Ex. No: 1	Linux Commands
16th July, 2024	

Aim:

To find and implement basic Linux Commands.

Programs and Outputs:

1. Display information about files in the current directory.

```
anantha@LAPTOP
$ ls
cat
```

2. Display the current working directory.

```
/cat$ pwd
/mnt/d/SAM/Sem-7/Big D/cat
```

3. Create a new directory.

```
anantha@LAPTOP
$ mkdir cat
anantha@LAPTOP
$ ls
cat
```

4. Navigate between different folders.

```
anantha@LAPTOP
$ cd cat
```

5. Remove empty directories from the directory lists.

```
anantha@LAPTOP-
$ ls
cat  dog
anantha@LAPTOP-
$ rmdir dog
anantha@LAPTOP-
$ ls
cat
```

6. a. Copy files from one directory to the same directory.

```
/cat$ cat pink.txt
Do pink cats even exist?
/cat$ cp orange.txt pink.txt
/cat$ cat pink.txt
Orange cats are not normal
```

6. b. Copy files from one directory to another directory.

```
/cow$ ls
orange.txt
/cat$ cp pink.txt /mnt/d/SAM/Sem-7/'Big D'/cow
/cow$ ls
orange.txt  pink.txt
```

7. a. Rename a filename to another name.

```
 mv orange.txt or.txt
anantha@LAPTOP-M
 ls
or.txt  pink.txt
```

7. b. Move a file from one directory to another.

```
cow$ ls
orange.txt  pink.txt
 cat$ mv or.txt /mnt/d/SAM/Sem-7/'Big D'/cow
cow$ ls
or.txt  orange.txt  pink.txt
```

8. a. Delete individual files from a directory

```
cow$ ls
or.txt  orange.txt  pink.txt
anantha@LAPTOP
cow$ rm or.txt
cow$ ls
orange.txt  pink.txt
```

8. b. Delete an entire directory which contains a few files.

```
anantha@LAPTOP
em-7/Big D$ ls
cat  cow
rm -r cow
/Sem-7/Big D$ ls
cat
```

9. Get basic information about the OS

```
em-7/Big D$ uname -a
rosoft-standard-WSL2 #1 SMP Fri Mar
29 23:14:13 UTC 2024 x86_64 x86_64 x
86_64 GNU/Linux
```

10. Find a file in the directory.

```
7/Big D$ find /mnt/d/SAM/Sem-7/'Big D'/
cat -name pink.txt
/mnt/d/SAM/Sem-7/Big D/cat/pink.txt
```

11. Create empty files

```
/cat$ touch orange.txt
```

12. Display file contents on terminal

```
 /cat$ cat orange.txt
 Orange cats are not normal
```

13. Clear terminal

```
Sem-7/Big D$ clear
```

14. Display the processes in terminal

```
/cow$ ps
    PID TTY          TIME CMD
    361 pts/0    00:00:00 bash
   7409 pts/0    00:00:00 ps
```

15. Access manual for all Linux commands

```
/cow$ man cd
```

```
cd(n)                          Tcl Built-In Commands                          cd(n)

NAME
       cd - Change working directory

SYNOPSIS
       cd ?dirName?

DESCRIPTION
       Change  the  current working directory to dirName, or to the home directory (as specified in the HOME environment variable) if dirName is not
       given.  Returns an empty string.  Note that the current working directory is a per-process resource; the cd command changes the  working  di-
       rectory for all interpreters and (in a threaded environment) all threads.
 Manual page cd(n) line 1 (press h for help or q to quit)
```

16. Search for a specific string in an output

```
cat$ grep cats pink.txt
Orange cats are not normal
```

17. Display active processes on the terminal

```
/cow$ top
top - 10:54:44 up 24 min,  1 user,  load average: 0.09, 0.08, 0.01
Tasks:  42 total,   1 running,  41 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.1 us,  0.0 sy,  0.0 ni, 99.8 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   7801.5 total,   6992.2 free,    569.8 used,    239.5 buff/cache
MiB Swap:   2048.0 total,   2048.0 free,      0.0 used.   7000.1 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
      1 root      20   0  167168  12376   8024 S   1.0   0.2   0:15.94 systemd
top - 10:54:50 up 24 min,  1 user,  load aver
Tasks:  34 total,   1 running,  33 sleeping,
```

18. Download files from the internet.

```
cat$ wget https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/
Resolving archive.ics.uci.edu (archive.ics.uci.edu)... 128.195.10.252
Connecting to archive.ics.uci.edu (archive.ics.uci.edu)|128.195.10.252|:443... connected.
```

19. Create or update passwords for existing users

```
/cat$ passwd anantha
Changing password for anantha.
Current password:
```

20. View the exact location of any tool/software installed

```
anantha@LAPTOP
which bash
/usr/bin/bash
```

21. Check the details of the file system

```
/cow$ df -h
Filesystem      Size  Used Avail Use% Mounted
on
none            3.9G  4.0K  3.9G   1% /mnt/wsl
drivers         379G  257G  122G  68% /usr/lib
/wsl/drivers
none            3.9G     0  3.9G   0% /usr/lib
/modules
none            3.9G     0  3.9G   0% /usr/lib
/modules/5.15.153.1-microsoft-standard-WSL2
/dev/sdc       1007G   17G  939G   2% /
```

22. Check the lines, word count, and characters in a file using different options

```
/cow$ wc pink.txt
 1  5 28 pink.txt
```

Result:

Successfully implemented and Tested Linux Commands.

Ex. No: 2	Install Hadoop
23rd July, 2024	

Aim:

To install Hadoop in Ubuntu.

Program:

Steps from this link : https://medium.com/@abhikdey06/apache-hadoop-3-3-6-installation-on-ubuntu-22-04-14516bceec85

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 3.4.0):

```
wget https://downloads.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz
```

- Extract the downloaded file:

```
tar -xzvf hadoop-3.4.0.tar.gz
```

3. Move Hadoop to the desired directory: `sudo mv hadoop-3.4.0 /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: `source ~/.bashrc`

5. Configure Hadoop:

- 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-MSOAV9AM:~$ 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-MSOAV9AM]
Starting resourcemanager
Starting nodemanagers
hadoop@LAPTOP-MSOAV9AM:~$ jps
1169 DataNode
1874 NodeManager
1427 SecondaryNameNode
2326 Jps
1003 NameNode
1724 ResourceManager
```

Result:

Successfully Installed Hadoop on Ubuntu Linux.

Ex. No: 3	MapReduce in Hadoop
30th July, 2024	

Aim:

To implement MapReduce for the word count problem in Hadoop.

Program:

```
import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
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<Object, Text, Text, IntWritable>{

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(Object key, Text value, Context context
                    ) throws IOException, InterruptedException {
      StringTokenizer itr = new StringTokenizer(value.toString());
      while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        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);
  }
}
```

Program (Python):

Mapper.py

```
#!/usr/bin/env python3

import sys


# Input comes from standard input (stdin)

for line in sys.stdin:

    # Remove leading and trailing whitespace

    line = line.strip()

    # Split the line into words

    words = line.split()

    # Increase counters

    for word in words:

        # Write the results to standard output (stdout)

        # What we output here will be the input for the Reducer

        # The tab-delimited format is <word, 1>

        print('%s\t%s' % (word, 1))
```

reducer.py

```
#!/usr/bin/env python3
```

```
import sys

current_word = None
current_count = 0
word = None

# Input comes from standard input (stdin)
for line in sys.stdin:
    # Remove leading and trailing whitespace
    line = line.strip()
    # Parse the input we got from mapper.py
    word, count = line.split('\t', 1)
    # Convert count (currently a string) to int
    try:
        count = int(count)
    except ValueError:
        # Count was not a number, so ignore/discard this line
        continue
    # This IF-switch only works because Hadoop sorts map output
    # by key (here: word) before it is passed to the reducer
    if current_word == word:
        current_count += count
    else:
        if current_word:
            # Write result to standard output (stdout)
            print('%s\t%s' % (current_word, current_count))
        current_count = count
        current_word = word

# Do not forget to output the last word if needed!
```

```
if current_word == word:
    print('%s\t%s' % (current_word, current_count))
```

commands:

```
chmod +x mapper.py
chmod +x reducer.py
```

```
$HADOOP_HOME/bin/hadoop jar $HADOOP_HOME/hadoop-streaming.jar \
-input myInputDirs \
-output myOutputDir \
-mapper /bin/cat \
-reducer /usr/bin/wc
```

```
mapred streaming -input /wordcount/input -output /wordcount/output4 -mapper
./mapper.py -reducer ./reducer.py
```

Output:

```
hadoop@LAPTOP-MSOAV9AM:~$ hdfs dfs -cat /output2/part-00000
1926    1
a       1
alexander       1
am      2
as      1
great   2
hello   1
i       3
in      1
king    1
my      1
name    1
suggests        1
the     1
was     1
```

Result:

Successfully implemented a MapReduce Function using Hadoop.

Ex. No: 4	MapReduce Problems
6th August, 2024	

Aim:

1. To Implement map reduce for NCDC weather dataset using Hadoop and find the max and min temperature.
2. Implement Apriori algorithm using map reduce paradigm.

Program 1:

TemperatureMapper.java

```
import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.FloatWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;


import java.io.IOException;


public class TemperatureMapper extends Mapper<LongWritable, Text, Text,
FloatWritable> {


    private final static FloatWritable temperature = new FloatWritable();

    private final Text tmaxKey = new Text("TMAX");

    private final Text tminKey = new Text("TMIN");


    @Override

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

        String line = value.toString();

        // Split the line on commas for CSV file

        String[] fields = line.split(",");
```

```
        // Skip header row and incomplete rows
        if (fields[0].equals("STATION") || fields.length < 6) {
            return;
        }

        // Safely parse TMAX and TMIN values
        try {
            String tmaxStr = fields[4].trim();
                      tmaxStr = tmaxStr.replace('"',' ');
            String tminStr = fields[5].trim();
            tminStr = tminStr.replace('"',' ');
            // System.out.println(tmaxStr);

            if (!tmaxStr.isEmpty() && !tminStr.isEmpty()) {
                float tmax = Float.parseFloat(tmaxStr);
                float tmin = Float.parseFloat(tminStr);

                temperature.set(tmax);
                context.write(tmaxKey, temperature);
                temperature.set(tmin);
                context.write(tminKey, temperature);
            }
        } catch (NumberFormatException e) {
            // Log parsing errors for debug purposes
            System.err.println("Error parsing temperature: " +
e.getMessage());
        }
    }
}
```

TemperatureReducer.java

