Ex. No: 1	Basic Linux Commands
16.7.2024	

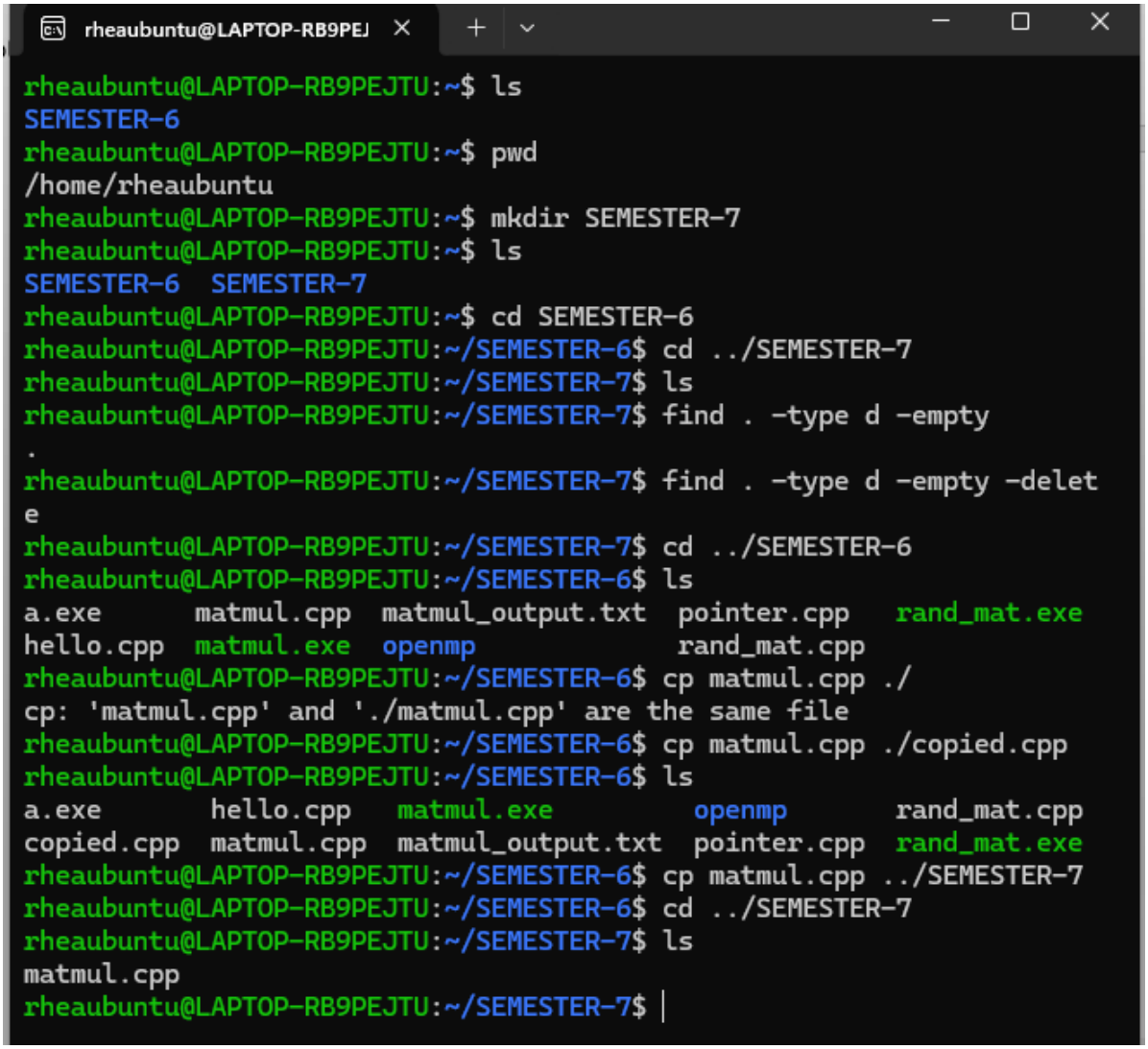
Aim:

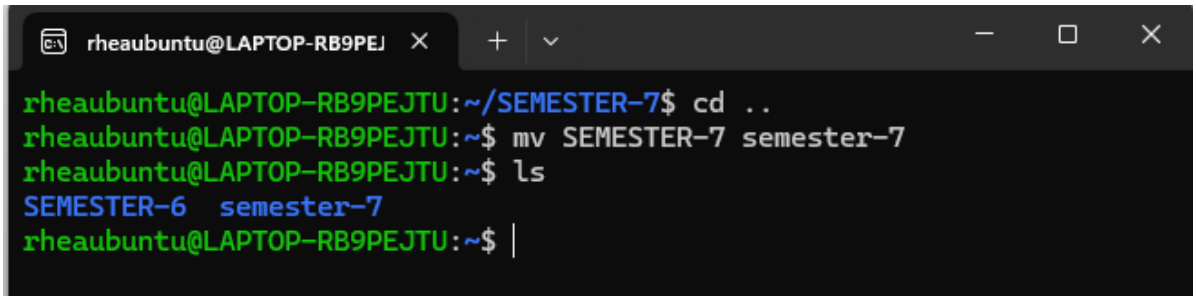
To execute various basic Linux commands

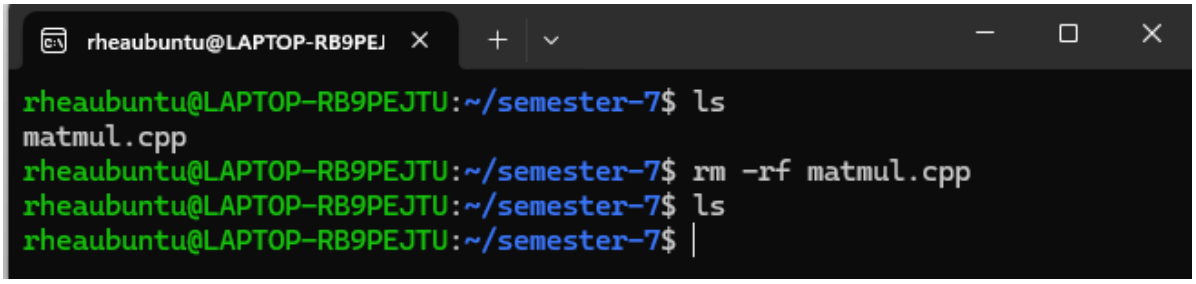
Program:

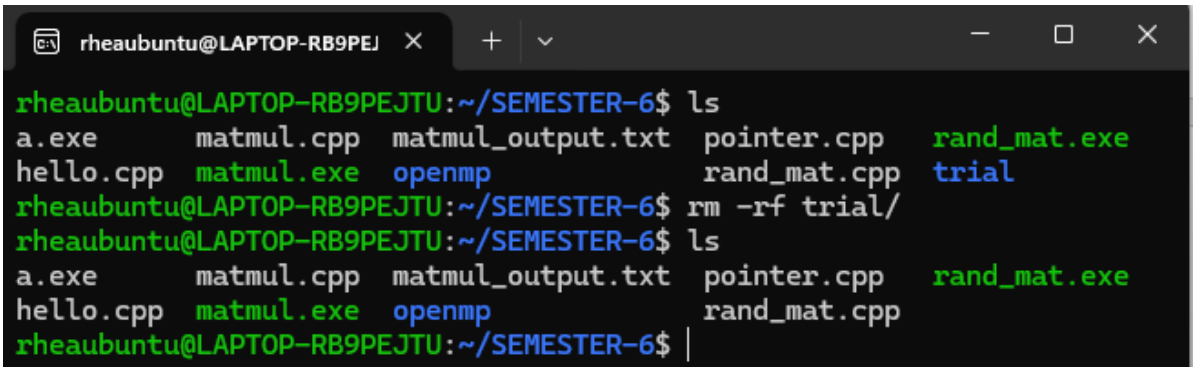
1. Display information about current directory - ls
2. Display the current working directory – pwd
3. Create a new directory – mkdir SEMESTER-7
4. Navigate between different folders – cd SEMESTER-6
5. Remove empty directories find . -type d -empty -delet
6. Copy files from one directory to same directory – cp matmul.cpp ./copied.cpp
7. Copy files from one directory to another directory – cp matmul.cpp ../SEMESTER-7
8. Rename a filename to another name – mv SEMESTER-7 semester-7
9. Move a file from one directory to another – mv copied.cpp trial/
10. Delete individual files from a directory – rm -rf matmul.cpp
11. Delete an unempty directory – rm -rf trial/
12. Get basic information about the OS – lscpu
13. Find a file in the directory – find *.cpp
14. Create empty files – touch empty.txt
15. Display file contents on terminal – cat empty.txt
16. Clear Terminal – clear
17. Display the processes in terminal – ps -A
18. Access manual for all Linux commands – help
19. Search for a specific string in an output – ls -l | grep “s/.*w://p”
20. 20.Display active processes on the terminal – ps
21. 21.Download files from the internet – wget <url>
22. Create or update passwords for existing users – passwd
23. View exact location of any tool/software installed – which bash
24. Check the details of the file system – df -Th
25. Check the lines, word count and characters in a file using different options :
    For characters – wc -c empty.txt
    For words – wc -w empty.txt
    For lines – wc -l empty.txt

Output:

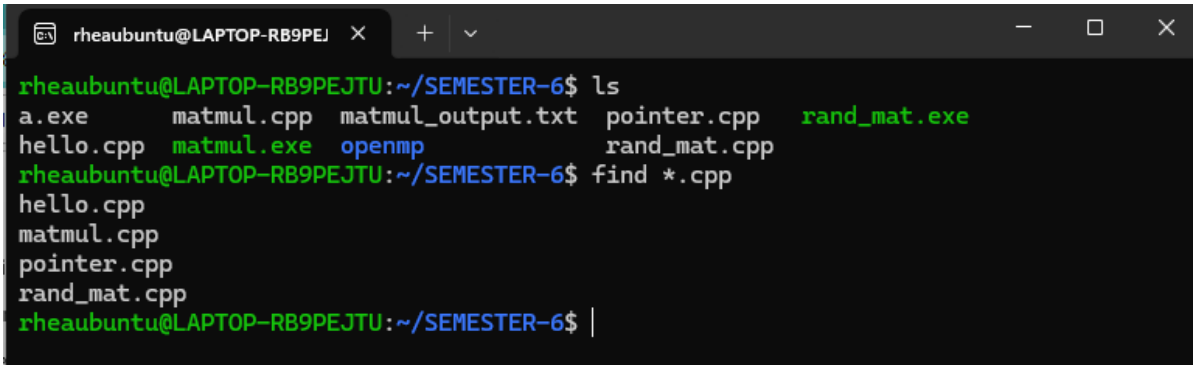

rheaubuntu@LAPTOP-RB9PEJ
rheaubuntu@LAPTOP-RB9PEJTU:~$ ls
SEMESTER-6
rheaubuntu@LAPTOP-RB9PEJTU:~$ pwd
/home/rheaubuntu
rheaubuntu@LAPTOP-RB9PEJTU:~$ mkdir SEMESTER-7
rheaubuntu@LAPTOP-RB9PEJTU:~$ ls
SEMESTER-6 SEMESTER-7
rheaubuntu@LAPTOP-RB9PEJTU:~$ cd SEMESTER-6
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-6$ cd ../SEMESTER-7
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-7$ ls
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-7$ find . -type d -empty
.
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-7$ find . -type d -empty -delet
e
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-7$ cd ../SEMESTER-6
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-6$ ls
a.exe matmul.cpp matmul_output.txt pointer.cpp rand_mat.exe
hello.cpp matmul.exe openmp rand_mat.cpp
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-6$ cp matmul.cpp ./
cp: 'matmul.cpp' and './matmul.cpp' are the same file
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-6$ cp matmul.cpp ./copied.cpp
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-6$ ls
a.exe hello.cpp matmul.exe openmp rand_mat.cpp
copied.cpp matmul.cpp matmul_output.txt pointer.cpp rand_mat.exe
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-6$ cp matmul.cpp ../SEMESTER-7
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-6$ cd ../SEMESTER-7
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-7$ ls
matmul.cpp
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-7$

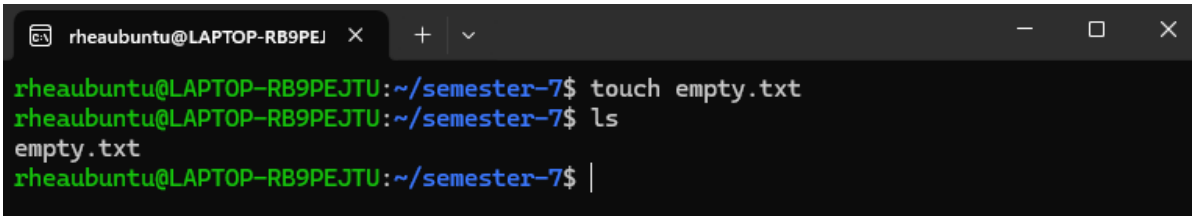

rheaubuntu@LAPTOP-RB9PEJ
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-7$ cd ..
rheaubuntu@LAPTOP-RB9PEJTU:~$ mv SEMESTER-7 semester-7
rheaubuntu@LAPTOP-RB9PEJTU:~$ ls
SEMESTER-6 semester-7
rheaubuntu@LAPTOP-RB9PEJTU:~$

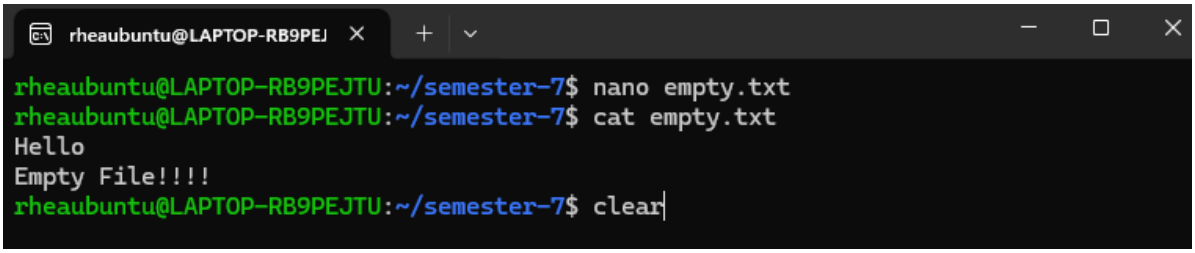

rheaubuntu@LAPTOP-RB9PEJ
rheaubuntu@LAPTOP-RB9PEJTU:~/semester-7$ ls
matmul.cpp
rheaubuntu@LAPTOP-RB9PEJTU:~/semester-7$ rm -rf matmul.cpp
rheaubuntu@LAPTOP-RB9PEJTU:~/semester-7$ ls
rheaubuntu@LAPTOP-RB9PEJTU:~/semester-7$

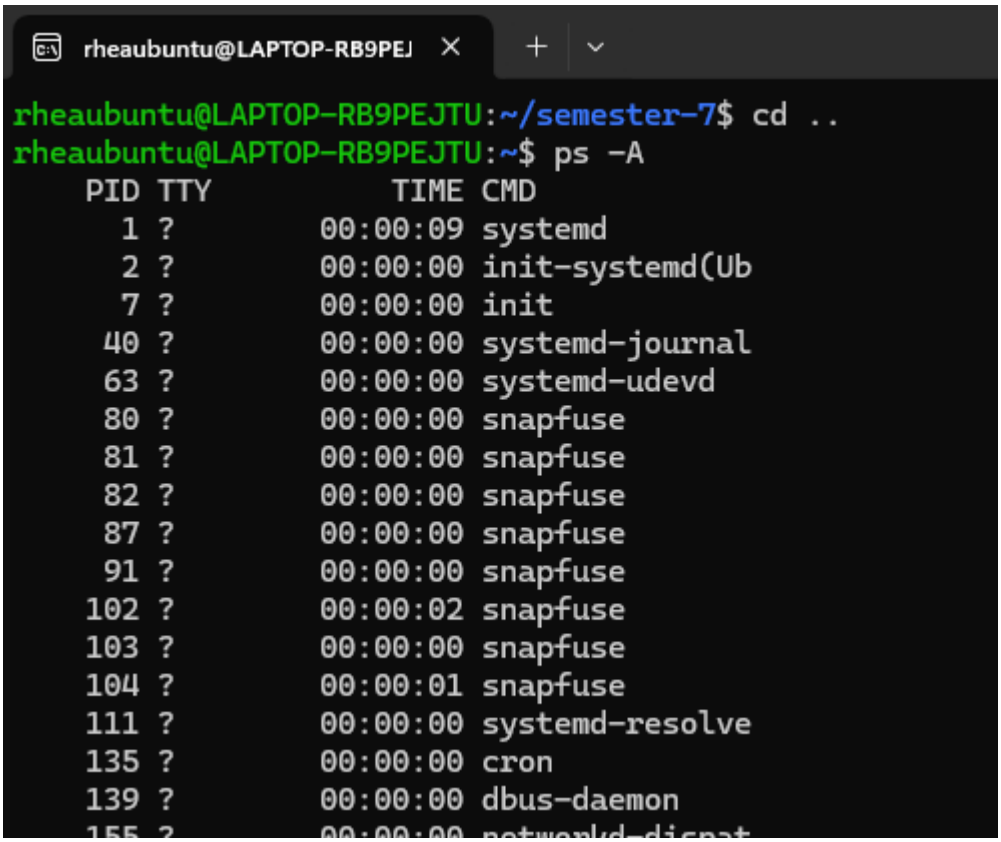

rheaubuntu@LAPTOP-RB9PEJ
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-6$ ls
a.exe matmul.cpp matmul_output.txt pointer.cpp rand_mat.exe
hello.cpp matmul.exe openmp rand_mat.cpp trial
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-6$ rm -rf trial/
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-6$ ls
a.exe matmul.cpp matmul_output.txt pointer.cpp rand_mat.exe
hello.cpp matmul.exe openmp rand_mat.cpp
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-6$

