ELS Bigdata 2021(EMR)

Jack Lin & Weiting Liang

2021/06/26

aws

Material & Lab answer submission

- https://github.com/jack555023/ELS_2021
- https://docs.google.com/forms/d/e/1FAIpQLSdehUs5pGKkEd11iX5Yv-TOI46RJXDumif23IcdDQTqv20trQ/viewform?usp=sf_link

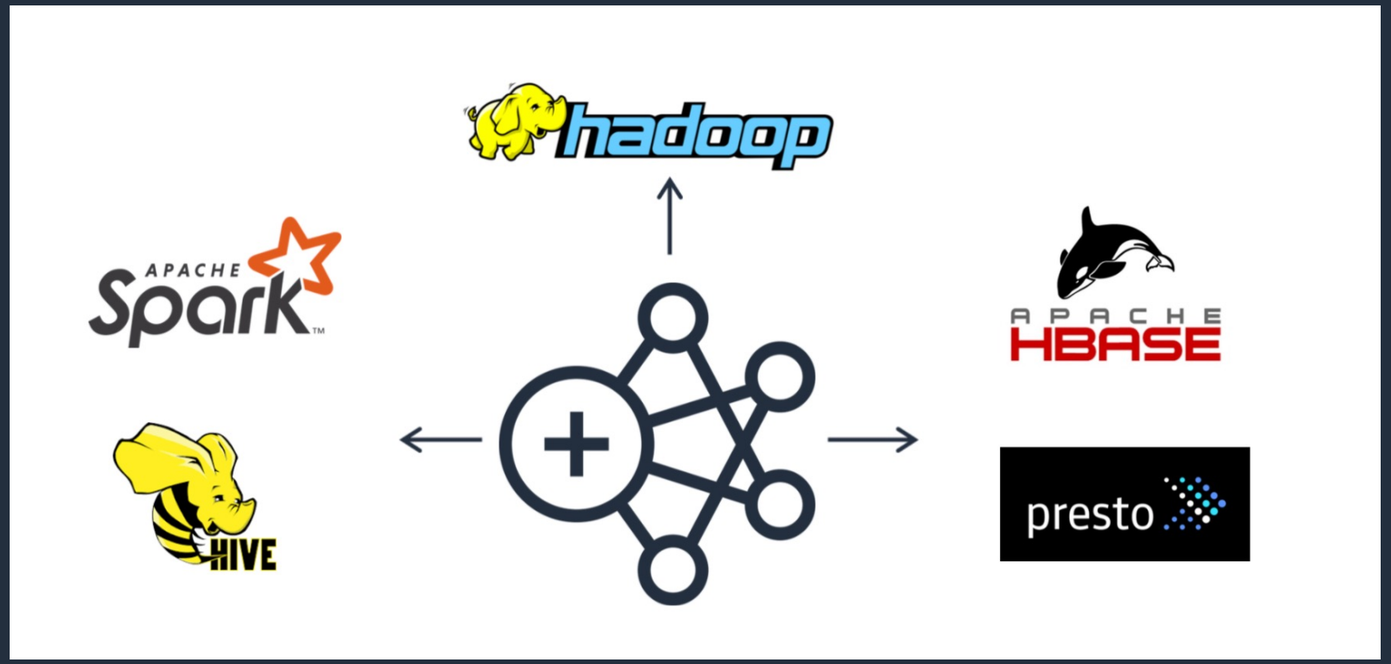
aws

EMR & Hadoop

What is EMR?

Elastic Map Reduce
Hadoop FrameWork

Why EMR :
Easily create/manage
Auto-scaling for nodes
Amazon S3 (EMRFS)

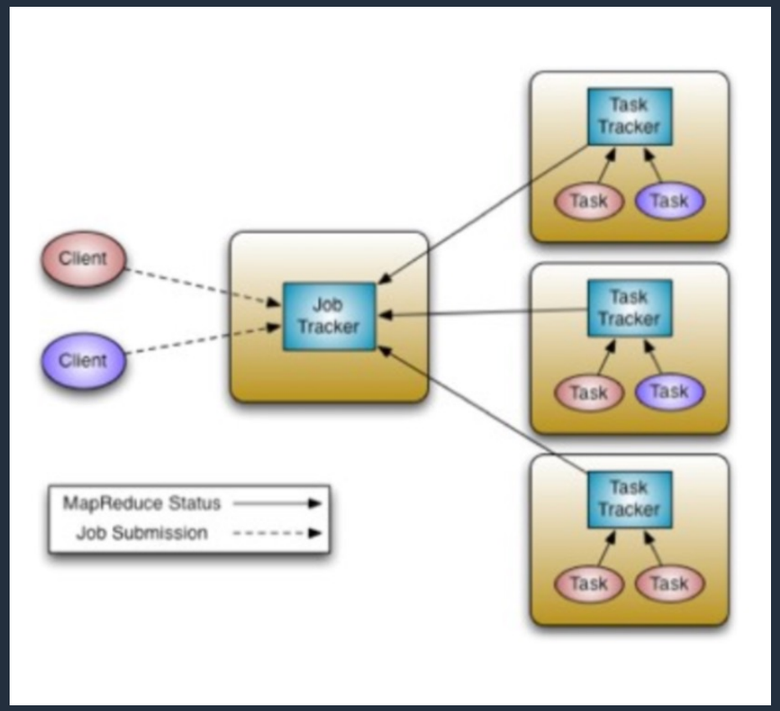


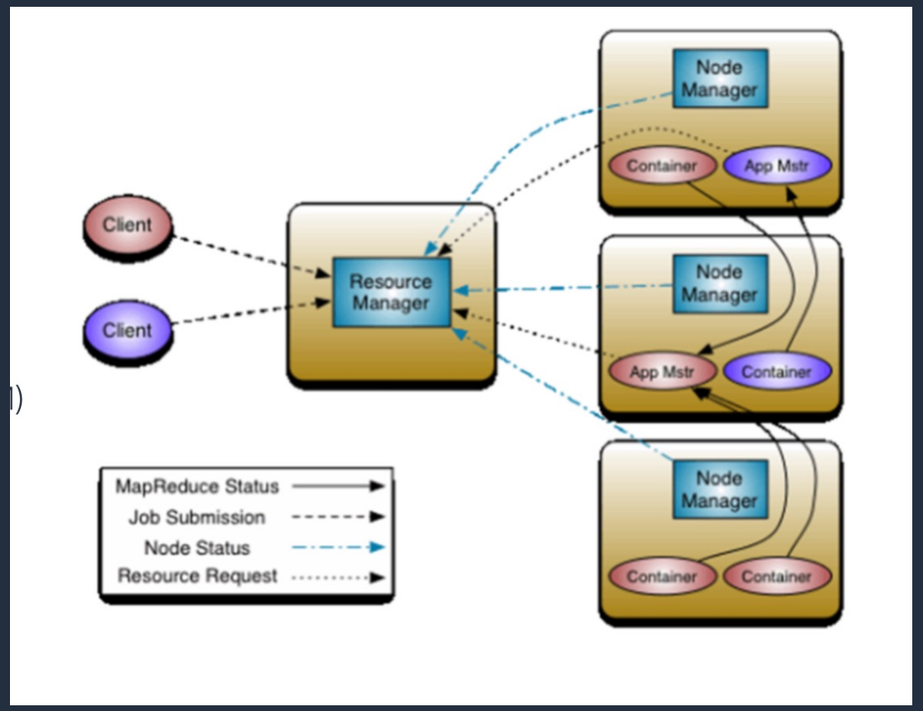
aws

Hadoop Framework - YARN

Job Execution

Hadoop 1.0 vs. Hadoop 2.0 (YARN)





Hadoop Framework - YARN

Resource Manager & Node Manager & Application Master

Yarn Application ID

Ex. application_1620035588649_0001

Resource Manager Start Epoch time + Sequence Number

Checking the YARN log (Start/Kill Yarn application)

aws

EMR’s Node.