```
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.FloatWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

import java.io.IOException;

public class TemperatureReducer extends Reducer<Text, FloatWritable,
Text, FloatWritable> {

    @Override
    protected void reduce(Text key, Iterable<FloatWritable> values,
Context context) throws IOException, InterruptedException {
        Float extremeValue;

        if (key.toString().equals("TMAX")) {
            extremeValue = Float.MIN_VALUE;
            for (FloatWritable value : values) {
                extremeValue = Math.max(extremeValue, value.get());
            }
            System.out.println("Maximum Temperature: " + extremeValue);
        } else {  // TMIN
            extremeValue = Float.MAX_VALUE;
            for (FloatWritable value : values) {
                extremeValue = Math.min(extremeValue, value.get());
            }
            System.out.println("Minimum Temperature: " + extremeValue);
        }

        // Emit the result
        context.write(key, new FloatWritable(extremeValue));
```

```
    }
}
```

TemperatureDriver.java

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.FloatWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class TemperatureDriver {
    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "Max and Min Temperature");

        job.setJarByClass(TemperatureDriver.class);
        job.setMapperClass(TemperatureMapper.class);
        job.setReducerClass(TemperatureReducer.class);

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(FloatWritable.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 python
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")
```

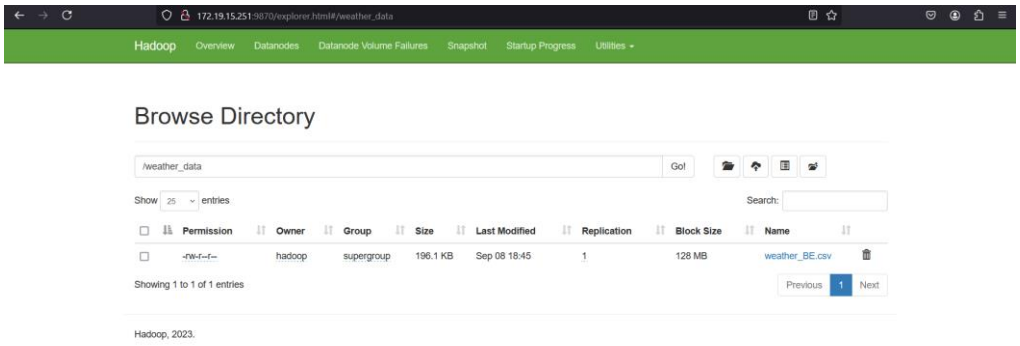
Reducer.py

```
#!/usr/bin/env python
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}")
```

Outputs :

```
hadoop@LAPTOP-MSOAV9AM:~/hadoop/bin$ hdfs dfs -cat /weather_data/output/part-00000
Min Temperature: -11.17
Max Temperature: 29.67
```



Result:

Successfully implemented a Weather Analysis and Apriori Algorithm using MapReduce

Ex. No: 5	Apache Spark
13th August, 2024	

Aim:

To install Spark and PySpark and run the WordCount Program and Distribution of Ratings in Movielens Dataset.

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 Program

```
import findspark

findspark.init()


from pyspark.sql import SparkSession

from pyspark.conf import SparkConf


spark =
SparkSession.builder.master("local[*]").appName("WordCount").getOrCreate(
)
```

```
sc = spark.sparkContext

text = "./words.txt"

text_read = sc.textFile(text)

text_read.take(20)

words = text_read.flatMap(lambda x: x.split())

words.take(15)

words = words.map(lambda word: word.rstrip('.'))

words.take(15)

word_count = words.map(lambda word: (word, 1))
word_count.take(20)

word_count_red = word_count.reduceByKey(lambda x, y: (x+y)).sortByKey()

word_count_red = word_count_red.map(lambda x: (x[1], x[0]))

word_count_red.take(15)
```

MovieLens Dataset

```
import findspark
findspark.init()
```

```
from pyspark.sql import SparkSession

from pyspark.conf import SparkConf


spark =
SparkSession.builder.master("local[*]").appName("MovieLens").getOrCreate(
)


sc = spark.sparkContext


text = "./u.data"


text_read = spark.read.csv(text)


import pyspark.pandas as ps


data = ps.read_csv(text, sep="\t", names = ["userid", "movieid",
"rating", "idid"])


data


len(data.groupby(['userid']).describe()['rating'])
```

Output:

```
[(1, 'college'),
 (1, 'i'),
 (1, 'in'),
 (1, 'is'),
 (1, 'my'),
 (2, 'nadar'),
 (2, 'shiv'),
 (1, 'study'),
 (1, 'university')]
```

	userid	movieid	rating	idid
0	196	242	3	881250949
1	186	302	3	891717742
2	22	377	1	878887116
3	244	51	2	880606923
4	166	346	1	886397596
5	298	474	4	884182806
6	115	265	2	881171488
7	253	465	5	891628467
8	305	451	3	886324817
9	6	86	3	883603013
10	62	257	2	879372434
11	286	1014	5	879781125
12	200	222	5	876042340
13	210	40	3	891035994
14	224	29	3	888104457
15	303	785	3	879485318
16	122	387	5	879270459

Result:

Successfully Installed Spark and PySpark, and implemented the WordCount Program and Distribution of Movie Reviews using them.

Ex. No: 6	PySpark 2
20th August, 2024	

Aim:

(a) To use the “friends_test” dataset. Col1 is ID, Col2 is name, Col 3 is Age, Col 4 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.

(b) 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 temperature for every StationID.

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

Program:

```
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)
```

RDD approach for average number of friends

```
age_friends_rdd = rdd_split.map(lambda row: (int(row[2]), (int(row[3]),
1)))
sum_count_rdd = age_friends_rdd.reduceByKey(lambda a, b: (a[0] + b[0],
a[1] + b[1]))
avg_friends_by_age = sum_count_rdd.mapValues(lambda x: x[0] / x[1])
sorted_avg_friends_by_age = avg_friends_by_age.sortByKey()
for age, avg_friends in sorted_avg_friends_by_age.collect():
      print(f"Age: {age}, Average Number of Friends: {avg_friends:.2f}")
age_friends_rdd = rdd_split.map(lambda row: (int(row[2]), int(row[3])))
max_friends_by_age = age_friends_rdd.reduceByKey(lambda a, b: a if a > b
else b)
min_friends_by_age = age_friends_rdd.reduceByKey(lambda a, b: a if a < b
else b)
max_min_friends_by_age = max_friends_by_age.join(min_friends_by_age)
sorted_max_min_friends_by_age = max_min_friends_by_age.sortByKey()
for age, (max_friends, min_friends) in
sorted_max_min_friends_by_age.collect():
      print(f"Age: {age}, Max Friends: {max_friends}, Min Friends:
{min_friends}")
sc.stop()
```

Temp dataset

```
conf = SparkConf().setAppName("temp_dataset").setMaster("local[*]")
sc = SparkContext(conf=conf)
rdd = sc.textFile("temp.csv")
rdd.first()
rdd_header = rdd.first()
rdd_filter = rdd.filter(lambda row: row != rdd_header)
rdd_data = rdd_filter.map(lambda row:row.split(","))
def rdd_display(x,threshold=5):
      count = 0
      for i in x.collect():
            print(i)
```

```
            count+=1
            if(count>threshold):
                  break
rdd_display(rdd_data)
rdd_TMIN_filter = rdd_data.filter(lambda row:row[2]=="TMIN")
rdd_display(rdd_TMIN_filter)


rdd_min_overall = rdd_TMIN_filter.map(lambda x:int(x[3])).reduce(lambda
a,b:a if a
print("Minimum temperature overall",rdd_min_overall)
rdd_min_itemID = rdd_TMIN_filter.map(lambda
x:(x[0],int(x[3]))).reduceByKey(lambda
print("minimum temperature by itemID")
rdd_display(rdd_min_itemID)
rdd_min_stationID = rdd_TMIN_filter.map(lambda
x:(x[1],int(x[3]))).reduceByKey(lamb
print("minimum temperature by StationID")
rdd_display(rdd_min_stationID,10)
rdd_TMAX_filter = rdd_data.filter(lambda row:row[2]=="TMAX")
rdd_display(rdd_TMAX_filter)
rdd_max_overall = rdd_TMAX_filter.map(lambda x:int(x[3])).reduce(lambda
a,b:a if a
print("Maximum temperature overall",rdd_max_overall)
rdd_max_itemID = rdd_TMAX_filter.map(lambda
x:(x[0],int(x[3]))).reduceByKey(lambda
print("maximum temperature by itemID")
rdd_display(rdd_max_itemID)
rdd_max_stationID = rdd_TMAX_filter.map(lambda
x:(x[1],int(x[3]))).reduceByKey(lamb
print("maximum temperature by StationID")
rdd_display(rdd_max_stationID,10)
sc.stop()
```