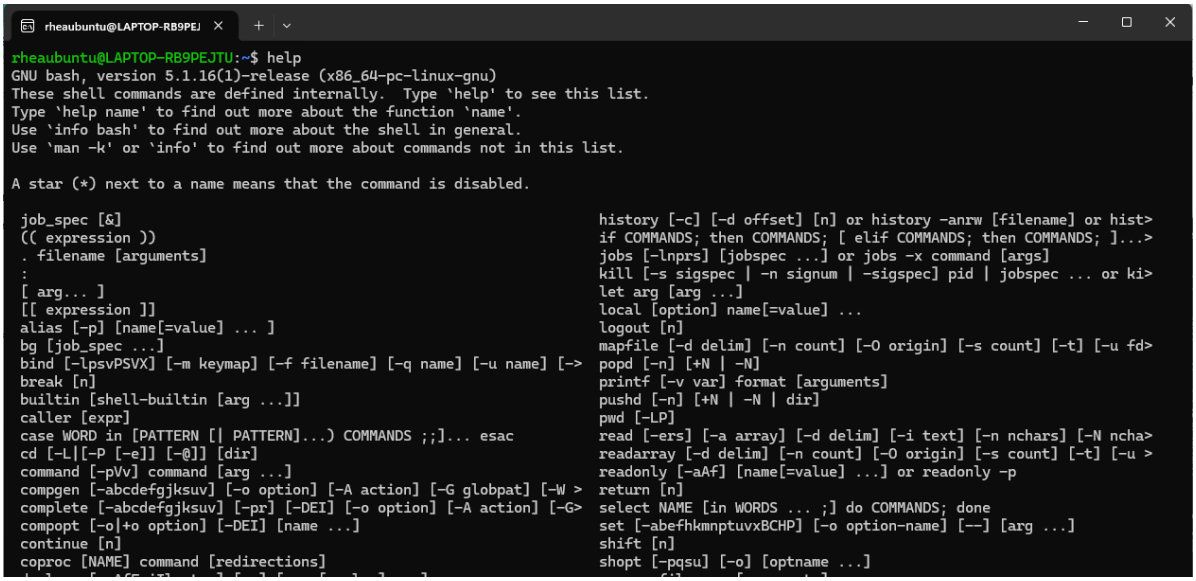

rheaubuntu@LAPTOP-RB9PEJ
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-6$ lscpu
Architecture: x86_64
CPU op-mode(s): 32-bit, 64-bit
Address sizes: 39 bits physical, 48 bits virtual
Byte Order: Little Endian
CPU(s): 8
On-line CPU(s) list: 0-7
Vendor ID: GenuineIntel
Model name: Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz
CPU family: 6
Model: 158
Thread(s) per core: 2
Core(s) per socket: 4
Socket(s): 1
Stepping: 10
BogoMIPS: 4800.01
Flags: fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon rep_good nopl xtopology cpuid pni pclmulqdq vmx ssse3 fma cx16 pdcm pcid sse4_1 sse4_2 movbe popcnt aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch invpcid_single pti ssbd ibrs ibpb stibp tpr_shadow vnmi ept vpid ept_ad fsgsbase bmi1 avx2 smep bmi2 erms invpcid rdseed adx smap clflushopt xsaveopt xsavec xgetbv1 xsaves md_clear flush_l1d arch_capabilities
Virtualization features:
Virtualization: VT-x
Hypervisor vendor: Microsoft
Virtualization type: full
Caches (sum of all):
L1d: 128 KiB (4 instances)
L1i: 128 KiB (4 instances)
L2: 1 MiB (4 instances)
L3: 8 MiB (1 instance)
Vulnerabilities:
Gather data sampling: Unknown: Dependent on hypervisor status
Itlb multihit: KVM: Mitigation: VMX disabled
L1tf: Mitigation; PTE Inversion; VMX conditional cache flushes, SMT vulnerable
Mds: Mitigation; Clear CPU buffers; SMT Host state unknown
Meltdown: Mitigation; PTI
Mmio stale data: Mitigation; Clear CPU buffers; SMT Host state unknown
Retbleed: Mitigation; IBRS

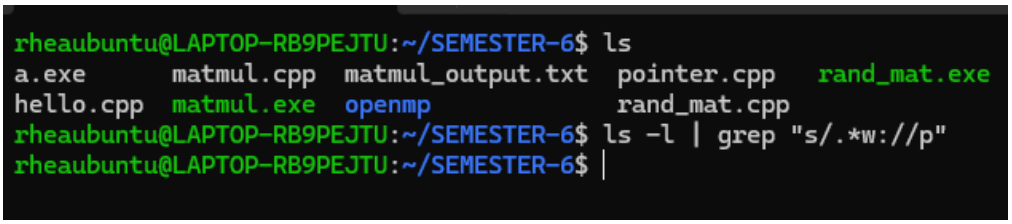

rheaubuntu@LAPTOP-RB9PEJ
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-6$ ls
a.exe matmul.cpp matmul_output.txt pointer.cpp rand_mat.exe
hello.cpp matmul.exe openmp rand_mat.cpp
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-6$ find *.cpp
hello.cpp
matmul.cpp
pointer.cpp
rand_mat.cpp
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-6$

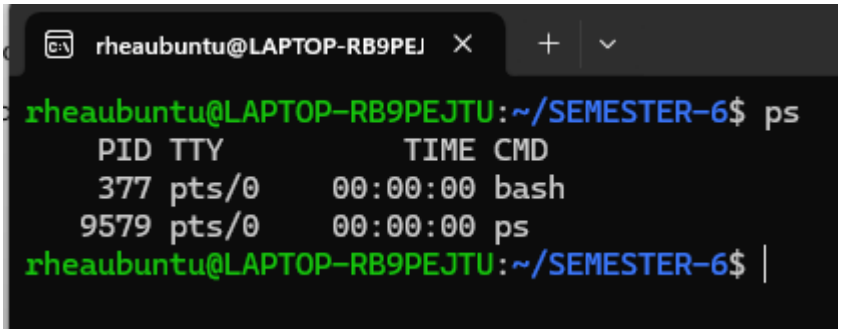

rheaubuntu@LAPTOP-RB9PEJ
rheaubuntu@LAPTOP-RB9PEJTU:~/semester-7$ touch empty.txt
rheaubuntu@LAPTOP-RB9PEJTU:~/semester-7$ ls
empty.txt
rheaubuntu@LAPTOP-RB9PEJTU:~/semester-7$

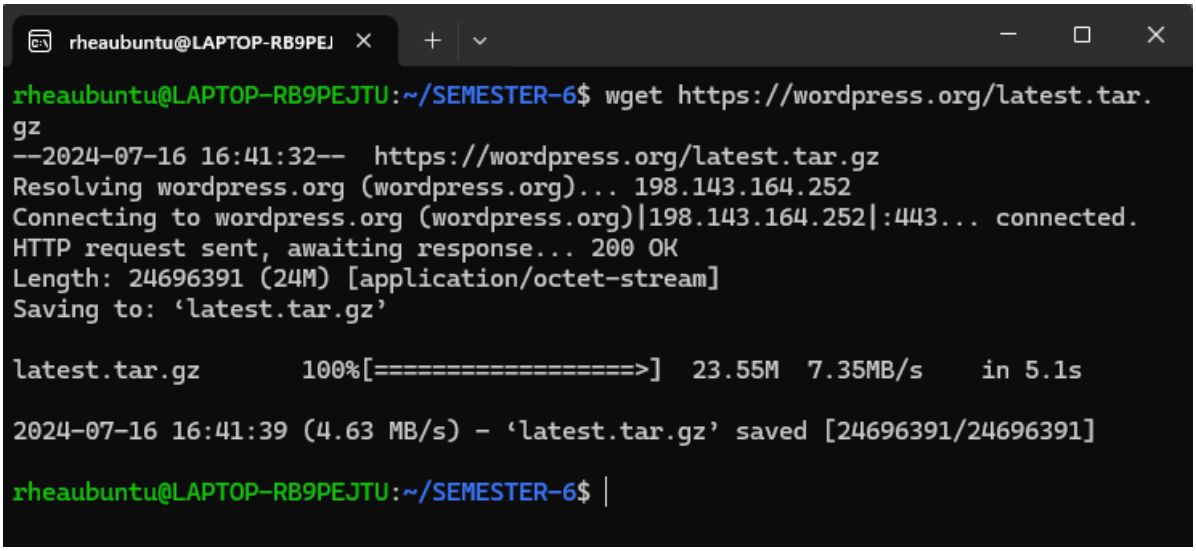

rheaubuntu@LAPTOP-RB9PEJ
rheaubuntu@LAPTOP-RB9PEJTU:~/semester-7$ nano empty.txt
rheaubuntu@LAPTOP-RB9PEJTU:~/semester-7$ cat empty.txt
Hello
Empty File!!!!
rheaubuntu@LAPTOP-RB9PEJTU:~/semester-7$ clear

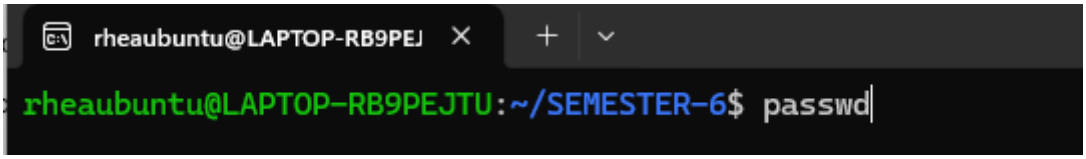

rheaubuntu@LAPTOP-RB9PEJ
rheaubuntu@LAPTOP-RB9PEJTU:~/semester-7$ cd ..
rheaubuntu@LAPTOP-RB9PEJTU:~$ ps -A
PID TTY TIME CMD
1 ? 00:00:09 systemd
2 ? 00:00:00 init-systemd(Ub
7 ? 00:00:00 init
40 ? 00:00:00 systemd-journal
63 ? 00:00:00 systemd-udevd
80 ? 00:00:00 snapfuse
81 ? 00:00:00 snapfuse
82 ? 00:00:00 snapfuse
87 ? 00:00:00 snapfuse
91 ? 00:00:00 snapfuse
102 ? 00:00:02 snapfuse
103 ? 00:00:00 snapfuse
104 ? 00:00:01 snapfuse
111 ? 00:00:00 systemd-resolve
135 ? 00:00:00 cron
139 ? 00:00:00 dbus-daemon

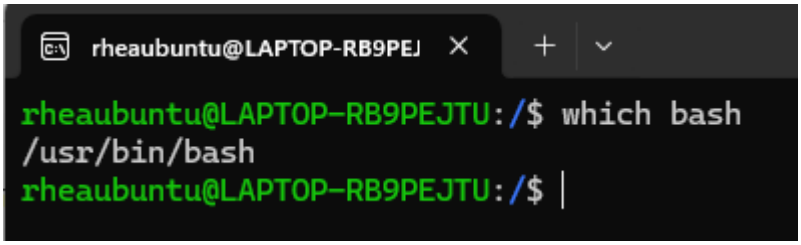

rheaubuntu@LAPTOP-RB9PEJ
rheaubuntu@LAPTOP-RB9PEJTU:~$ help
GNU bash, version 5.1.16(1)-release (x86_64-pc-linux-gnu)
These shell commands are defined internally. Type 'help' to see this list.
Type 'help name' to find out more about the function 'name'.
Use 'info bash' to find out more about the shell in general.
Use 'man -k' or 'info' to find out more about commands not in this list.
A star (*) next to a name means that the command is disabled.
job_spec [&]
((expression))
. filename [arguments]
:
[arg...]
[[expression]]
alias [-p] [name[=value] ...]
bg [job_spec ...]
bind [-lpsvPSVX] [-m keymap] [-f filename] [-q name] [-u name] [->
break [n]
builtin [shell-builtin [arg ...]]
caller [expr]
case WORD in [PATTERN [| PATTERN]...) COMMANDS ;;]... esac
cd [-L|[-P [-e]] [-@]] [dir]
command [-pVv] command [arg ...]
compgen [-abcdefgjksuv] [-o option] [-A action] [-G globpat] [-W >
complete [-abcdefgjksuv] [-pr] [-DEI] [-o option] [-A action] [-G>
compopt [-o|+o option] [-DEI] [name ...]
continue [n]
coproc [NAME] command [redirections]
history [-c] [-d offset] [n] or history -anrw [filename] or hist>
if COMMANDS; then COMMANDS; [elif COMMANDS; then COMMANDS;]...>
jobs [-lnprs] [jobspec ...] or jobs -x command [args]
kill [-s sigspec | -n signum | -sigspec] pid | jobspec ... or ki>
let arg [arg ...]
local [option] name[=value] ...
logout [n]
mapfile [-d delim] [-n count] [-O origin] [-s count] [-t] [-u fd>
popd [-n] [+N | -N]
printf [-v var] format [arguments]
pushd [-n] [+N | -N | dir]
pwd [-LP]
read [-ers] [-a array] [-d delim] [-i text] [-n nchars] [-N ncha>
readarray [-d delim] [-n count] [-O origin] [-s count] [-t] [-u >
readonly [-aAf] [name[=value] ...] or readonly -p
return [n]
select NAME [in WORDS ... ;] do COMMANDS; done
set [-abefhkmnptuvxBCHP] [-o option-name] [--] [arg ...]
shift [n]
shopt [-pqsu] [-o] [optname ...]

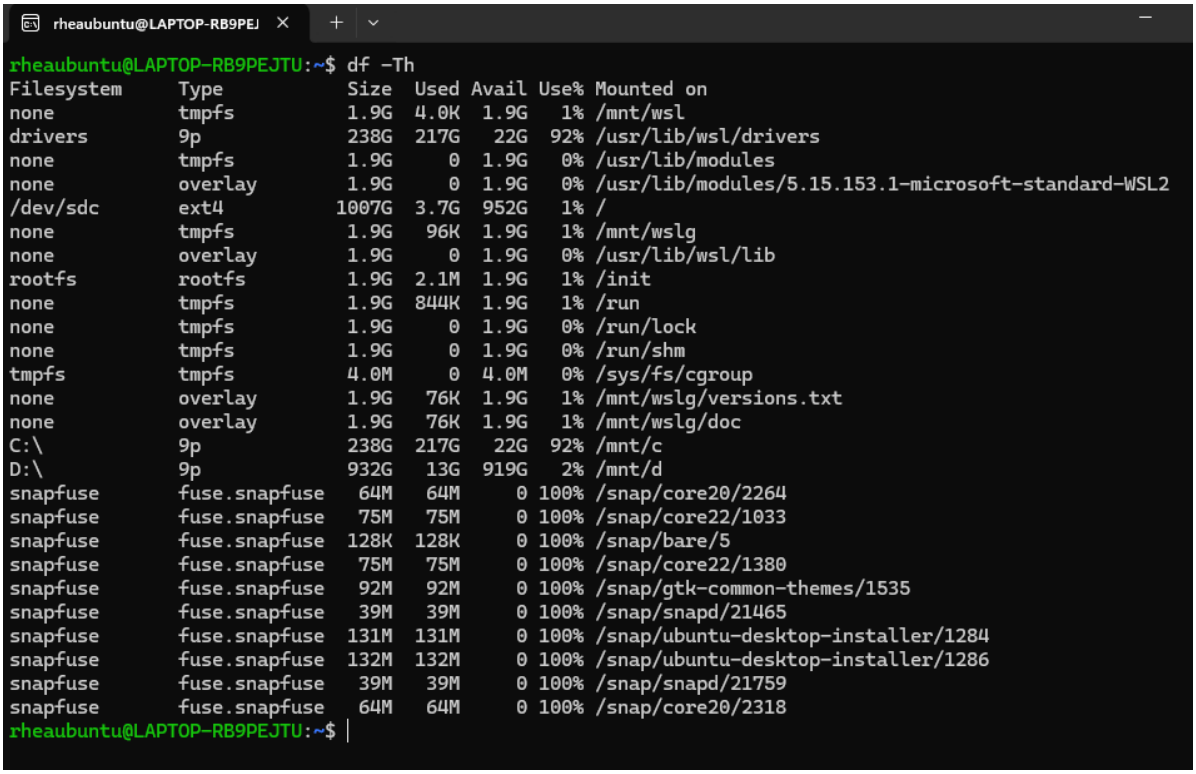

rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-6$ ls
a.exe matmul.cpp matmul_output.txt pointer.cpp rand_mat.exe
hello.cpp matmul.exe openmp rand_mat.cpp
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-6$ ls -l | grep "s/.*w://p"
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-6$


rheaubuntu@LAPTOP-RB9PEJ
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-6$ ps
PID TTY TIME CMD
377 pts/0 00:00:00 bash
9579 pts/0 00:00:00 ps
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-6$


rheaubuntu@LAPTOP-RB9PEJ
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-6$ wget https://wordpress.org/latest.tar.gz
--2024-07-16 16:41:32-- https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24696391 (24M) [application/octet-stream]
Saving to: 'latest.tar.gz'
latest.tar.gz 100%[==========>] 23.55M 7.35MB/s in 5.1s
2024-07-16 16:41:39 (4.63 MB/s) - 'latest.tar.gz' saved [24696391/24696391]
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-6$


rheaubuntu@LAPTOP-RB9PEJ
rheaubuntu@LAPTOP-RB9PEJTU:~/SEMESTER-6$ passwd


rheaubuntu@LAPTOP-RB9PEJ
rheaubuntu@LAPTOP-RB9PEJTU:/$ which bash
/usr/bin/bash
rheaubuntu@LAPTOP-RB9PEJTU:/$



