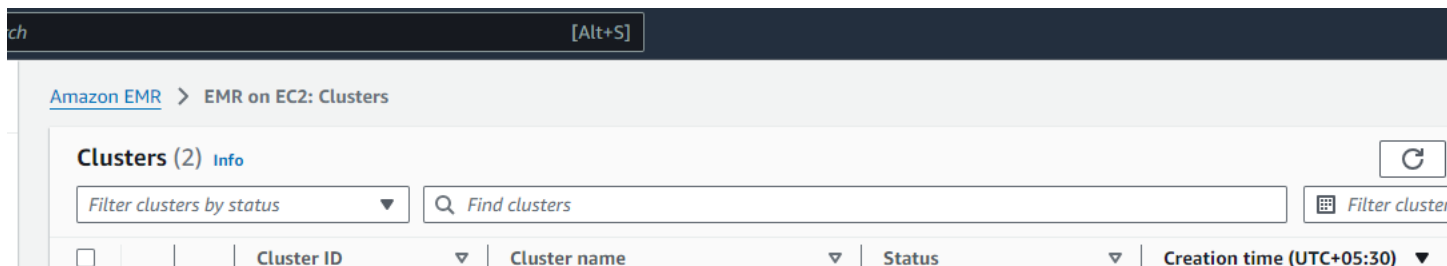


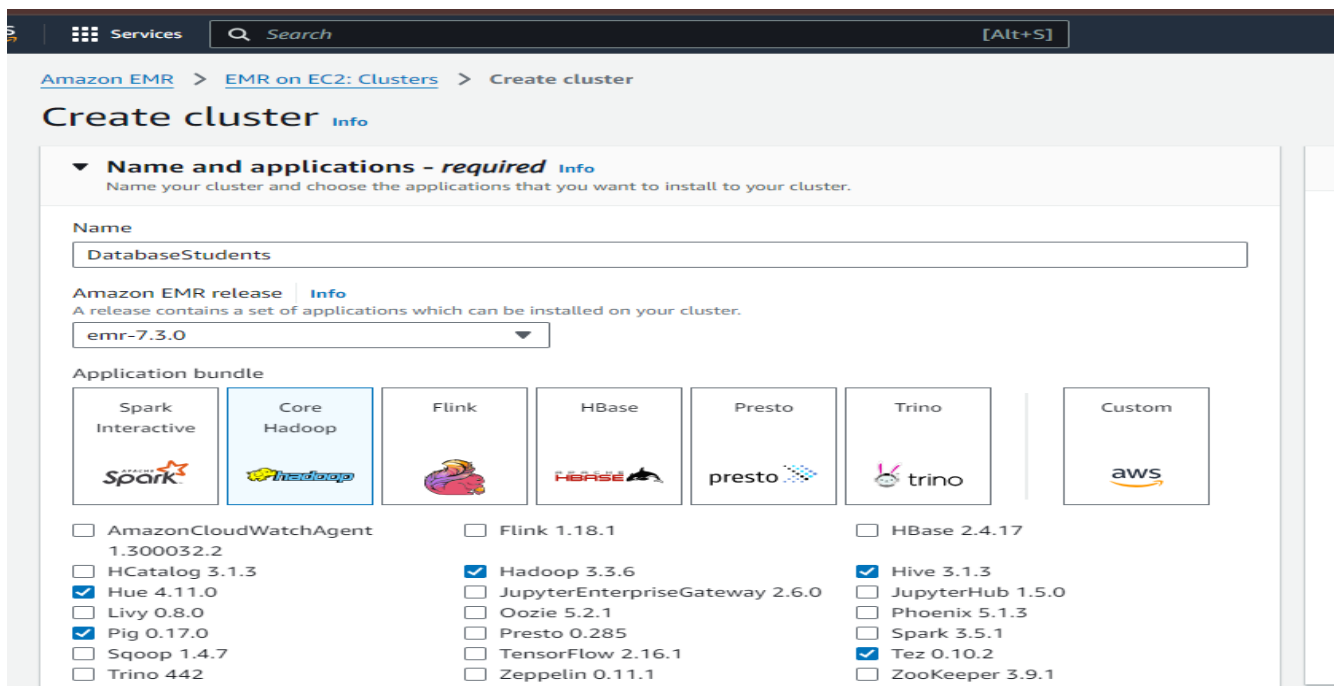
# CLOUD COMPUTING PRACTICAL ASSIGNMENT NO:10

Installation and configuration of cloud Hadoop and demonstrate simple query Prepare Screen shots file and also write down the steps. Make single word or PDF file .

Step I : Go to EMR service and click on to create Cluster.



Step II: give the name to cluster and in application Bundle select the Core Hadoop.



Step III: In Cluster Configuration keep uniform instance group as it is.