Output:

```
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
```

```
Age: 18, Max Friends: 499, Min Friends: 24
Age: 19, Max Friends: 404, Min Friends: 5
Age: 20, Max Friends: 384, Min Friends: 1
Age: 21, Max Friends: 491, Min Friends: 89
Age: 22, Max Friends: 478, Min Friends: 6
Age: 23, Max Friends: 392, Min Friends: 65
Age: 24, Max Friends: 492, Min Friends: 49
Age: 25, Max Friends: 485, Min Friends: 1
Age: 26, Max Friends: 492, Min Friends: 2
Age: 27, Max Friends: 471, Min Friends: 53
Age: 28, Max Friends: 378, Min Friends: 32
Age: 29, Max Friends: 367, Min Friends: 11
Age: 30, Max Friends: 487, Min Friends: 17
Age: 31, Max Friends: 481, Min Friends: 15
Age: 32, Max Friends: 412, Min Friends: 24
Age: 33, Max Friends: 471, Min Friends: 74
Age: 34, Max Friends: 423, Min Friends: 48
Age: 35, Max Friends: 428, Min Friends: 13
Age: 36, Max Friends: 493, Min Friends: 49
Age: 37, Max Friends: 471, Min Friends: 46
Age: 38, Max Friends: 459, Min Friends: 2
Age: 39, Max Friends: 275, Min Friends: 68
Age: 40, Max Friends: 465, Min Friends: 7
Age: 41, Max Friends: 397, Min Friends: 62
Age: 42, Max Friends: 467, Min Friends: 95
Age: 43, Max Friends: 428, Min Friends: 48
Age: 44, Max Friends: 499, Min Friends: 61
Age: 45, Max Friends: 497, Min Friends: 54
Age: 46, Max Friends: 462, Min Friends: 63
Age: 47, Max Friends: 488, Min Friends: 4
Age: 48, Max Friends: 439, Min Friends: 57
Age: 49, Max Friends: 476, Min Friends: 17
Age: 50, Max Friends: 436, Min Friends: 119
Age: 51, Max Friends: 493, Min Friends: 81
Age: 52, Max Friends: 487, Min Friends: 77
Age: 53, Max Friends: 451, Min Friends: 86
Age: 54, Max Friends: 462, Min Friends: 7
Age: 55, Max Friends: 474, Min Friends: 57
Age: 56, Max Friends: 444, Min Friends: 15
Age: 57, Max Friends: 465, Min Friends: 8
Age: 58, Max Friends: 348, Min Friends: 6
Age: 59, Max Friends: 439, Min Friends: 14
Age: 60, Max Friends: 324, Min Friends: 2
Age: 61, Max Friends: 469, Min Friends: 2
Age: 62, Max Friends: 496, Min Friends: 12
Age: 63, Max Friends: 469, Min Friends: 342
Age: 64, Max Friends: 499, Min Friends: 65
Age: 65, Max Friends: 443, Min Friends: 101
Age: 66, Max Friends: 496, Min Friends: 41
Age: 67, Max Friends: 445, Min Friends: 35
Age: 68, Max Friends: 490, Min Friends: 21
Age: 69, Max Friends: 491, Min Friends: 9
```

-

```
['ITE00100554', '18000101', 'TMAX', '-75']
['ITE00100554', '18000101', 'TMIN', '-148']
['GM000010962', '18000101', 'PRCP', '0']
['EZE00100082', '18000101', 'TMAX', '-86']
['EZE00100082', '18000101', 'TMIN', '-135']
['ITE00100554', '18000102', 'TMAX', '-60']
```

-

```
['ITE00100554', '18000101', 'TMIN', '-148']
['EZE00100082', '18000101', 'TMIN', '-135']
['ITE00100554', '18000102', 'TMIN', '-125']
['EZE00100082', '18000102', 'TMIN', '-130']
['ITE00100554', '18000103', 'TMIN', '-46']
['EZE00100082', '18000103', 'TMIN', '-73']
```

Minimum temperature overall -148

```
minimum temperature by itemID
('ITE00100554', -148)
('EZE00100082', -135)
```

```
minimum temperature by StationID
('18000102', -130)
('18000104', -74)
('18000106', -57)
('18000110', -75)
('18000111', -62)
('18000112', -60)
('18000114', -35)
('18000115', -23)
('18000116', -37)
('18000117', -35)
('18000118', 9)
```

```
['ITE00100554', '18000101', 'TMAX', '-75']
['EZE00100082', '18000101', 'TMAX', '-86']
['ITE00100554', '18000102', 'TMAX', '-60']
['EZE00100082', '18000102', 'TMAX', '-44']
['ITE00100554', '18000103', 'TMAX', '-23']
['EZE00100082', '18000103', 'TMAX', '-10']
```

```
maximum temperature by StationID
('18000102', -44)
('18000104', 0)
('18000106', 13)
('18000110', 46)
('18000111', 66)
('18000112', 41)
('18000114', 41)
('18000115', 54)
('18000116', 56)
('18000117', 84)
('18000118', 59)
```

Result:

Successfully understood map value functions and utilised them for various tasks

Ex. No: 7	Hadoop and Docker
27th August, 2024	

Aim:

To install hadoop in docker containers

Output:


```
SUBCOMMAND may print help when invoked w/o parameters or with -h.
hadoop@85dfacaf27ee:~/ex_docker$ hdfs dfs -cat /user/hadoop/output/part-r-00000
I       1
a       1
above   1
are     1
diamond 1
high    1
how     1
in      1
like    1
little  2
sky     1
so      1
star    2
the     2
twinkle 4
up      1
what    1
wonder  1
workd   1
you     1
hadoop@85dfacaf27ee:~/ex_docker$ s
```

Result:

Successfully installed hadoop in docker containers

Ex. No: 8	Public and Private Keys
3rd September, 2024	

Aim:

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

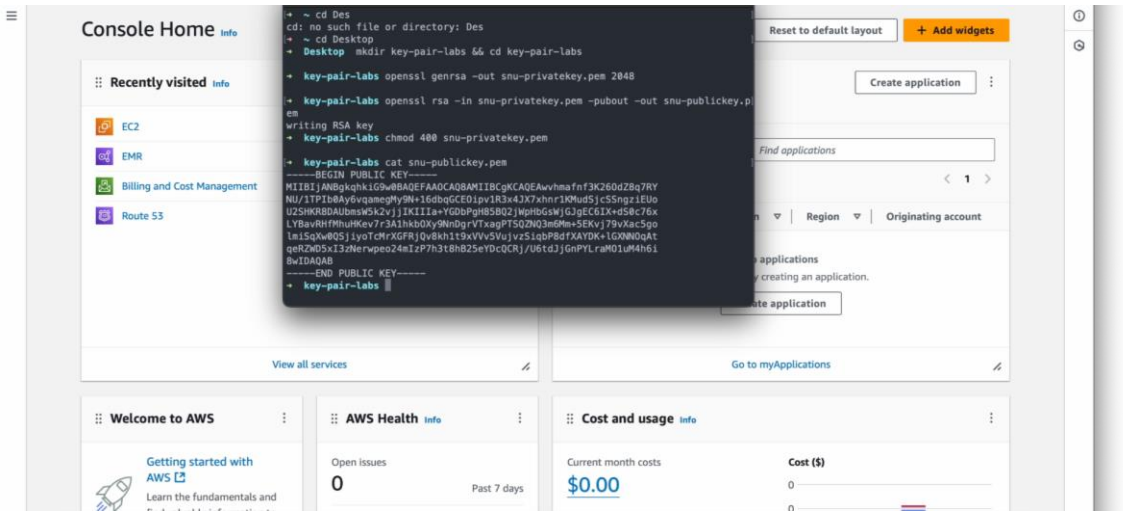
Program:

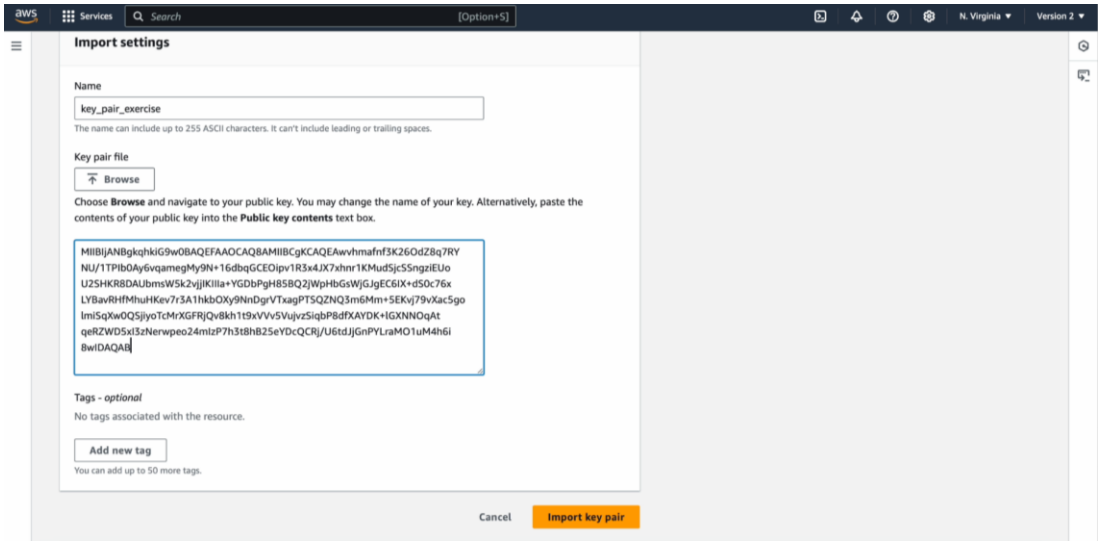
1. **Generate Private and Public Key Pairs**
    - **Linux/Mac:**
        - Create a new directory for key pairs: mkdir key-pair-labs && cd key-pair-labs
        - Generate a private key: openssl genrsa -out snu-privatekey.pem 2048
        - Generate a public key from the private key: openssl rsa -in 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@<public_ip_address>
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.

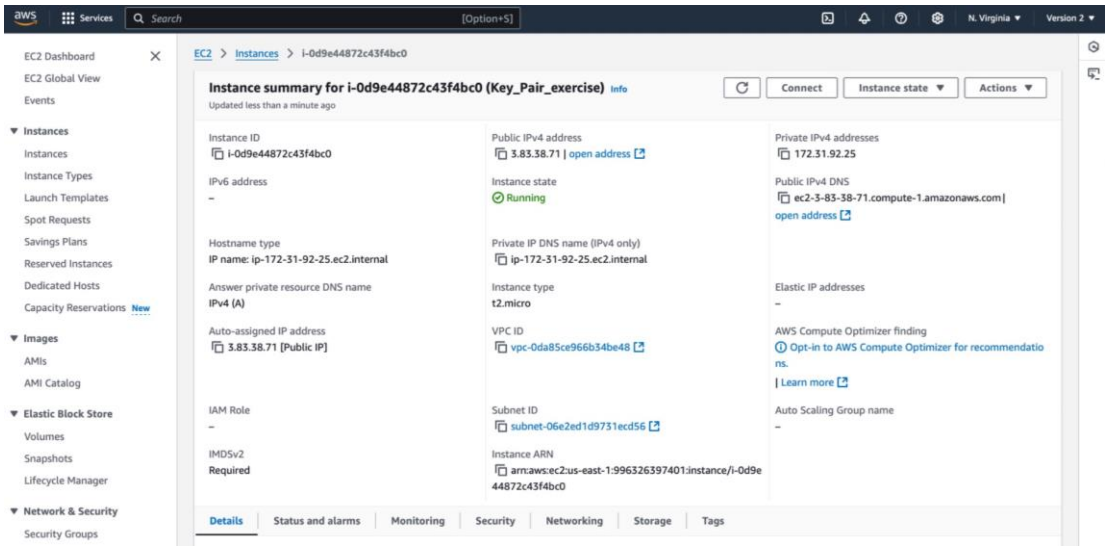
- You should be able to ping the instance successfully.