Result:

Successfully implemented various basic linux commands

Ex. No: 2	Install and Configure Hadoop
23.7.2024	

Aim:

To install and configure Hadoop in Ubuntu

Program:

1. Install Java:
   - Check if Java is installed: java -version
   - If Java is not installed, download and install it:
     sudo apt update
     sudo apt install openjdk-8-jdk

2. Download Hadoop:
   - Go to the Apache Hadoop releases page and download the binary distribution (e.g., Hadoop 3.3.6):

     wget https://downloads.apache.org/hadoop/common/hadoop 3.3.0/hadoop-3.4.0.tar.gz
   - Extract the downloaded file:
     tar -xzvf hadoop-3.3.6.tar.gz

3. Move Hadoop to the desired directory:
   sudo mv hadoop-3.3.6 /opt/hadoop

4. Set environment variables:
   - Open the `.bashrc` file for editing: nano ~/.bashrc
   - Add the following lines at the end of the file:
     export HADOOP_HOME=/opt/hadoop export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
   - Save and exit, then refresh your terminal

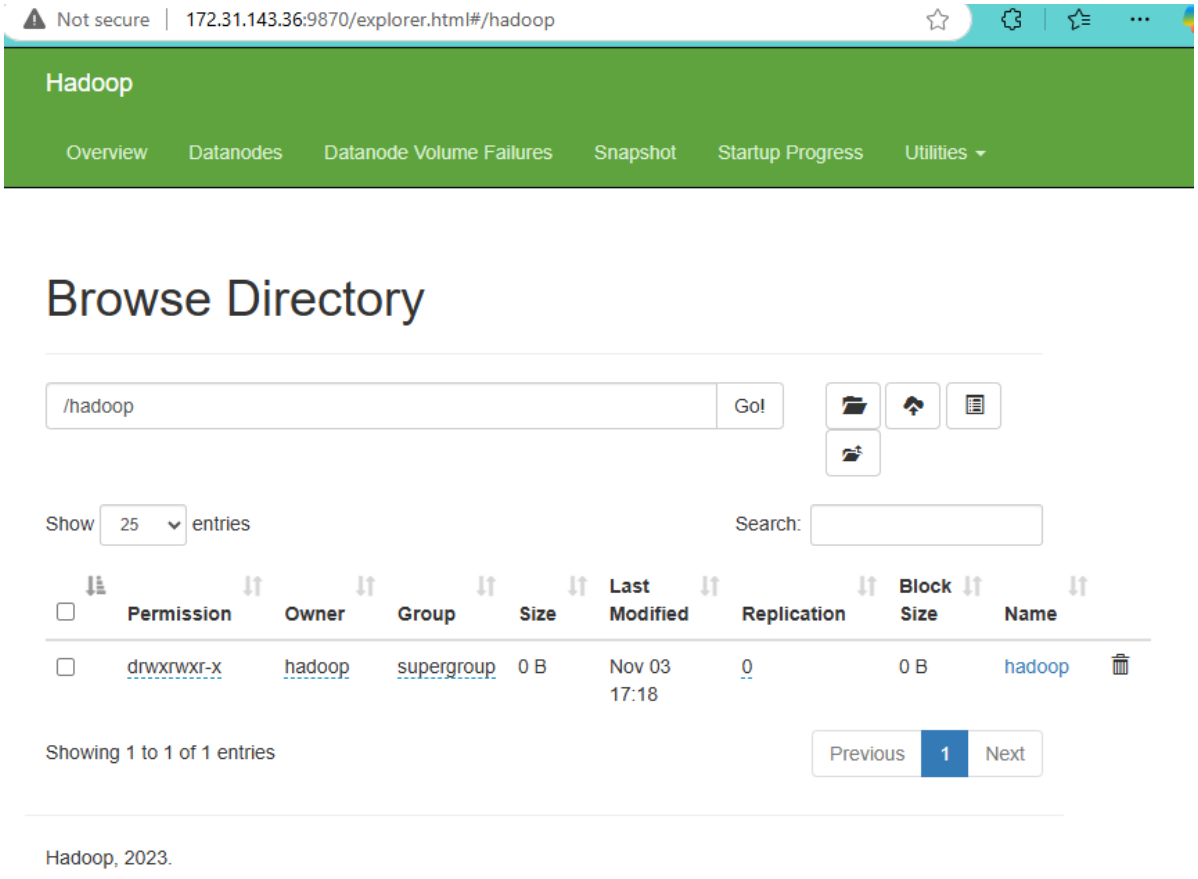
5. Configure Hadoop: source ~/.bashrc
   - Navigate to the Hadoop configuration directory:
     cd $HADOOP_HOME/etc/hadoop/
   - Edit `hadoop-env.sh` to set the Java home path:
     nano hadoop-env.sh
   - Add or modify the line to include your Java installation path:
     export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64

6. Format the NameNode: hadoop namenode -format
7. Start YARN: start-yarn.sh
8. Start all Hadoop services: start-all.sh
9. Check running processes: jps
10. Access Hadoop web interface: http://localhost:9870
11. Stop all Hadoop services when needed: stop-all.sh

Output:

```
hadoop@LAPTOP-RB9PEJTU:~$ hadoop version
Hadoop 3.3.6
Source code repository https://github.com/apache/hadoop.git -r 1be78238728da9266a4f88195058f08fd012bf9c
Compiled by ubuntu on 2023-06-18T08:22Z
Compiled on platform linux-x86_64
Compiled with protoc 3.7.1
From source with checksum 5652179ad55f76cb287d9c633bb53bbd
This command was run using /home/hadoop/hadoop/share/hadoop/common/hadoop-common-3.3.6.jar
hadoop@LAPTOP-RB9PEJTU:~$
```

```
hadoop@LAPTOP-RB9PEJTU:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [LAPTOP-RB9PEJTU]
Starting resourcemanager
Starting nodemanagers
```



Result:

Successfully installed and configured Hadoop in Ubuntu

Ex. No: 3	Implementing MapReduce
30.7.2024	

Aim:

To implement a simple map-reduce code for the wordcount problem in Hadoop.

Program:

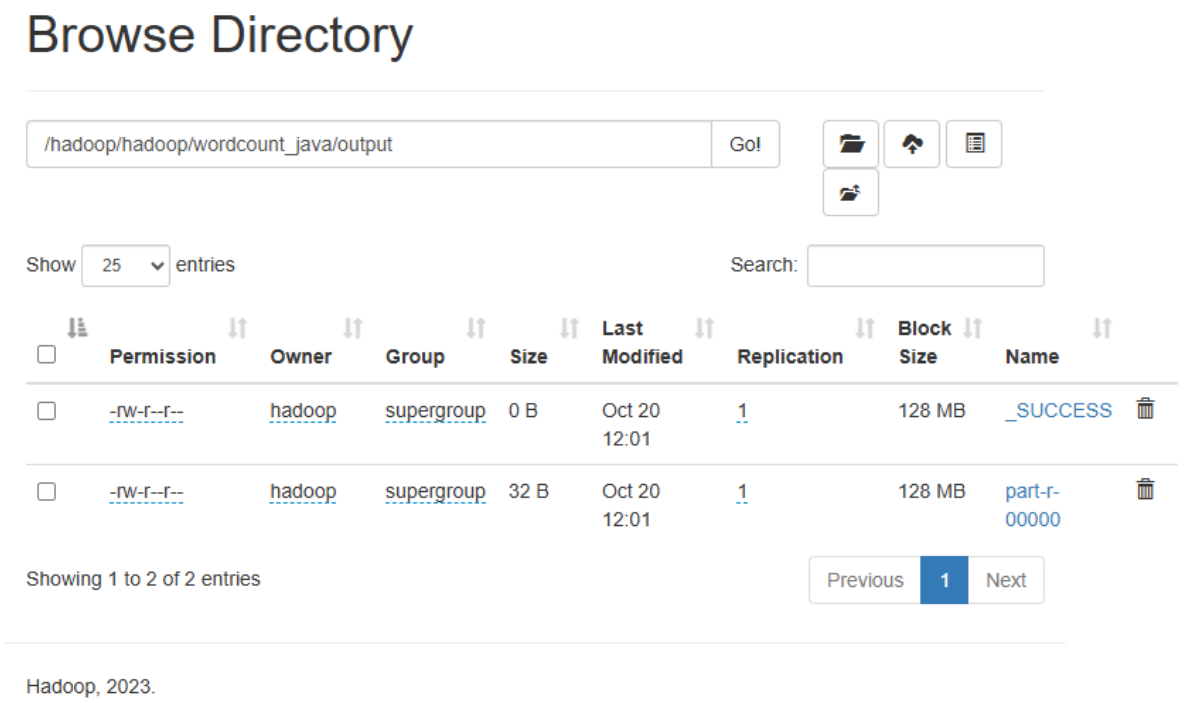
WordCount.java

```
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;


public class WordCount {


    public static class TokenizerMapper extends Mapper<LongWritable,
Text, Text, IntWritable> {
        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();


        public void map(LongWritable key, Text value, Context context)
throws IOException, InterruptedException {
            String[] words = value.toString().split("\\s+");
            for (String str : words) {
                word.set(str);
                context.write(word, one);
            }
        }
```

```
    }


    public static class IntSumReducer extends Reducer<Text, IntWritable,
Text, IntWritable> {
        private IntWritable result = new IntWritable();


        public void reduce(Text key, Iterable<IntWritable> values,
Context context) throws IOException, InterruptedException {
            int sum = 0;
            for (IntWritable val : values) {
                sum += val.get();
            }
            result.set(sum);
            context.write(key, result);
        }
    }


    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "word count");
        job.setJarByClass(WordCount.class);
        job.setMapperClass(TokenizerMapper.class);
        job.setCombinerClass(IntSumReducer.class);
        job.setReducerClass(IntSumReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

Output:

```
hadoop@LAPTOP-RB9PEJTU:~/hadoop/wordcount_java$ hdfs dfs -cat /hadoop/hadoop/wordcount_java/output/part-r-00000
Hadoop  1
Hello   3
Rhea    1
World   1
```



Result:

Successfully implemented map reduce for wordcount problem in Hadoop

Ex. No: 4	Implementing MapReduce 2
6.8.2024	

Aim:

1.Implement map reduce for NCDC weather dataset using Hadoop – find the max and min temperature

2.Implement Apriori algorithm using map reduce paradigm

Program 1:
Temperature.java

```
import java.io.IOException;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.io.*;

import org.apache.hadoop.mapreduce.*;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;


public class Temperatures {


    public static class TempMapper extends Mapper<LongWritable, Text,
Text, IntWritable> {


        public void map(LongWritable key, Text value, Context context)
throws IOException, InterruptedException {

            String line = value.toString();

            String[] fields = line.split("\t");


            if (fields.length >= 8) {  // Ensure there are enough fields

                try {

                    // Extract the minimum and maximum temperature fields
```

```
                    int minTemp = Integer.parseInt(fields[6].trim());  // 
MLY-TMIN-NORMAL
                    int maxTemp = Integer.parseInt(fields[7].trim());  // 
MLY-TMAX-NORMAL


                    // Write the min and max temperatures to the context
                    context.write(new Text("Min Temperature"), new 
IntWritable(minTemp));
                    context.write(new Text("Max Temperature"), new 
IntWritable(maxTemp));
                } catch (NumberFormatException e) {
                    // Ignore invalid data
                }
            }
        }
    }


    public static class TempReducer extends Reducer<Text, IntWritable, 
Text, IntWritable> {


        public void reduce(Text key, Iterable<IntWritable> values, 
Context context) throws IOException, InterruptedException {
            int extremeTemp = key.toString().equals("Min Temperature") ? 
Integer.MAX_VALUE : Integer.MIN_VALUE;


            for (IntWritable value : values) {
                int temp = value.get();
                if (key.toString().equals("Min Temperature")) {
                    if (temp < extremeTemp) {
                        extremeTemp = temp;
                    }
                } else {  // Max Temperature
                    if (temp > extremeTemp) {
```

```
                        extremeTemp = temp;
                    }
                }
            }
            context.write(key, new IntWritable(extremeTemp));
        }
    }

    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "min and max temperatures");
        job.setJarByClass(Temperatures.class);
        job.setMapperClass(TempMapper.class);
        job.setReducerClass(TempReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

Program 2:

Mapper.py

```
#!/usr/bin/env python3
import sys
from itertools import combinations
def generate_combinations(item_list, length):
return list(combinations(item_list, length))

# Input comes from standard input (stdin)
```

```
for line in sys.stdin:
      line = line.strip()
      items = line.split()
      for length in range(1, len(items) + 1):
            for combination in generate_combinations(items, length):
                  print(f"{','.join(combination)}\t1")
```

Reducer.py

```
#!/usr/bin/env python3
import sys
current_itemset = None
current_count = 0 #
Input comes from standard input (stdin)
for line in sys.stdin:
      line = line.strip()
      itemset, count = line.split('\t', 1)
      count = int(count)
if current_itemset == itemset:
      current_count += count
else:
      if current_itemset:
            print(f"{current_itemset}\t{current_count}")
      current_count = count
      current_itemset = itemset
if current_itemset == itemset:
      print(f"{current_itemset}\t{current_count}")
```

Output 1 : NCDC Dataset

```
hadoop@LAPTOP-RB9PEJTU:~/hadoopdata/mapreduce_code$ hdfs dfs -cat /temperature/output/part-r-
00000
Max Temperature 793
Min Temperature -43
```

Browse Directory

/temperature/output Go!

Show 25 entries Search:

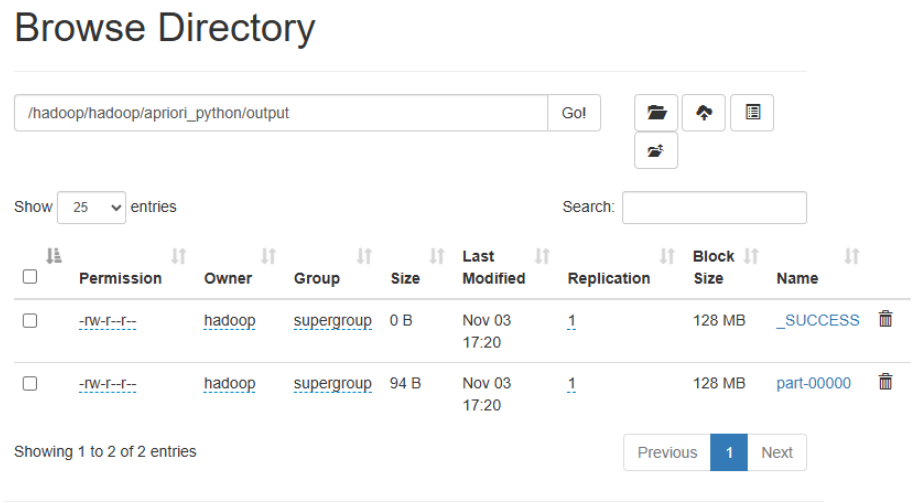
	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
☐	-rw-r--r--	hadoop	supergroup	0 B	Sep 10 14:47	1	128 MB	_SUCCESS
☐	-rw-r--r--	hadoop	supergroup	40 B	Sep 10 14:47	1	128 MB	part-r-00000

Showing 1 to 2 of 2 entries Previous 1 Next

Hadoop, 2023.

Output 2 : Apriori Algorithm

```
        File Input Format Counters
                Bytes Read=67
        File Output Format Counters
                Bytes Written=94
2024-11-03 17:20:55,898 INFO streaming.StreamJob: Output directory: /hadoop/hadoop/apriori_python/output
hadoop@LAPTOP-RB9PEJTU:~/hadoop/apriori_python$ hdfs dfs -cat /hadoop/hadoop/apriori_python/output/part-00000
apple   3
apple,banana    2
apple,banana,orange     1
apple,orange    1
banana  4
banana,orange   3
orange  3
hadoop@LAPTOP-RB9PEJTU:~/hadoop/apriori_python$
```



Result:

Successfully implemented map reduce for NCDC Dataset and executed apriori algorithm using map reduce paradigm.

Ex. No: 5	Spark and PySpark
13.8.2024	

Aim:

Install spark and pyspark. Run a spark shell and test the installation. Run the wordcount program that you did using Hadoop using pyspark. Use the movielens dataset and try to find out for each movie, how are the ratings distributed.

