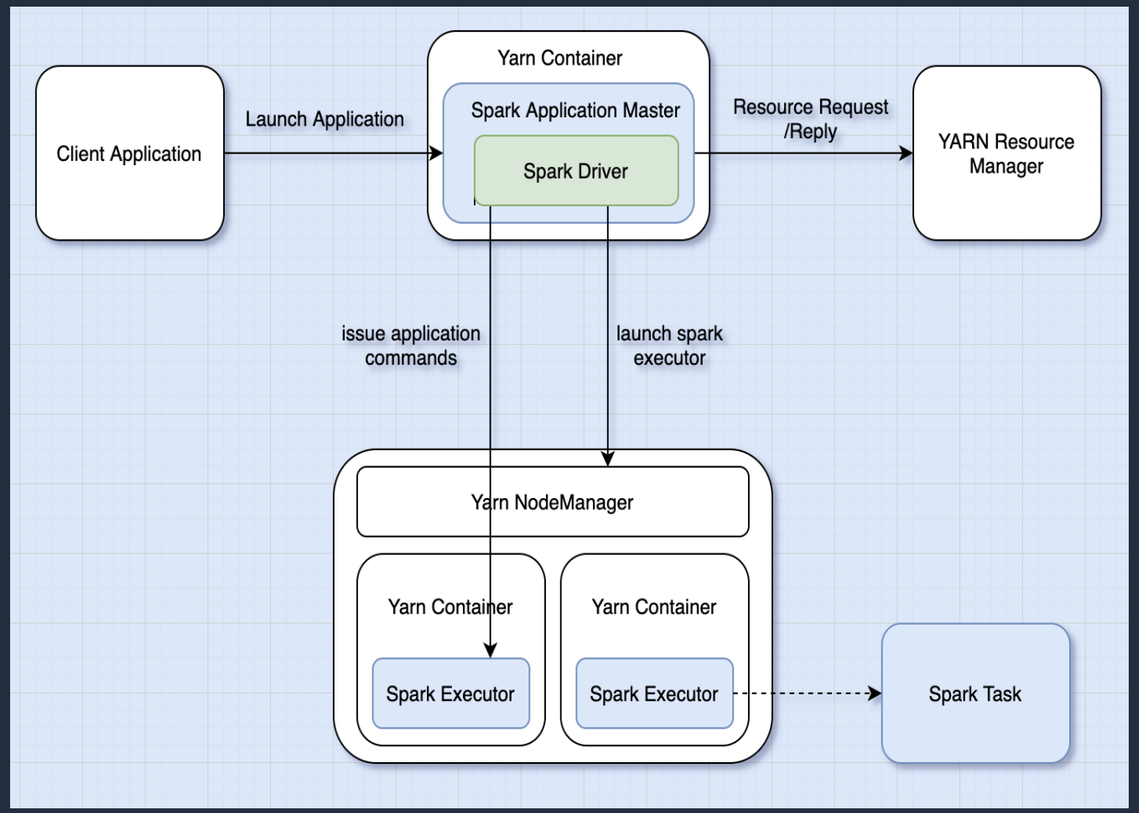
What are two different modes for spark to run on YARN ?

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Spark on YARN – client mode vs cluster mode



client mode



cluster mode

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

YARN ResourceManager and NodeManagers

Mode	YARN client mode	YARN cluster mode
Driver runs in	Client	ApplicationMaster
Request Resources	ApplicationMaster	ApplicationMaster
Starts executor processes	YARN NodeManager	YARN NodeManager
Persistent services	YARN ResourceManager and NodeManagers	YARN ResourceManager and NodeManagers
Support Spark shell	Yes	No

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Spark Lab 1

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Lets Submit Spark Job on EMR

 # Run on a YARN cluster client mode

```
/usr/bin/spark-submit \
--class org.apache.spark.examples.SparkPi \
--master yarn \
--deploy-mode client \
/usr/lib/spark/examples/jars/spark-examples.jar \
10
```

 # Run on a YARN cluster Cluster mode

```
/usr/bin/spark-submit \
--class org.apache.spark.examples.SparkPi \
--master yarn \
--deploy-mode cluster \
/usr/lib/spark/examples/jars/spark-examples.jar \
10
```

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Question-III

What's the difference between Persistent user interfaces and On-cluster user interfaces on EMR ?

