

ELS Bigdata 2021(EMR)

Jack Lin & Weiting Liang
2021/05/08

Material & Lab answer submission

- https://github.com/jack555023/ELS_2021
- [https://docs.google.com/forms/d/e/1FAIpQLSdg3_6tqvOR4MEKtzqvp
gMxOuyst14NRhyvx-vwf-dAQ5eaAQ/viewform](https://docs.google.com/forms/d/e/1FAIpQLSdg3_6tqvOR4MEKtzqvpgMxOuyst14NRhyvx-vwf-dAQ5eaAQ/viewform)

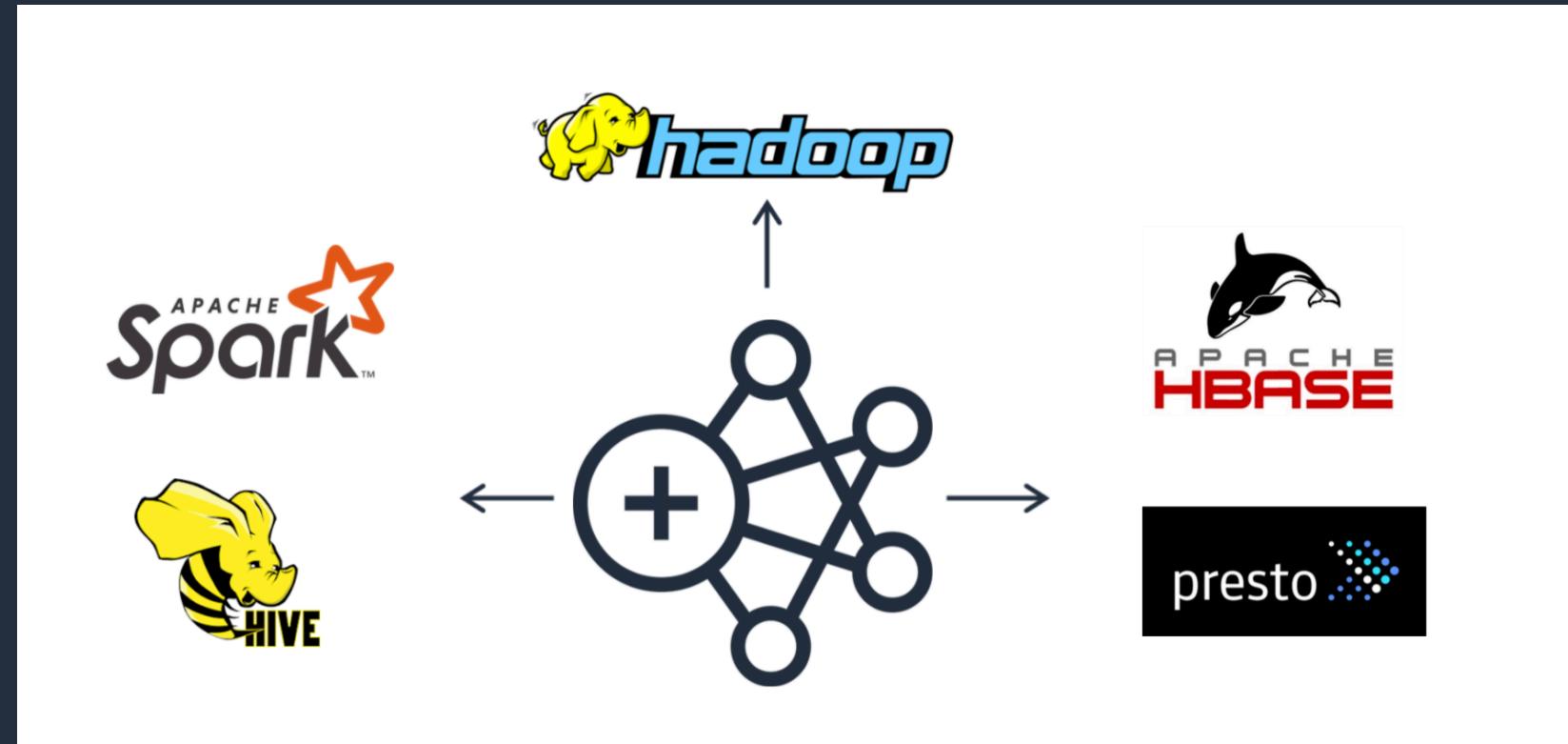
EMR & Hadoop



What is EMR?

Elastic Map Reduce
Hadoop FrameWork

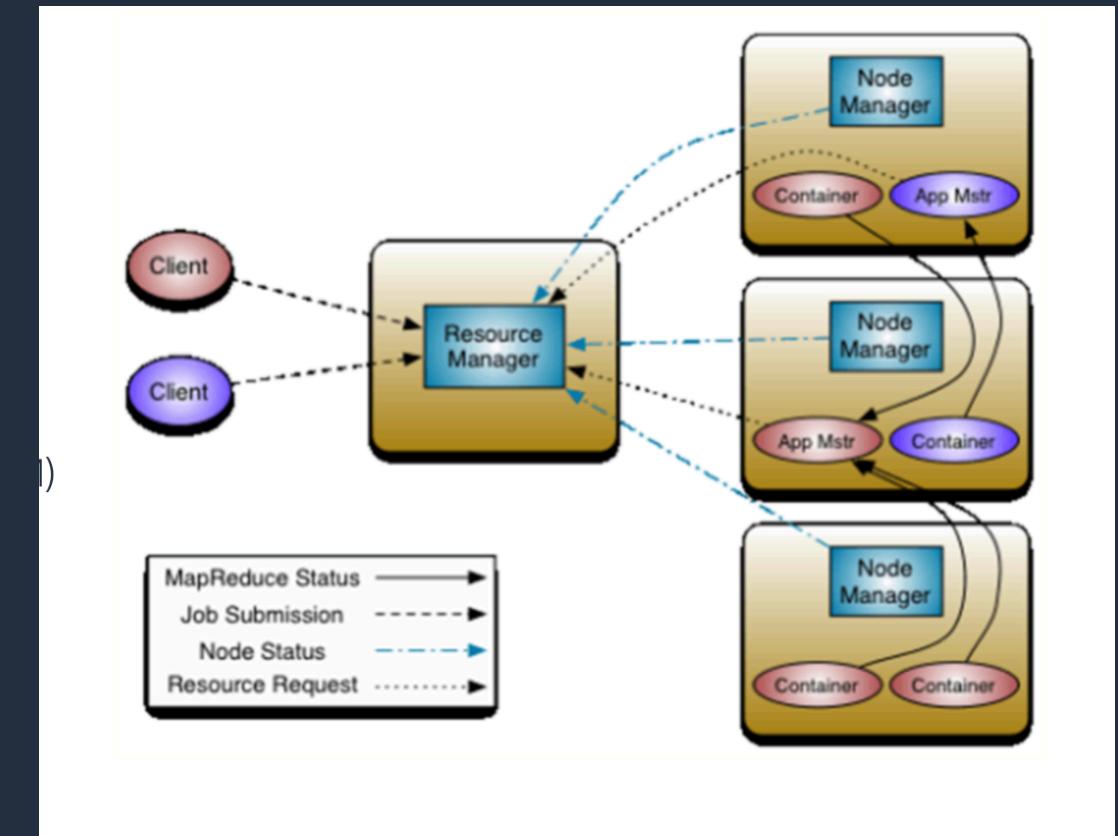
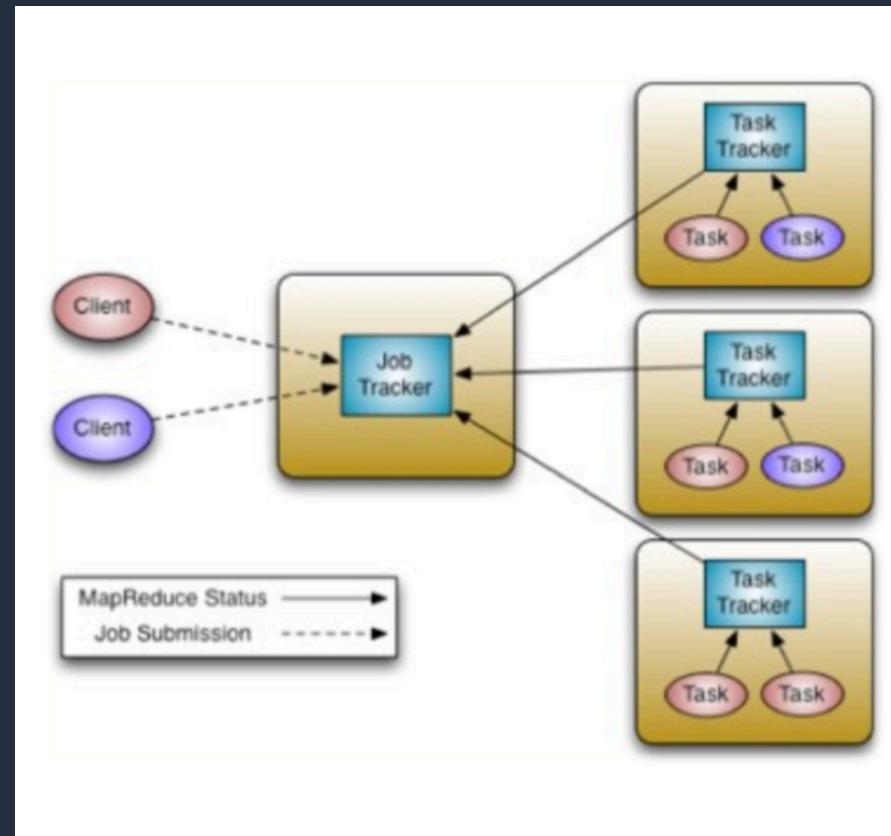
Why EMR :
Easily create/manage
Auto-scaling for nodes
Amazon S3 (EMRFS)