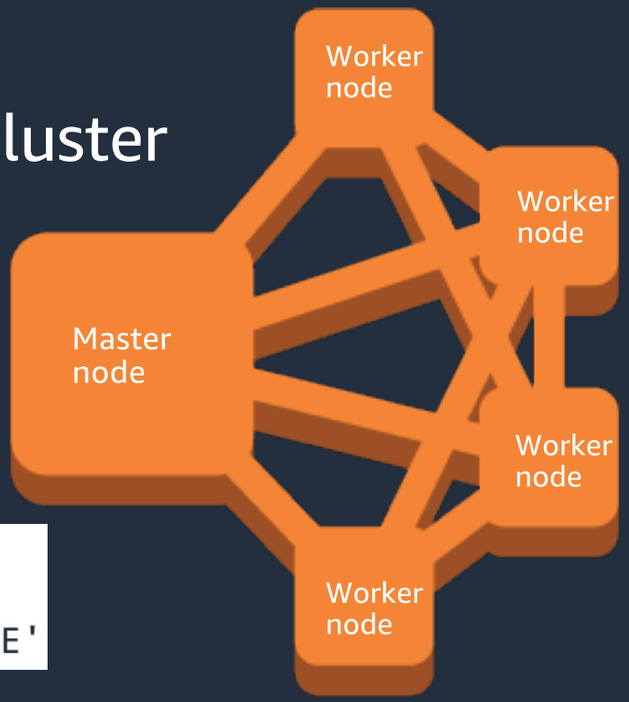
Infrastructure

Master: Management & Monitoring the cluster

Core: Storage & Job execution

Task: Job execution

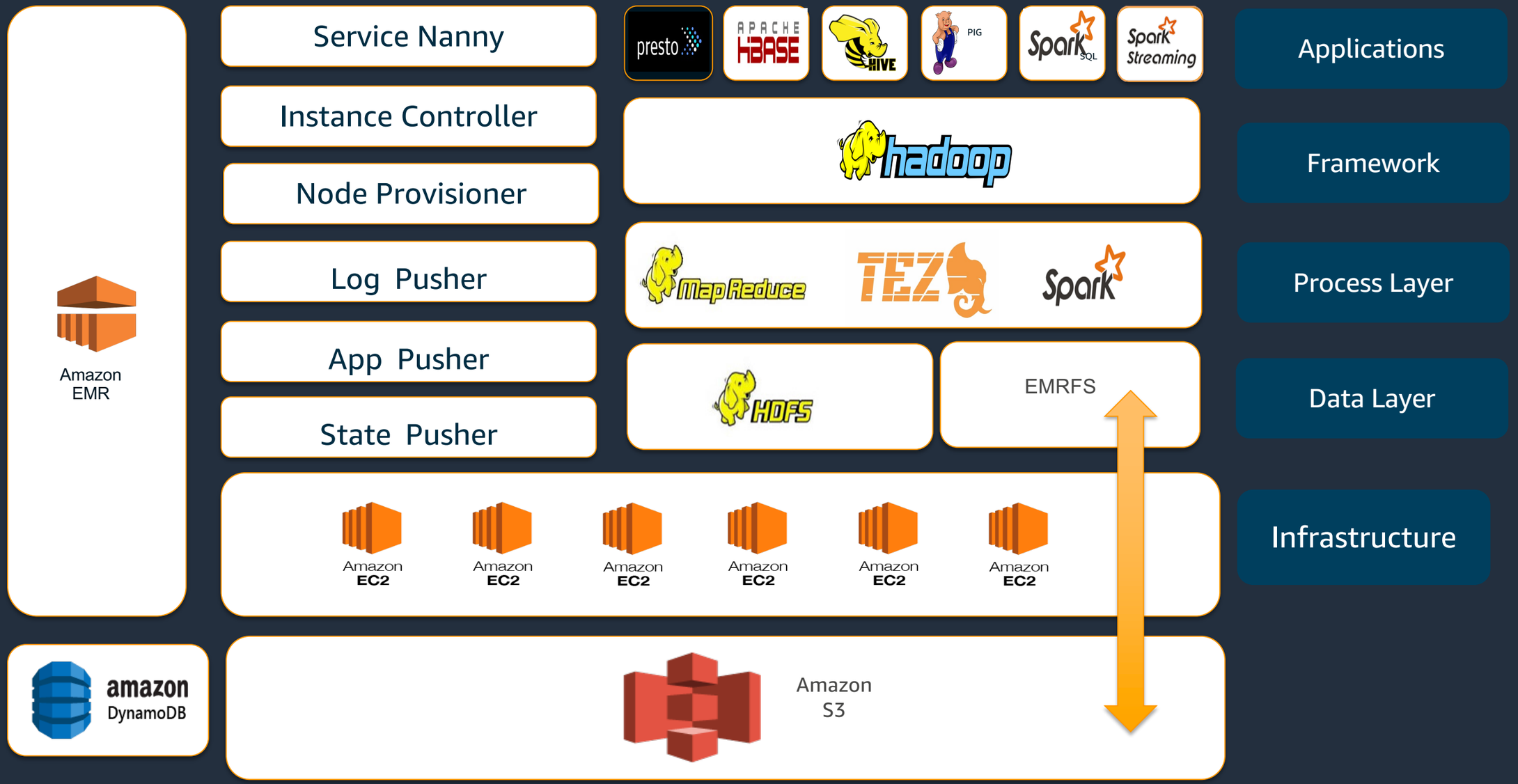
- `YARN.node-labels.enabled: true`
- `YARN.node-labels.am.default-node-label-expression: 'CORE'`



Amazon EMR cluster

aws

EMR Structure



EMR Daemons

- Daemons /etc/init.d/
- Service Nanny, Instance Controller, Node provisioner, Log Pusher, App Pusher State Pusher

- Service Nanny (Start & babysit other daemons)
- Instance controller (IC Master IC & Slave IC)
- Node Provisioner (Staring Service …etc)
- LogPusher (Push the log !!!)
- AppPusher(Spark History Server Log)
- StatePusher

aws

EMR Bootstrap Action & Steps

- Bootstrap Action
    - Running to all nodes
    - Only when Launching ….
- Steps
    - Submit to cluster via console / CLI
    - Streaming, Hive, Pig, Spark, Custom JAR ….etc.
    - Launching Cluster & Running Cluster