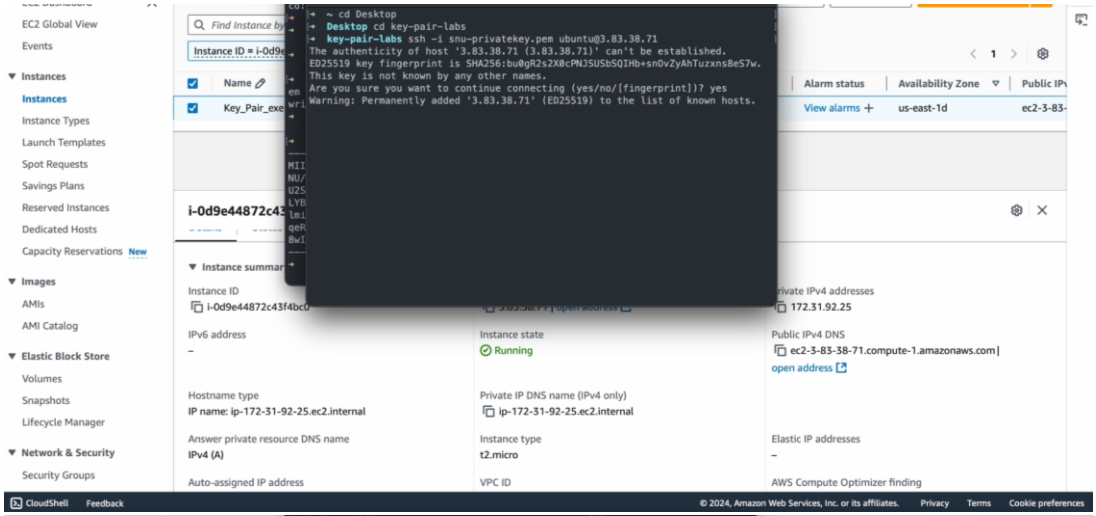
6. **Edit Default NACL:**
   - Open the Network Access Control List (NACL) settings for the instance.
   - Edit the default NACL to block ICMP traffic.
   - Add an egress rule to deny ICMP traffic:
     - Type: Custom TCP Rule
     - Protocol: ICMP
     - Port Range: All
     - Destination: 0.0.0.0/0
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.

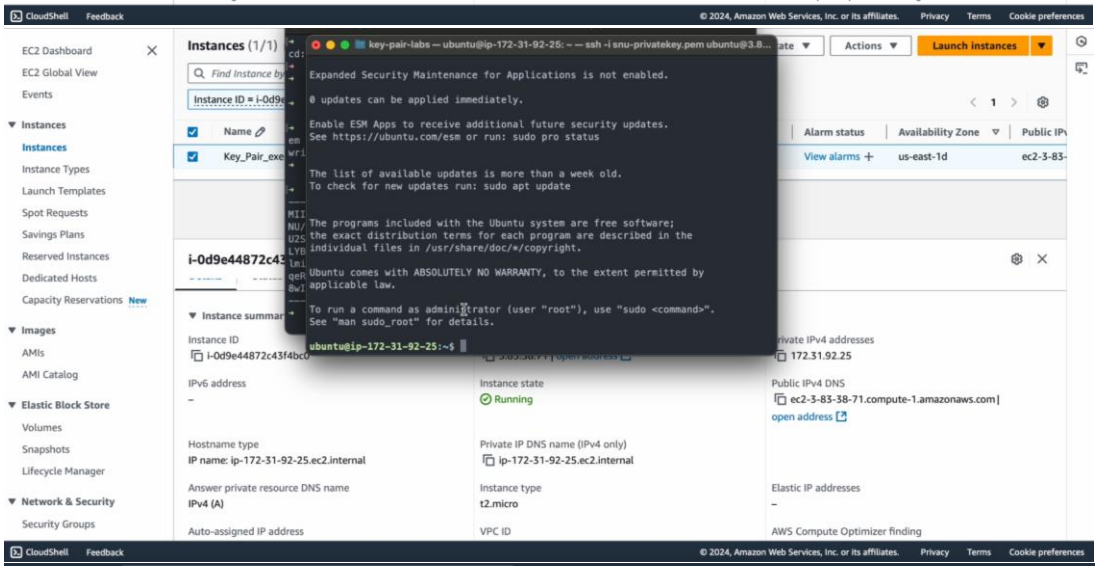
Output:

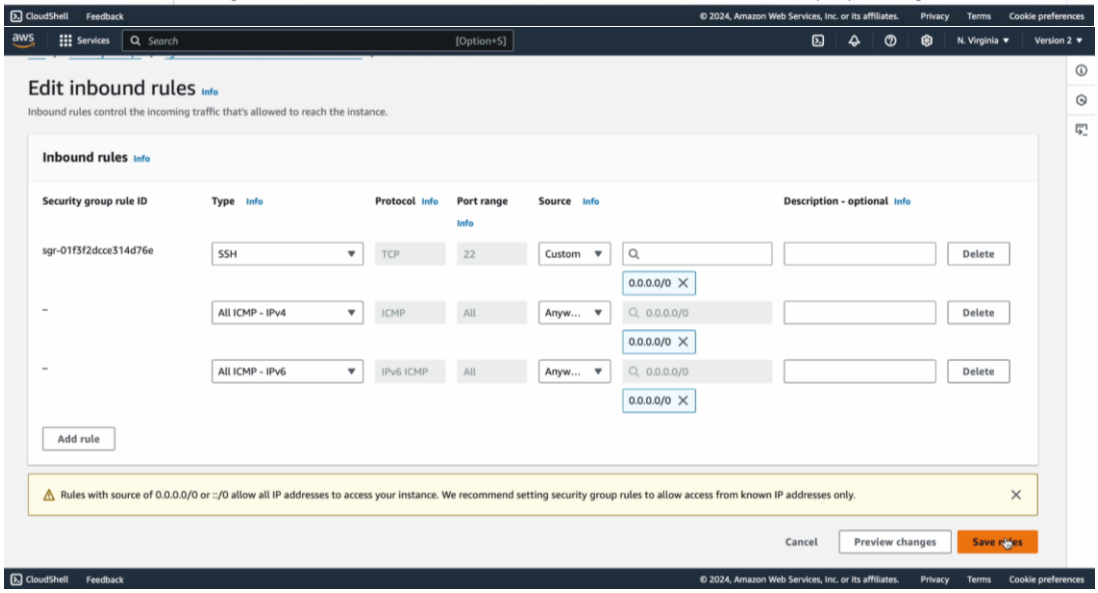


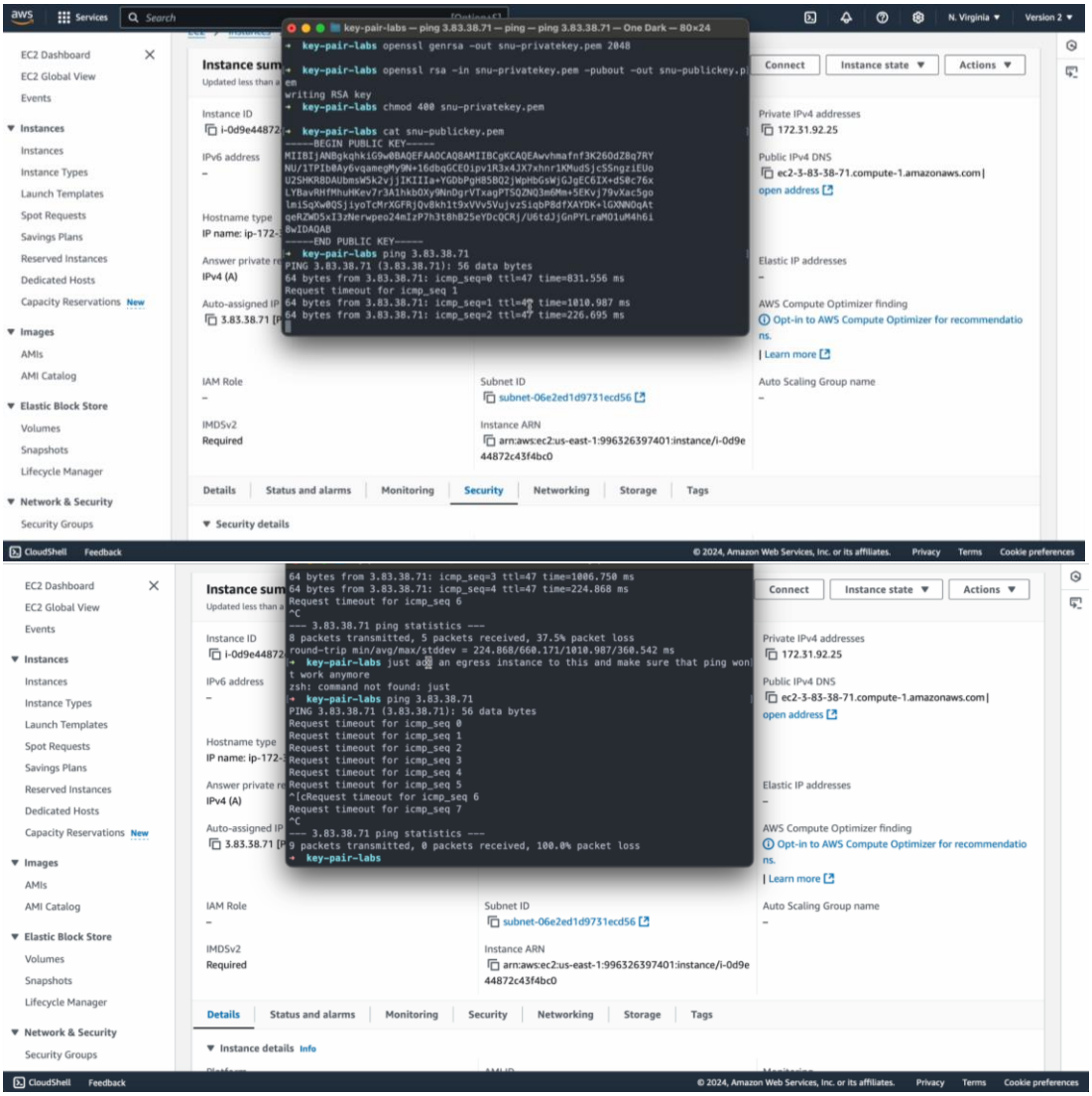

Services
Search
[Option+S]
N. Virginia
Version 2
Import settings
Name
key_pair_exercise
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.
Key pair file
Browse
Choose Browse and navigate to your public key. You may change the name of your key. Alternatively, paste the contents of your public key into the Public key contents text box.
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwvhmafnf3K26OdZ8q7RY
NU/1TPIb0Ay6vqamegMy9N+16dbqGCEOipv1R3x4JX7xhnr1KMud5jcSSngziEUo
U2SHKR8DAUbmsW5k2vjjIKIIa+YGDbPgH85BQ2jWpHbGsWjGJgEC6IX+dS0c76x
LYBavRHfMhuHKev7r3A1hkbOXy9NnDgrVTxagPTSQZNQ3m6Mm+5EKvj79vXac5go
lmiSqXw0QSjiyoTcMrXGFRjQv8kh1t9xVVv5VujvzSiqbP8dfXAYDK+IGXNNOqAt
qeRZWD5xI3zNerwpeo24mIzP7h3t8hB25eYDcQCRj/U6tdJjGnPYLraMO1uM4h6i
8wIDAQAB
Tags - optional
No tags associated with the resource.
Add new tag
You can add up to 50 more tags.
Cancel
Import key pair