Program

1. Download and Install Spark

    a. wget https://www.apache.org/dyn/closer.lua/spark/spark 3.5.0/spark-3.5.0-bin-hadoop3.tgz

    b. tar -xzf spark-3.5.0-bin-hadoop3.tgz

    c. mv spark-3.5.0-bin-hadoop3 spark

2. Set Environment Variables

    a. nano ~/.bashrc

    b. export SPARK_HOME=~/spark

    c. export PATH=$PATH:$SPARK_HOME/bin

    d. export PYTHONPATH=$SPARK_HOME/python:$PYTHONPATH

    e. export PYSPARK_PYTHON=python3

    f. source ~/.bashrc

3. Running Programs : spark-submit file.py

Wordcount :

```
from pyspark import SparkContext


# Initialize SparkContext

sc = SparkContext("local", "Word Count")
```

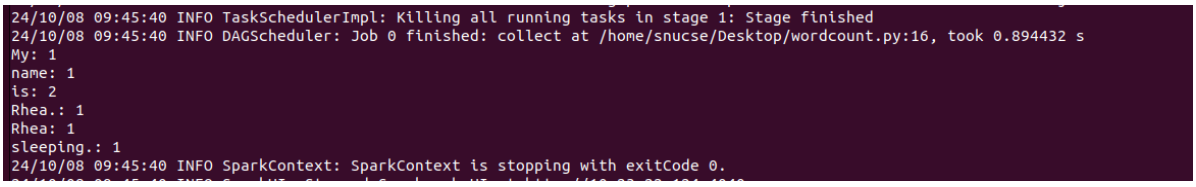
```
# Read input file
input_file = "/home/snucse/Desktop/wordcount_input.txt"
text_file = sc.textFile(input_file)

# Count words
counts = text_file.flatMap(lambda line: line.split(" ")) \
                  .map(lambda word: (word, 1)) \
                  .reduceByKey(lambda a, b: a + b)

# Collect and print results
for word, count in counts.collect():
    print(f"{word}: {count}")

# Stop the SparkContext
sc.stop()
```

Program Movie-lens :

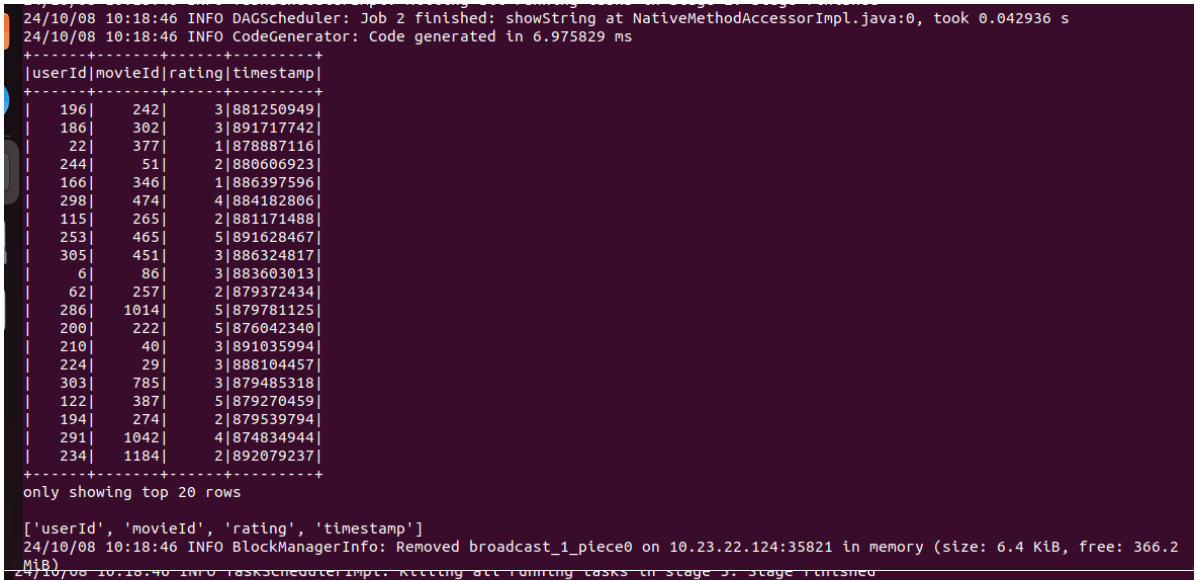
```
from pyspark.sql import SparkSession
from pyspark.sql import functions as F

# Initialize Spark session
spark = SparkSession.builder \
    .appName("MovieLens Ratings Distribution") \
    .getOrCreate()

# Load the MovieLens dataset (change the path accordingly)
ratings = spark.read.csv("/home/snucse/Desktop/movielens.csv",
header=True, inferSchema=True)

# Show the original DataFrame and column names
ratings.show()
```

```
print(ratings.columns)

# Clean up column names (if needed)
ratings = ratings.toDF(*[c.strip() for c in ratings.columns])

# Calculate the count of ratings and average rating for each movieId
rating_distribution = ratings.groupBy("movieId").agg(
    F.count("rating").alias("rating_count"),
    F.avg("rating").alias("average_rating")
)

# Show the results
rating_distribution.show()

# Stop the Spark session
spark.stop()
```

Output Wordcount :



Output Movie-lens:



Result:

Successfully installed Spark and Pyspark, and implemented the Wordount problem and distribution of movie reviews using them.

Ex. No: 6	PySpark 2
20.8.2024	

Aim:

1.Use the friends_test datastet. Col1 is ID, Col2 is name, Col3 is Age and Col4 is num of friends. Understand mapvalues function of RDD in spark and find the average number of friends for each unique age present in the dataset.

2.Use the temp_csv dataset. Column headers are present in the dataset. Understand filter operations and filter out only the “TMIN” values from the “desc” column. With the resultant data (RDD), find the following:

a. Minimum temperature (overall)
b. Minimum temperature for every ItemID
c. Minimum emperature for every StationID

3. Use the same dataset, filter only “TMAX” column and find the maximum temperatures just like the ones mentioned above.

Program:

Friends_test_analysis.py

```
from pyspark import SparkConf, SparkContext

from pyspark.sql

import * from pyspark.sql.functions

import * from pyspark.sql.types import *


spark = SparkSession.builder.appName("friends
test").config("spark.memory.offHeap.e

df = spark.read.csv('friends_test.csv',header=False)

df.explain()

spark.stop()

conf = SparkConf().setAppName("Basicapp").setMaster("local[*]")

sc = SparkContext(conf=conf)

rdd = sc.textFile("friends_test.csv")

rdd.first()

rdd_split = rdd.map(lambda line: line.split(","))

for row in rdd_split.take(5):
```

```
    print(row)
```

Temp_analysis.py

```
from pyspark import SparkConf, SparkContext


# Initialize Spark Context
conf = SparkConf().setAppName("TempDataset").setMaster("local[*]")
sc = SparkContext(conf=conf)
rdd = sc.textFile("temp.csv")


# Split data and remove header
rdd_header = rdd.first()
rdd_data = rdd.filter(lambda row: row != rdd_header).map(lambda row:
row.split(","))


# Filter for TMIN and compute minimum temperatures
rdd_TMIN_filter = rdd_data.filter(lambda row: row[2] == "TMIN")
rdd_min_overall = rdd_TMIN_filter.map(lambda x: int(x[3])).reduce(lambda
a, b: a if a < b else b)
print("Minimum temperature overall:", rdd_min_overall)


rdd_min_itemID = rdd_TMIN_filter.map(lambda x: (x[0],
int(x[3]))).reduceByKey(lambda a, b: a if a < b else b)
print("Minimum temperature by ItemID:", rdd_min_itemID.collect())


rdd_min_stationID = rdd_TMIN_filter.map(lambda x: (x[1],
int(x[3]))).reduceByKey(lambda a, b: a if a < b else b)
print("Minimum temperature by StationID:", rdd_min_stationID.collect())


# Filter for TMAX and compute maximum temperatures
rdd_TMAX_filter = rdd_data.filter(lambda row: row[2] == "TMAX")
```

```
rdd_max_overall = rdd_TMAX_filter.map(lambda x: int(x[3])).reduce(lambda
a, b: a if a > b else b)

print("Maximum temperature overall:", rdd_max_overall)


rdd_max_itemID = rdd_TMAX_filter.map(lambda x: (x[0],
int(x[3]))).reduceByKey(lambda a, b: a if a > b else b)

print("Maximum temperature by ItemID:", rdd_max_itemID.collect())


rdd_max_stationID = rdd_TMAX_filter.map(lambda x: (x[1],
int(x[3]))).reduceByKey(lambda a, b: a if a > b else b)

print("Maximum temperature by StationID:", rdd_max_stationID.collect())


sc.stop()
```

Output : Friends_test_analysis.py (part a)

```
hadoop@LAPTOP-RB9PEJTU:~/hadoop/pyspark_friends$ spark-submit friends_test_analysis.py
24/11/03 18:03:06 WARN Utils: Your hostname, LAPTOP-RB9PEJTU resolves to a loopback address: 127.0.1.1; using 10.255.255.254 instead (on interface lo)
24/11/03 18:03:06 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
24/11/03 18:03:07 INFO SparkContext: Running Spark version 3.2.0
```

```
24/11/03 18:03:11 INFO TaskSchedulerImpl: Removed TaskSet 6.0, whose tasks have all completed, from pool
24/11/03 18:03:11 INFO DAGScheduler: ResultStage 6 (collect at /home/hadoop/hadoop/pyspark_friends/friends_test_analysis.py:17) finished in 0.068 s
24/11/03 18:03:11 INFO DAGScheduler: Job 2 is finished. Cancelling potential speculative or zombie tasks for this job
24/11/03 18:03:11 INFO TaskSchedulerImpl: Killing all running tasks in stage 6: Stage finished
24/11/03 18:03:11 INFO DAGScheduler: Job 2 finished: collect at /home/hadoop/hadoop/pyspark_friends/friends_test_analysis.py:17, took 0.156635 s
Age: 18, Average Number of Friends: 343.38
Age: 19, Average Number of Friends: 213.27
Age: 20, Average Number of Friends: 165.00
Age: 21, Average Number of Friends: 350.88
Age: 22, Average Number of Friends: 206.43
Age: 23, Average Number of Friends: 246.30
Age: 24, Average Number of Friends: 233.80
Age: 25, Average Number of Friends: 197.45
Age: 26, Average Number of Friends: 242.06
Age: 27, Average Number of Friends: 228.12
Age: 28, Average Number of Friends: 209.10
Age: 29, Average Number of Friends: 215.92
Age: 30, Average Number of Friends: 235.82
Age: 31, Average Number of Friends: 267.25
Age: 32, Average Number of Friends: 207.91
Age: 33, Average Number of Friends: 325.33
Age: 34, Average Number of Friends: 245.50
Age: 35, Average Number of Friends: 211.62
Age: 36, Average Number of Friends: 246.60
Age: 37, Average Number of Friends: 249.33
Age: 38, Average Number of Friends: 193.53
Age: 39, Average Number of Friends: 169.29
Age: 40, Average Number of Friends: 250.82
Age: 41, Average Number of Friends: 268.56
Age: 42, Average Number of Friends: 303.50
Age: 43, Average Number of Friends: 230.57
Age: 44, Average Number of Friends: 282.17
Age: 45, Average Number of Friends: 309.54
Age: 46, Average Number of Friends: 223.69
Age: 47, Average Number of Friends: 233.22
Age: 48, Average Number of Friends: 281.40
Age: 49, Average Number of Friends: 184.67
Age: 50, Average Number of Friends: 254.60
Age: 51, Average Number of Friends: 302.14
```

```
Age: 52, Average Number of Friends: 340.64
Age: 53, Average Number of Friends: 222.86
Age: 54, Average Number of Friends: 278.08
Age: 55, Average Number of Friends: 295.54
Age: 56, Average Number of Friends: 306.67
Age: 57, Average Number of Friends: 258.83
Age: 58, Average Number of Friends: 116.55
Age: 59, Average Number of Friends: 220.00
Age: 60, Average Number of Friends: 202.71
Age: 61, Average Number of Friends: 256.22
Age: 62, Average Number of Friends: 220.77
Age: 63, Average Number of Friends: 384.00
Age: 64, Average Number of Friends: 281.33
Age: 65, Average Number of Friends: 298.20
Age: 66, Average Number of Friends: 276.44
Age: 67, Average Number of Friends: 214.62
Age: 68, Average Number of Friends: 269.60
Age: 69, Average Number of Friends: 235.20
24/11/03 18:03:11 INFO SparkUI: Stopped Spark web UI at http://10.255.255.254:4040
24/11/03 18:03:11 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
24/11/03 18:03:11 INFO MemoryStore: MemoryStore cleared
24/11/03 18:03:11 INFO BlockManager: BlockManager stopped
24/11/03 18:03:11 INFO BlockManagerMaster: BlockManagerMaster stopped
24/11/03 18:03:11 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
24/11/03 18:03:11 INFO SparkContext: Successfully stopped SparkContext
24/11/03 18:03:12 INFO ShutdownHookManager: Shutdown hook called
24/11/03 18:03:12 INFO ShutdownHookManager: Deleting directory /tmp/spark-bb2dc12a-5411-48e4-b790-a529bdf63ffc
24/11/03 18:03:12 INFO ShutdownHookManager: Deleting directory /tmp/spark-47d3c0d8-4f6d-468d-bb3c-029f245e1e42/pyspark-dda7de57-8078-4f52-be24-e258d9a
9e689
24/11/03 18:03:12 INFO ShutdownHookManager: Deleting directory /tmp/spark-47d3c0d8-4f6d-468d-bb3c-029f245e1e42
```

Output : Test_analysis.py (part b and c)

```
hadoop@LAPTOP-RB9PEJTU:~/hadoop/pyspark_friends$ spark-submit temp_analysis.py
24/11/03 18:03:32 WARN Utils: Your hostname, LAPTOP-RB9PEJTU resolves to a loopback a
24/11/03 18:03:32 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another addre
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
24/11/03 18:03:33 INFO SparkContext: Running Spark version 3.2.0
24/11/03 18:03:33 WARN NativeCodeLoader: Unable to load native-hadoop library for you
```

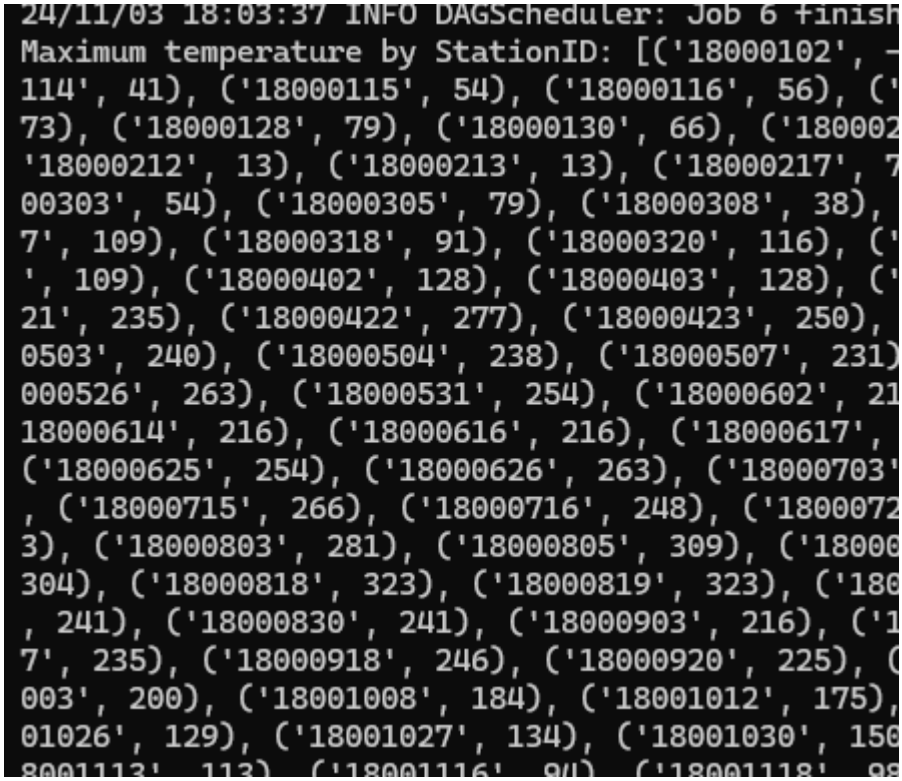
```
Minimum temperature overall: -148
```

```
Minimum temperature by ItemID: [('ITE00100554', -148), ('EZE00100082', -135)]
```

```
Minimum temperature by StationID: [('18000102', -13
('18000114', -35), ('18000115', -23), ('18000116',
18000127', 15), ('18000128', 33), ('18000130', 3),
0211', -102), ('18000212', -78), ('18000213', -42),
8000228', -43), ('18000303', -30), ('18000305', -47
, ('18000315', -52), ('18000317', -52), ('18000318'
, ('18000331', 11), ('18000401', 50), ('18000402',
18000420', 125), ('18000421', 153), ('18000422', 15
('18000502', 139), ('18000503', 154), ('18000504',
 ('18000519', 105), ('18000526', 160), ('18000531',
), ('18000611', 104), ('18000614', 126), ('18000616
('18000624', 115), ('18000625', 100), ('18000626',
, ('18000714', 144), ('18000715', 124), ('18000716'
0), ('18000802', 154), ('18000803', 183), ('1800080
181), ('18000816', 198), ('18000818', 185), ('18000
, 146), ('18000829', 149), ('18000830', 154), ('180
4', 104), ('18000917', 110), ('18000918', 118), ('1
1', 110), ('18001003', 94), ('18001008', 104), ('18
52), ('18001026', 50), ('18001027', 45), ('18001030
```

```
Maximum temperature overall: 323
```

```
Maximum temperature by ItemID: [('ITE00100554', 323), ('EZE00100082', 323)]
```



Result:

Successfully utilized map value functions for various tasks in pyspark.

