Alia Mohamed
Project 2 Report
April 23, 2023

Task 1: Launch a cluster of virtual machines in a cloud environment (e.g., AWS, Azure, or GCP). You will need to have one node as the master and at least two nodes as workers (slaves).

For this task I chose to work with AWS EMR clusters. In order to launch the cluster with 1 master and 2 worker nodes I first needed to create a S3 Bucket to store files, create an IAM role for EMR from the IAM console, and create Instance Profile and add the role to it before finally launching the cluster.

create s3 bucket

Command:

```
aws s3api create-bucket --bucket pleaseletmehavethisbucketname --region us-east-1
```

Output:

```
{
   "Location": "/pleaseletmehavethisbucketname"
}
```

Create instance profile

Command:

```
aws iam create-instance-profile --instance-profile-name my-instance-profile
```

Output:

```
{
   "InstanceProfile": {
      "Path": "/",
      "InstanceProfileName": "my-instance-profile",
      "InstanceProfileId": "AIPAZQV43TNZUTHFVYPUU",
      "Arn": "arn:aws:iam::654304844659:instance-profile/my-instance-profile",
      "CreateDate": "2023-04-24T00:00:35+00:00",
      "Roles": []
   }
}
```

Add role to the instance profile

Command:

```
aws iam add-role-to-instance-profile --instance-profile-name my-instance-profile --role-name emr-role
```

Create cluster

Command:

```
aws emr create-cluster \
--name "my-cluster" \
--release-label emr-6.3.0 \
--instance-groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m5.xlarge
InstanceGroupType=CORE,InstanceCount=2,InstanceType=m5.xlarge \
--applications Name=Hadoop Name=Spark \
--ec2-attributes KeyName=p0KeyPair,InstanceProfile=my-instance-profile \
--log-uri s3://pleaseletmehavethisbucketname/logs/ \
--region us-east-1 \
--service-role arn:aws:iam::654304844659:role/emr-role
```

Output:

```
{
   "ClusterId": "j-24CVMTERYPG8D",
   "ClusterArn": "arn:aws:elasticmapreduce:us-east-1:654304844659:cluster/j-24CVMTERYPG8D"
}
```

Describing the cluster

```
aws emr describe-cluster --cluster-id j-24CVMTERYPG8D
{
    "Cluster": {
        "Id": "j-24CVMTERYPG8D",
        "Name": "my-cluster",
        "Status": {
            "State": "STARTING",
            "StateChangeReason": {
                "Message": "Configuring cluster software"
            },
            "Timeline": {
                "CreationDateTime": "2023-04-23T20:13:52.189000-04:00"
            }
        },
        "Ec2InstanceAttributes": {
            "Ec2KeyName": "p0KeyPair",
            "RequestedEc2SubnetIds": [],
            "Ec2AvailabilityZone": "us-east-1b",
            "RequestedEc2AvailabilityZones": [],
            "IamInstanceProfile": "my-instance-profile",
            "EmrManagedMasterSecurityGroup": "sg-020aa14e8ef46e5e4",
            "EmrManagedSlaveSecurityGroup": "sg-0466108a35bbad1d8"
        },
        "InstanceCollectionType": "INSTANCE_GROUP",
:...skipping...
```

Task 2: Deploy the HDFS service on the cluster.

This part of the project was already taken care of when creating the cluster above, namely the line: **--applications Name=Hadoop Name=Spark \**

Task 3: Download the text version of Pride and Prejudice from Project Gutenberg, and save it to the HDFS cluster.

For this task I downloaded Pride and Prejudice as a txt file on my local machine. To be able to save this file to hdf/emrfs on the cluster I needed to ssh into the master node first and upload the file to the s3 bucket using S3 console.

Ssh into master node using public dns

Command:

```
ssh -i p0KeyPair.pem hadoop@ec2-107-21-146-18.compute-1.amazonaws.com
```

Output:

```
The authenticity of host 'ec2-107-21-146-18.compute-1.amazonaws.com (107.21.146.18)' can't be established.
ED25519 key fingerprint is SHA256:GXhweGWuVuI7LnmfBIch7I+gQx+n+mAHAkRH1ySbj9Y.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-107-21-146-18.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
80 package(s) needed for security, out of 130 available
Run "sudo yum update" to apply all updates.

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR
```

Upload file to s3 bucket from s3 console
Save file to hdfs from s3 bucket

Command:

```
aws s3 cp s3://pleaseletmehavethisbucketname/Pride_And_Prejudice.txt hdfs:/ --region us-east-1
```

Output:

```
download: s3://pleaseletmehavethisbucketname/Pride_And_Prejudice.txt to hdfs:/Pride_And_Prejudice.txt
```

Task 4: Deploy the Spark service on the cluster.

This part of the project was also already taken care of when creating the cluster above, namely the line: **--applications Name=Hadoop Name=Spark **

Task 5: Use the file in HDFS as input, run a wordcount program in Spark to count the number of occurrences of each word. Sort the words by count, in descending order, and return a list of the (word, count) pairs for the 20 most used words.

For this part I constructed the code needed for the application in a file named wordcount.py that was then used with input file from hdfs to obtain results

Command:

```
spark-submit --master yarn --deploy-mode client
s3://pleaseletmehavethisbucketname/wordcount.py hdfs:/Pride_And_Prejudice.txt
hdfs:/Pride_And_Prejudice_output.txt
```

Code:

```
from pyspark import SparkContext

sc = SparkContext(appName="WordCount")
lines = sc.textFile("hdfs:/Pride_And_Prejudice.txt")
words = lines.flatMap(lambda line: line.split(" "))
wordsCount = words.map(lambda word: (word, 1)).reduceByKey(lambda a,b:
a+b).sortBy(lambda x: x[1], False)
print(wordsCount.take(20))
sc.stop()
```

Results:

```
[('', 10420), ('the', 4509), ('to', 4275), ('of', 3897), ('and', 3443), ('a', 2021), ('in', 1923), ('her',
1905), ('was', 1817), ('I', 1764), ('that', 1458), ('not', 1432), ('she', 1341), ('be', 1227), ('his', 1196),
('as', 1165), ('had', 1131), ('with', 1086), ('he', 1054), ('for', 1041)]
```

Task 6: Task 6: Write a Spark program that uses Monte Carlo methods to estimate the value of $π$.

For this part I constructed the code needed for the application in a file named EstimatePi.py that I uploaded to the S3 bucket on the cluster before submitting it

Command:

```
 spark–submit ––master yarn
s3://pleaseletmehavethisbucketname/EstimatePi.py
```

Code:

```
from random import random
from pyspark import SparkContext


def point(p):
    x = random()
    y = random()
    return x*x + y*y < 1


sc = SparkContext(appName="EstimatePi")
n = 10000
count = sc.parallelize(range(0, n)).map(point).reduce(lambda a,b: a+b)
pi = 4 * count / n
print("Pi =", pi)
sc.stop()
```

Results: Pi = 3.1648

At the end of the project the cluster was terminated properly using the EMR console.

References

Amazon EMR from https://aws.amazon.com/emr/
Getting Started With EMR from
https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-gs.html
Apache Hadoop from https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-hadoop.html
Apache Spark from https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-spark.html
Hadoop documentation from https://hadoop.apache.org/docs/current/
Spark documentation from https://spark.apache.org/docs/latest/submitting-applications.html
Create your first S3 Bucket from
https://docs.aws.amazon.com/AmazonS3/latest/userguide/creating-bucket.html
Using Instance Profiles from
https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-ec2_instance-profiles.html
Command Line Interface Commands for EMR from
https://docs.aws.amazon.com/cli/latest/reference/emr/index.html