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-0d9e44872c43f4bc0
Instance summary for i-0d9e44872c43f4bc0 (Key_Pair_exercise) Info
Updated less than a minute ago
Connect
Instance state
Actions
Instance ID
i-0d9e44872c43f4bc0
IPv6 address
–
Hostname type
IP name: ip-172-31-92-25.ec2.internal
Answer private resource DNS name
IPv4 (A)
Auto-assigned IP address
3.83.38.71 [Public IP]
IAM Role
–
IMDSv2
Required
Public IPv4 address
3.83.38.71 | open address
Instance state
Running
Private IP DNS name (IPv4 only)
ip-172-31-92-25.ec2.internal
Instance type
t2.micro
VPC ID
vpc-0da85ce966b34be48
Subnet ID
subnet-06e2ed1d9731ecd56
Instance ARN
arn:aws:ec2:us-east-1:996326397401:instance/i-0d9e44872c43f4bc0
Private IPv4 addresses
172.31.92.25
Public IPv4 DNS
ec2-3-83-38-71.compute-1.amazonaws.com | open address
Elastic IP addresses
–
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations.
| Learn more
Auto Scaling Group name
–
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags


~ cd Desktop
Desktop cd key-pair-labs
key-pair-labs ssh -i snu-privatekey.pem ubuntu@3.83.38.71
The authenticity of host '3.83.38.71 (3.83.38.71)' can't be established.
ED25519 key fingerprint is SHA256:bu0gR2s2X0cPNJSUSbSQIHb+snOvZyAhTuzxnsBeS7w.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.83.38.71' (ED25519) to the list of known hosts.
EC2 Global View
Events
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
Alarm status
Availability Zone
Public IPv
View alarms
us-east-1d
Instance ID
IPv6 address
Instance state
Running
Public IPv4 DNS
ec2-3-83-38-71.compute-1.amazonaws.com
open address
Hostname type
IP name: ip-172-31-92-25.ec2.internal
Private IP DNS name (IPv4 only)
ip-172-31-92-25.ec2.internal
Answer private resource DNS name
IPv4 (A)
Instance type
t2.micro
Elastic IP addresses
Auto-assigned IP address
VPC ID
AWS Compute Optimizer finding
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Instances (1/1)
Actions
Launch instances
key-pair-labs — ubuntu@ip-172-31-92-25: ~ — ssh -i snu-privatekey.pem ubuntu@3.8...
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
ubuntu@ip-172-31-92-25:~$
EC2 Dashboard
EC2 Global View
Events
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
Alarm status
Availability Zone
Public IPv
View alarms
us-east-1d
Instance ID
IPv6 address
Instance state
Running
Public IPv4 DNS
ec2-3-83-38-71.compute-1.amazonaws.com
open address
Hostname type
IP name: ip-172-31-92-25.ec2.internal
Private IP DNS name (IPv4 only)
ip-172-31-92-25.ec2.internal
Answer private resource DNS name
IPv4 (A)
Instance type
t2.micro
Elastic IP addresses
Auto-assigned IP address
VPC ID
AWS Compute Optimizer finding
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


aws
Services
Search
[Option+5]
N. Virginia
Version 2
Edit inbound rules Info
Inbound rules control the incoming traffic that's allowed to reach the instance.
Inbound rules Info
Security group rule ID
Type Info
Protocol Info
Port range Info
Source Info
Description - optional Info
sgr-01f3f2dcce314d76e
SSH
TCP
22
Custom
0.0.0.0/0
Delete
All ICMP - IPv4
ICMP
All
Anyw...
0.0.0.0/0
Delete
All ICMP - IPv6
IPv6 ICMP
All
Anyw...
0.0.0.0/0
Delete
Add rule
Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.
Cancel
Preview changes
Save rules
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



Result:

Successfully Created Public and Private Keys and Utilised it to secure EC2 instances.

Ex. No: 9	EC2
10th September, 2024	

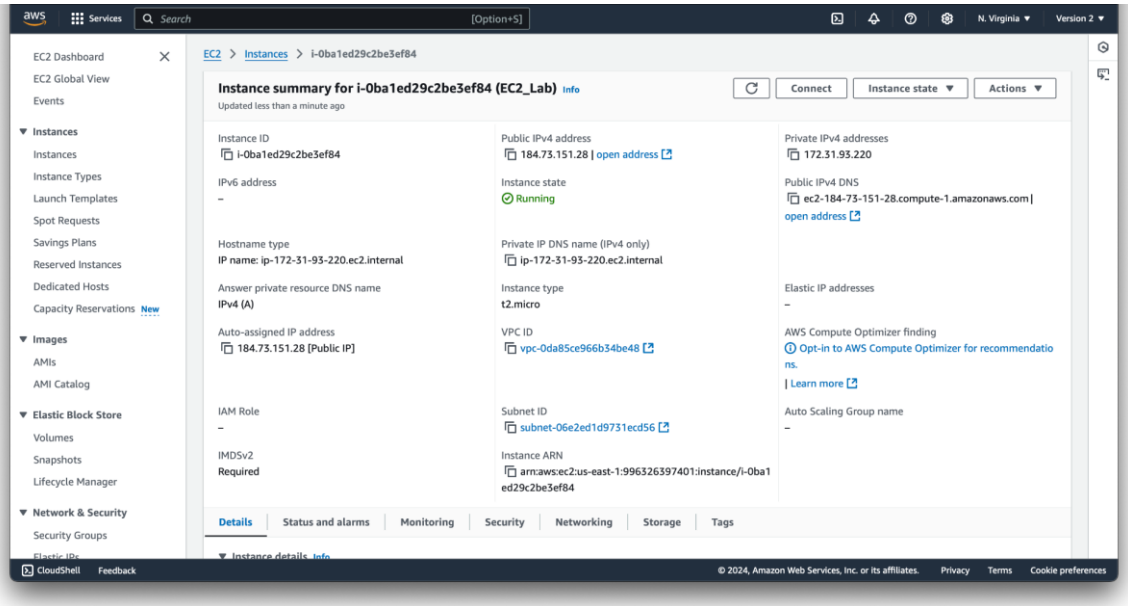
Aim:

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:

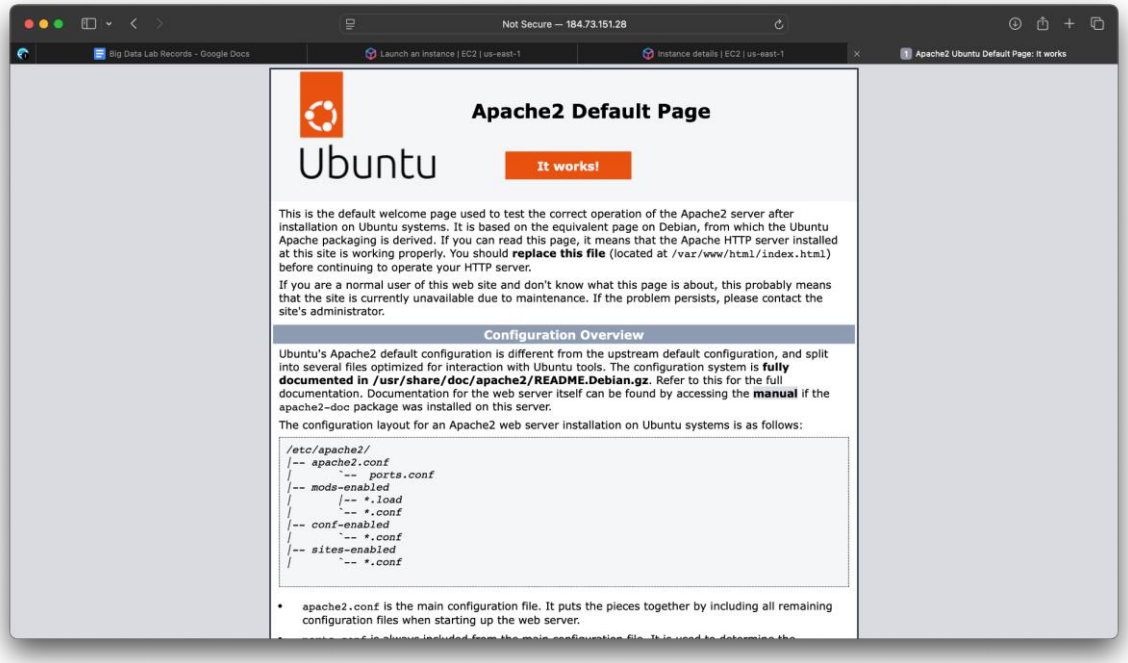



```
➜  Desktop mkdir key-pair-labs && cd key-pair-labs

➜  key-pair-labs openssl genrsa -out snu-privatekey.pem 2048

➜  key-pair-labs openssl rsa -in snu-privatekey.pem -pubout -out snu-publickey.pem
writing RSA key
➜  key-pair-labs chmod 400 snu-privatekey.pem

➜  key-pair-labs cat snu-publickey.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAn5V9u25EVk5nNfsOqMoi
shWlxxwjP9A1MGYOHIQjQdInUdk1t4U1NR57m+9YlRwaYyYojiMen1IJcVdHGzTl
R3BbGmPSKhR7ilawYKwb3IoCP3s1NPbg/Mq88PJzEdKPPjUP9xisnncNKlNkoTwo
tnpXEtWUXdJ/8pWvwaGVQsUWHjN8YHbL3/oRoVndhWps9BGn4cqRV1DMpQ5QwkkI
+iApDy4sj3xg24fhaOpnbGzDYJSikarqLvT4TVISuNqSuxwj0qoMGZhkmold8Tcy
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:O447y6SkAs3URJwfWj4XN8W9P1GDvUeMPMAO4QdgfnY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

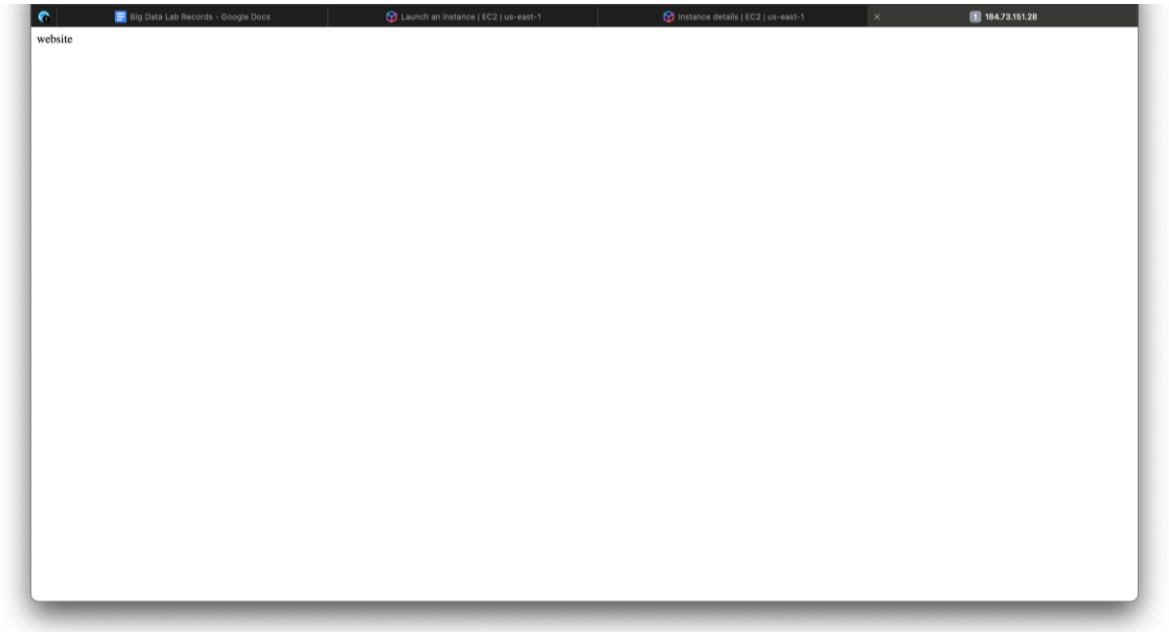
```
ubuntu@ip-172-31-93-220:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [12
6 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [
126 kB]
Get:4 http://security.ubuntu.com/ubuntu 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
```

```
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]
```

```
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)
```




```
21 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Sun Oct 27 07:57:17 2024 from 49.43.251.85
ubuntu@ip-172-31-93-220:~$ cd /var/www/html/
ubuntu@ip-172-31-93-220:/var/www/html$ suo chmod 777 index.html
Command 'suo' not found, did you mean:
  command 'su' from deb util-linux (2.39.3-9ubuntu6.1)
  command 'sur' from deb subtle (0.11.3224-xi-2.2build5)
  command 'sup' from deb sup (20100519-3)
  command 'su1' from deb hxtools (20231101-1)
  command 'sudo' from deb sudo (1.9.14p2-1ubuntu1)
  command 'sudo' from deb sudo-ldap (1.9.14p2-1ubuntu1)
  command 'sumo' from deb sumo (1.18.0+dfsg-3build2)
  command 'zuo' from deb zuo (1.9-1)
  command 'sum' from deb coreutils (9.4-2ubuntu2)
Try: sudo apt install <deb name>
ubuntu@ip-172-31-93-220:/var/www/html$ sudo chmod 777 index.html
ubuntu@ip-172-31-93-220:/var/www/html$ echo "website" > index.html
ubuntu@ip-172-31-93-220:/var/www/html$
```



Result:

Successfully deployed an EC2 instance and tested using Apache Webservices.

Ex. No: 10	Route53
1st October, 2024	

Aim:

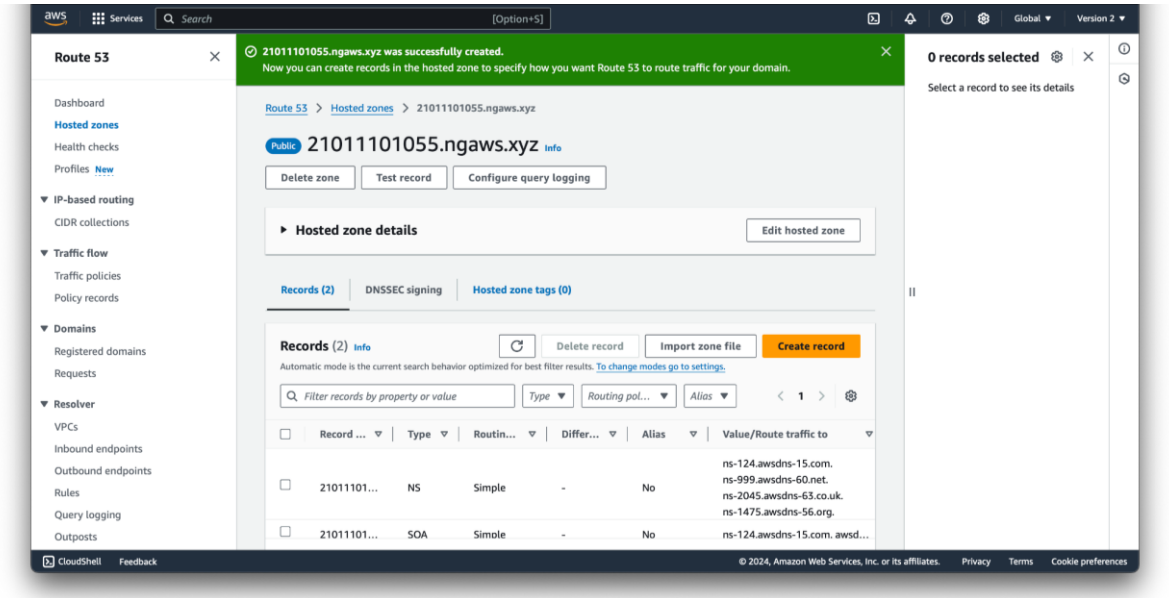
To set up a web server on an AWS EC2 instance and configure a domain name using GoDaddy and Route 53

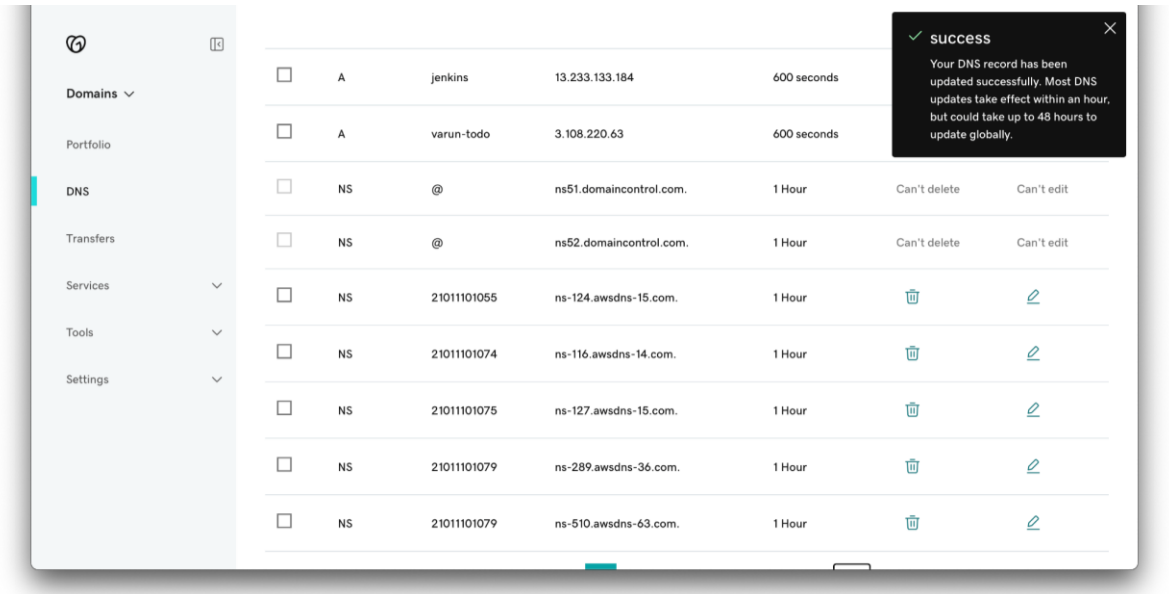
Program:

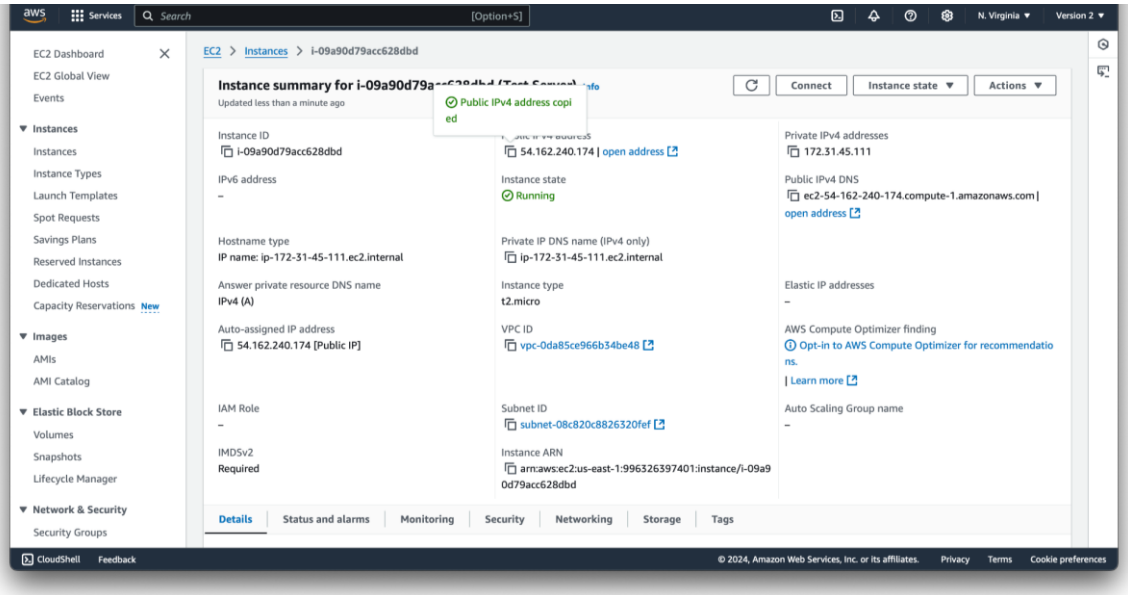
1. **Login to GoDaddy and Create a subdomain:**
    - Use your 12-digit registration number.
    - Example: 21100101001.ngaws.xyz
2. **Create a Hosted Zone in Route 53:**
    - Navigate to the Route 53 service in the AWS console.
    - Create a new Hosted Zone for your subdomain.
3. **Get the Name Server 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

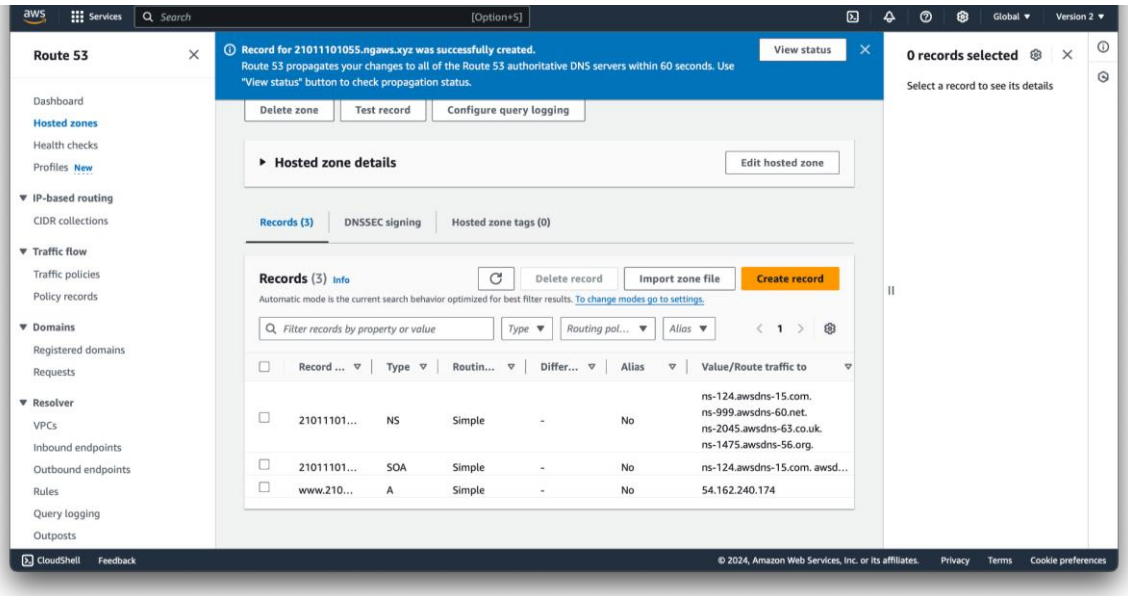
8. Create an A record in Route 53:
    - Navigate to your Hosted Zone in Route 53.
    - Create a new A record with the following details:
        - Name: @ (for the root domain)
        - Value: The public IP address of your EC2 instance
        - TTL: 3600 (1 hour)

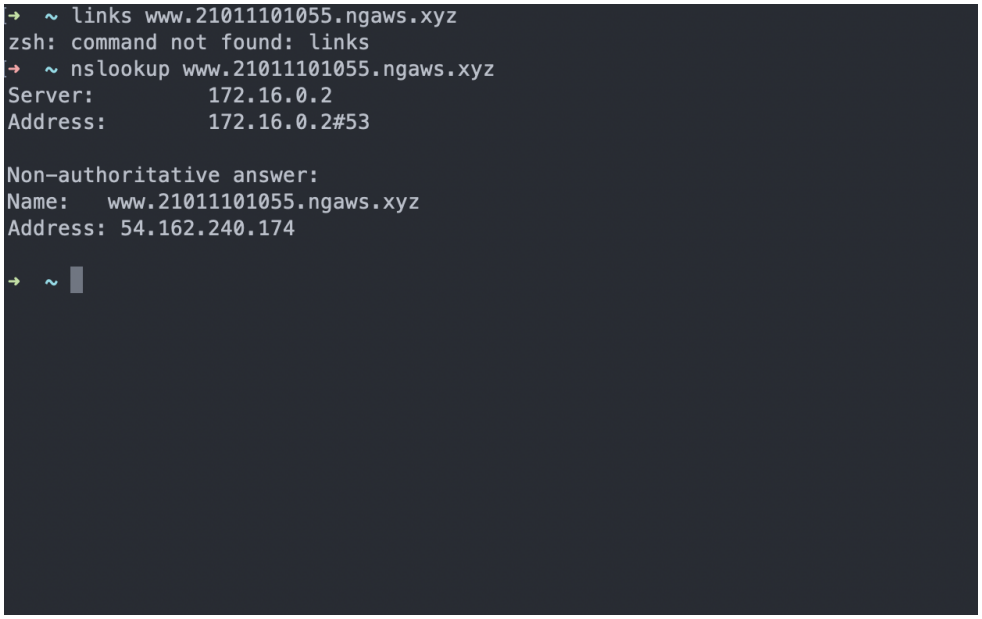
Output:






aws
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-09a90d79acc628dbd
Updated less than a minute ago
Public IPv4 address copied
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 recommendations.
| 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-09a90d79acc628dbd
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


aws
Services
Search
[Option+S]
Global
Version 2
Route 53
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
Record for 21011101055.ngaws.xyz was successfully created.
Route 53 propagates your changes to all of the Route 53 authoritative DNS servers within 60 seconds. Use "View status" button to check propagation status.
View status
Delete zone
Test record
Configure query logging
Hosted zone details
Edit hosted zone
Records (3)
DNSSEC signing
Hosted zone tags (0)
Records (3) Info
Delete record
Import zone file
Create record
Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.
Filter records by property or value
Type
Routing pol...
Alias
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...
www.210...
A
Simple
-
No
54.162.240.174
0 records selected
Select a record to see its details
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
8th October, 2024	

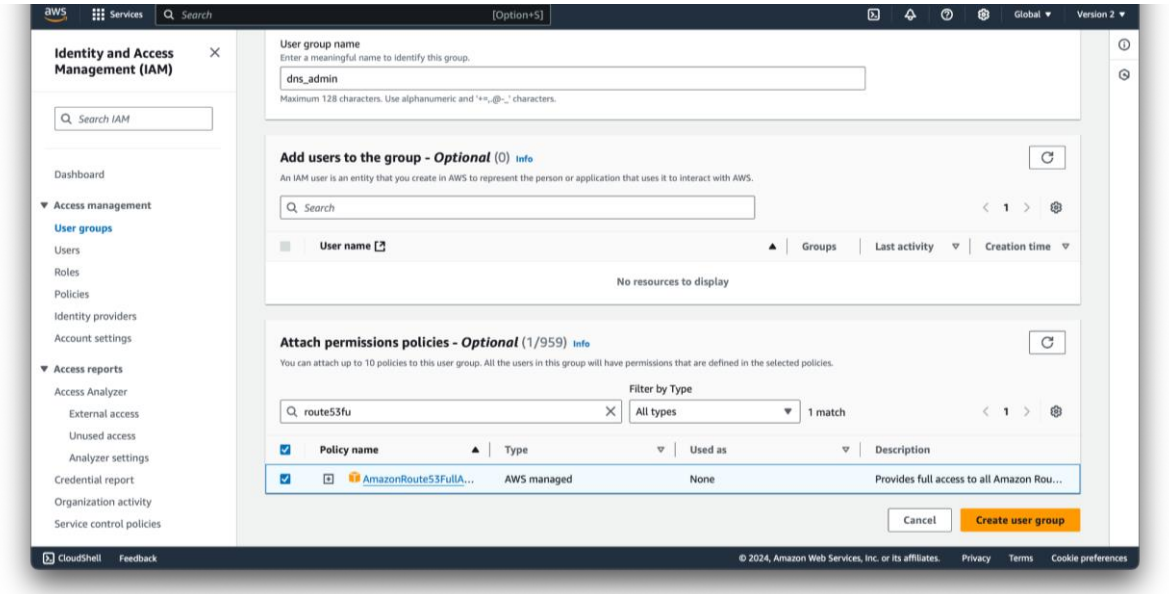
Aim:

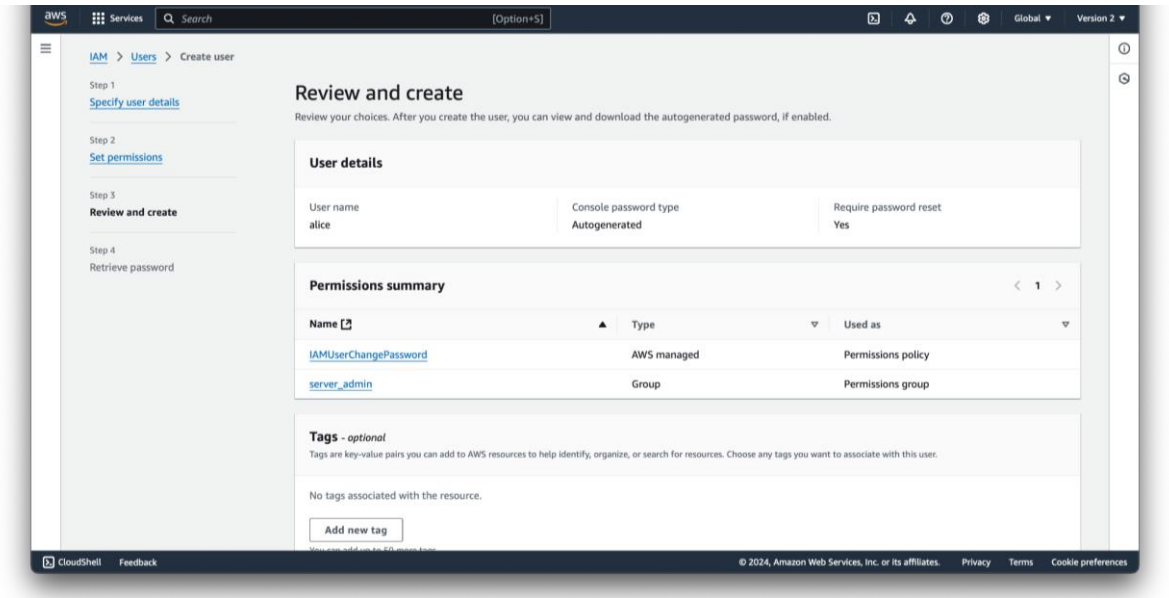
To implement IAM user access in AWS Console.

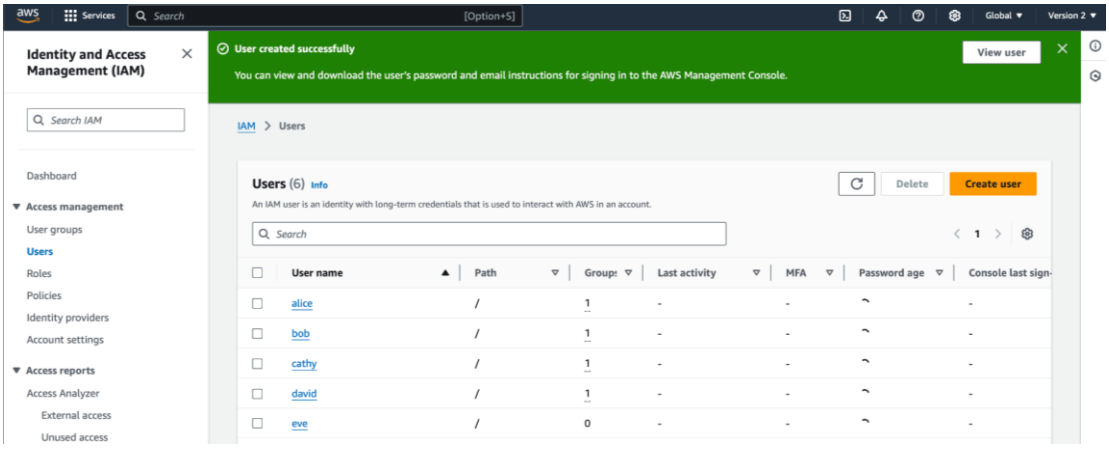
Program:

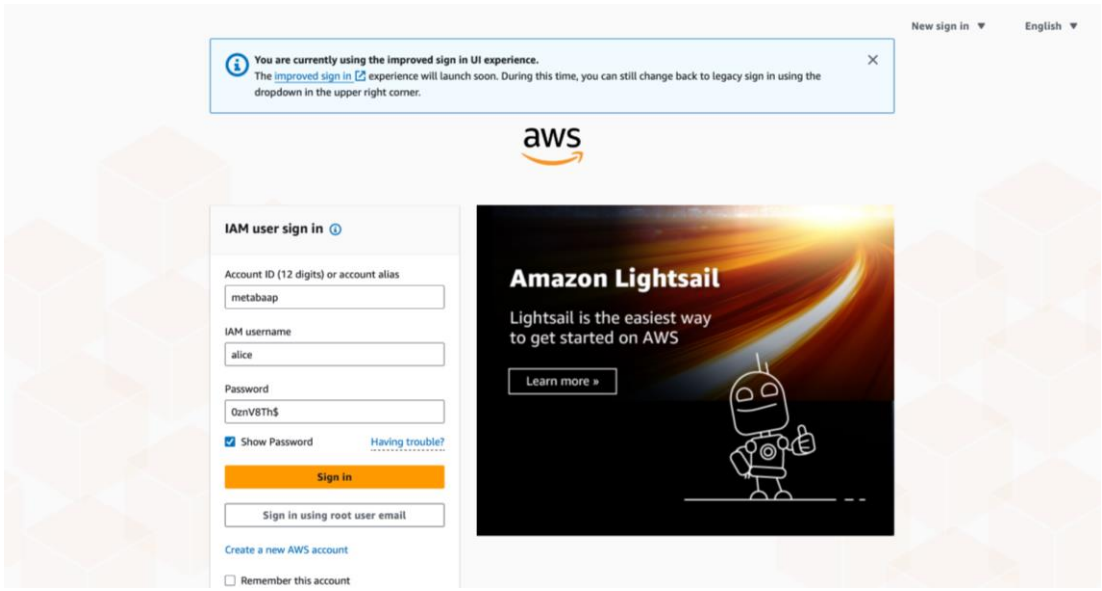
1. Create User Groups server_admin and dns_admin. Give them full access to EC2 Service and Route53 services.
2. Create users, alice, bob, cathay and david. Set passwords for them
3. Add alice and bob to user group server_adin and cathy and david to user group dns_admin.
4. Create user eve and give him Billing access.
5. Create user vishwa 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.

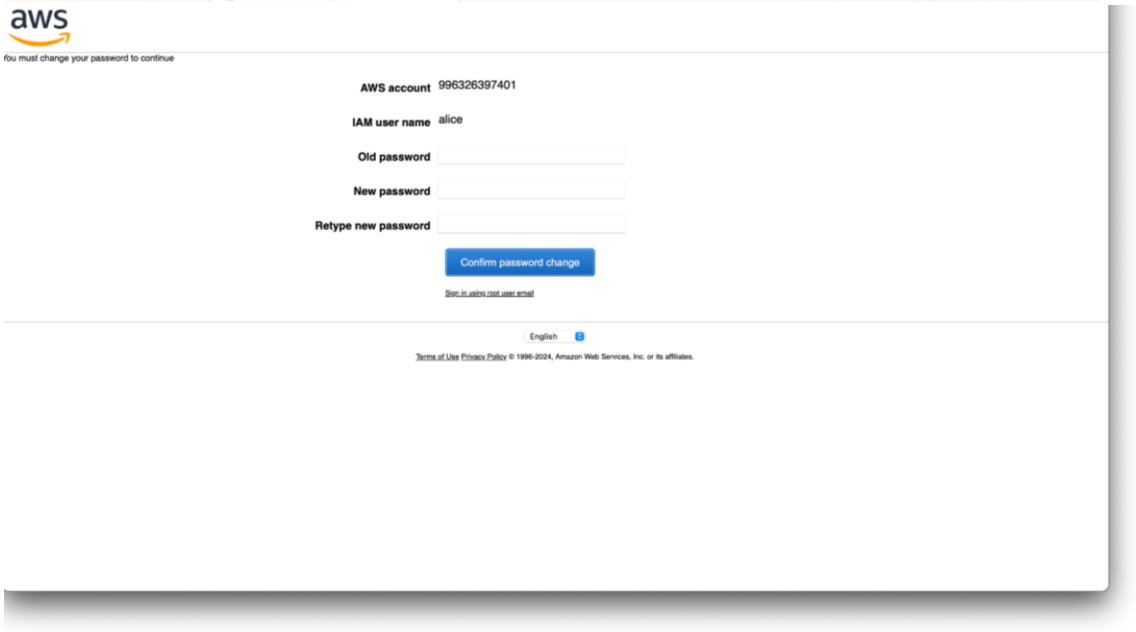
Output:

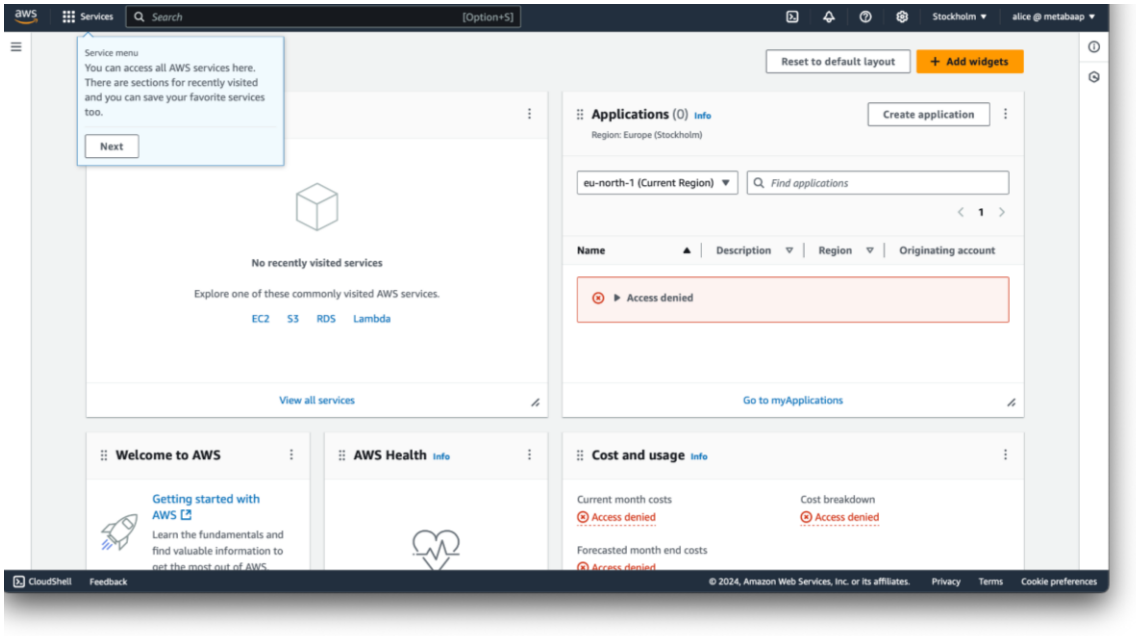

Identity and Access Management (IAM)
User group name
Enter a meaningful name to identify this group.
dns_admin
Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.
Add users to the group - Optional (0) Info
An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.
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.
route53fu
All types
1 match
AmazonRoute53FullA...
AWS managed
None
Provides full access to all Amazon Rou...
Cancel
Create user group


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
IAMUserChangePassword
AWS managed
Permissions policy
server_admin
Group
Permissions group
Tags - optional
No tags associated with the resource.
Add new tag


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
alice
bob
cathy
david
eve


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
0znV8Th$
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 »


aws
You must change your password to continue
AWS account 996326397401
IAM user name alice
Old password
New password
Retype new password
Confirm password change
Sign in using root user email
English
Terms of Use Privacy Policy © 1996-2024, Amazon Web Services, Inc. or its affiliates.



Result:

Successfully implemented IAM labs in AWS