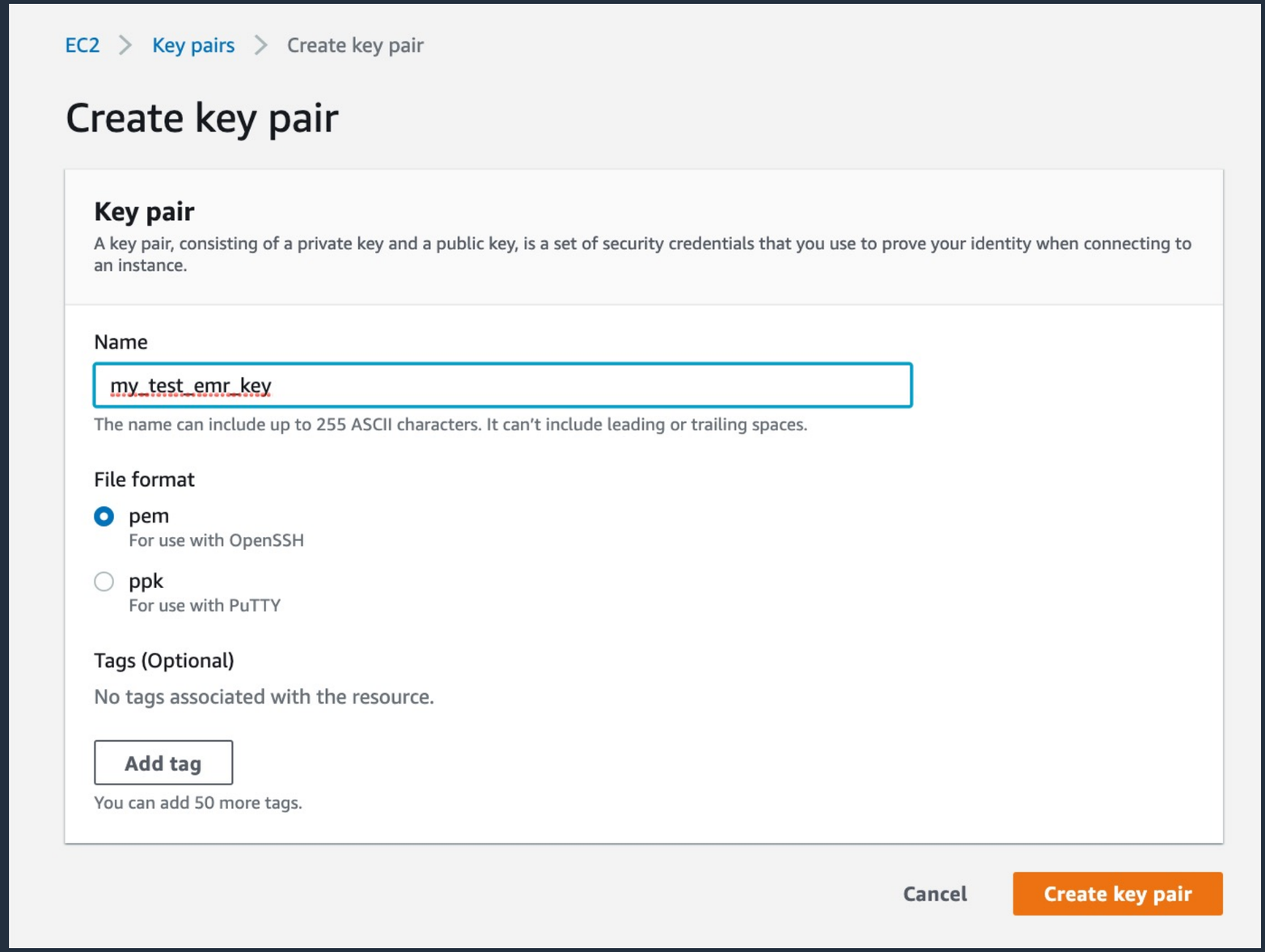
aws

Launching EMR Cluster

aws

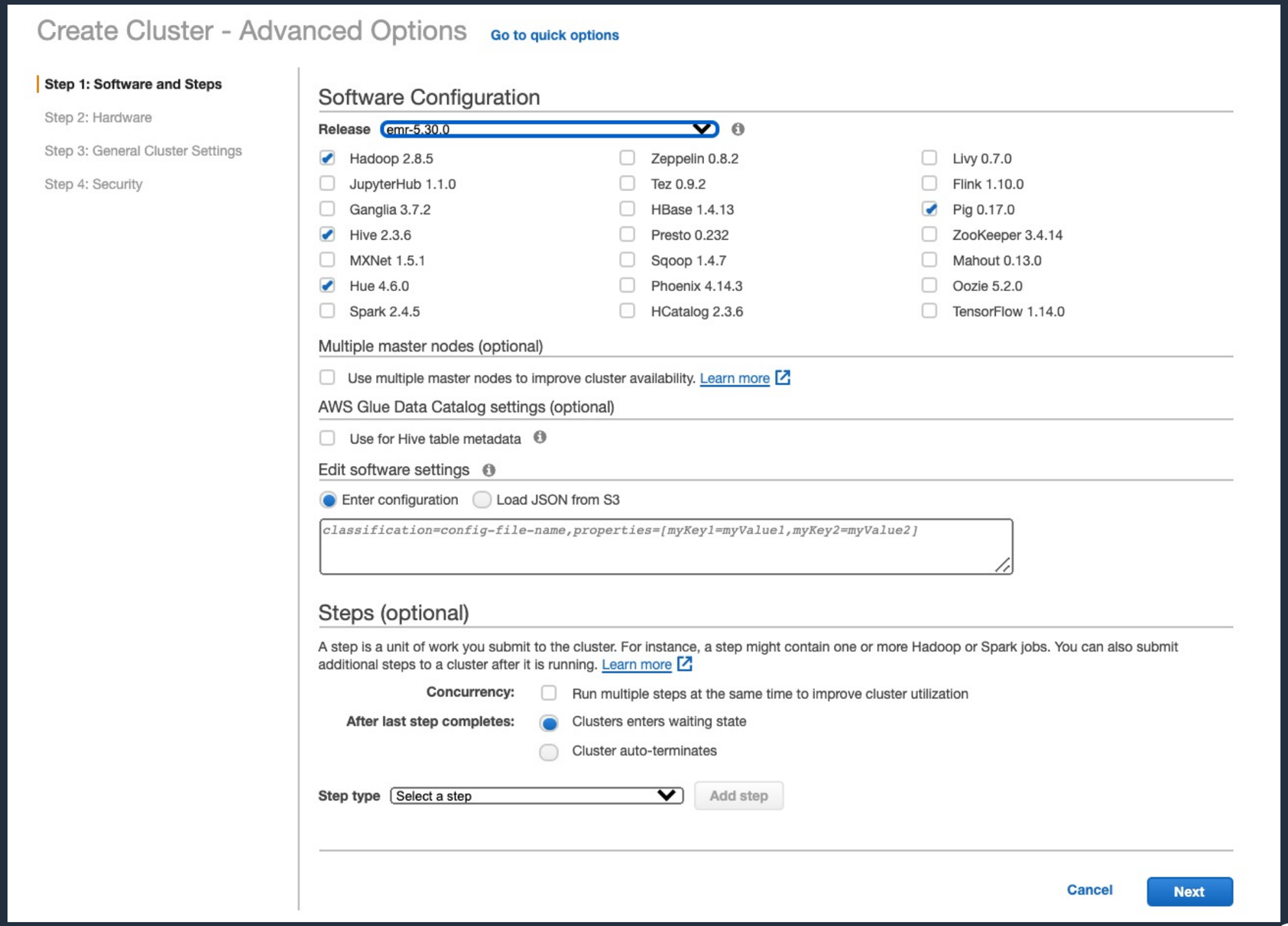
Launch EMR

- chmod 400

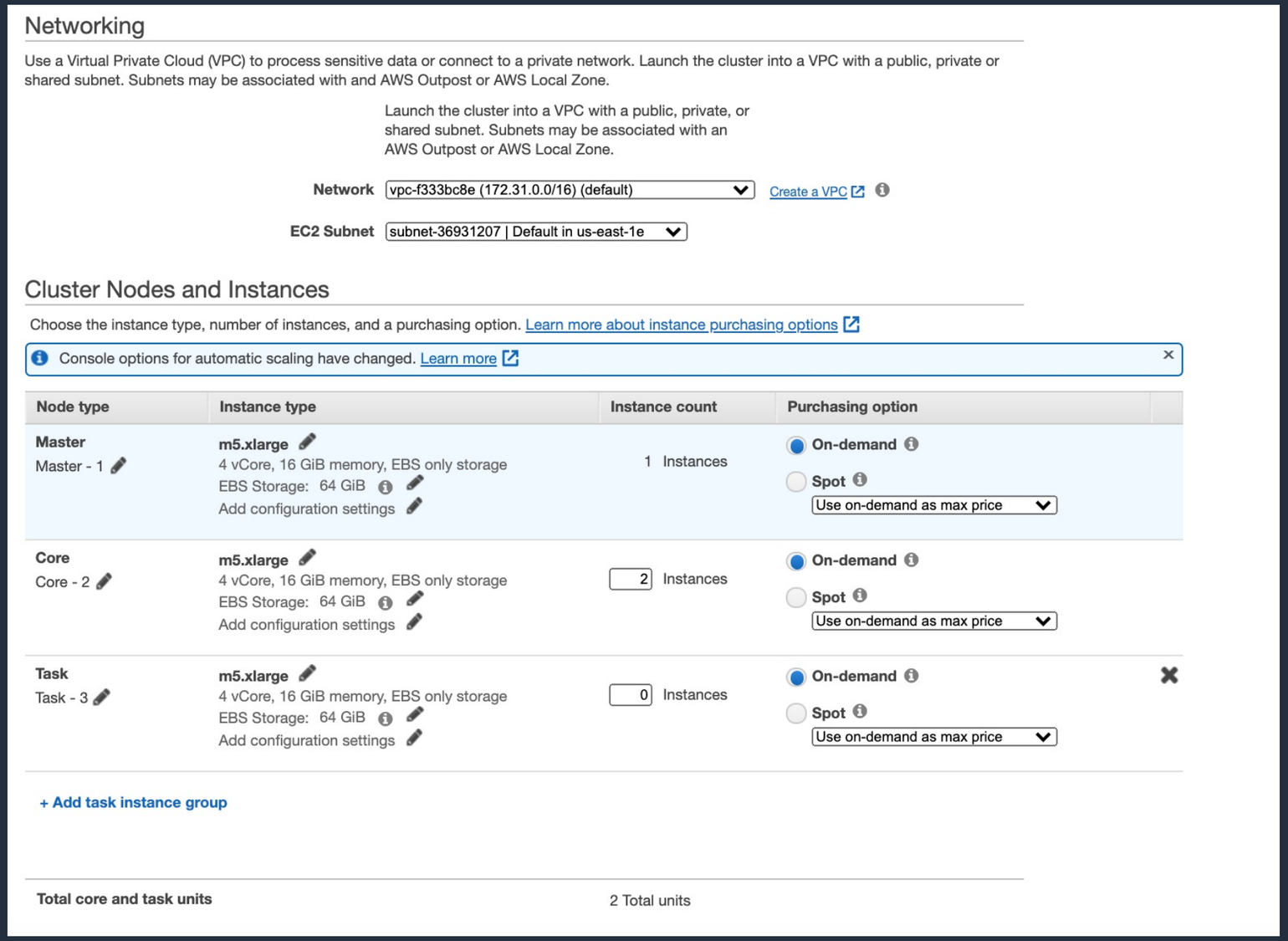


aws

Launch EMR

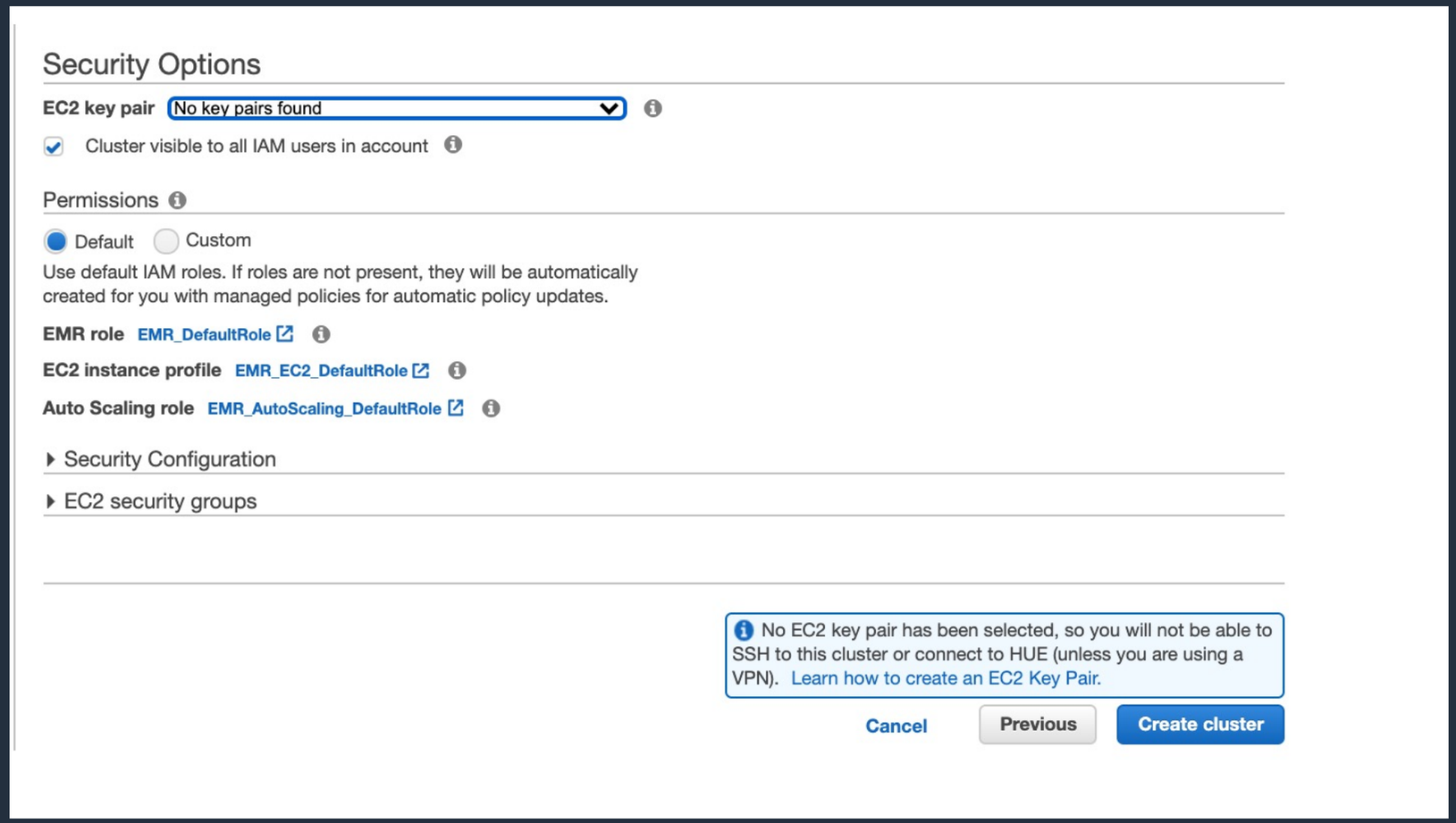


Launch EMR



aws

Launch EMR



aws

Demo & Cluster creation.

aws

Introduction to Hive and Tez

aws

Introduction to Hive

- Open source Big Data framework in the Hadoop ecosystem
- SQL-like interface to query data in HDFS
- HQL (Hive QL) is used to query data stored in a Hadoop cluster
- Maps high level SQL operations to low level MR Java API or Tez API

Hive Use Case

- Data warehouse applications.
- Optimizes storage and processing of data in cost effective ways.
- Easy integration with Spark and HBase.
- Allows users to run SQL.
- Supports querying data from different formats (e.g. JSON)

aws

Running Hive on EMR

```
[hadoop@ip-172-31-18-67 ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties
hive> show databases;
OK
default
Time taken: 0.513 seconds, Fetched: 1 row(s)
hive> use default;
OK
Time taken: 0.038 seconds
hive> create table foo(x int);
OK
Time taken: 0.502 seconds
hive> show tables;
OK
foo
Time taken: 0.04 seconds, Fetched: 1 row(s)
```


aws

Running Hive on EMR

```
hive> INSERT INTO TABLE foo VALUES ('1');
Query ID = hadoop_20210502063608_1447216c-bba9-4a38-9144-44b0e263c97f
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1619937144070_0001)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 01/01  [==========================>>] 100%  ELAPSED TIME: 4.01 s
--------------------------------------------------------------------------------
Loading data to table default.foo
OK
Time taken: 7.294 seconds
hive> select * from default.foo;
OK
1
Time taken: 0.131 seconds, Fetched: 1 row(s)
```

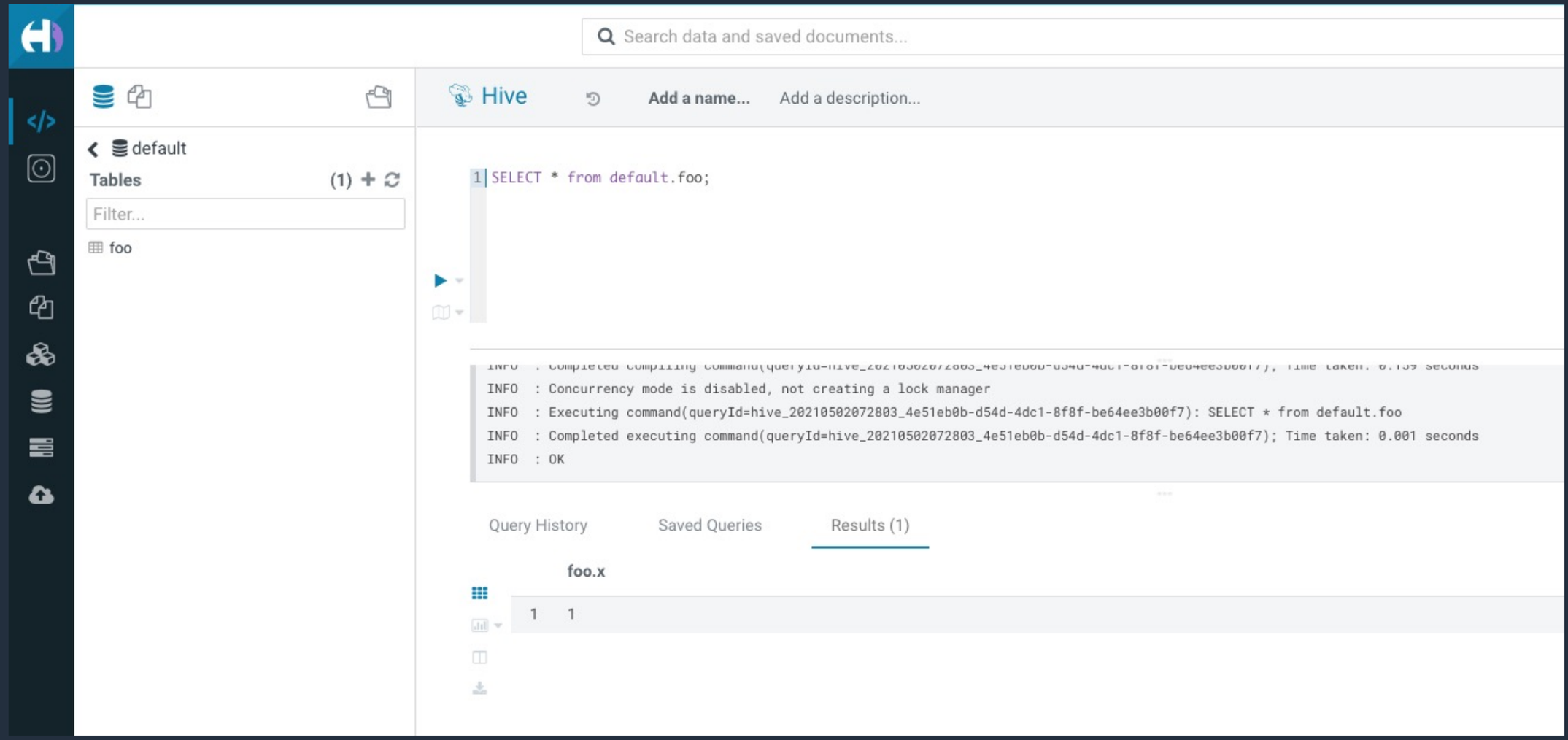

aws

Using Beeline to Run Your Hive Scripts

```
[hadoop@ip-172-31-18-67 ~]$ beeline -u 'jdbc:hive2://localhost:10000' -n hive -e 'select * from default.foo;'
Connecting to jdbc:hive2://localhost:10000
Connected to: Apache Hive (version 2.3.7-amzn-4)
Driver: Hive JDBC (version 2.3.7-amzn-4)
Transaction isolation: TRANSACTION_REPEATABLE_READ
+--------+
| foo.x  |
+--------+
| 1      |
+--------+
1 row selected (0.231 seconds)
Beeline version 2.3.7-amzn-4 by Apache Hive
Closing: 0: jdbc:hive2://localhost:10000
```


aws

Using Hue to Run Hive Scripts



aws

Hive Features

- Ad-hoc inserts and updates
- Support for columnar formats like ORC / Parquet
- Partitions
    - Divides data into folders allowing you to query something without going through all the data
    - Supports dynamic partitioning allowing you to output to different partitions based on what yout data contains.
- Allows different techniques for joining 2+ datasets together
- Supports extensions of the job
    - Java UDF (user defined function)

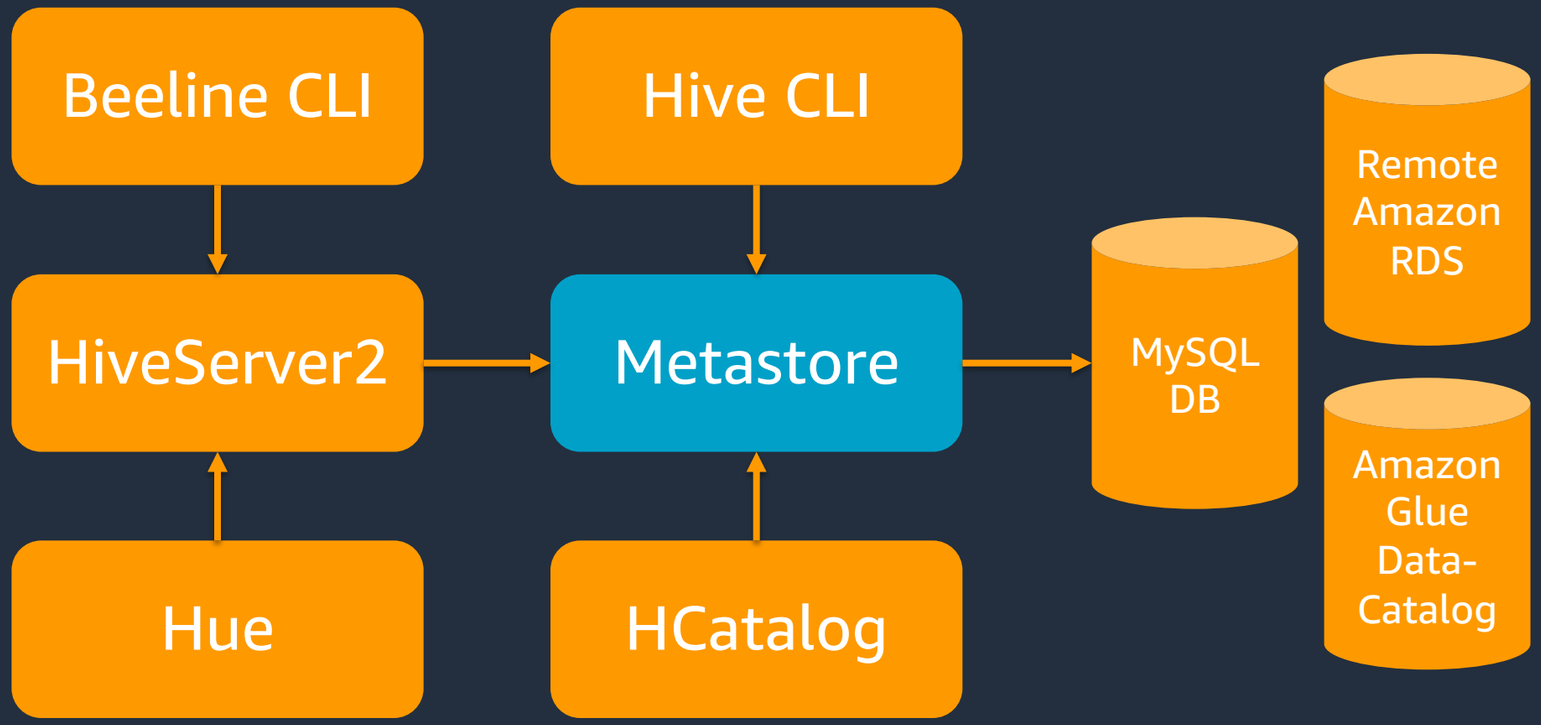
aws

Hive Metastore

- The Metastore contains a description of the table and the underlying data on which it is built, but does not contain actual user data.
- Two options for external Metastore:
    - Using AWS Glue Data Catalog as Hive Metastore
    - Using external RDBMS (MySQL, Oracle, etc.) or Amazon RDS

Remote Metastore Mode

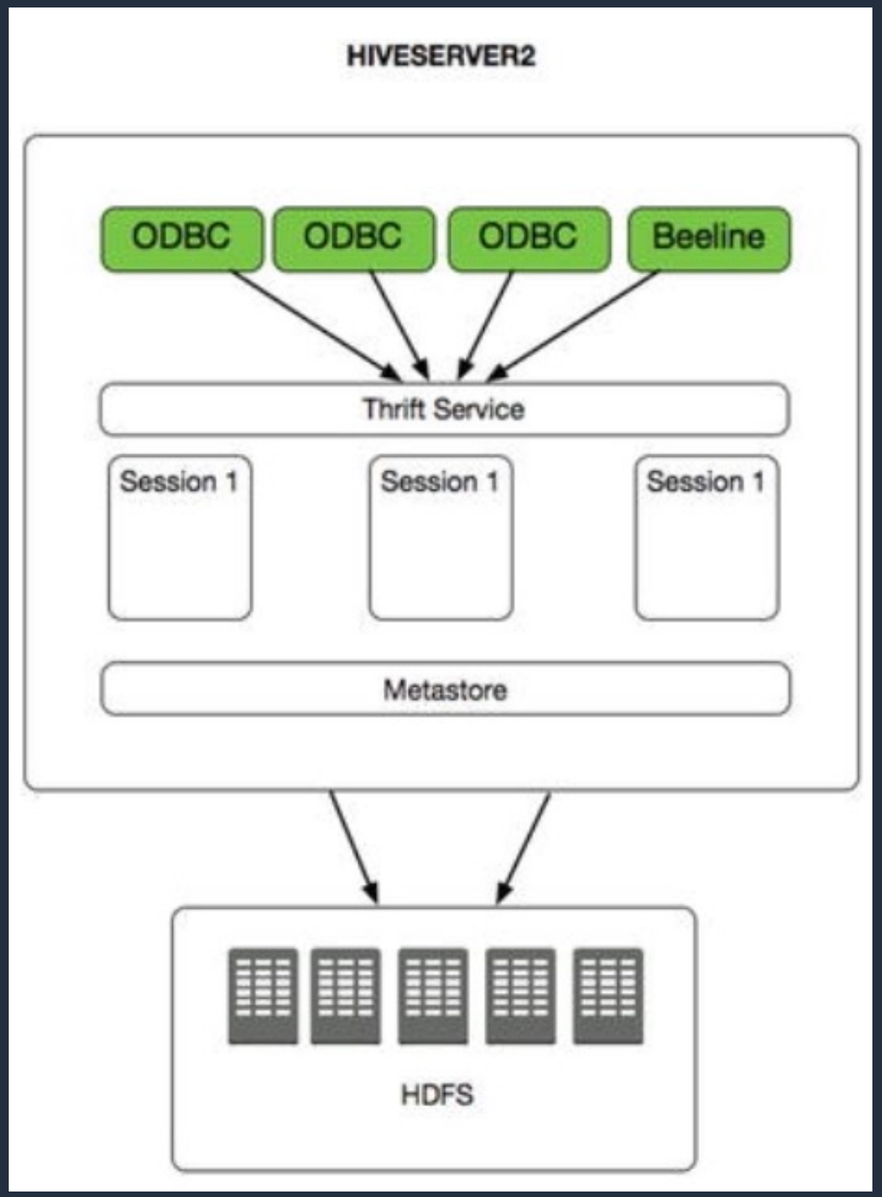
- Runs in its own JVM process.
- Other processes communicate with it using Thrift network API.
- By default, Metastore database is a mysql engine running on the master node listening on port 3306.



aws

HiveServer2 (HS2)

- Service that enables clients to execute queries against Hive.
- Supports multi-client concurrency and aithentication.
- Large concurrent connections cause OutOfMemory.
- Listens on thrift 10000 port to connect with Beeline and Hue.



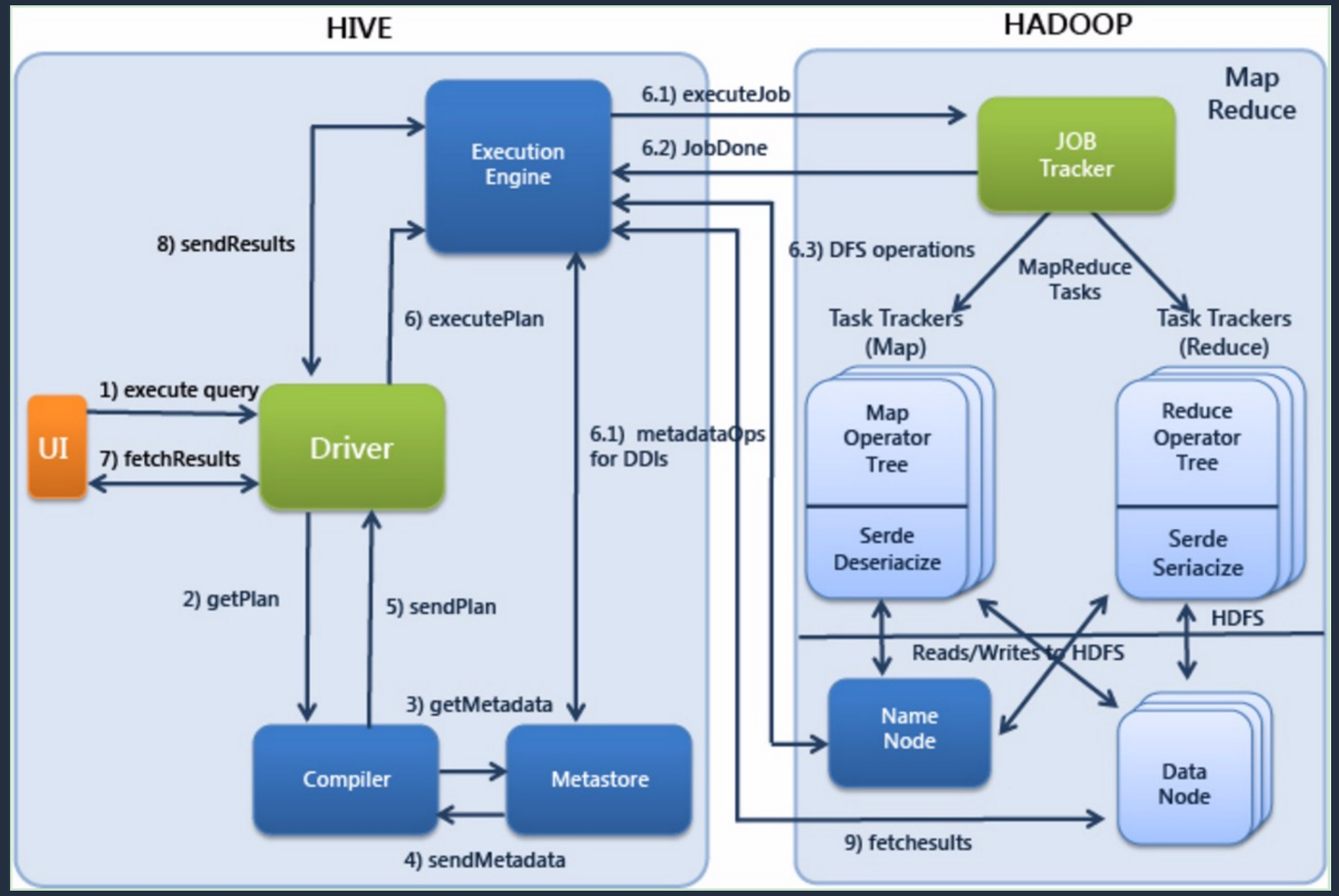
aws

HiveQL

- HiveQL integrates SQL-like queries into underlying Jave without the need to implement queries in low-level Java API.
- Divided into two languages:
    - Data Definition Language (DDL)
        - Used for creating, altering, and droping databases, tables, views, functions and indexes
    - Data Manipulation Language (DML)
        - Used to put data into Hive tables
        - Extract data to filesystem
        - Allows you to manipulate data

Hive on MapReduce (MR) Engine

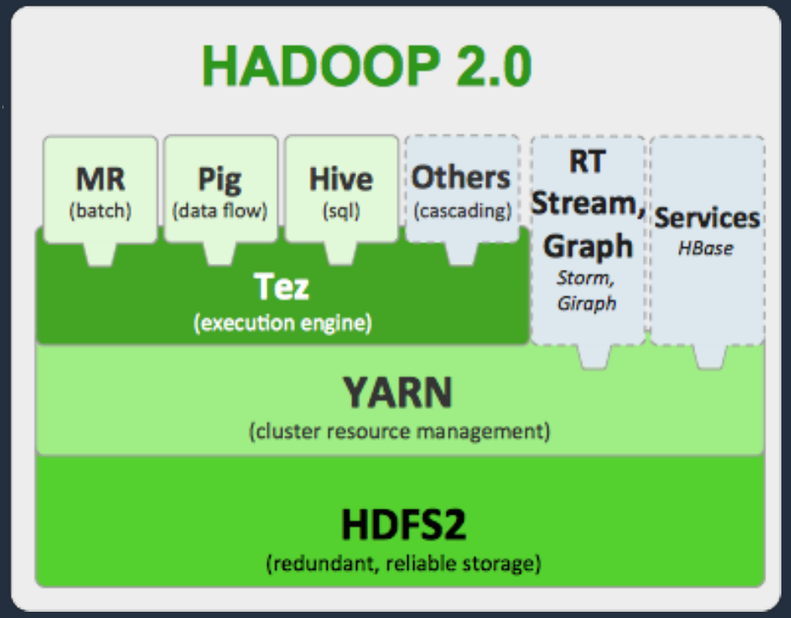
- Runs the Hive query as a traditional MR job.
- Safest fallback option.
- Select this engine by setting value: hive.execition.engine=mr



aws

Hive on Tez

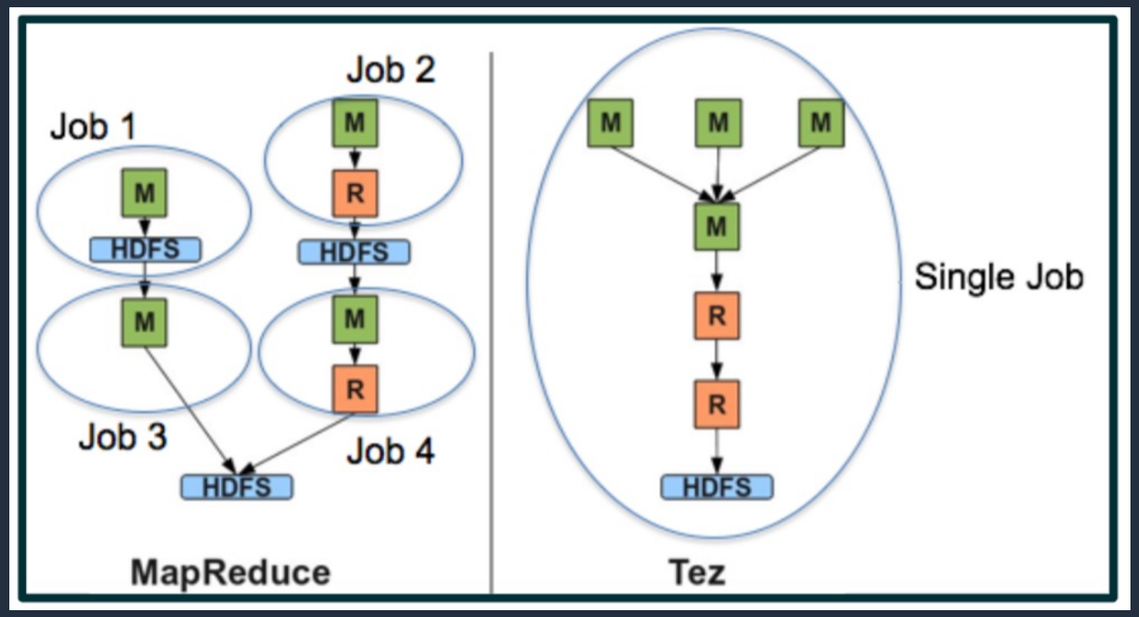
- Tez is the default execution engine starting from EMR versions 5.0.0 and later.
- App framework built on YARN that can execute complex DAGs for data processing.
- Built for high performance batch and interactive data-processing applications.



aws

Tez Engine v.s. MR Engine

- Tez solves the interactive query slowness problem.
- Solves MR shortcomings by balacing performance and throughput.
- Can split map and reduce jobs into smaller tasks.

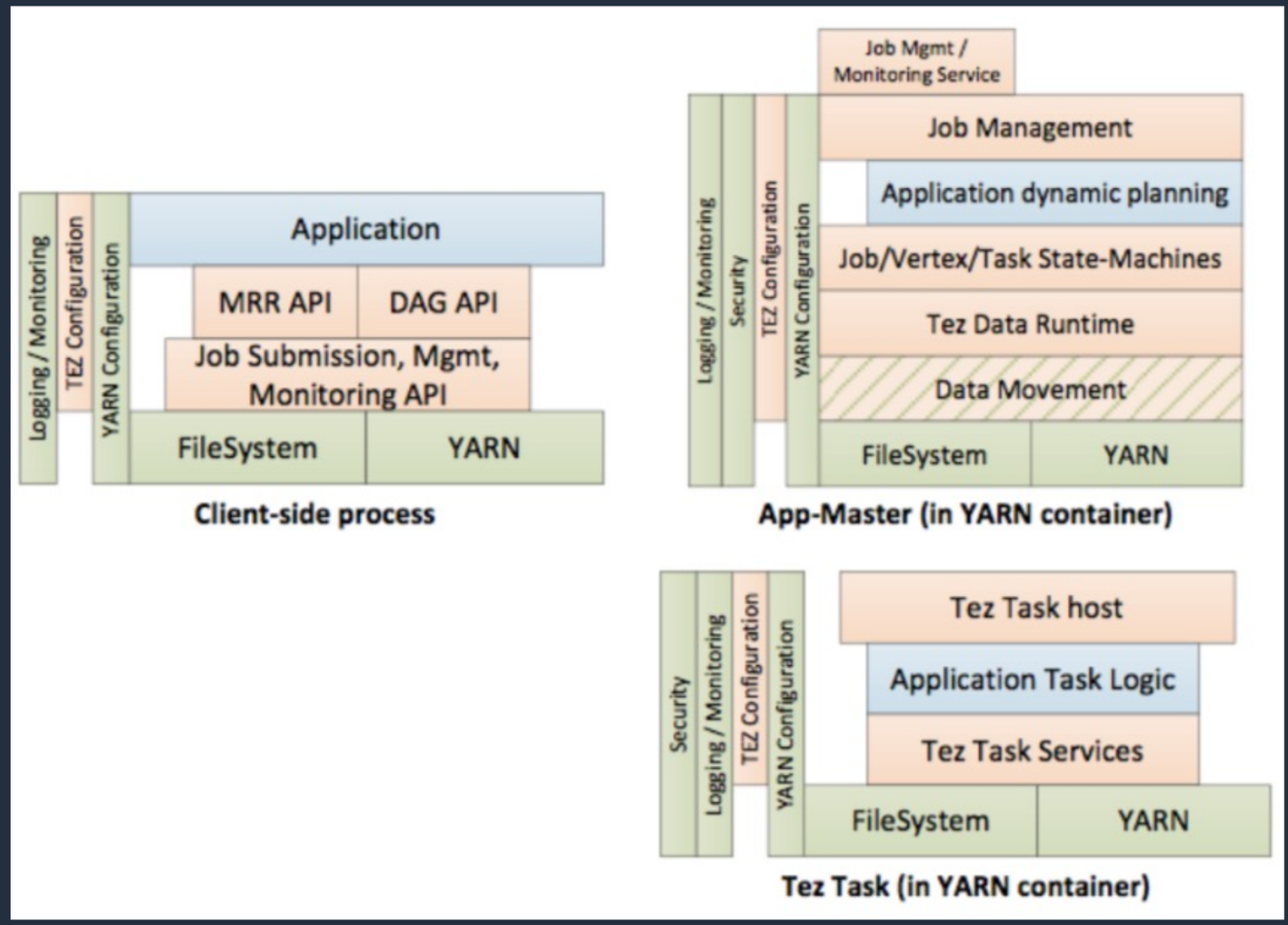


aws

Tez Components

Tez AppMaster is responsible for

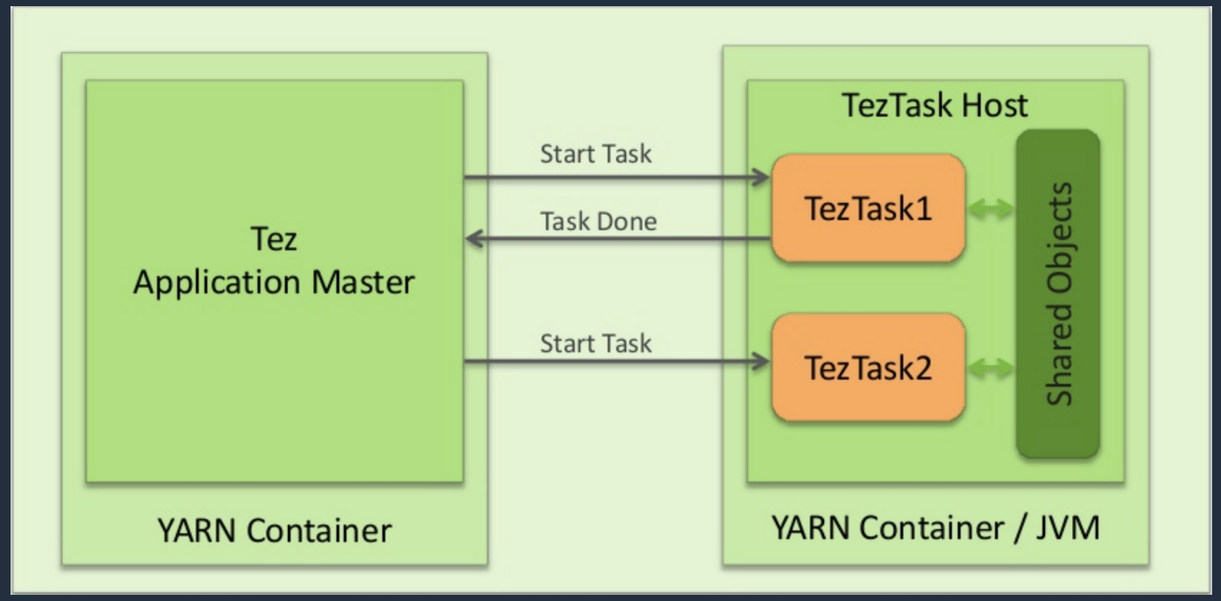
- Using the containers to implement an effective job runtime.
- Dealing with transient container execution failures.
- Responding to RM requests about allocated or deallocated continers.



aws

Tez – Container Re-Use

- Set the tez.am.container.reuse.enable = true
- Reuse YARN containers/JVMs to launch new tasks.
- Reduce scheduling and launching delays.
- Shard in-memory data across tasks.

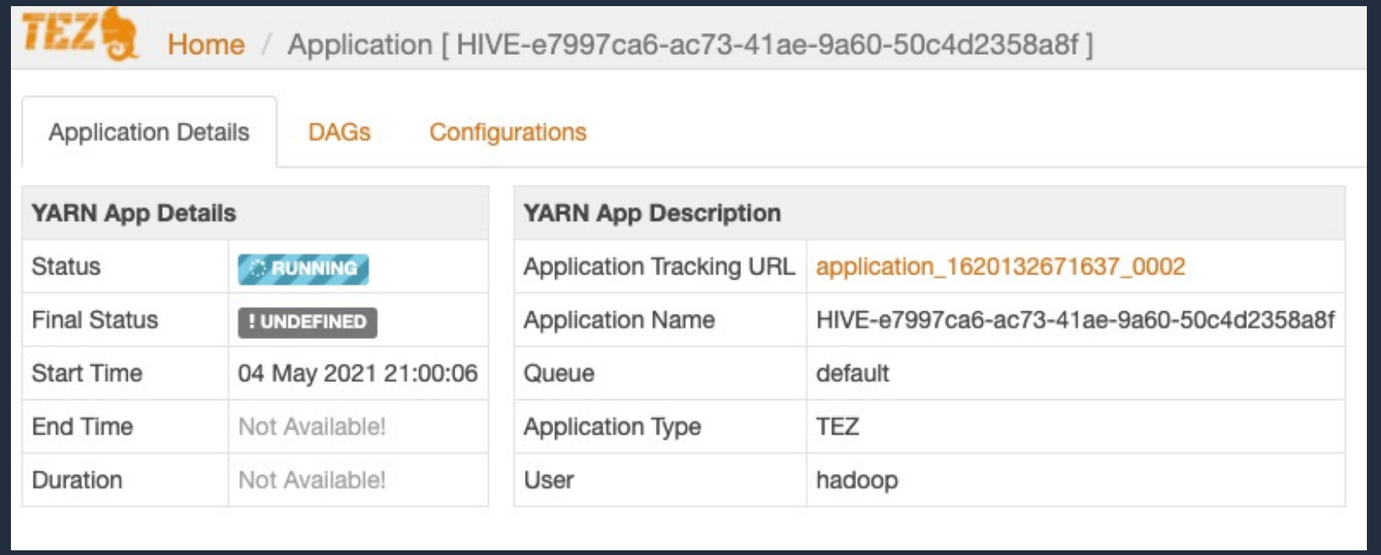


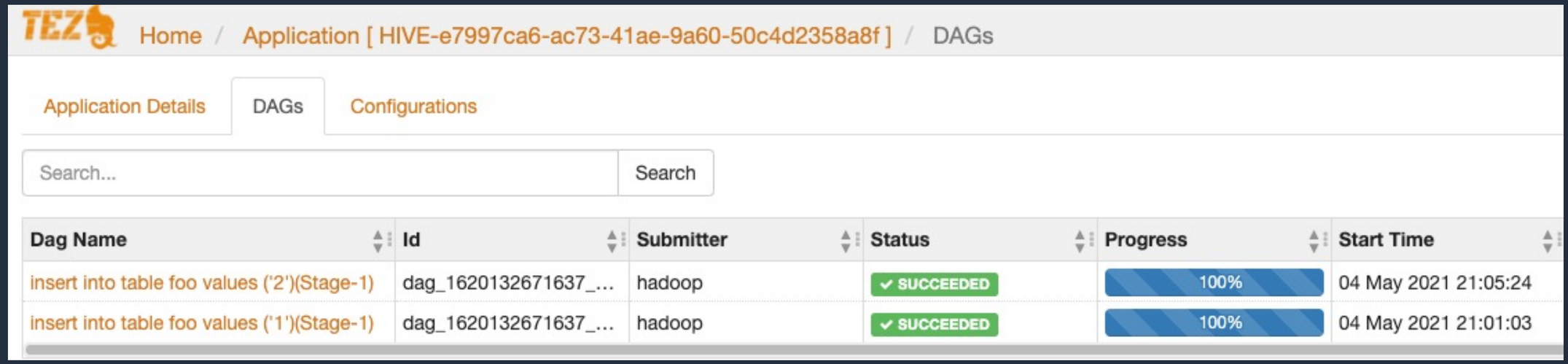
aws

Tez - Sessions

- Non-Session Mode
    - Each DAG is executed in a different AppMaster that exits after the DAG execution completes.
- Session Mode
    - A single instance of AppMaster is created, and all DAGs are submitted to the same AppMaster.
    - Better performance when a series of DAGs need to be executed, since it enables resource reuse across DAGs.

Tez UI





aws

Troubleshooting guide

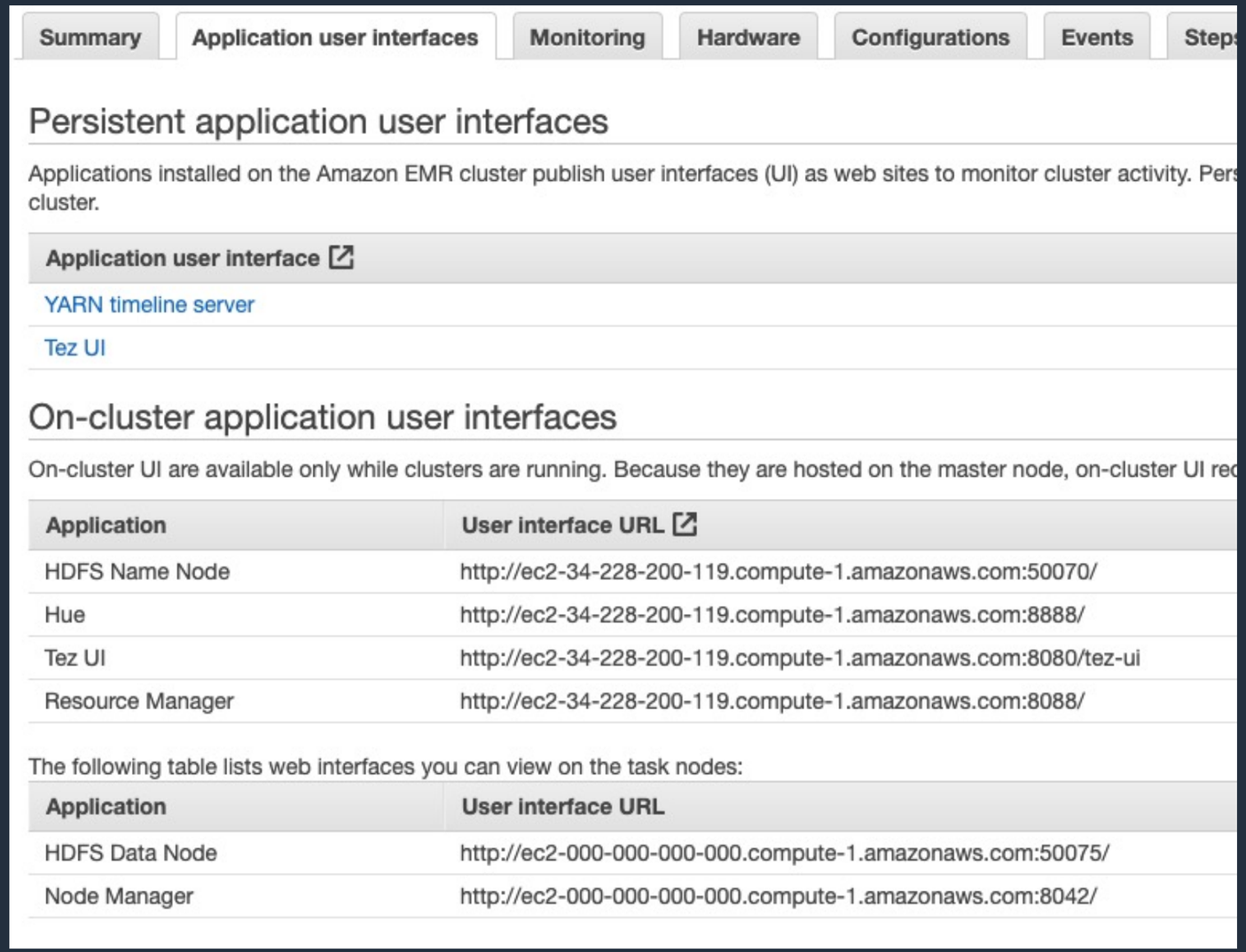
Web Interfaces Hosted on Amazon EMR Clusters

Name of Interface	URI
Hadoop HDFS NameNode	http://master-public-dns-name:50070/
Hadoop HDFS DataNode	http://coretask-public-dns-name:50075/
Resource Manager	http://master-public-dns-name:8088/
Node Manager	http://coretask-public-dns-name:8042/
Hue	http://master-public-dns-name:8888/
Tez UI	http://master-public-dns-name:8080/tez-ui

https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-web-interfaces.html

aws

Web Interfaces Hosted on Amazon EMR Clusters



Summary | Application user interfaces | Monitoring | Hardware | Configurations | Events | Steps

Persistent application user interfaces

Applications installed on the Amazon EMR cluster publish user interfaces (UI) as web sites to monitor cluster activity. Pers
cluster.

Application user interface
YARN timeline server
Tez UI

On-cluster application user interfaces

On-cluster UI are available only while clusters are running. Because they are hosted on the master node, on-cluster UI rec

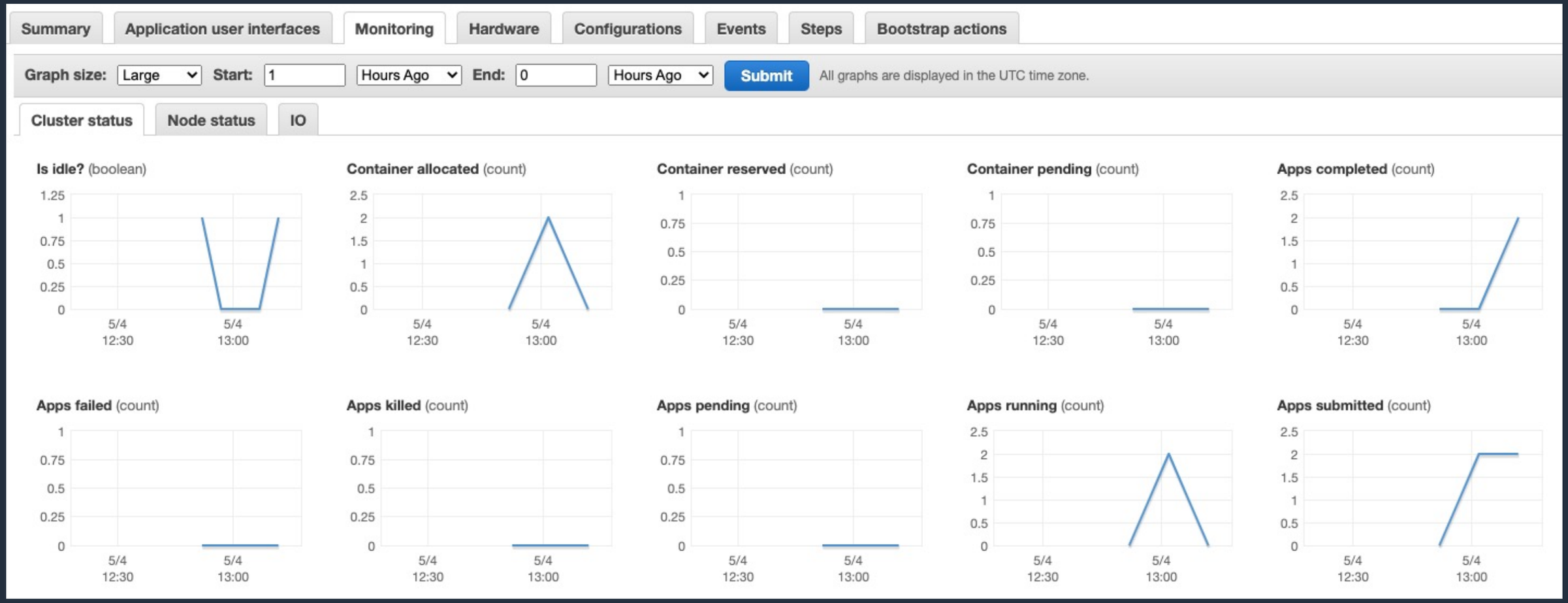
Application	User interface URL
HDFS Name Node	http://ec2-34-228-200-119.compute-1.amazonaws.com:50070/
Hue	http://ec2-34-228-200-119.compute-1.amazonaws.com:8888/
Tez UI	http://ec2-34-228-200-119.compute-1.amazonaws.com:8080/tez-ui
Resource Manager	http://ec2-34-228-200-119.compute-1.amazonaws.com:8088/

The following table lists web interfaces you can view on the task nodes:

Application	User interface URL
HDFS Data Node	http://ec2-000-000-000-000.compute-1.amazonaws.com:50075/
Node Manager	http://ec2-000-000-000-000.compute-1.amazonaws.com:8042/

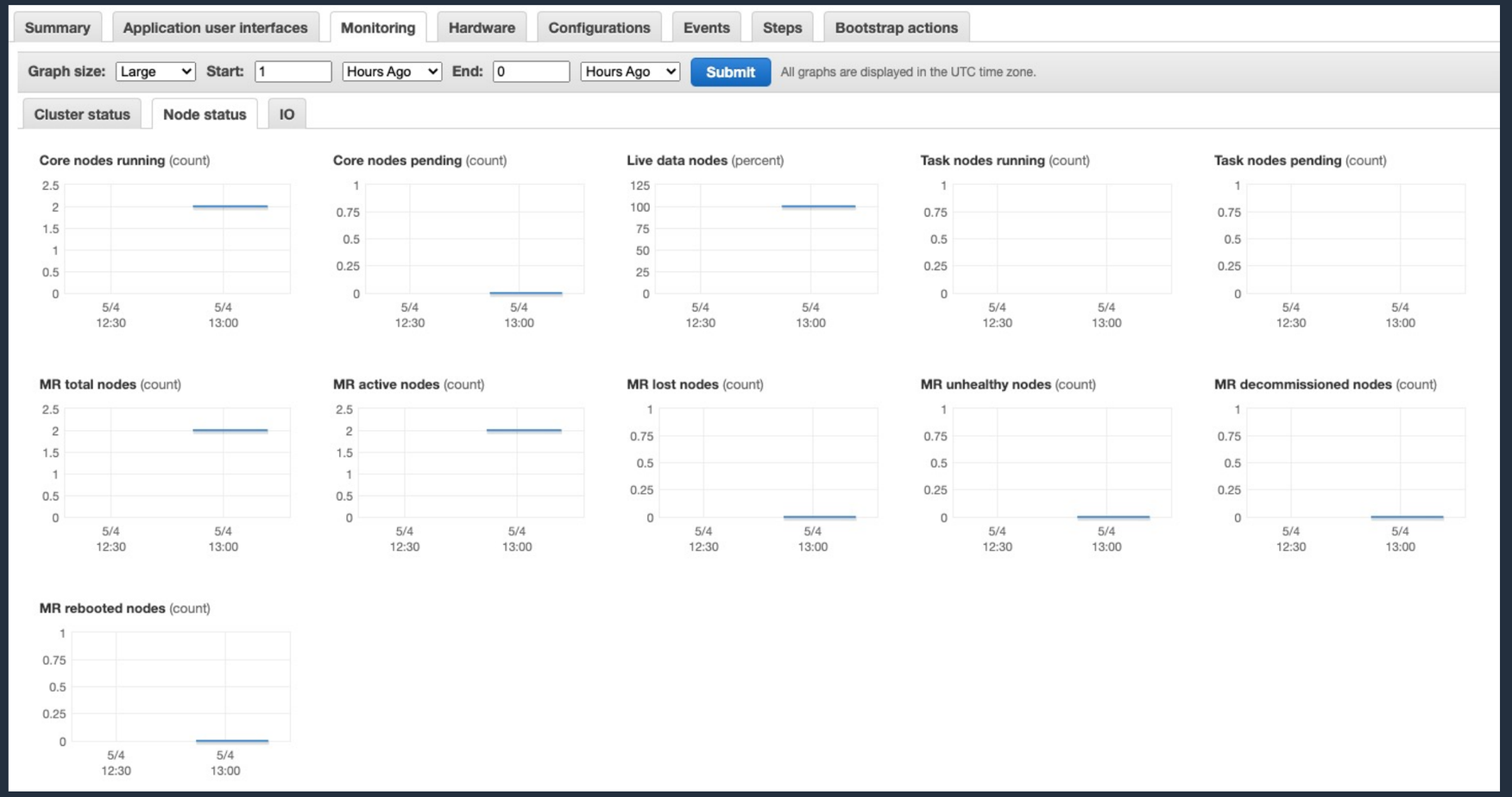
aws

CloudWatch Metric



https://docs.aws.amazon.com/emr/latest/ManagementGuide/UsingEMR_ViewingMetrics.html

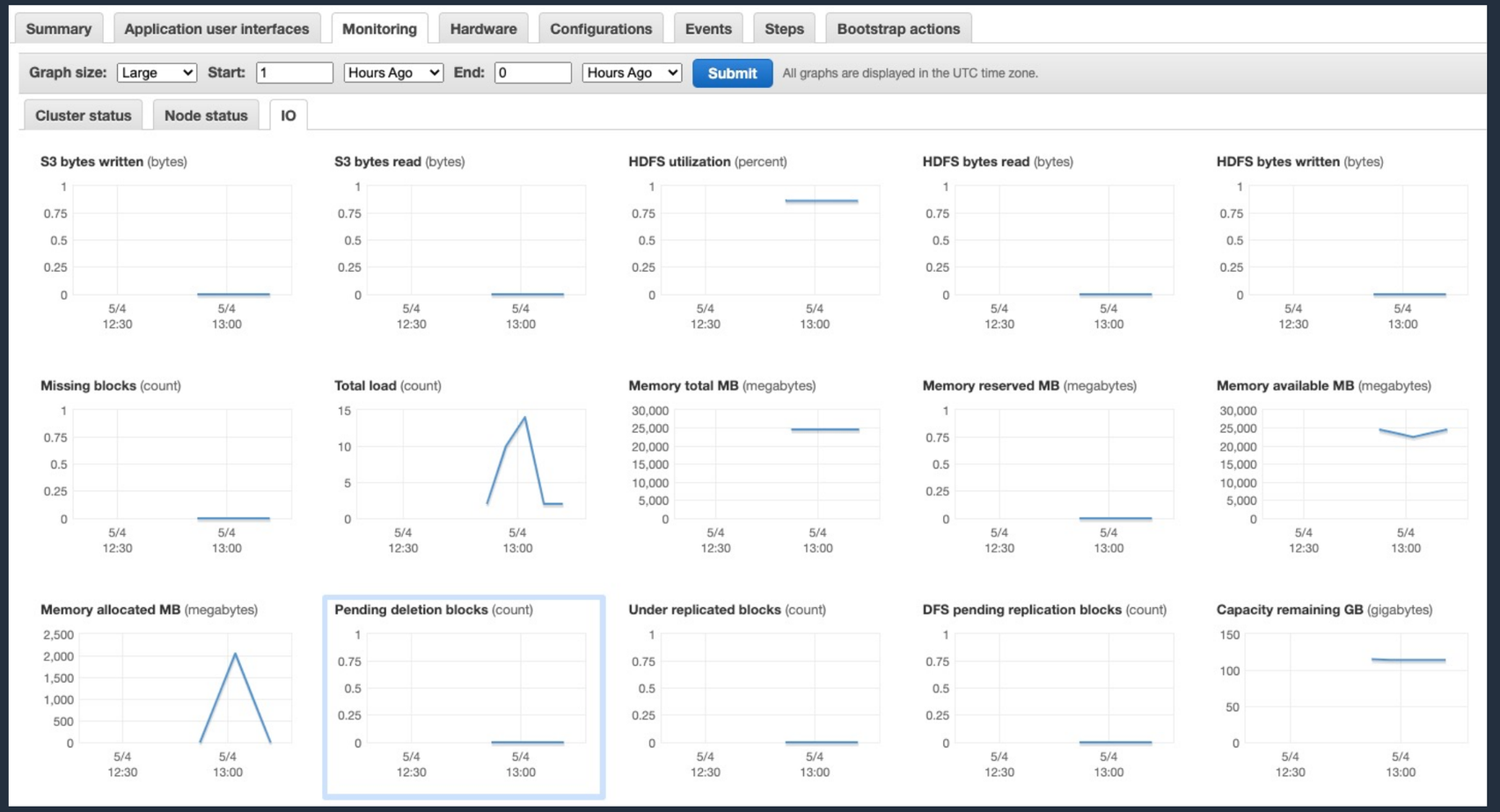
CloudWatch Metric



https://docs.aws.amazon.com/emr/latest/ManagementGuide/UsingEMR_ViewingMetrics.html

aws

CloudWatch Metric

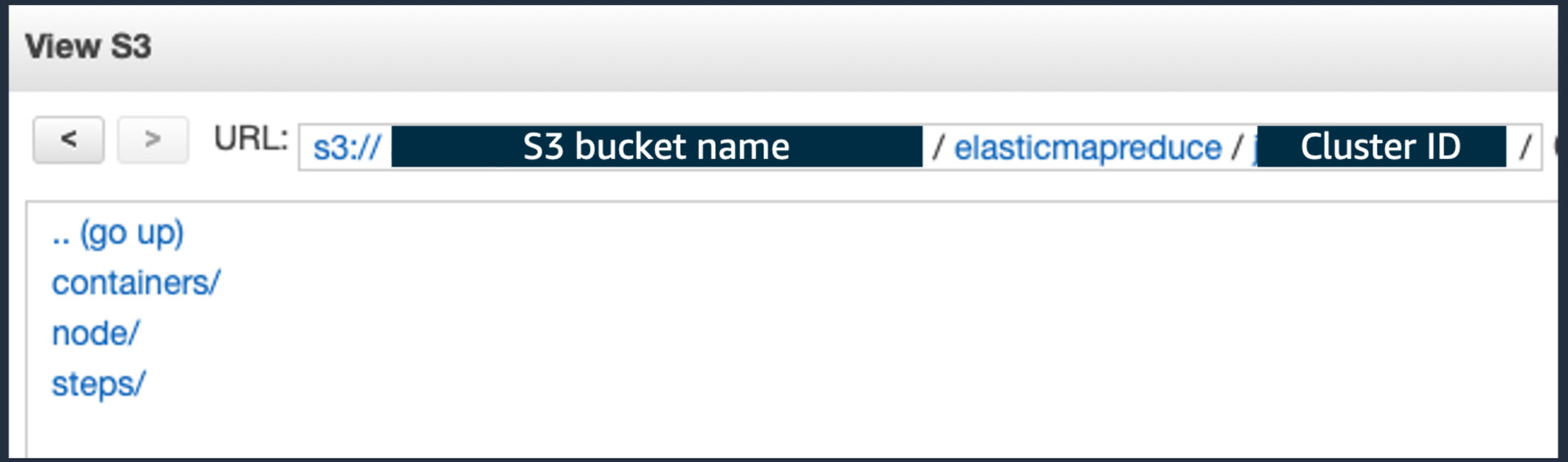


https://docs.aws.amazon.com/emr/latest/ManagementGuide/UsingEMR_ViewingMetrics.html

aws

Logging in S3

- containers/
- node/
- steps/



aws

Logging in S3 – containers/

- application_1620132671637_0001
  - container_1620132671637_0001_01_000001
    - Application Master log
  - container_1620132671637_0001_01_000002
  - …

- In cluster:
    - $ yarn logs --applicationId application_1620132671637_0001

Logging in S3 – Nodes/

- Node (EC2 instance ID)
    - applications
        - hadoop-hdfs
        - hadoop-yarn
        - application you selected (hive, hue etc)
    - daemons
        - instance-state
    - …
- In cluster:
    - $ ls /var/log/
    - $ ls /emr/

aws

Logging in S3 – steps/

- s-12L84L9JH7H8E
    - stderr.gz
    - …
- In cluster:
    - $ ls /var/log/ hadoop/steps/

aws

Lab Time Q1. Hive log level

Support 您好, 我們正在測試使用Hive, 但是日誌都是INFO Level 的日誌, 想請教如何開啟Hive 的日誌至Debug Level 呢?

Requirement: 請提供開啟Debug 的日誌的方式, 以及您如何確認日誌的Level (請寫出日誌位置及範例日誌內容)

aws

Lab Time Q2. Hive Trouble Shooting

- Material: github 中的hive_log.zip
- Support 您好:
- 我們開啟了一個集群, 並於Hive CLI 以及 Beeline 都提交了一個Hive 的 Query, 但是兩個都失敗了, 可以麻煩幫忙排查失敗的原因嗎?

- Requirement: 請分別寫出於Hive CLI 以及Beeline 中的
- 1) 執行時間 2) 失敗的Query 語句 3) 失敗的原因

aws

Lab Time Q3. Job Troubleshooting -1

- Material: github 中的 j-LTO0HUCBSSEC.zip
- Support 您好:
- 我們開啟了一個集群(ID: j-LTO0HUCBSSEC), 並且執行了一個MapReduce 的任務 (application_1620035588649_0001), 但是執行失敗,可以麻煩幫忙排查失敗的原因嗎?

- 集群資訊:
- Master Node: i-044984285ce9bbdd6
- Core Node: i-03c1484aa1febf850

- Requirement: 請寫出任務失敗的原因以及您排查的所有過程(請詳細寫下思路以及觀察過哪一些日誌內容)

aws

Lab Time Q4. Job Troubleshooting -2

Material: github 中的 j-NB7V3VXMXQ4P.zip & emr_new.zip

Support 您好:

我開啟了一個集群(j-NB7V3VXMXQ4P), 並執行了一個任務, 任務他卡住了過了很久後 我手動把任務關閉, 集群的日誌已經上傳(j-NB7V3VXMXQ4P.zip), 主節點的/emr 的日誌, 也已經打包上傳 (emr_new.zip),可以麻煩幫忙排查失敗的原因嗎?

集群資訊:

Master Node: i-0cbe55221fb1986a6

Core Node: i-0568cf5d4e1f0152c

Task Node: i-0cfeef686a4e1ed95

Requirement: 請寫出任務卡住的原因以及您排查的所有過程(請詳細寫下思路以及觀察過哪一些日誌內容)

aws