The screenshot shows the 'Cluster configuration - required' section of the Amazon EMR console. It has a sub-header 'Choose a configuration method for the primary, core, and task node groups for your cluster.' There are two radio button options: 'Uniform instance groups' (selected) and 'Flexible instance fleets'. The 'Uniform instance groups' option is highlighted with a blue border and contains the text: 'Choose the same EC2 instance type and purchasing option (On-Demand or Spot) for all nodes in your node group. [Learn more](#)'. Below this, the 'Uniform instance groups' section is expanded, showing 'Primary' configuration. It includes a dropdown menu for 'Choose EC2 instance type' with 'm5.xlarge' selected, showing details: '4 vCore 16 GiB memory', 'EBS only storage', 'On-Demand price: -', and 'Lowest Spot price: -'. There is an 'Actions' button with a dropdown arrow. Below the dropdown is a checkbox for 'Use high availability' with the text: 'Launch highly available, more resilient cluster with three primary nodes on On-Demand Instances. This configuration applies for the lifetime of your cluster. [Learn more](#)'. At the bottom, there is a section for 'Node configuration - optional'.

Step IV: Keep all as it is only in Security Configuration and EC2 key pair select the key-pair.

The screenshot shows the 'Security configuration and EC2 key pair' section of the Amazon EMR console. It has a sub-header 'Choose a security configuration or create a new one that you can reuse with other clusters.' Under 'Security configuration', it says 'Select your cluster encryption, authentication, authorization, and instance metadata service settings.' There are three buttons: 'Choose a security configuration' (with a search icon), 'Browse' (with an external link icon), and 'Create security configuration' (with an external link icon). Below this, under 'Amazon EC2 key pair for SSH to the cluster', there is a search bar with 'radhe-Krishn' entered, a close button (X), and two buttons: 'Browse' and 'Create key pair' (with an external link icon).

Step V: In Amazon EMR service role and ec2 instance profile for Amazon EMR select EMR\_DefaultRole and EMR\_EC2\_DefaultRole respectively. and click on to create cluster.

Services

Search

[Alt+S]

Amazon EC2 key pair for SSH to the cluster

radhe-Krishn

Browse

Create key pair

Identity and Access Management (IAM) roles - required

Choose or create a service role and instance profile for the EC2 instances in your cluster.

Amazon EMR service role

The service role is an IAM role that Amazon EMR assumes to provision resources and perform service-level actions with other AWS services.

Choose an existing service role

Select a default service role or a custom role with IAM policies attached so that your cluster can interact with other AWS services.

Create a service role

Let Amazon EMR create a new service role so that you can grant and restrict access to resources in other AWS services.

Service role

EMR\_DefaultRole

EC2 instance profile for Amazon EMR

The instance profile assigns a role to every EC2 instance in a cluster. The instance profile must specify a role that can access the resources for your steps and bootstrap actions.

Choose an existing instance profile

Select a default role or a custom instance profile with IAM policies attached so that your cluster can interact with your resources in Amazon S3.

Create an instance profile

Let Amazon EMR create a new instance profile so that you can specify a custom set of resources for it to access in Amazon S3.

Instance profile

EMR\_EC2\_DefaultRole

Custom automatic scaling role - optional

When a custom automatic scaling rule triggers, Amazon EMR assumes this role to add and terminate EC2 instances. [Learn more](#)

Custom automatic scaling role

Summary

Name and applications - required

Name

DatabaseStudents

Amazon EMR release

emr-7.3.0

Application bundle

Core Hadoop (Hadoop 3.3.6, Hive 3.1.3, Hue 4.11.0, Pig 0.17.0, Tez 0.10.2)

Cluster configuration - required

Uniform instance groups

Primary (m5.xlarge), Core (m5.xlarge), Task (m5.xlarge)

Cluster scaling and provisioning - required

Provisioning configuration

Core size: 1 instance

Cancel

Create cluster

Step VI: Wait for few minute for creation of cluster.

Step VII: Once Instances are in running state then go to EC2 Service and connect the Ec2 Instance.

Services

Search

[Alt+S]

j-21EMGTBFSBDD

Cluster configuration

Instance groups

Capacity

1 Primary 1 Core 1 Task

emr-7.3.0

Installed applications

Hadoop 3.3.6, Hive 3.1.3, Hue 4.11.0, Pig 0.17.0, Tez 0.10.2

aws-logs-246110268600-us-west-2/elasticmapreduce

Persistent application UIs

YARN timeline server

Tez UI

Primary node public DNS

ec2-54-203-209-24.us-west-2.compute.amazonaws.com

Connect to the Primary node using SSH

Connect to the Primary node using SSM

Waiting

Creation time

October 03, 2024, 17:17 (UTC+05:30)

Elapsed time

9 minutes, 46 seconds

Properties

Bootstrap actions

Instances (Hardware)

Steps

Applications

Configurations

Monitoring

Events

Tags (0)

Instance group settings

Cluster scaling option

Manually set cluster size

Core

Name and Maximum core nodes in the cluster

Core 1 instances

Task

Name and Maximum task nodes in the cluster

Task - 1 1 instances

Edit cluster scaling option

Instance groups (3)

With the instance groups configuration, each node type consists of the same instance type and the same purchasing option for instances: On-Demand or Spot.