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)



EMR's Node.



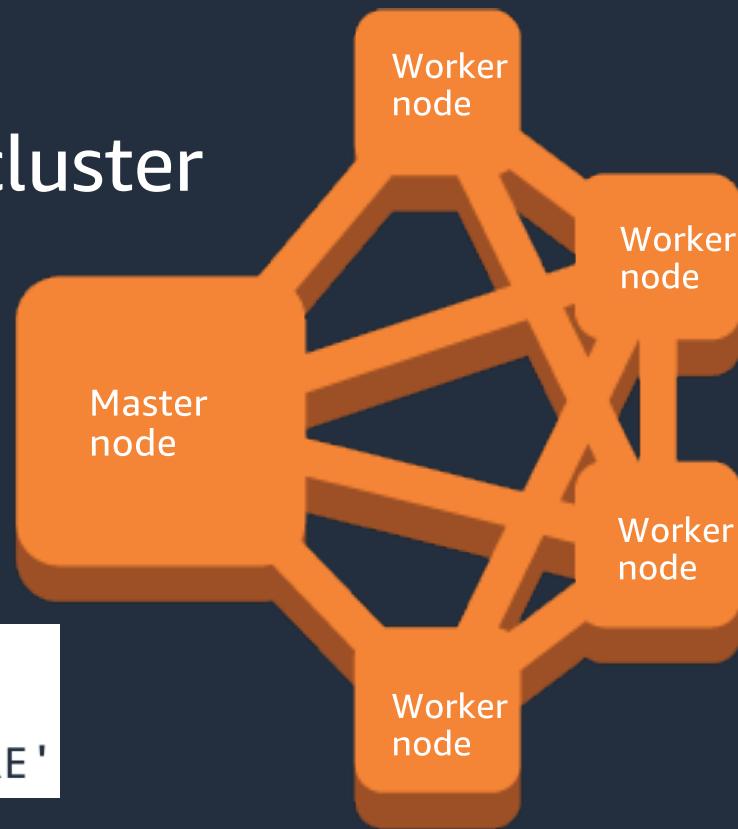
Infrastructure

Master: Management & Monitoring the cluster

Core: Storage & Job execution

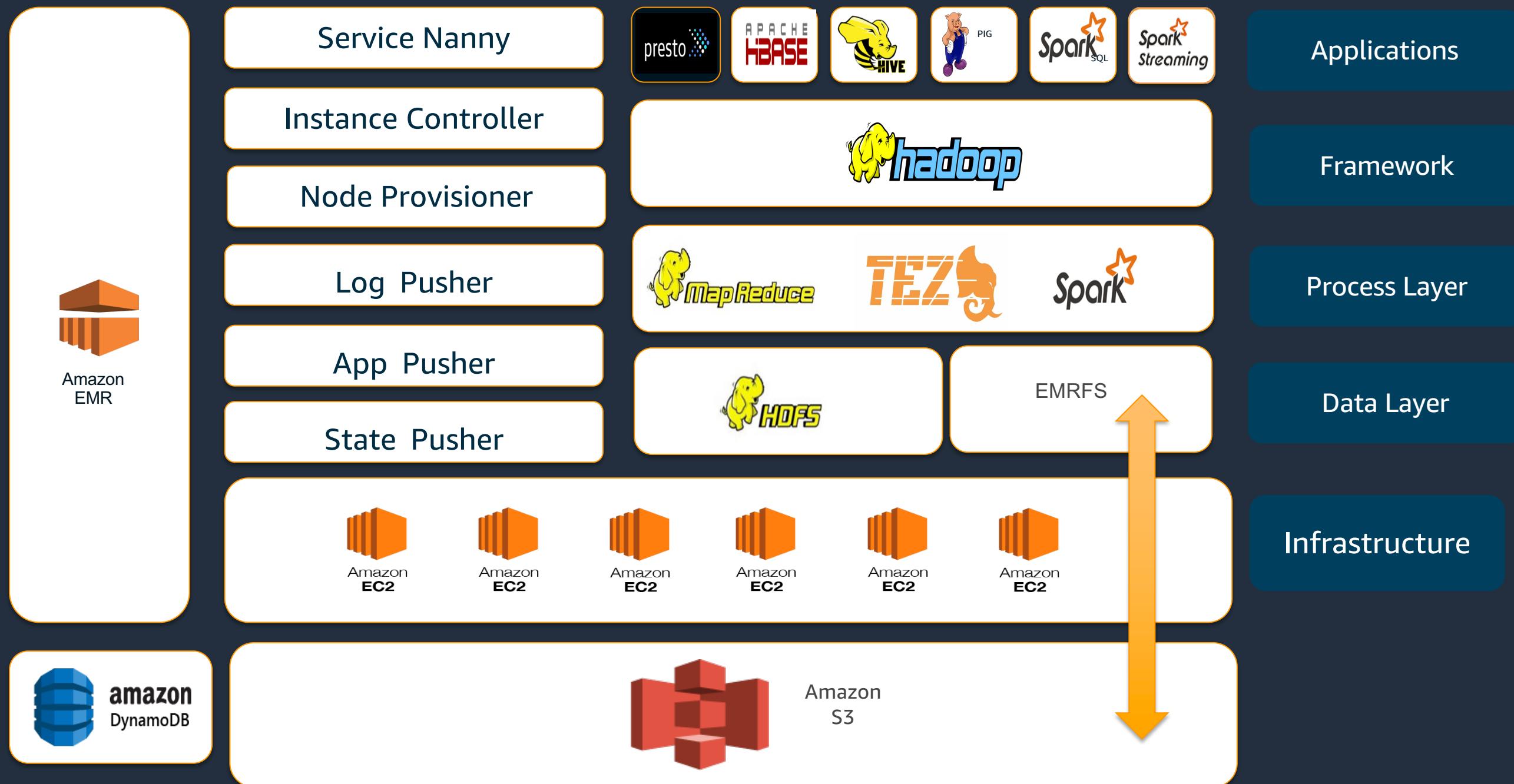
Task: Job execution

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



Amazon EMR cluster

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 (Starting Service ...etc)
- LogPusher (Push the log !!!)
- AppPusher(Spark History Server Log)
- StatePusher

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 JARetc.
 - Launching Cluster & Running Cluster

Launching EMR Cluster



Launch EMR

- chmod 400

EC2 > Key pairs > Create key pair

Create key pair

Key pair
A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name
 The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

File format
 pem
For use with OpenSSH
 ppk
For use with PuTTY

Tags (Optional)
No tags associated with the resource.
[Add tag](#)
You can add 50 more tags.