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Monitoring and Debugging

- **Spark UI**
  Job Performance, Task breakdown of job, information about cached DataFrames, and more

- **Application History**
  Application history from EMR console

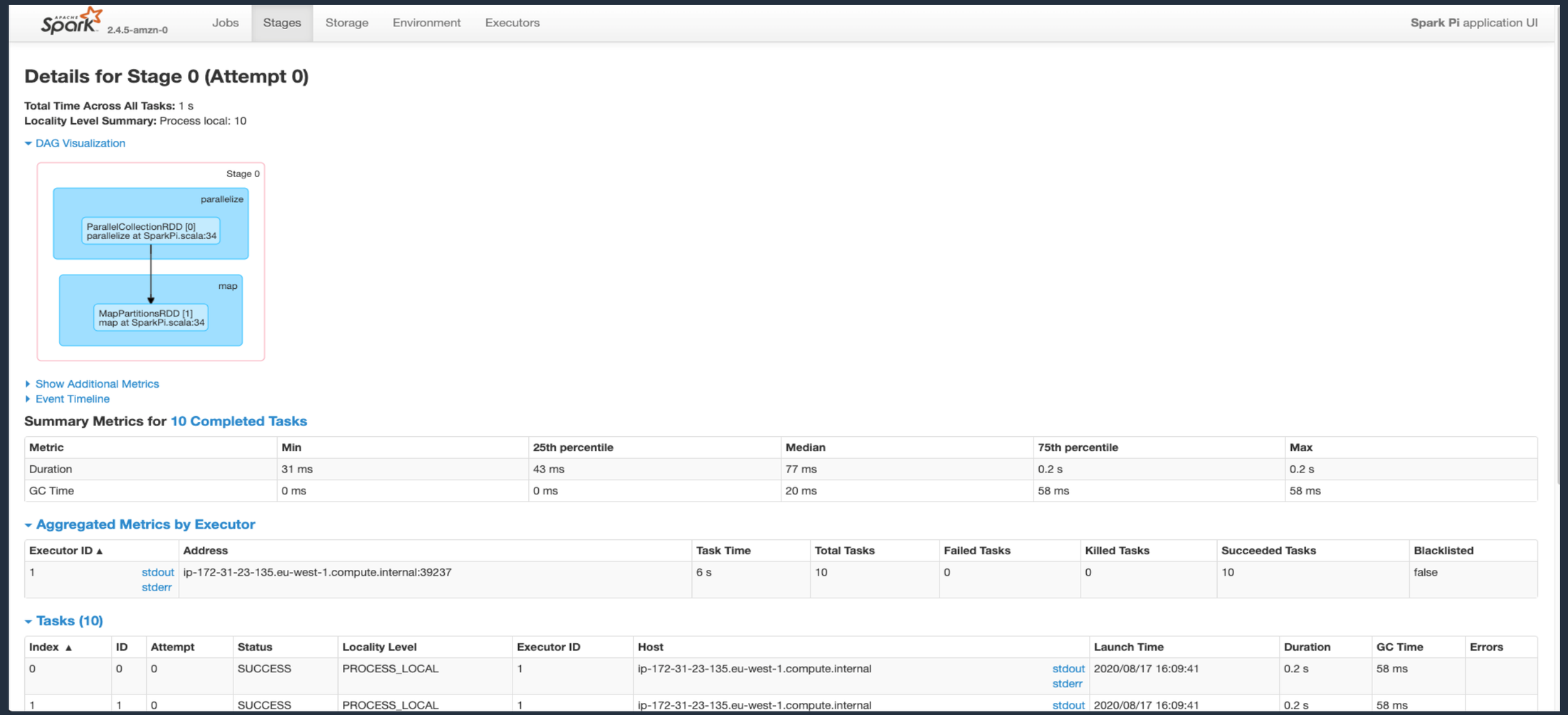
- **Logs pushing to s3**
  logs produced by driver and executors on each node can browse through log folders in EMR console