Find instances by status

Find resources by ID or type; or search for text within loaded results

< 1 >

Type and name	ID	Status	Instances	Purchasing option and p...	EBS size (GiB)	EC2 Instance ID	Public DNS name	Publ
		Last state change reason						
Primary	ig-1UF9GWN8WFEH	Running	1	On-Demand	-	-	-	-
Core	ig-3DOBRWYUDZL6P	Running	1	On-Demand	-	-	-	-
Task (Task - 1)	ig-2EP8FMVCOM16X	Running	1	On-Demand	-	-	-	-



managing data stored in Hadoop's distributed file system (HDFS) using SQL-like syntax.

Step X: in that Create the Student Table

CREATE TABLE student (

id INT,

name STRING,

age INT,

grade STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ',';

```
[root@ip-172-31-36-22 ~]# hive
Hive Session ID = 9eeac9c4-fd9d-4b5c-9191-3d646497bbd2

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
hive> CREATE TABLE student (
>   id INT,
>   name STRING,
>   age INT,
>   grade STRING
> )
> ROW FORMAT DELIMITED
> FIELDS TERMINATED BY ',';
OK
Time taken: 0.561 seconds
```

Step XI: then Insert the data in it

```
aws Services Q Search [Alt+S]
logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
hive> CREATE TABLE student (
  > id INT,
  > name STRING,
  > age INT,
  > grade STRING
  > )
  > ROW FORMAT DELIMITED
  > FIELDS TERMINATED BY ',';
K
time taken: 0.561 seconds
hive> INSERT INTO TABLE student VALUES (1, 'John', 20, 'A');
Query ID = root_20241003120904_c6b5de44-9dd3-419b-86f1-164a96d24336
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1727956398633_0001)

-----
VERTICES      MODE      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
app 1 ..... container  SUCCEEDED    1         1         0         0         0         0
Reducer 2 ..... container  SUCCEEDED    1         1         0         0         0         0
-----
VERTICES: 02/02 [=====>>>] 100% ELAPSED TIME: 4.98 s
-----
Loading data to table default.student
K
time taken: 7.758 seconds
hive> INSERT INTO TABLE student VALUES (2, 'Jane', 22, 'B');
Query ID = root_20241003120912_74886f32-ced2-4b21-8644-d7c472b535d1
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1727956398633_0001)

-----
VERTICES      MODE      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
app 1 ..... container  SUCCEEDED    1         1         0         0         0         0
Reducer 2 ..... container  SUCCEEDED    1         1         0         0         0         0
-----
VERTICES: 02/02 [=====>>>] 100% ELAPSED TIME: 1.15 s
```

Step XII: run Queries on terminal.

to fetch specific student data. For instance, to retrieve students with an 'A' grade:

SELECT \* FROM student WHERE grade='A';

```

-----
VERTICES      MODE      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  SUCCEEDED    1         1         0         0         0         0
Reducer 2 ..... container  SUCCEEDED    1         1         0         0         0         0
-----
VERTICES: 02/02 [=====>>>] 100% ELAPSED TIME: 0.13 s
-----

Loading data to table default.student
OK
Time taken: 1.039 seconds
hive> INSERT INTO TABLE student VALUES (4, 'Bob', 23, 'C');
Query ID = root_20241003120915_f7fc186a-107c-4c4b-8348-093dff164686
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1727956398633_0001)

-----
VERTICES      MODE      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  SUCCEEDED    1         1         0         0         0         0
Reducer 2 ..... container  SUCCEEDED    1         1         0         0         0         0
-----
VERTICES: 02/02 [=====>>>] 100% ELAPSED TIME: 1.01 s
-----

Loading data to table default.student
OK
Time taken: 1.851 seconds
hive> SELECT * FROM student WHERE grade='A';
OK
1      John    20      A
3      Alice    21      A
Time taken: 0.294 seconds, Fetched: 2 row(s)
hive>

```

## Step XIII: Retrieve All Students Ordered by Age:

This query will return all students and order them by their age in ascending order.