[Cancel](#) [Create key pair](#)

Launch EMR

Create Cluster - Advanced Options [Go to quick options](#)

Step 1: Software and Steps

Step 2: Hardware

Step 3: General Cluster Settings

Step 4: Security

Software Configuration

Release **emr-5.30.0**

- | | | |
|--|---|--|
| <input checked="" type="checkbox"/> Hadoop 2.8.5 | <input type="checkbox"/> Zeppelin 0.8.2 | <input type="checkbox"/> Livy 0.7.0 |
| <input type="checkbox"/> JupyterHub 1.1.0 | <input type="checkbox"/> Tez 0.9.2 | <input type="checkbox"/> Flink 1.10.0 |
| <input type="checkbox"/> Ganglia 3.7.2 | <input type="checkbox"/> HBase 1.4.13 | <input checked="" type="checkbox"/> Pig 0.17.0 |
| <input checked="" type="checkbox"/> Hive 2.3.6 | <input type="checkbox"/> Presto 0.232 | <input type="checkbox"/> ZooKeeper 3.4.14 |
| <input type="checkbox"/> MXNet 1.5.1 | <input type="checkbox"/> Sqoop 1.4.7 | <input type="checkbox"/> Mahout 0.13.0 |
| <input checked="" type="checkbox"/> Hue 4.6.0 | <input type="checkbox"/> Phoenix 4.14.3 | <input type="checkbox"/> Oozie 5.2.0 |
| <input type="checkbox"/> Spark 2.4.5 | <input type="checkbox"/> HCatalog 2.3.6 | <input type="checkbox"/> TensorFlow 1.14.0 |

Multiple master nodes (optional)

Use multiple master nodes to improve cluster availability. [Learn more](#)

AWS Glue Data Catalog settings (optional)

Use for Hive table metadata

Edit software settings

Enter configuration Load JSON from S3

```
classification=config-file-name,properties=[myKey1=myValue1,myKey2=myValue2]
```

Steps (optional)

A step is a unit of work you submit to the cluster. For instance, a step might contain one or more Hadoop or Spark jobs. You can also submit additional steps to a cluster after it is running. [Learn more](#)

Concurrency: Run multiple steps at the same time to improve cluster utilization

After last step completes: Clusters enters waiting state

Cluster auto-terminates

Step type **Select a step**

Add step

Cancel

Next

Launch EMR

Networking

Use a Virtual Private Cloud (VPC) to process sensitive data or connect to a private network. Launch the cluster into a VPC with a public, private or shared subnet. Subnets may be associated with an AWS Outpost or AWS Local Zone.

Launch the cluster into a VPC with a public, private, or shared subnet. Subnets may be associated with an AWS Outpost or AWS Local Zone.

Network

EC2 Subnet

Cluster Nodes and Instances

Choose the instance type, number of instances, and a purchasing option. [Learn more about instance purchasing options](#)

Console options for automatic scaling have changed. [Learn more](#)

Node type	Instance type	Instance count	Purchasing option
Master Master - 1 <input type="button" value="i"/>	m5.xlarge <input type="button" value="i"/> 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 64 GiB <input type="button" value="i"/> <input type="button" value="i"/> Add configuration settings <input type="button" value="i"/>	1 Instances	<input checked="" type="radio"/> On-demand <input type="button" value="i"/> <input type="radio"/> Spot <input type="button" value="i"/> Use on-demand as max price <input type="button" value="▼"/>
Core Core - 2 <input type="button" value="i"/>	m5.xlarge <input type="button" value="i"/> 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 64 GiB <input type="button" value="i"/> <input type="button" value="i"/> Add configuration settings <input type="button" value="i"/>	2 Instances	<input checked="" type="radio"/> On-demand <input type="button" value="i"/> <input type="radio"/> Spot <input type="button" value="i"/> Use on-demand as max price <input type="button" value="▼"/>
Task Task - 3 <input type="button" value="i"/>	m5.xlarge <input type="button" value="i"/> 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 64 GiB <input type="button" value="i"/> <input type="button" value="i"/> Add configuration settings <input type="button" value="i"/>	0 Instances	<input checked="" type="radio"/> On-demand <input type="button" value="i"/> <input type="radio"/> Spot <input type="button" value="i"/> Use on-demand as max price <input type="button" value="▼"/>

[+ Add task instance group](#)

Total core and task units

2 Total units

Launch EMR

Security Options

EC2 key pair No key pairs found i

Cluster visible to all IAM users in account i

Permissions i

Default Custom

Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.

EMR role [EMR_DefaultRole](#) i

EC2 instance profile [EMR_EC2_DefaultRole](#) i

Auto Scaling role [EMR_AutoScaling_DefaultRole](#) i

► Security Configuration

► EC2 security groups

i No EC2 key pair has been selected, so you will not be able to SSH to this cluster or connect to HUE (unless you are using a VPN). [Learn how to create an EC2 Key Pair](#).

[Cancel](#) [Previous](#) [Create cluster](#)

Demo & Cluster creation.



Introduction to Hive and Tez



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)

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)
```

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)
```

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
```

Using Hue to Run Hive Scripts

The screenshot shows the Hue web interface for running Hive scripts. The left sidebar contains icons for various data sources and operations. The main area has a search bar at the top. Below it, there are tabs for 'Hive' (selected), 'Add a name...', and 'Add a description...'. A tree view on the left shows a 'default' database with a single 'Tables' node containing '(1)' and a 'foo' folder. The central pane displays a query editor with the following content:

```
1 SELECT * from default.foo;
```

Below the query, the terminal output shows the execution results:

```
INFO : Completed compiling command(queryId=hive_20210502072803_4e51eb0b-d54d-4dc1-8f8f-be64ee3b00f7); Time taken: 0.001 seconds
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Executing command(queryId=hive_20210502072803_4e51eb0b-d54d-4dc1-8f8f-be64ee3b00f7): SELECT * from default.foo
INFO : Completed executing command(queryId=hive_20210502072803_4e51eb0b-d54d-4dc1-8f8f-be64ee3b00f7); Time taken: 0.001 seconds
INFO : OK
```

At the bottom, there are tabs for 'Query History', 'Saved Queries', and 'Results (1)'. The 'Results (1)' tab is selected, showing a table titled 'foo.x' with one row containing two columns labeled '1' and '1'.

Hive Features

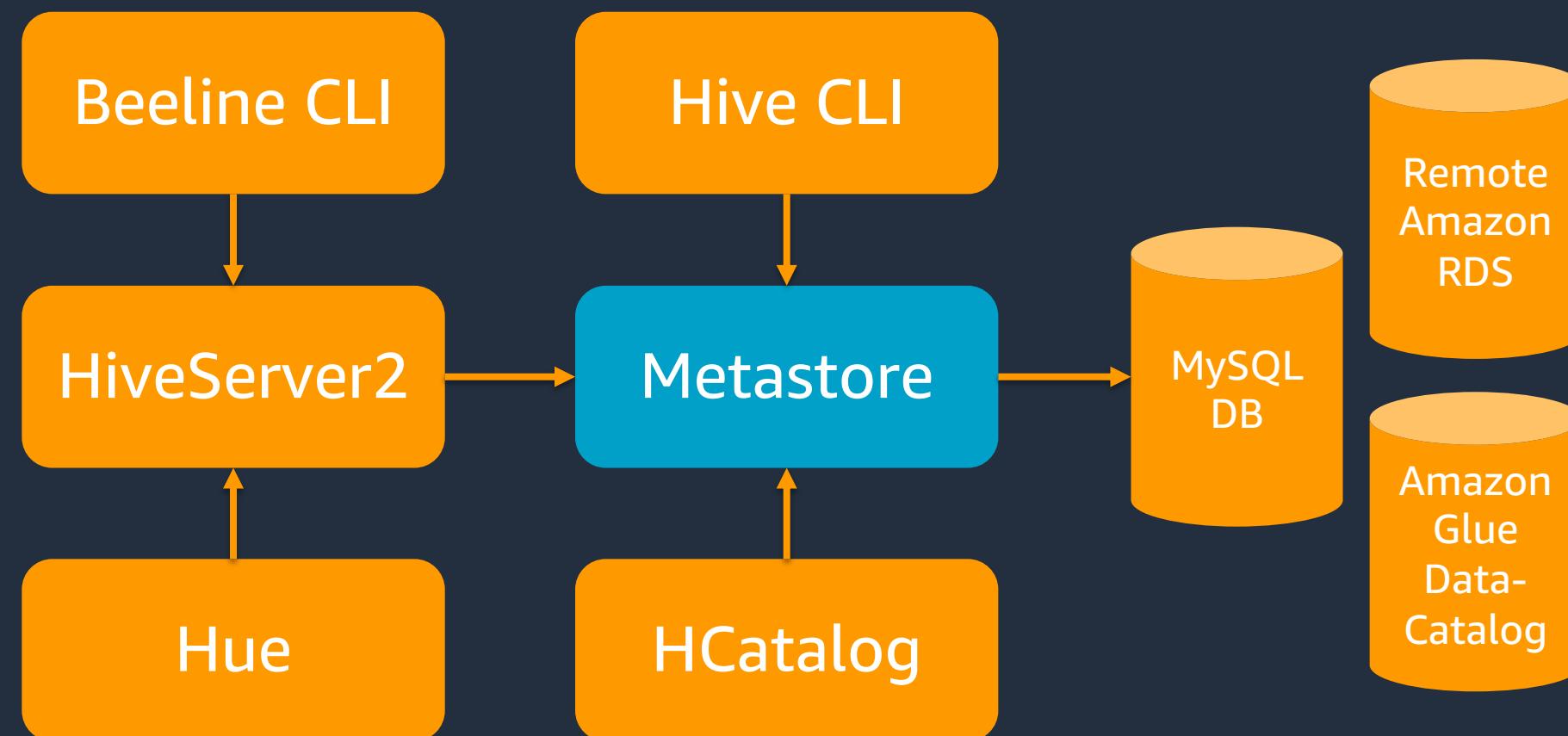
- Ad-hoc inserts and updates
- Support for columnar formats like ORC / Parquet
- Partitions
 - Divides data into folders allowing you to query something without going through all the data
 - Supports dynamic partitioning allowing you to output to different partitions based on what your data contains.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

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