- **Cloud watch metrics**
  Cloud watch metrics in the EMR console

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

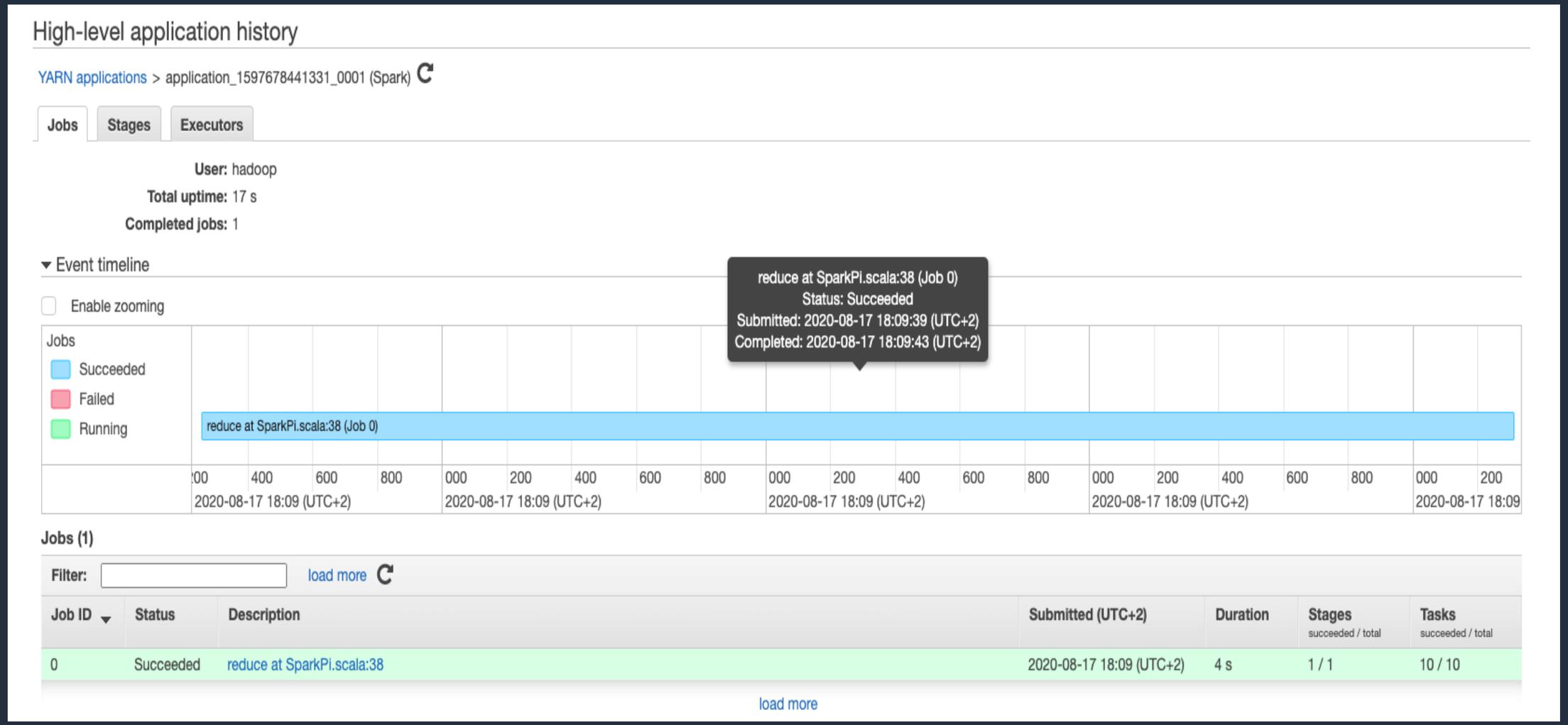
aws