Ex. No: 7	Hadoop and Docker
27.8.2024	

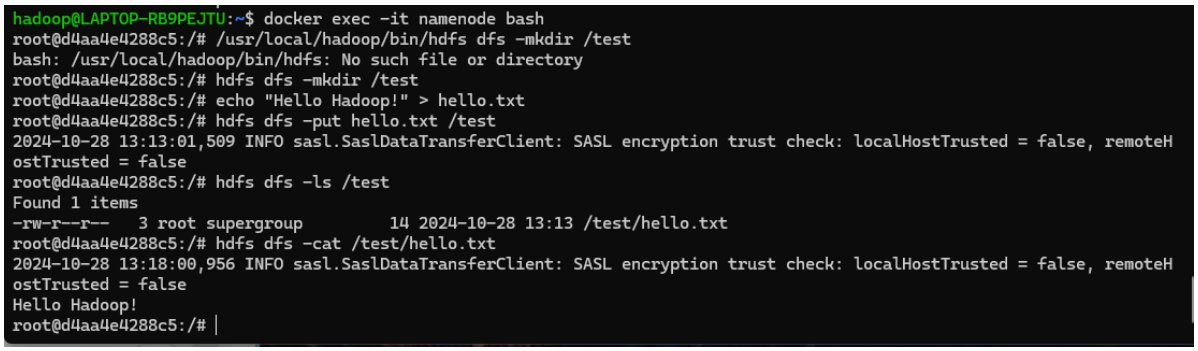
Aim:

Set up a simple Hadoop environment using Docker container, including atleast one NameNode and one DataNode. Ensure the containers are properly configured to interact with each other. After the setup, verify the Hadoop cluster is operational by running a simle HDFS file operation (e,g, uploading a file to HDFS)

Output:

```
hadoop@LAPTOP-RB9PEJTU:~$ docker network create hadoop-net
c2c3697ab95ac230b5addbcaf905420439251c7df0f024ad2a30145dc8803369
hadoop@LAPTOP-RB9PEJTU:~$ docker run -d --name namenode \
  --network hadoop-net \
  -e CLUSTER_NAME=test-cluster \
  -e CORE_CONF_fs_defaultFS=hdfs://namenode:9000 \
  bde2020/hadoop-namenode
Unable to find image 'bde2020/hadoop-namenode:latest' locally
latest: Pulling from bde2020/hadoop-namenode
3192219afd04: Pull complete
7127a1d8cced: Pull complete
883a89599900: Pull complete
77920a3e82af: Pull complete
92329e81aec4: Pull complete
f373218fec59: Pull complete
aa53513fe997: Pull complete
8b1800105b98: Pull complete
c3a84a3e49c8: Pull complete
a65640a64a76: Pull complete
a29cc756d786: Pull complete
abf352b16046: Pull complete
dddd5a449e99: Pull complete
Digest: sha256:fdf74110805132d646cf6f12635efc0919e1fb2ac5bd376c5366272fc261301e
Status: Downloaded newer image for bde2020/hadoop-namenode:latest
d4aa4e4288c5f71009642a02e01a2a3a1a1cca455c525581c0d98c4c162447dd
```

```
hadoop@LAPTOP-RB9PEJTU:~$ docker run -d --name datanode \
  --network hadoop-net \
  -e CORE_CONF_fs_defaultFS=hdfs://namenode:9000 \
  -e SERVICE_PRECONDITION="namenode:9000" \
  bde2020/hadoop-datanode
Unable to find image 'bde2020/hadoop-datanode:latest' locally
latest: Pulling from bde2020/hadoop-datanode
3192219afd04: Already exists
7127a1d8cced: Already exists
883a89599900: Already exists
77920a3e82af: Already exists
92329e81aec4: Already exists
f373218fec59: Already exists
aa53513fe997: Already exists
8b1800105b98: Already exists
c3a84a3e49c8: Already exists
a65640a64a76: Already exists
4bf0ae3d5cc8: Pull complete
b91d0b0b68c8: Pull complete
5e185246c615: Pull complete
Digest: sha256:35f899bcbe9f983825a8a3bdc135ed0e8e0eaf3b58f9b08bf257b5e86bae3b47
Status: Downloaded newer image for bde2020/hadoop-datanode:latest
ae8eec62ba572dd934f228a09179876d0d8aa6295568bd220b31179a4797cbb6
```



Result:

Successfully setup Hadoop environment and ran a Hadoop code in Docker

Ex. No: 8	Public and Private Keys
3.9.2024	

Aim:

To secure an EC2 instance using SSH Keys and Network Access Control.

Program:

1. Generate Private and Public Key Pairs

○ Linux/Mac: ■ Create a new directory for key pairs: mkdir key-pair-labs && cd key-pair-labs

■ Generate a private key: openssl genrsa-out snu-privatekey.pem 2048

■ Generate a public key from the private key: openssl rsa-in snu-privatekey.pem-pubout-out snu-publickey.pem

■ Set permissions on the private key: chmod 400 snu-privatekey.pem

■ Copy the public key: cat snu-publickey.pem

○ Paste the public key into the AWS console.

2. Launch an Ubuntu EC2 Instance:

○ Launch a new EC2 instance using the public key.

○ Choose an appropriate instance type and AMI.

○ Allocate an Elastic IP address and attach it to the instance.

3. Login to the Instance: ○ Use the private key generated in step 1 to SSH into the instance.

○ Linux / Mac : ssh-i snu-privatekey.pem ubuntu@

4. Edit Security Group: ○ Open the Security Group settings for the instance.

○ Add an inbound rule to allow ICMP traffic (ping) from anywhere:

■ Type: Custom TCP Rule

■ Protocol: ICMP

■ Port Range: All

■ Source: 0.0.0.0/0

5. Ping the Instance:

○ Ping the public IP address of the instance to verify connectivity.

○ You should be able to ping the instance successfully.

6. Edit Default NACL: ○ Open the Network Access Control List (NACL) settings for the instance.

○ Edit the default NACL to block ICMP traffic.

○ Add an egress rule to deny ICMP traffic:

■ Type: Custom TCP Rule

■ Protocol: ICMP

■ Port Range: All

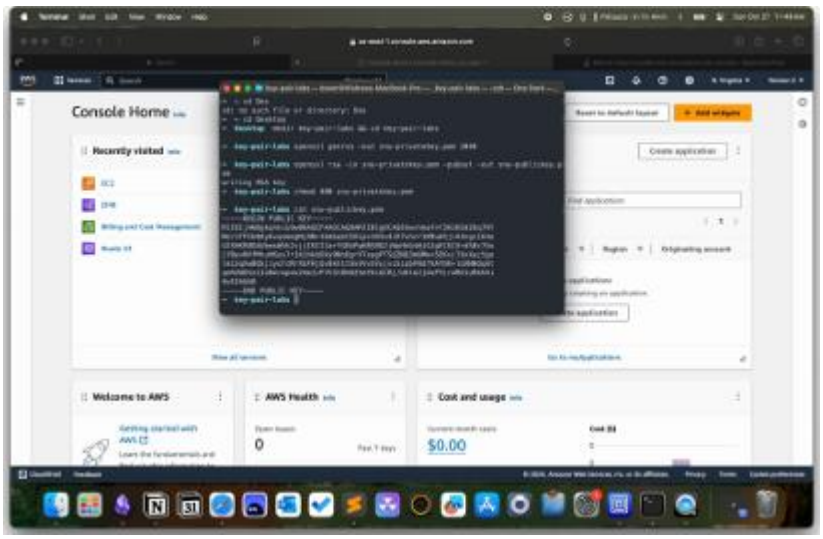
■ Destination: 0.0.0.0/0

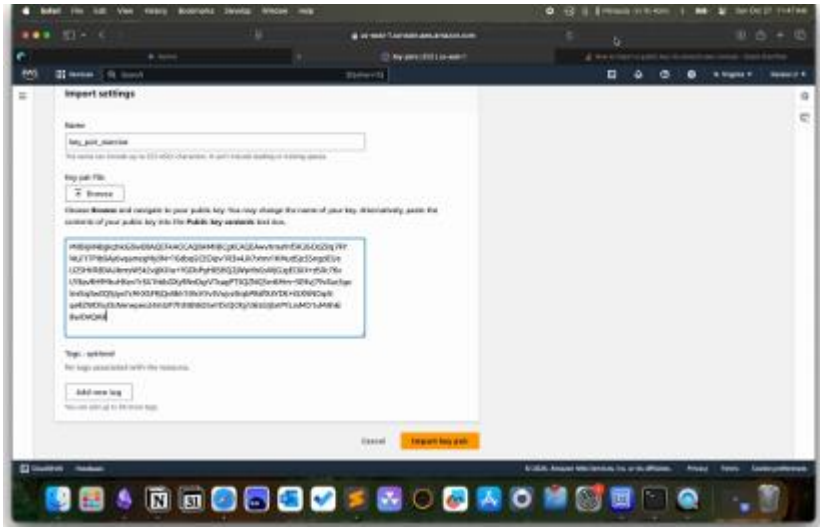
7. Ping the Instance (Again):

○ Ping the public IP address of the instance again.

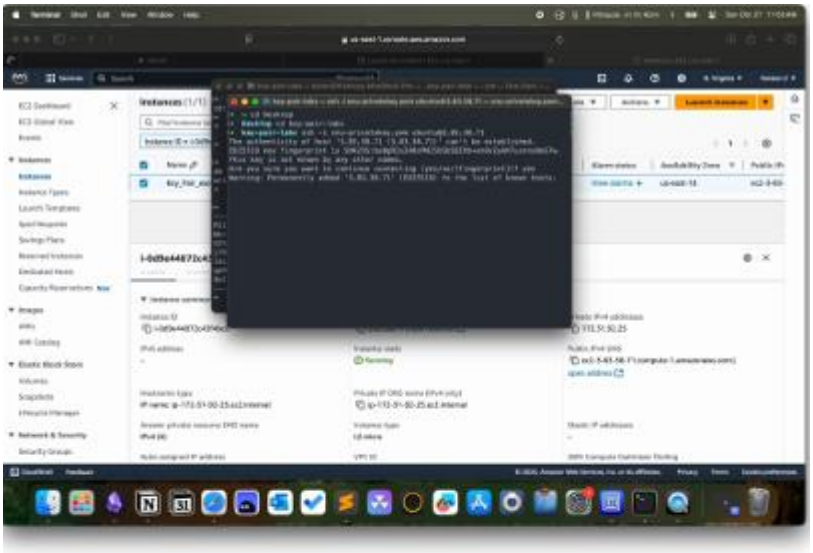
○ You should not be able to ping the instance this time, as ICMP traffic is now blocked

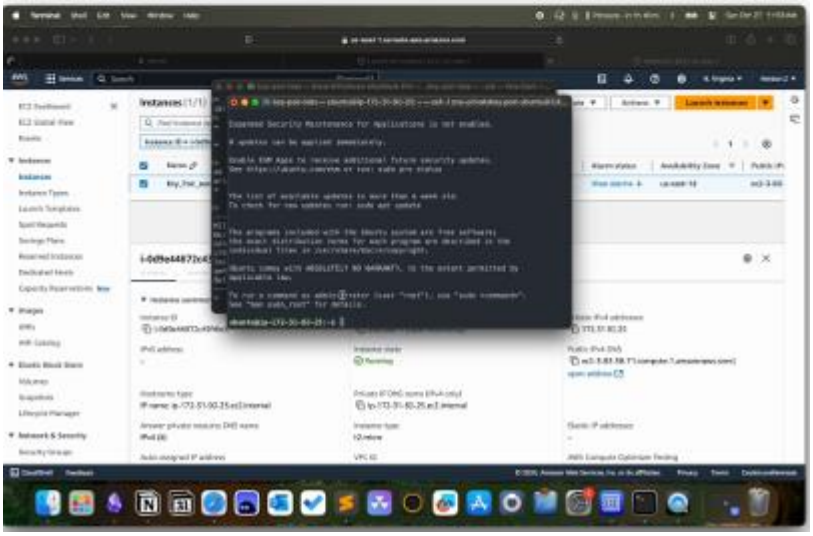
Output:

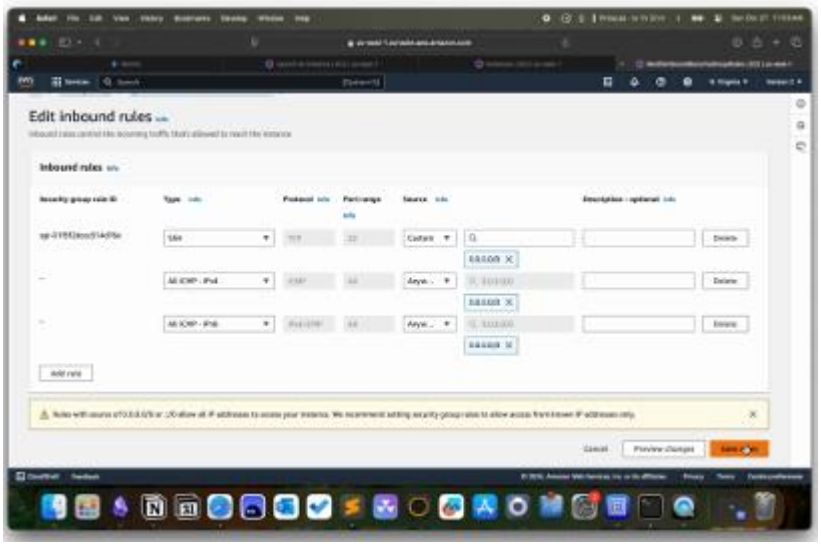


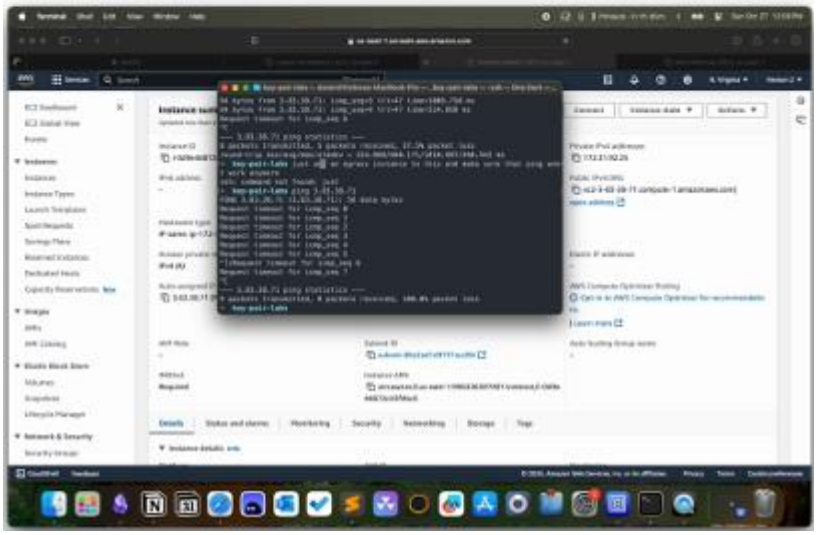

Import settings
Import key pair


Instance summary for i-0d9e44872c43f4bc0 (Key_Pair_exercise)






Edit inbound rules
Inbound rules
Add rule
Cancel
Preview changes
Save rules



Result:

Successfully created public and private keys and utilized it to secure EC2 instances

Ex. No: 9	EC2
10.8.2024	

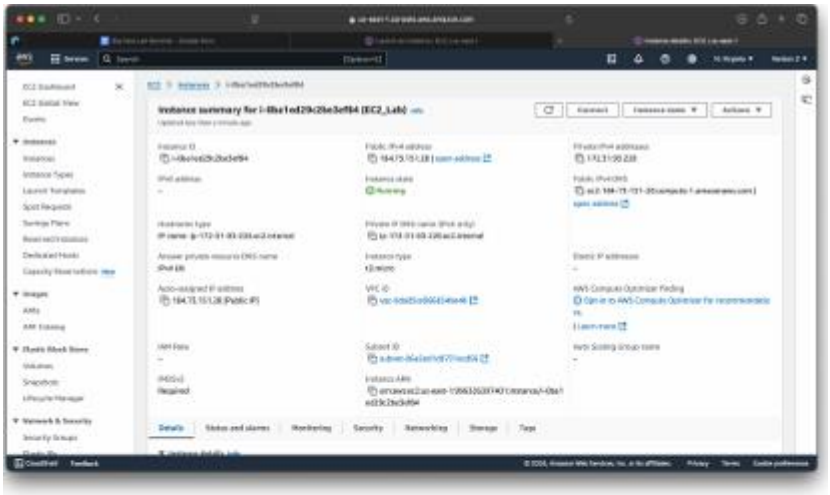
Aim:

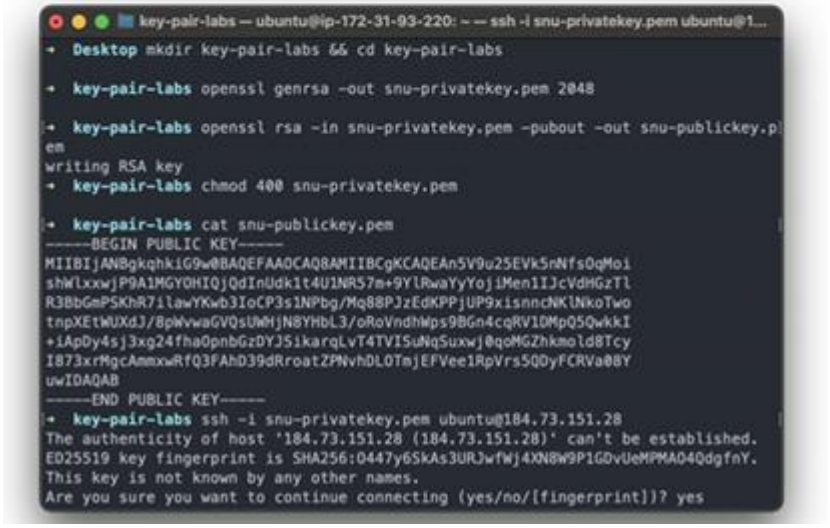
To create an EC2 instance and Deploy a web server on it using Apache2

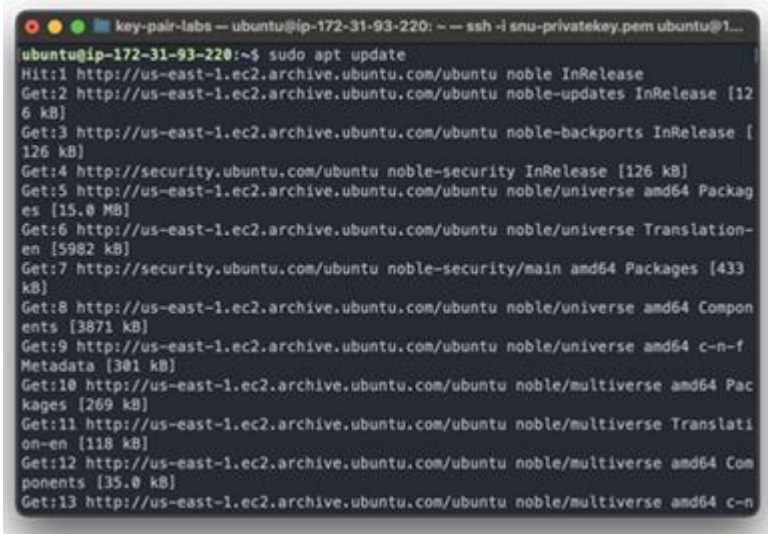
Program:

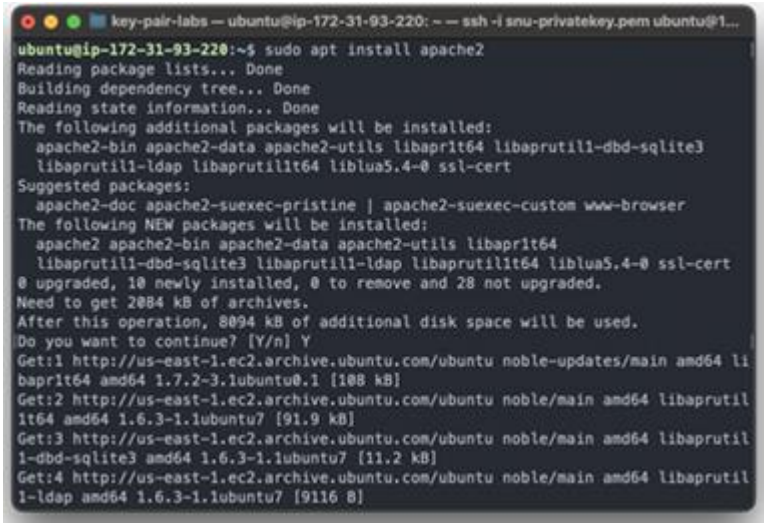
1. Create a Key Pair:
    - Use the AWS Management Console to create a new key pair.
    - Download the private key file (.pem) and save it securely.
2. Launch EC2 Instance:
    - Ubuntu AMI, ○ Choose an instance type based on your requirements.
    - Configure security groups to allow SSH and HTTP traffic.
    - Launch the instance and provide the private key file during the launch process.
3. Connect to Instance:
    - Use the SSH client to connect to the instance using the public IP address and private key file.
4. Update Package Lists:
    - Run the following command to update the package lists : sudo apt update
5. Install Apache2: sudo apt install apache2
6. Check Apache Status:sudo service apache2 status
7. Test Apache:
    - Open a web browser and enter the public IP address of the instance.
    - You should see the default Apache2 welcome page.
8. Change the Contents of the Webpage:
    - cd /var/www/html/ ○ sudo chmod 777 index.html
    - echo "website" > index.html
9. Test the website to show new content

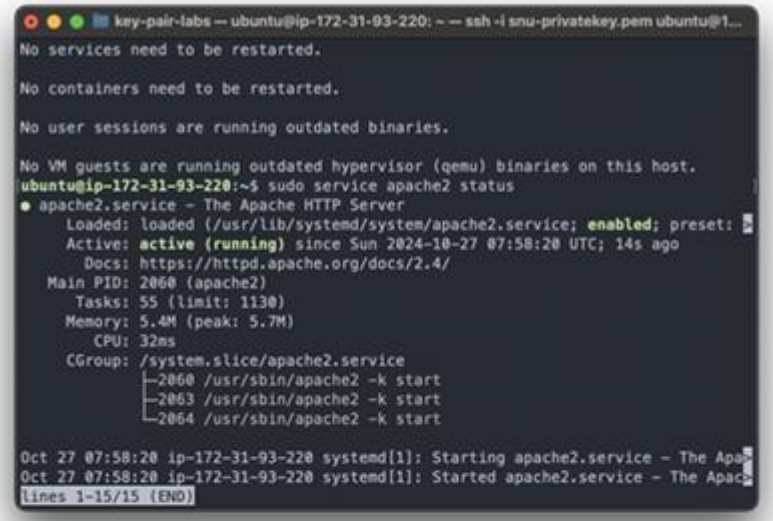
Output:



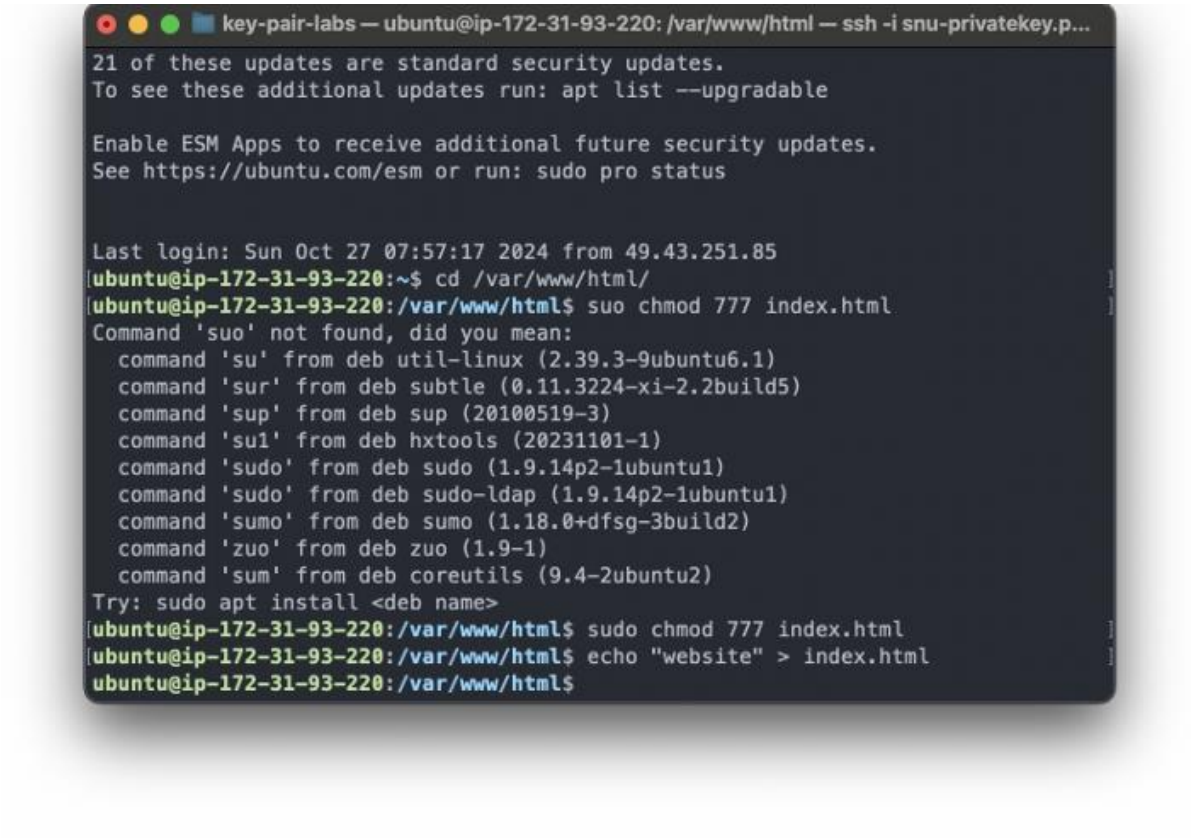

key-pair-labs — ubuntu@ip-172-31-93-220: ~ — ssh -i snu-privatekey.pem ubuntu@1...
→ Desktop mkdir key-pair-labs && cd key-pair-labs
→ key-pair-labs openssl genrsa -out snu-privatekey.pem 2048
→ key-pair-labs openssl rsa -in snu-privatekey.pem -pubout -out snu-publickey.p
em
writing RSA key
→ key-pair-labs chmod 400 snu-privatekey.pem
→ key-pair-labs cat snu-publickey.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAn5V9u25EVk5nNfsOqMoi
shWlxxwjP9A1MGYOHIQjQdInUdk1t4U1NR57m+9YlRwaYyYojiMen1IJcVdMGzTl
R3BbGmPSKhR7ilawYKwb3IoCP3s1NPbg/Mq88PJzEdKPPjUP9xisnncNKlNkoTwo
tnpXEtWUXdJ/8pWvwaGVQsUWHjN8YHbL3/oRoVndhWps9BGn4cqRV1DMpQ5QwkkI
+iApDy4sj3xg24fhaOpnbGzDYJSikarqLvT4TVI5uNqSuxwj0qoMGZhkmold8Tcy
I873xrMgcAmmxwRfQ3FAhD39dRroatZPNvhDLOTmjEFVee1RpVrs5QDyFCRVa08Y
uwIDAQAB
-----END PUBLIC KEY-----
→ key-pair-labs ssh -i snu-privatekey.pem ubuntu@184.73.151.28
The authenticity of host '184.73.151.28 (184.73.151.28)' can't be established.
ED25519 key fingerprint is SHA256:0447y6SkAs3URJwfWj4XN8W9P1GDvUeMPMAO4QdgfnY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

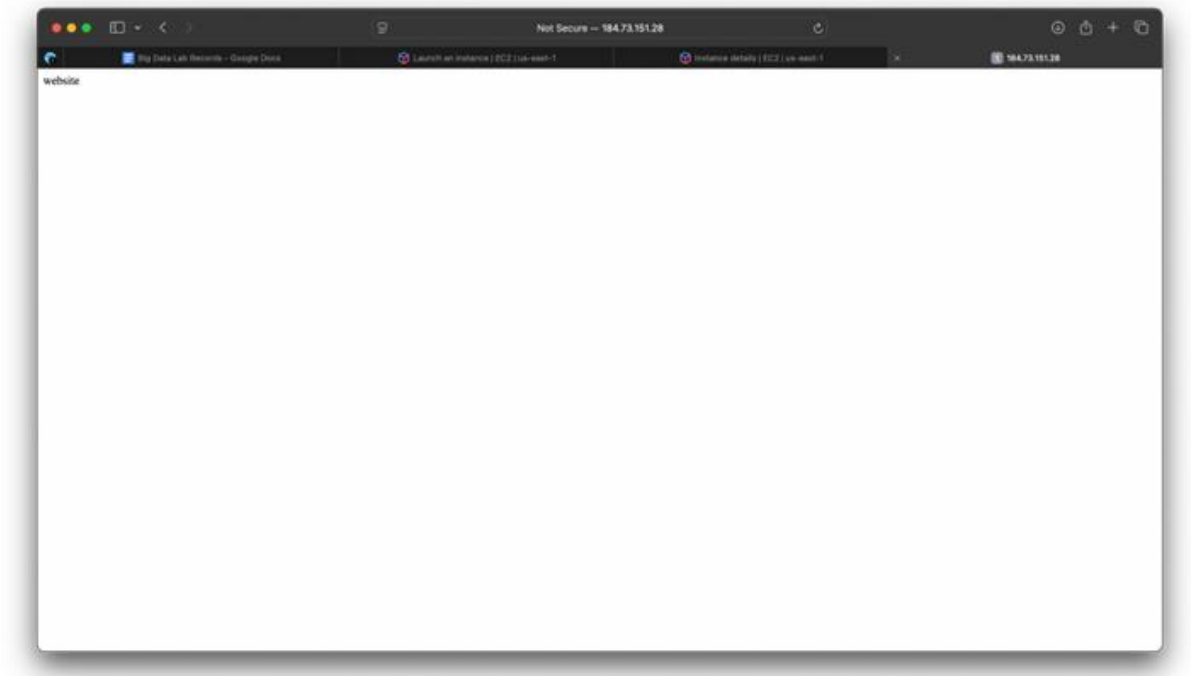

key-pair-labs — ubuntu@ip-172-31-93-220: ~ — ssh -i snu-privatekey.pem ubuntu@1...
ubuntu@ip-172-31-93-220:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [12
6 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [
126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packag
es [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-
en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [433
kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Compon
ents [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f
Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Pac
kages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translati
on-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Com
ponents [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n


key-pair-labs — ubuntu@ip-172-31-93-220: ~ — ssh -i snu-privatekey.pem ubuntu@1...
ubuntu@ip-172-31-93-220:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3
libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
Suggested packages:
apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
apache2 apache2-bin apache2-data apache2-utils libapr1t64
libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 28 not upgraded.
Need to get 2084 kB of archives.
After this operation, 8094 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 li
bapr1t64 amd64 1.7.2-3.1ubuntu0.1 [108 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil
1t64 amd64 1.6.3-1.1ubuntu7 [91.9 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil
1-dbd-sqlite3 amd64 1.6.3-1.1ubuntu7 [11.2 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil
1-ldap amd64 1.6.3-1.1ubuntu7 [9116 B]


key-pair-labs — ubuntu@ip-172-31-93-220: ~ — ssh -i snu-privatekey.pem ubuntu@1...
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-93-220:~$ sudo service apache2 status
● apache2.service - The Apache HTTP Server
Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset:
Active: active (running) since Sun 2024-10-27 07:58:20 UTC; 14s ago
Docs: https://httpd.apache.org/docs/2.4/
Main PID: 2060 (apache2)
Tasks: 55 (limit: 1130)
Memory: 5.4M (peak: 5.7M)
CPU: 32ms
CGroup: /system.slice/apache2.service
├─2060 /usr/sbin/apache2 -k start
├─2063 /usr/sbin/apache2 -k start
└─2064 /usr/sbin/apache2 -k start
Oct 27 07:58:20 ip-172-31-93-220 systemd[1]: Starting apache2.service - The Apa
Oct 27 07:58:20 ip-172-31-93-220 systemd[1]: Started apache2.service - The Apac
lines 1-15/15 (END)


Apache2 Default Page
Ubuntu
It works!
Configuration Overview





Result:

Successfully deployed an EC2 instance and tested using Apache Webservices.

Ex. No: 10	Route53
1.10.2024	

Aim:

To set up a web server on an AWS EC2 instance and configure a domain name using GoDaddy and Route 53

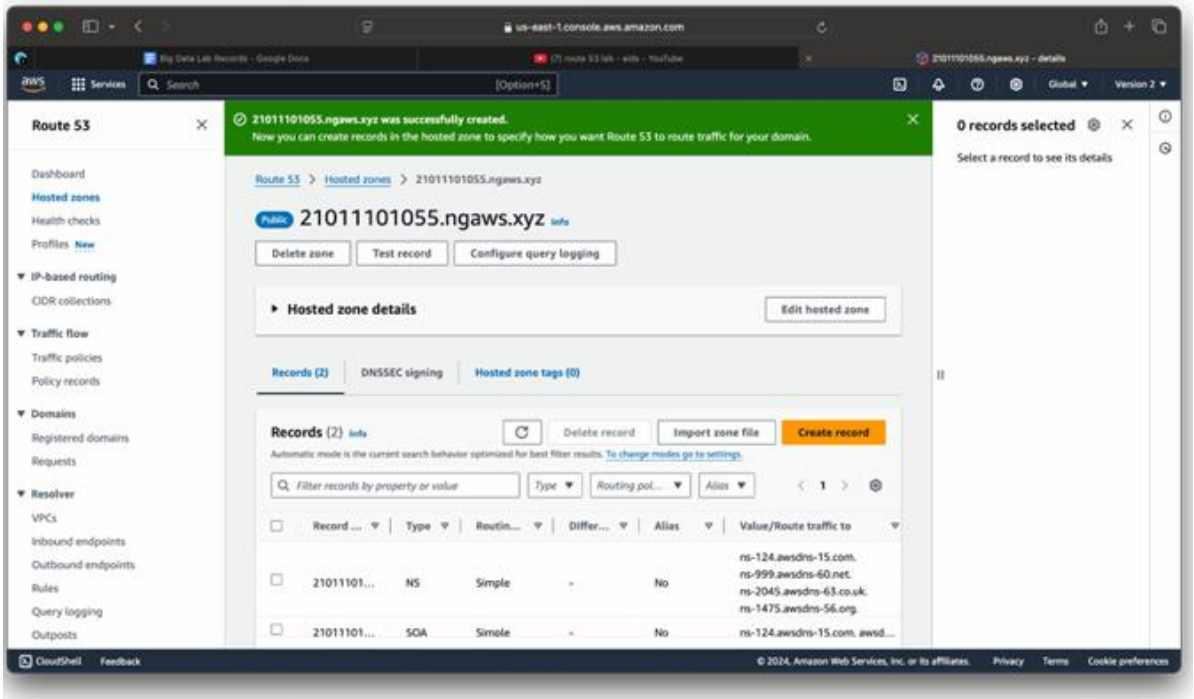
Program:

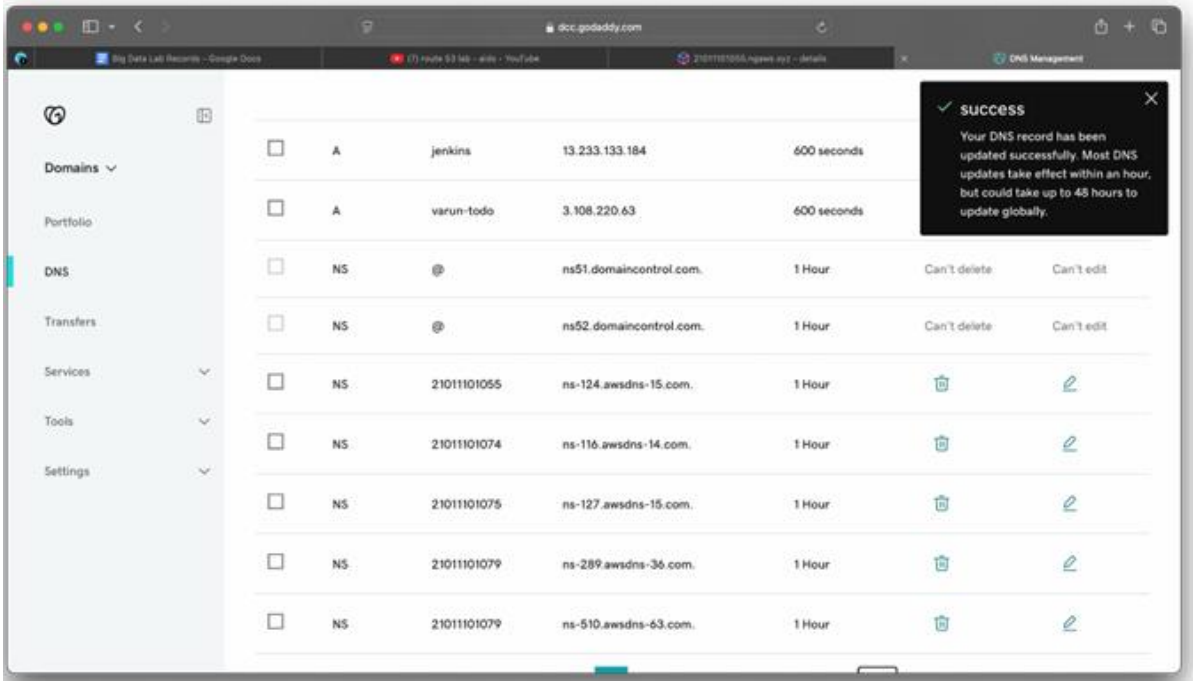
1. Login to GoDaddy and Create a subdomain:
    - Use your 12-digit registration number.
    - Example: 21100101001.ngaws.xyz
2. Create a Hosted Zone in Route 53:
    - Navigate to the Route 53 service in the AWS console.
    - Create a new Hosted Zone for your subdomain.
3. Get the Name Server information:
    - Retrieve the name server information from the Route 53 dashboard.
    - Example: ns-1234.awsdns-12.org, [invalid URL removed]
4. Update the NS record in GoDaddy:
    - Log in to the GoDaddy portal.
    - Navigate to the DNS settings for your subdomain.
    - Update the NS records with the name servers obtained from Route 53.
5. Create an EC2 instance:
    - Launch an EC2 instance with an elastic public IP address.
6. Install Apache:
    - Connect to the EC2 instance using SSH.
    - Update the package lists: sudo apt update
    - Install Apache: sudo apt install apache2
7. Configure Apache:
    - Edit the Apache configuration file: sudo nano /etc/apache2/sites-available/000-default.conf
    - Modify the DocumentRoot and ServerName directives to point to your desired web content directory and domain name.

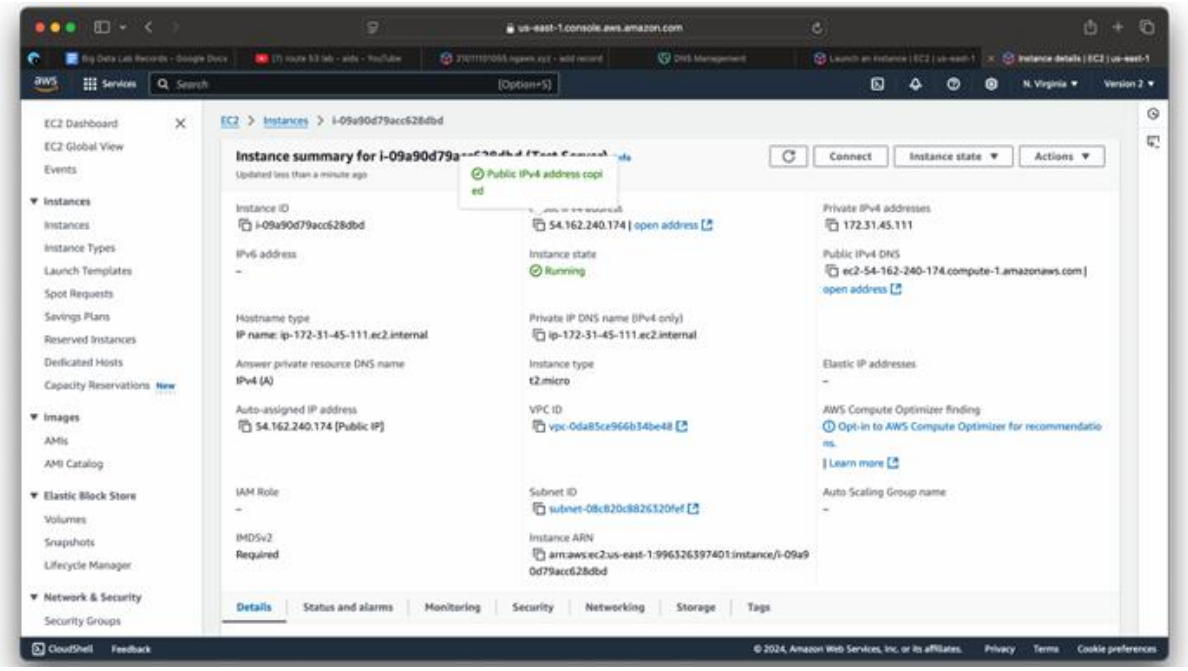
- Save the configuration file and restart Apache: sudo systemctl restart apache2

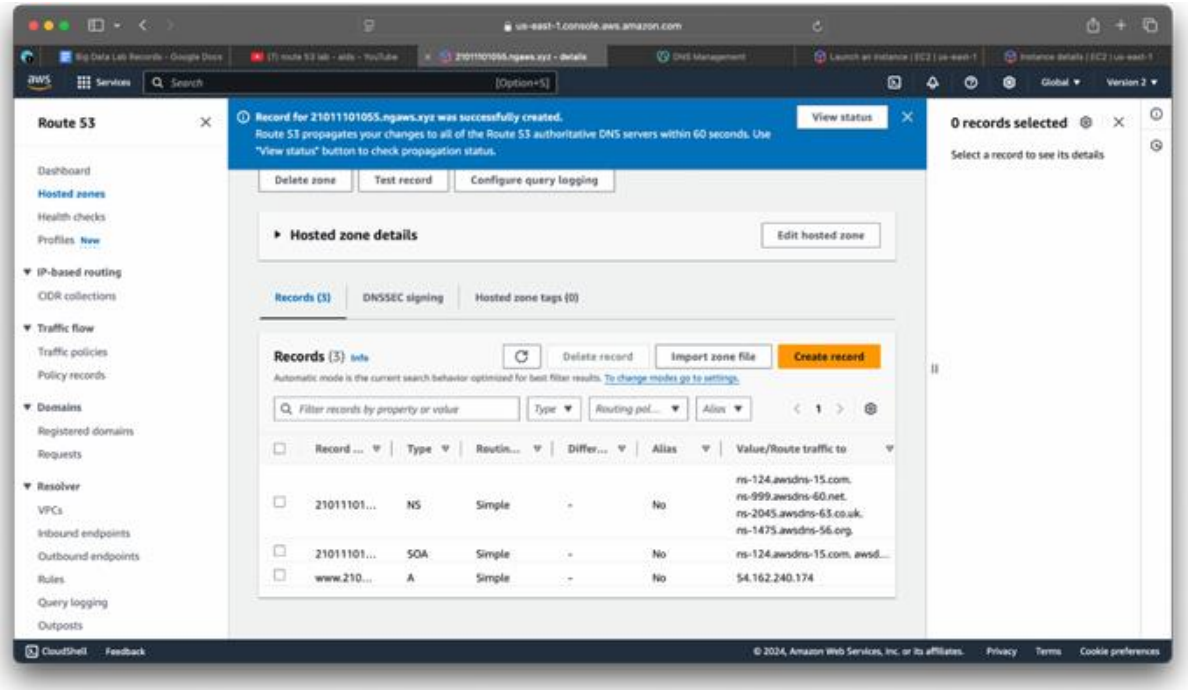
8. Create an A record in Route 53:
   - Navigate to your Hosted Zone in Route 53.
   - Create a new A record with the following details:
     - Name: @ (for the root domain)
     - Value: The public IP address of your EC2 instance
     - TTL: 3600 (1 hour)

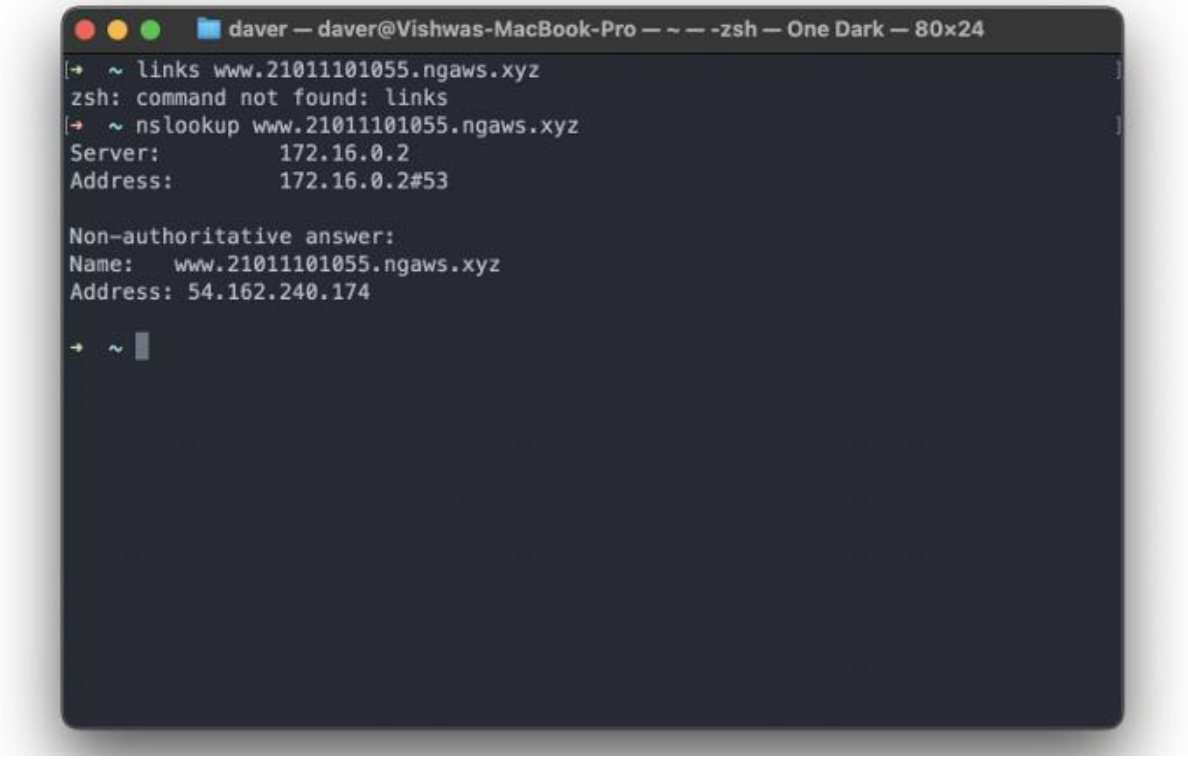
Output:


us-east-1.console.aws.amazon.com
Route 53
21011101055.ngaws.xyz was successfully created.
Now you can create records in the hosted zone to specify how you want Route 53 to route traffic for your domain.
0 records selected
Select a record to see its details
Route 53 > Hosted zones > 21011101055.ngaws.xyz
Public 21011101055.ngaws.xyz info
Delete zone
Test record
Configure query logging
Hosted zone details
Edit hosted zone
Records (2)
DNSSEC signing
Hosted zone tags (0)
Records (2) info
Delete record
Import zone file
Create record
Filter records by property or value
Record ...
Type
Routin...
Differ...
Alias
Value/Route traffic to
21011101...
NS
Simple
No
ns-124.awsdns-15.com.
ns-999.awsdns-60.net.
ns-2045.awsdns-63.co.uk.
ns-1475.awsdns-56.org.
21011101...
SOA
Simple
No
ns-124.awsdns-15.com. awsd...
Dashboard
Hosted zones
Health checks
Profiles New
IP-based routing
CIDR collections
Traffic flow
Traffic policies
Policy records
Domains
Registered domains
Requests
Resolver
VPCs
Inbound endpoints
Outbound endpoints
Rules
Query logging
Outposts
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


dcc.godaddy.com
DNS Management
success
Your DNS record has been updated successfully. Most DNS updates take effect within an hour, but could take up to 48 hours to update globally.
Domains
Portfolio
DNS
Transfers
Services
Tools
Settings
A jenkins 13.233.133.184 600 seconds
A varun-todo 3.108.220.63 600 seconds
NS @ ns51.domaincontrol.com. 1 Hour Can't delete Can't edit
NS @ ns52.domaincontrol.com. 1 Hour Can't delete Can't edit
NS 21011101055 ns-124.awsdns-15.com. 1 Hour
NS 21011101074 ns-116.awsdns-14.com. 1 Hour
NS 21011101075 ns-127.awsdns-15.com. 1 Hour
NS 21011101079 ns-289.awsdns-36.com. 1 Hour
NS 21011101079 ns-510.awsdns-63.com. 1 Hour


us-east-1.console.aws.amazon.com
Services
Search
[Option+S]
N. Virginia
Version 2
EC2 Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
EC2 > Instances > i-09a90d79acc628dbd
Instance summary for i-09a90d79a
Updated less than a minute ago
Public IPv4 address copi
ed
Connect
Instance state
Actions
Instance ID
i-09a90d79acc628dbd
54.162.240.174 | open address
Private IPv4 addresses
172.31.45.111
IPv6 address
–
Instance state
Running
Public IPv4 DNS
ec2-54-162-240-174.compute-1.amazonaws.com |
open address
Hostname type
IP name: ip-172-31-45-111.ec2.internal
Private IP DNS name (IPv4 only)
ip-172-31-45-111.ec2.internal
Answer private resource DNS name
IPv4 (A)
Instance type
t2.micro
Elastic IP addresses
–
Auto-assigned IP address
54.162.240.174 [Public IP]
VPC ID
vpc-0da85ce966b34be48
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendatio
ns.
| Learn more
IAM Role
–
Subnet ID
subnet-08c820c8826320fef
Auto Scaling Group name
–
IMDSv2
Required
Instance ARN
arn:aws:ec2:us-east-1:996326397401:instance/i-09a9
0d79acc628dbd
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences





Result:

Successfully implemented a Route53 DNS Lookup on AWS.

Ex. No: 11	IAM
8.10.2024	

Aim:

To implement IAM user access in AWS Console.

Program:

1. Create User Groups server_admin and dns_admin. Give them full access to EC2 Service and Route53 services.

2. Create users, alice, bob, cathay and david. Set passwords for them

3. Add alice and bob to user group server_adin and cathy and david to user group dns_admin.

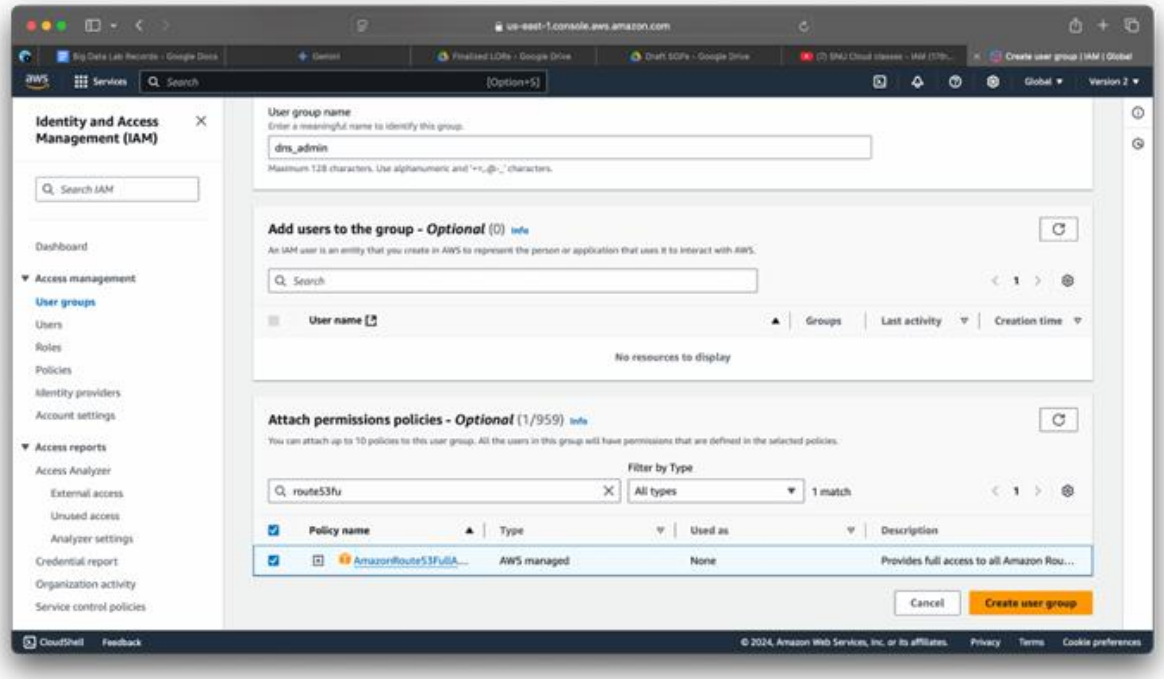
4. Create user eve and give him Billing access.

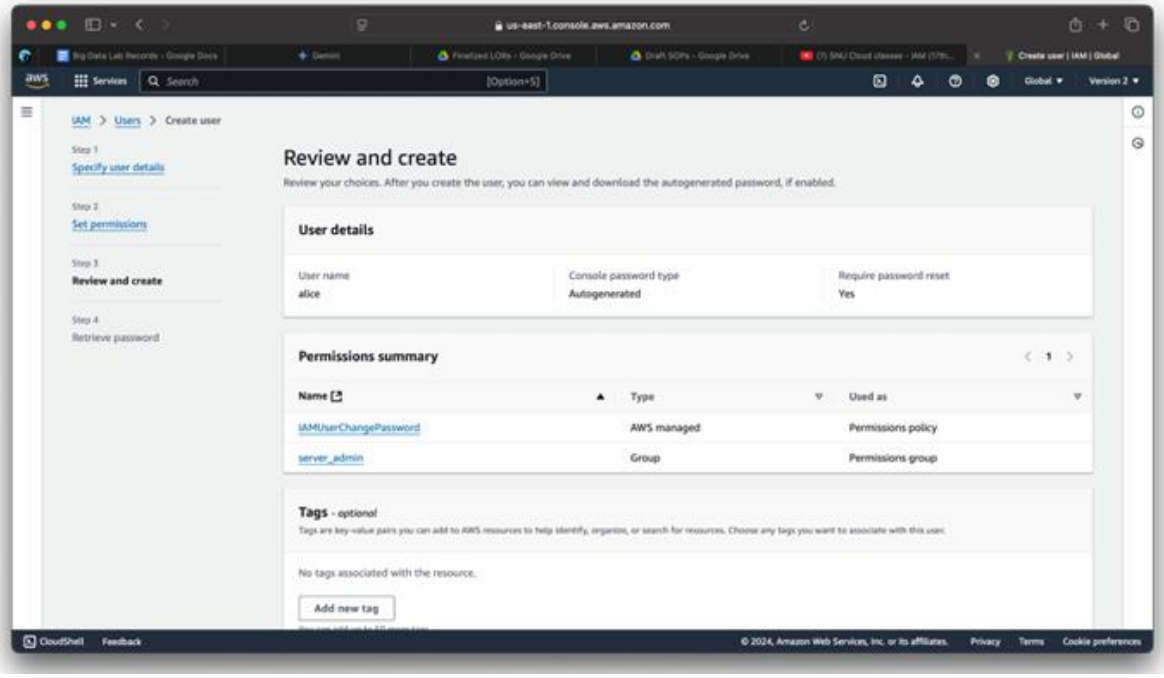
5. Create user hadoop and give full (admin) access to the services.

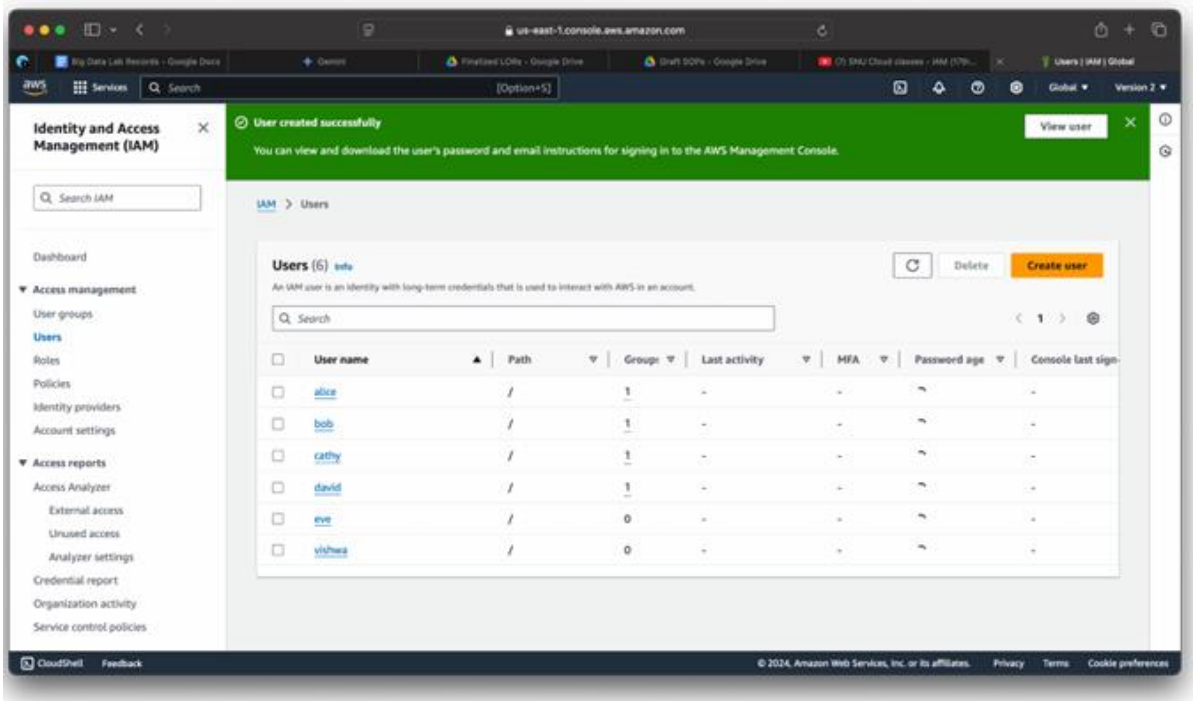
6. Create an alias name for your account.

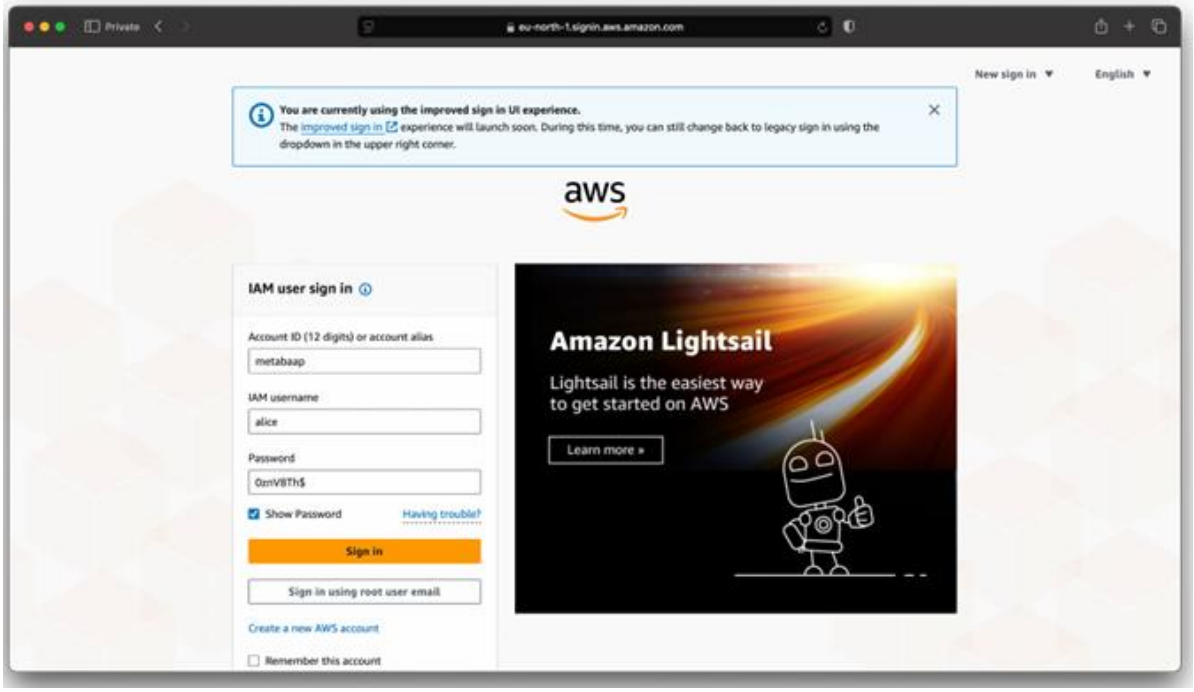
7. Use the alias URL to login to your account instead of the account ID.

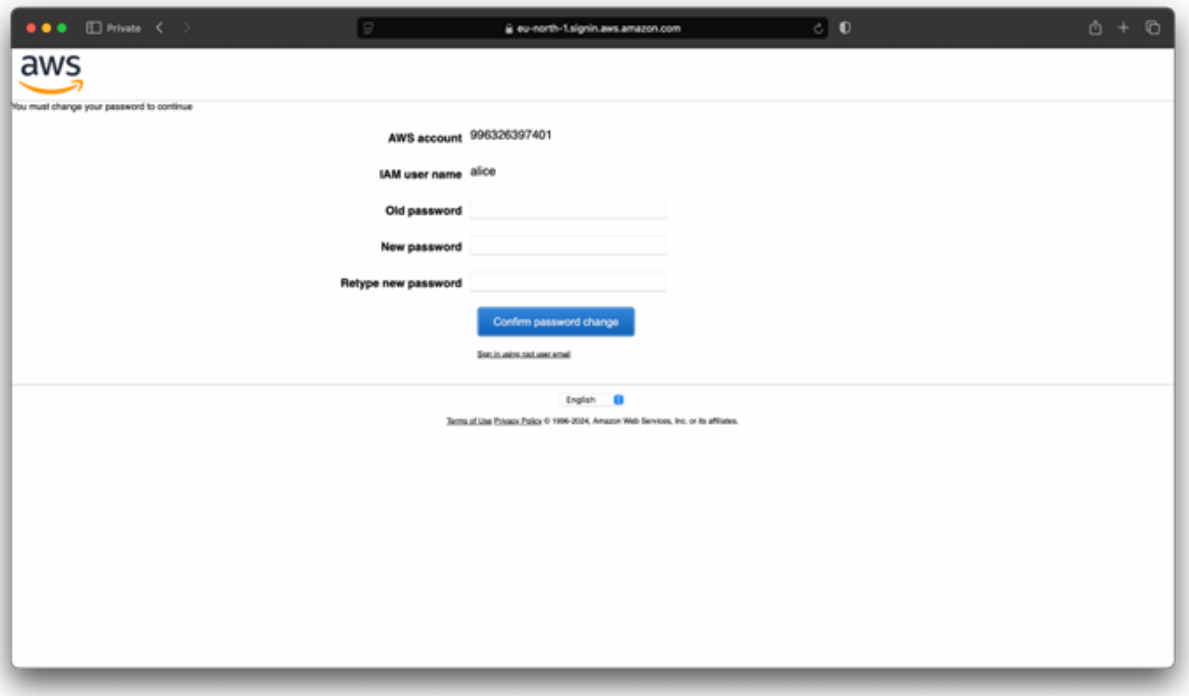
Output:

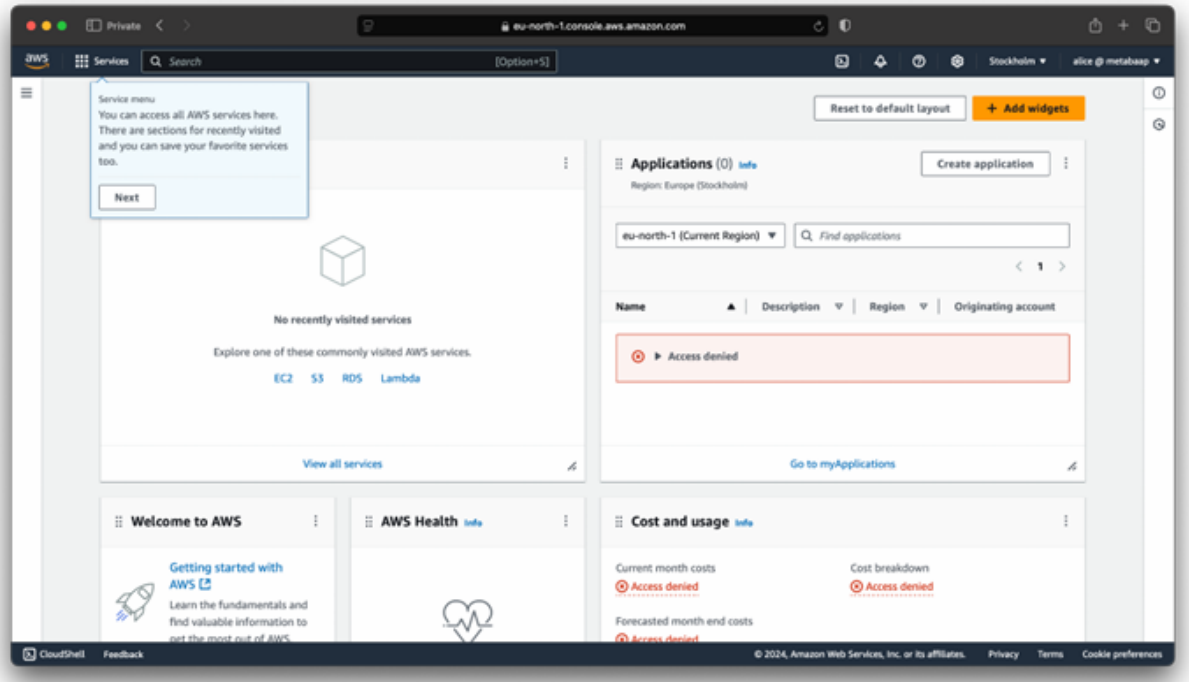

us-east-1.console.aws.amazon.com
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access Analyzer
External access
Unused access
Analyzer settings
Credential report
Organization activity
Service control policies
User group name
Enter a meaningful name to identify this group.
dns_admin
Maximum 128 characters. Use alphanumeric and '=,.@-_' characters.
Add users to the group - Optional (0) Info
An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.
Search
User name
Groups
Last activity
Creation time
No resources to display
Attach permissions policies - Optional (1/959) Info
You can attach up to 10 policies to this user group. All the users in this group will have permissions that are defined in the selected policies.
Filter by Type
route53fu
All types
1 match
Policy name
Type
Used as
Description
AmazonRoute53FullA...
AWS managed
None
Provides full access to all Amazon Rou...
Cancel
Create user group
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


us-east-1.console.aws.amazon.com
IAM > Users > Create user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password
Review and create
Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.
User details
User name
alice
Console password type
Autogenerated
Require password reset
Yes
Permissions summary
Name
Type
Used as
IAMUserChangePassword
AWS managed
Permissions policy
server_admin
Group
Permissions group
Tags - optional
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.
No tags associated with the resource.
Add new tag
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


us-east-1.console.aws.amazon.com
Identity and Access Management (IAM)
User created successfully
You can view and download the user's password and email instructions for signing in to the AWS Management Console.
View user
IAM > Users
Users (6) info
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.
Delete
Create user
User name
Path
Groups
Last activity
MFA
Password age
Console last sign-
alice
bob
cathy
david
eve
vishwa
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access Analyzer
External access
Unused access
Analyzer settings
Credential report
Organization activity
Service control policies
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


eu-north-1.signin.aws.amazon.com
New sign in
English
You are currently using the improved sign in UI experience.
The improved sign in experience will launch soon. During this time, you can still change back to legacy sign in using the dropdown in the upper right corner.
aws
IAM user sign in
Account ID (12 digits) or account alias
metabaap
IAM username
alice
Password
OznV8Th$
Show Password
Having trouble?
Sign in
Sign in using root user email
Create a new AWS account
Remember this account
Amazon Lightsail
Lightsail is the easiest way to get started on AWS
Learn more »





Result:

Successfully implemented IAM labs in AWS