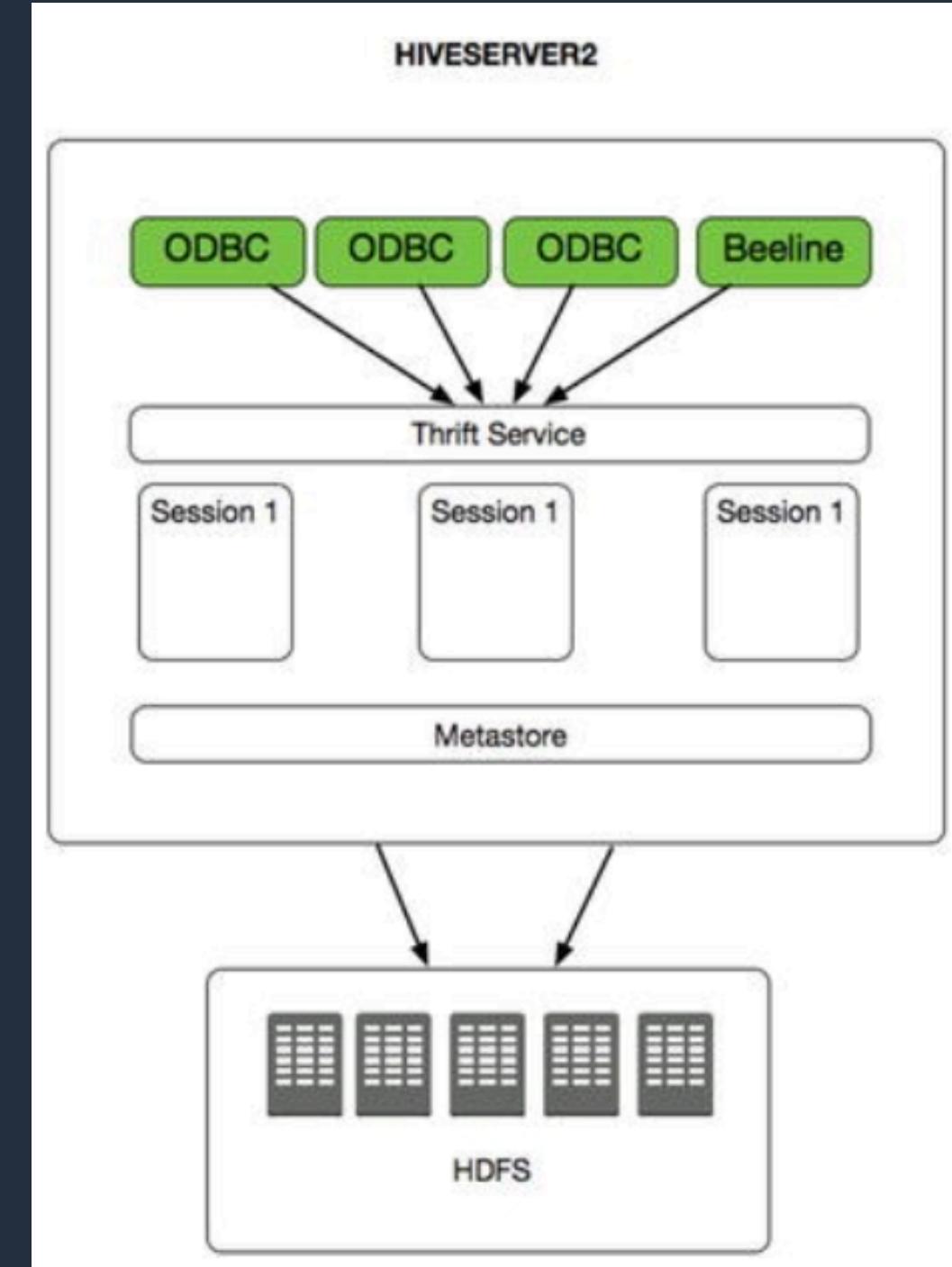
Remote Metastore Mode

- 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.



HiveServer2 (HS2)

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



How Does HiveServer2 Work?

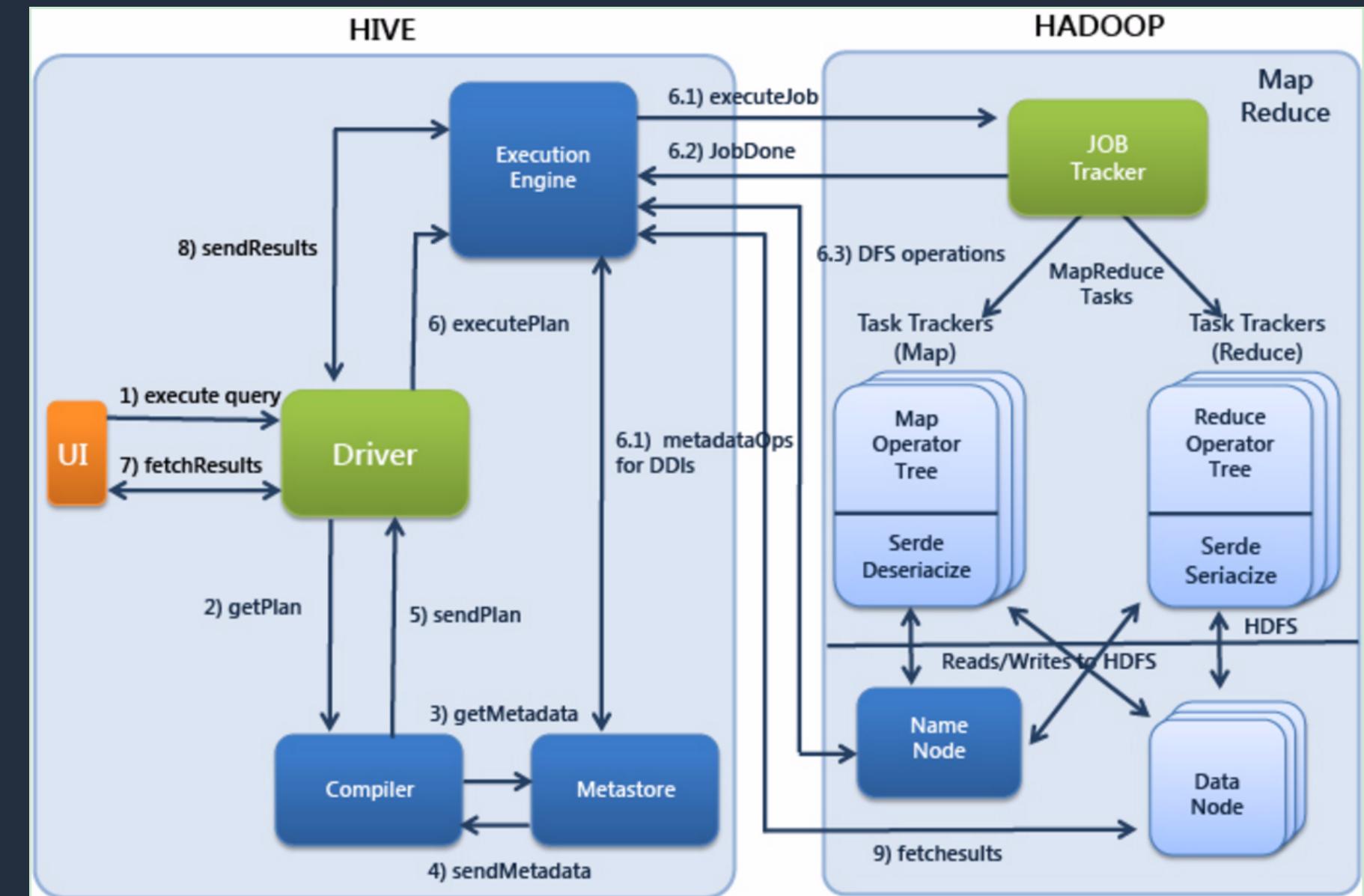
1. JDBC client (e.g. Beeline) creates a HiveConnection by initiating a transport connection (e.g. TCP connection) followed by an OpenSession API call to get a SessionHandle.
2. HiveStatement is executed and an ExecuteStatement API call is made from the Thrift client. In the API call, SessionHandle info is passed to the server with query info.
3. HS2 server receives the request and ask the driver (CommandProcessor) for query parsing and compilation.
4. Driver kicks off background job to talk to Hadoop, and immediately returns a response to the client. Response contains an OperationHandle created from the server side.
5. Client uses the OperationHandle to talk to HS2 to poll the status of the query execution.

HiveQL

- HiveQL integrates SQL-like queries into underlying Java without the need to implement queries in low-level Java API.
- Divided into two languages:
 - Data Definition Language (DDL)
 - Used for creating, altering, and dropping 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

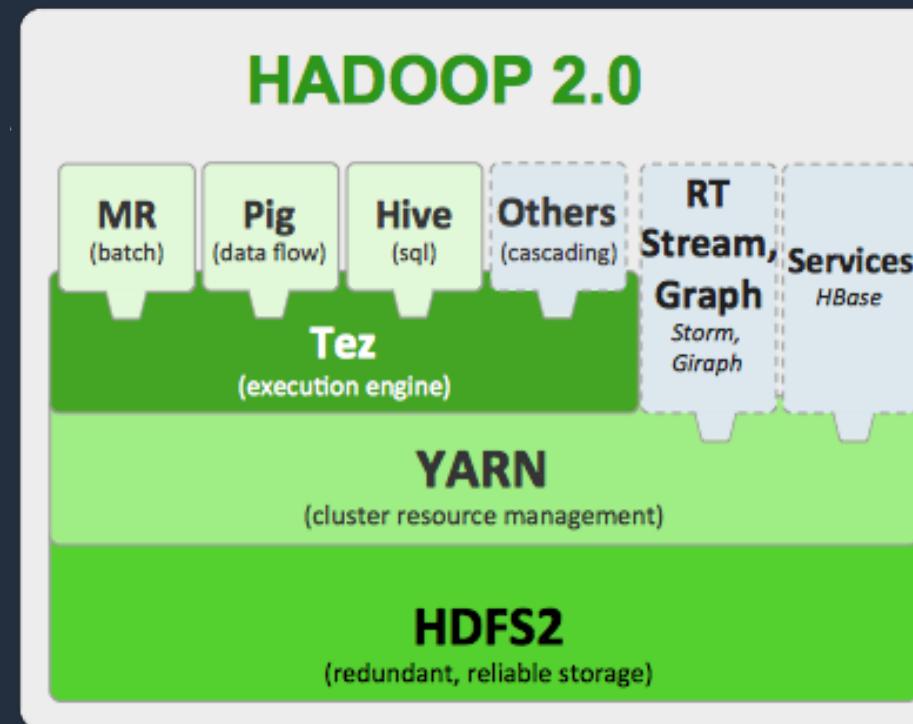
Hive on MapReduce (MR) Engine

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



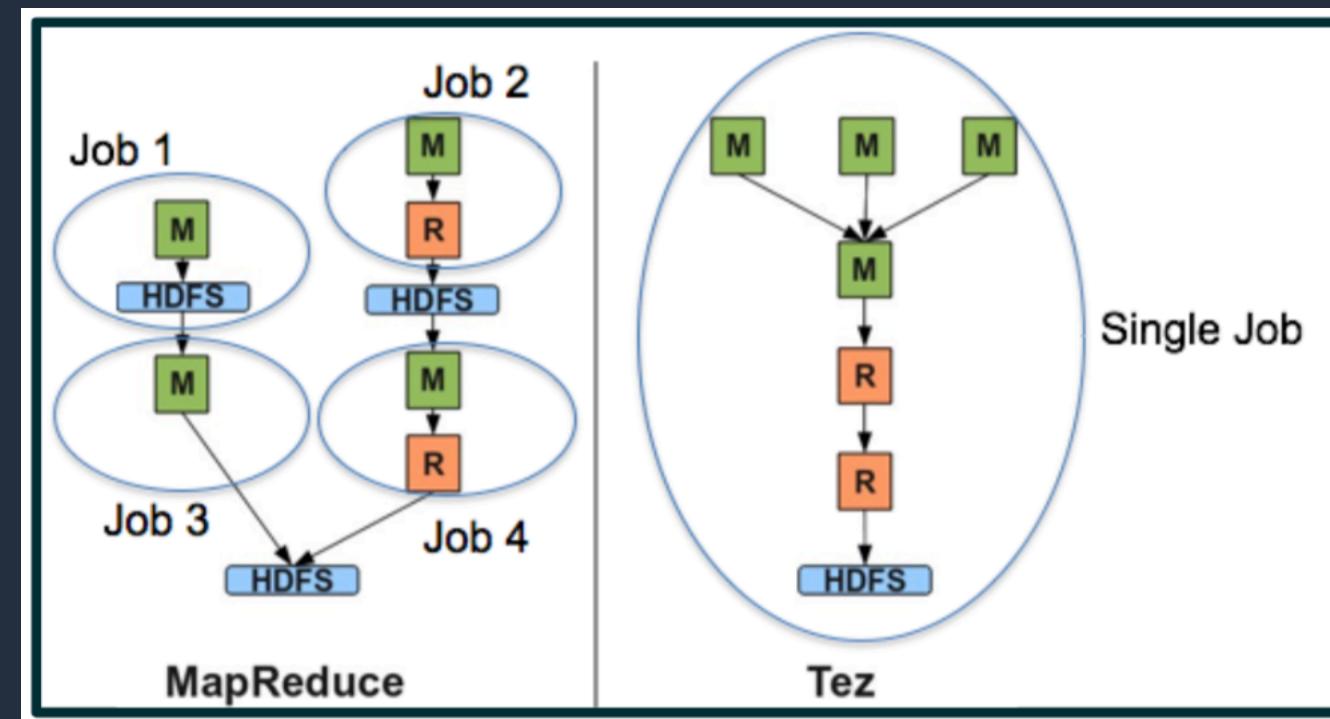
Hive on Tez

- 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.



Tez Engine v.s. MR Engine

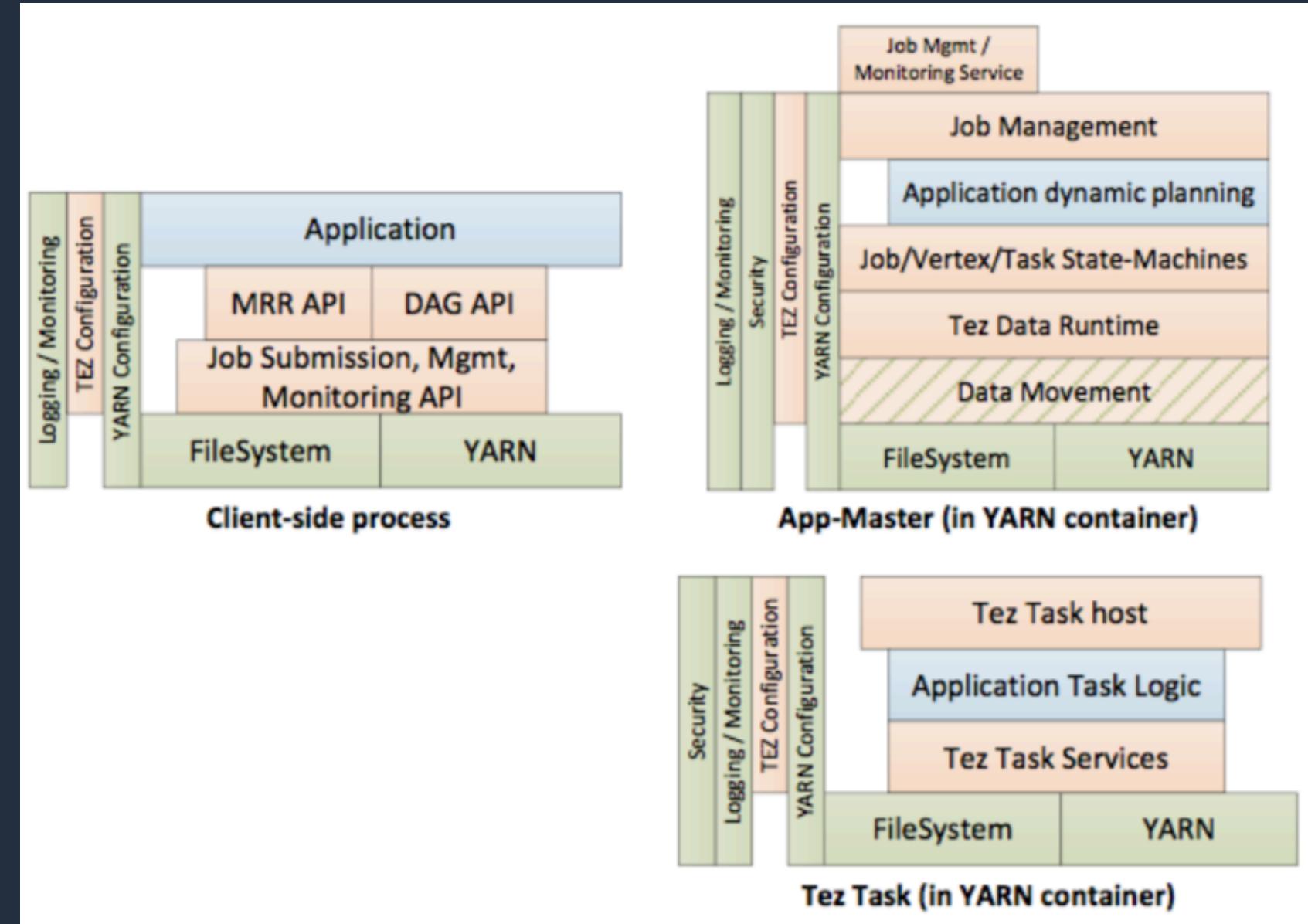
- Tez solves the interactive query slowness problem.
- Solves MR shortcomings by balancing performance and throughput.
- Can split map and reduce jobs into smaller tasks.



Tez Components

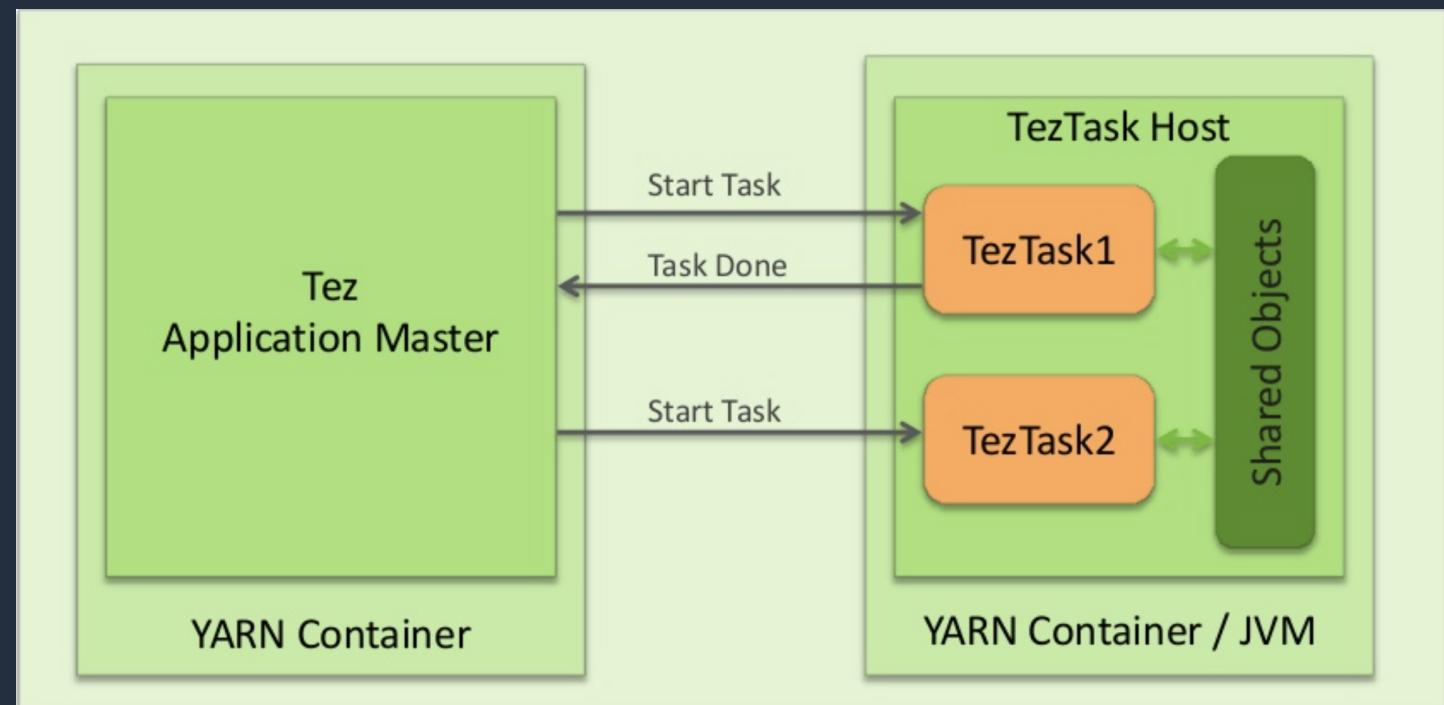
Tez AppMaster is responsible for

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



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.



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

TEZ Home / Application [HIVE-e7997ca6-ac73-41ae-9a60-50c4d2358a8f]

Application Details DAGs Configurations

YARN App Details		YARN App Description	
Status	RUNNING	Application Tracking URL	application_1620132671637_0002
Final Status	! UNDEFINED	Application Name	HIVE-e7997ca6-ac73-41ae-9a60-50c4d2358a8f
Start Time	04 May 2021 21:00:06	Queue	default
End Time	Not Available!	Application Type	TEZ
Duration	Not Available!	User	hadoop

TEZ Home / Application [HIVE-e7997ca6-ac73-41ae-9a60-50c4d2358a8f] / DAGs

Application Details DAGs Configurations

Search...

Dag Name	Id	Submitter	Status	Progress	Start Time
insert into table foo values ('2')(Stage-1)	dag_1620132671637_...	hadoop	SUCCEEDED	100%	04 May 2021 21:05:24
insert into table foo values ('1')(Stage-1)	dag_1620132671637_...	hadoop	SUCCEEDED	100%	04 May 2021 21:01:03

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>

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. Persistent UIs are available even when the cluster is stopped.

Application user interface
YARN timeline server
Tez UI

On-cluster application user interfaces

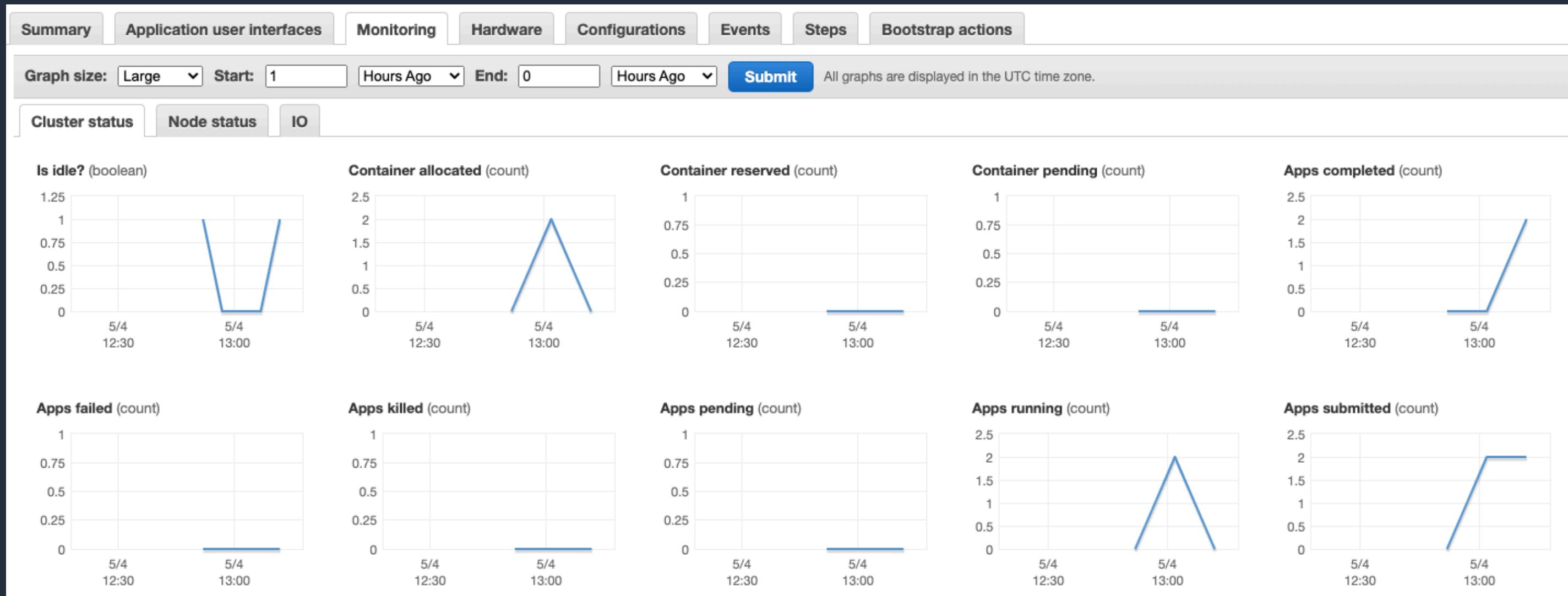
On-cluster UI are available only while clusters are running. Because they are hosted on the master node, on-cluster UI require SSH access to the master node.

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:

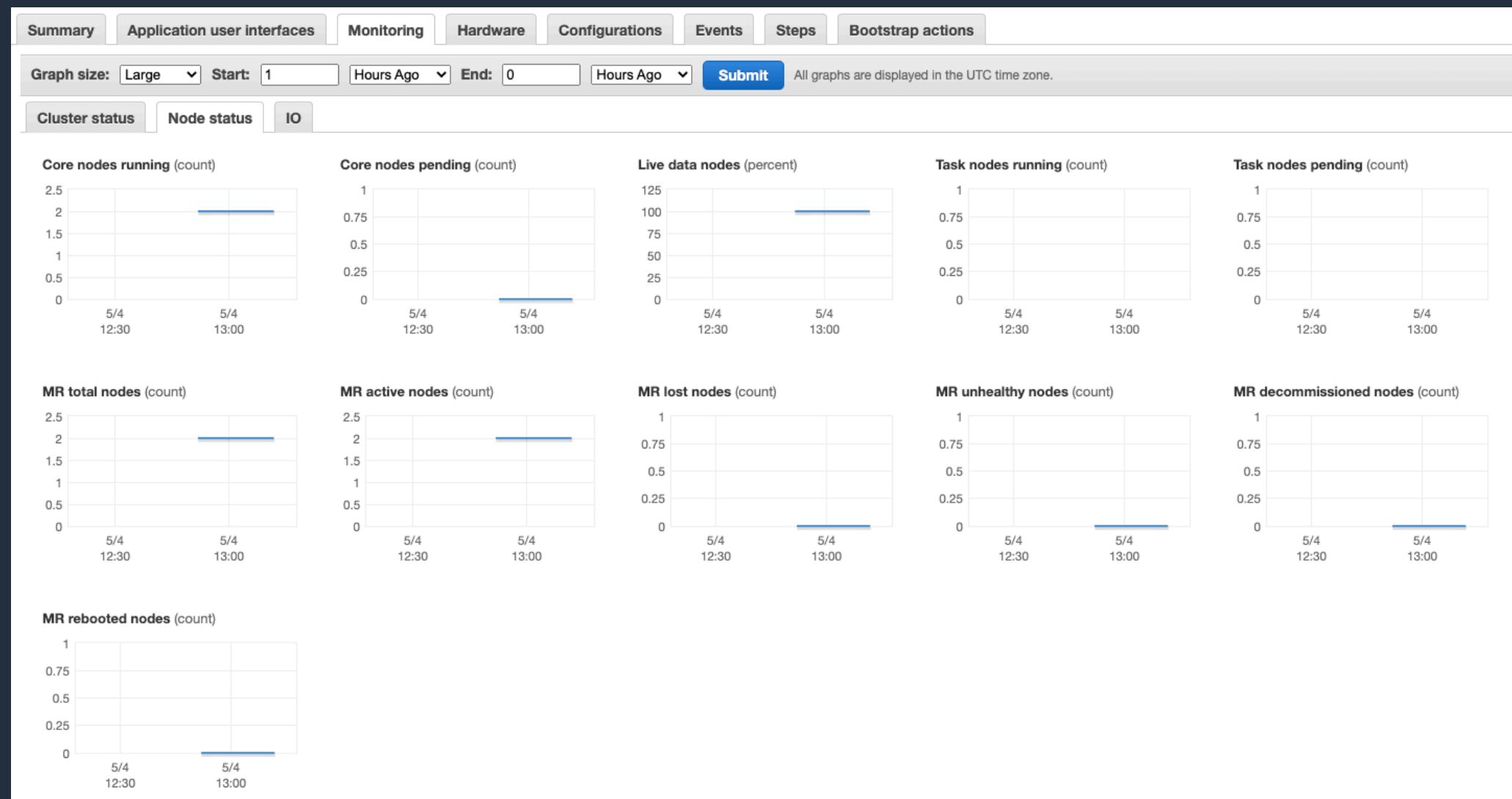
Application	User interface URL
HDFS Data Node	http://ec2-000-000-000-000.compute-1.amazonaws.com:50075/
Node Manager	http://ec2-000-000-000-000.compute-1.amazonaws.com:8042/

CloudWatch Metric



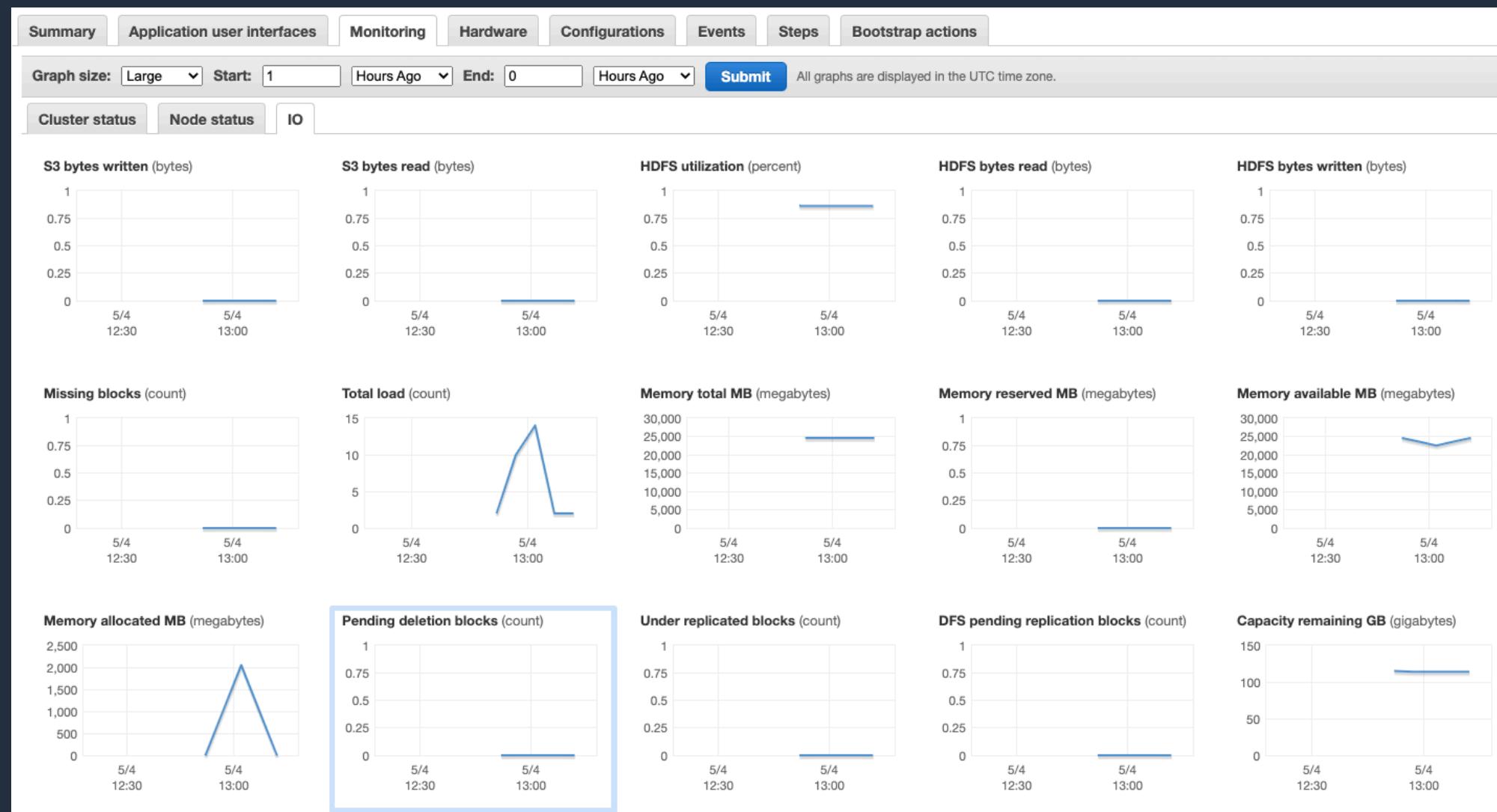
https://docs.aws.amazon.com/emr/latest/ManagementGuide/UsingEMR_ViewingMetrics.html

CloudWatch Metric



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

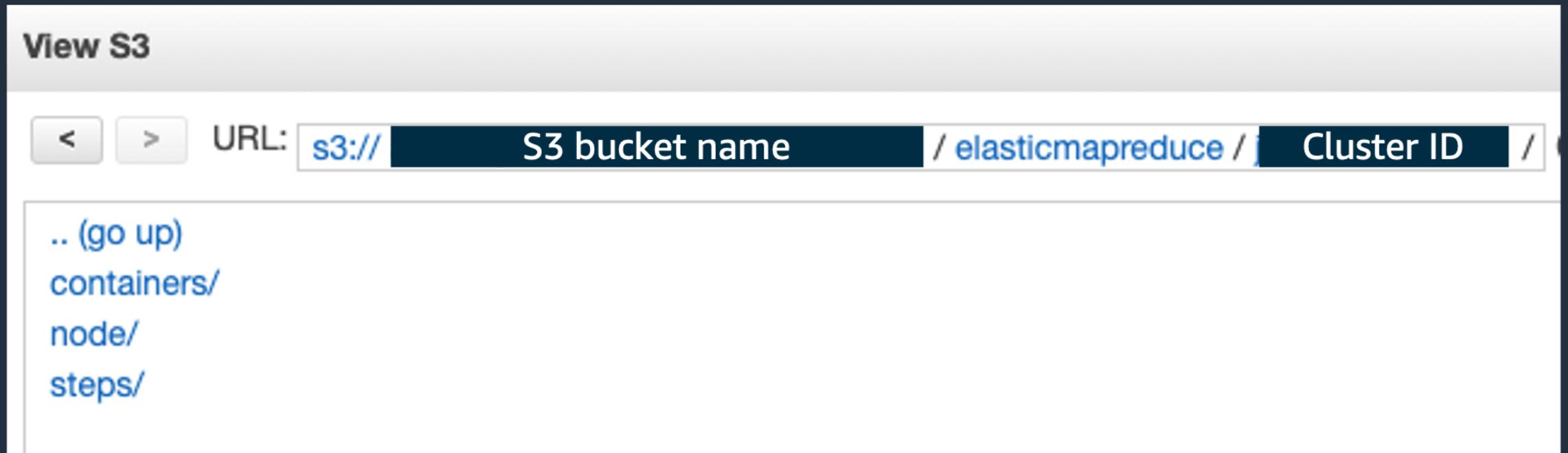
CloudWatch Metric



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

Logging in S3

- containers/
- node/
- steps/



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/

Logging in S3 – steps/

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

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)

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

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-03c1484aa1febfb850
- 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)

Lab Time Q4. Job Troubleshooting -2

Material: github 中的 j-NB7V3VXMXQ4P.zip & emr.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.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)