Monitoring and Debugging – Using Spark UI



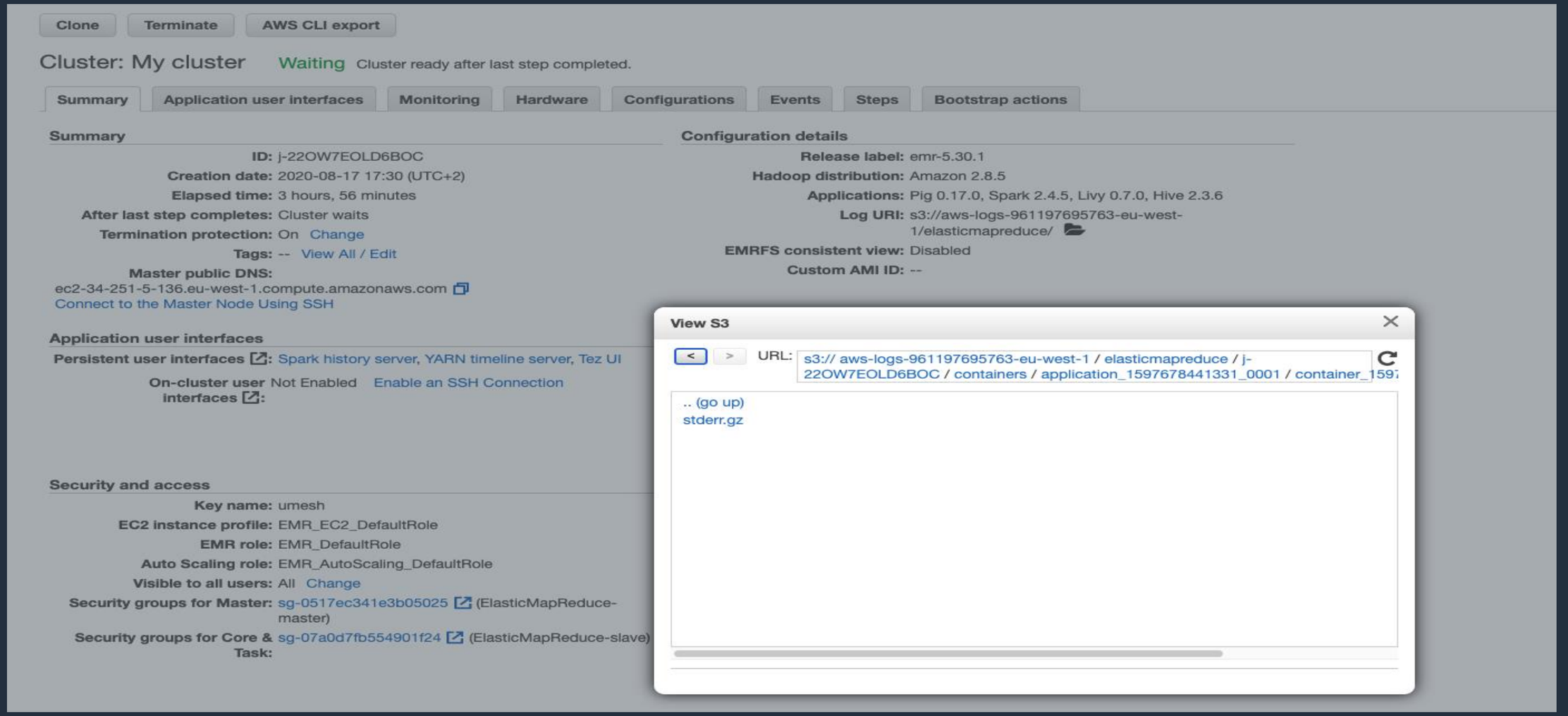
© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Monitoring and Debugging – Using EMR console