**SELECT \* FROM student ORDER BY age ASC;**

```

Time taken: 0.294 seconds, Fetched: 2 row(s)
hive> SELECT * FROM student ORDER BY age ASC;
Query ID = root_20241003121619_bb5d4f2b-f6f8-474b-9f84-a4c99f2dc9af
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1727956398633_0002)

-----
VERTICES      MODE      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  SUCCEEDED    1         1         0         0         0         0
Reducer 2 ..... container  SUCCEEDED    1         1         0         0         0         0
-----
VERTICES: 02/02 [=====>>>] 100% ELAPSED TIME: 4.36 s
-----

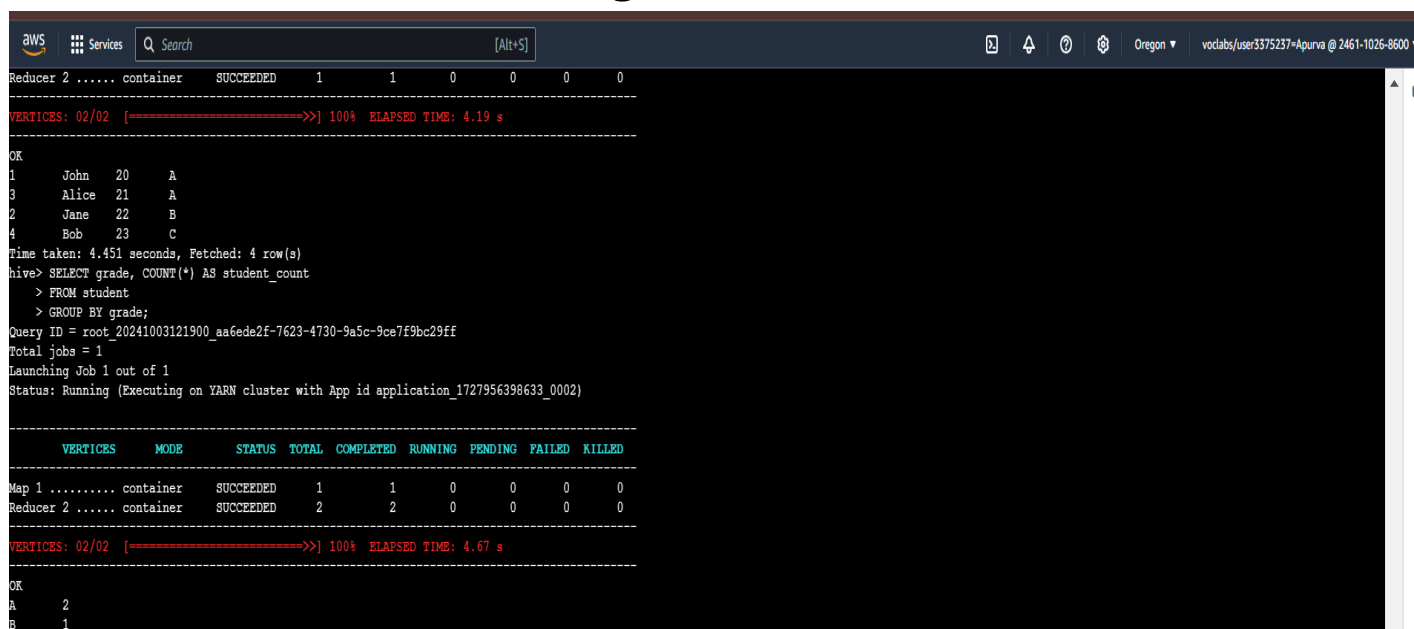
OK
1      John    20      A
3      Alice    21      A
2      Jane    22      B
4      Bob     23      C
Time taken: 9.567 seconds, Fetched: 4 row(s)

```

## Step XIV: Count the Number of Students in Each Grade:

This query will give you a count of students grouped by their grade.

```
SELECT grade, COUNT(*) AS student_count  
FROM student GROUP BY grade;
```



The screenshot shows an AWS console terminal window with the following content:

```
Reducer 2 ..... container SUCCEEDED 1 1 0 0 0 0  
-----  
VERTICES: 02/02 [=====>>>] 100% ELAPSED TIME: 4.19 s  
-----  
OK  
1 John 20 A  
3 Alice 21 A  
2 Jane 22 B  
4 Bob 23 C  
Time taken: 4.451 seconds, Fetched: 4 row(s)  
hive> SELECT grade, COUNT(*) AS student_count  
> FROM student  
> GROUP BY grade;  
Query ID = root_20241003121900_aa6ede2f-7623-4730-9a5c-9ce7f9bc29ff  
Total jobs = 1  
Launching Job 1 out of 1  
Status: Running (Executing on YARN cluster with App id application_1727956398633_0002)  
-----  
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: 4.67 s  
-----  
OK  
A 2  
B 1
```

Then delete the instances and cluster properly.