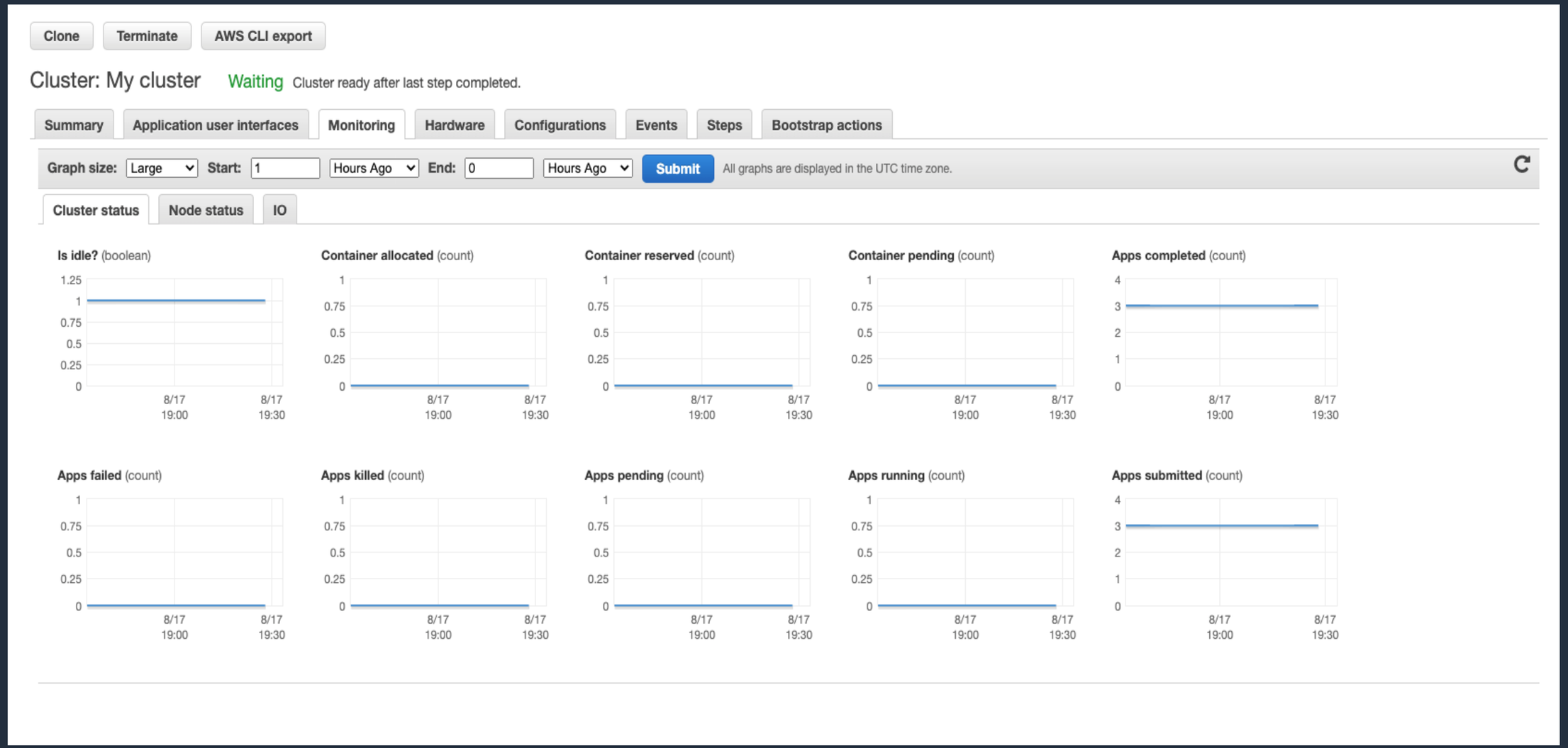
© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Monitoring and Debugging – Using logs on EMR console



© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Monitoring and Debugging – Using CW metrics on EMR console



© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

RDD, Data Frames, Datasets

Resilient Distributed Dataset (RDD)

RDD was the primary user-facing API in Spark since its inception. At the core, an RDD is an immutable distributed collection of elements of your data, partitioned across nodes in your cluster that can be operated in parallel with a low-level API that offers transformations and actions.

DataFrames

Like an RDD, a DataFrame is an immutable distributed collection of data. Unlike an RDD, data is organized into named columns, like a table in a relational database.

Datasets

Dataset was introduced in spark-1.6 as experimental feature. Dataset takes on two distinct APIs characteristics: a strongly-typed API and an untyped API.

Note : you can move between DataFrame or Dataset and RDDs — by simple API method calls— and DataFrames and Datasets are built on top of RDDs.

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Spark SQL

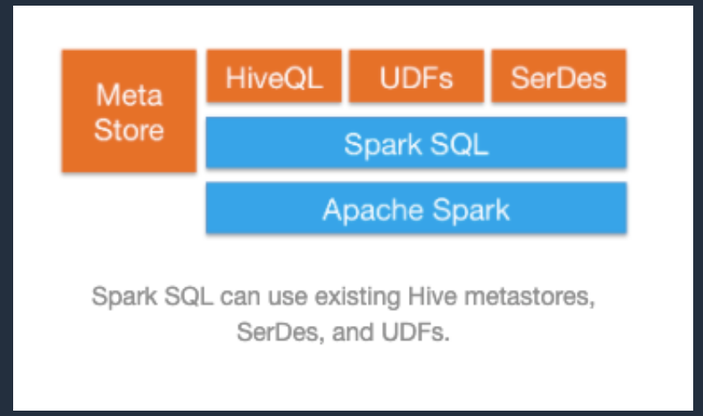
Spark SQL is Apache Spark's module for working with structured data.

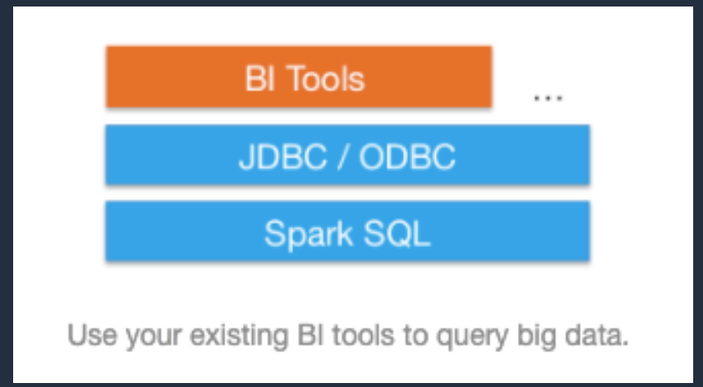
- Integrated
  Seamlessly mix SQL queries with Spark programs.

- Uniform Data Access
  Connect to any data source the same way.

- Hive Integration
  Run SQL or HiveQL queries on existing warehouses.

- Standard Connectivity
  Connect through JDBC or ODBC.





© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

SQL vs Data Frames vs Datasets

	SQL	DataFrames	Datasets
Syntax Errors	Runtime	Compile Time	Compile Time
Analysis Errors	Runtime	Runtime	Compile Time

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Spark Lab 2

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Question-I

Why spark is faster than MR ?

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Questions?

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

References

https://cwiki.apache.org/confluence/display/Hive/Home

https://tez.apache.org/

https://spark.apache.org/docs/latest/

https://docs.aws.amazon.com/emr/latest/ReleaseGuide/tez-using.html

https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-spark-application.html

https://github.com/apache/spark/tree/master/examples/src/main

https://tez.apache.org/talks.html

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Thank You

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved.