CLOUD COMPUTING EXPERIMENTS

Submitted By

Smriti Priya

101765006

Under Supervision of

Dr. Gitanjali Chandwani Manocha

Assistant Professor
Electronics & Communication

Department of Electronics and Communication Engineering

THAPAR INSTITUTE OF ENGINEERING & TECHNOLOGY,

PATIALA, PUNJAB

July 2020-Dec 2020

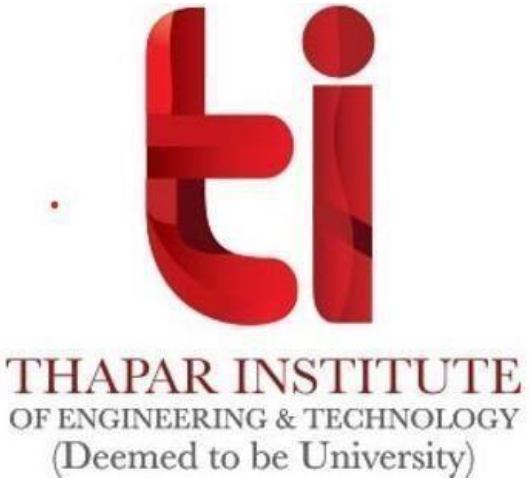
Table of Contents

A Practical Activity Report For
Cloud Computing

Submitted By:

101765006 Smriti(ENC-7)

Submitted To:
Dr. Gitanjali Chandwani



Depatment of Electronics and Communication Engineering

THAPAR INSTITUTE OF ENGINEERING & TECHNOLOGY, (DEEMED TO BE UNIVERSITY), PATIALA, PUNJAB

Experiment 10

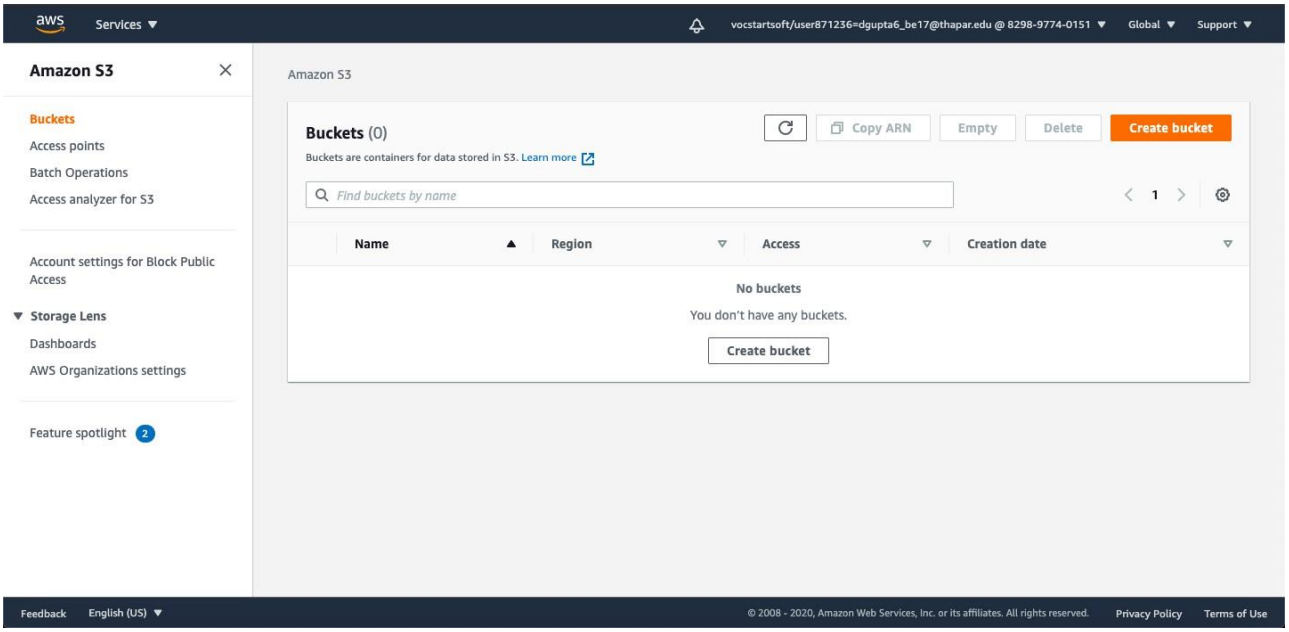
Aim: To Create CloudFront Distribution using Amazon S3.

Introduction: Amazon CloudFront is a web service that speeds up distribution of your static and dynamic web content, such as .html, .css, .js, and image files, to your users. CloudFront delivers your content through a worldwide network of data centres called edge locations. When a user requests content that you're serving with CloudFront, the user is routed to the edge location that provides the lowest latency (time delay), so that content is delivered with the best possible performance.

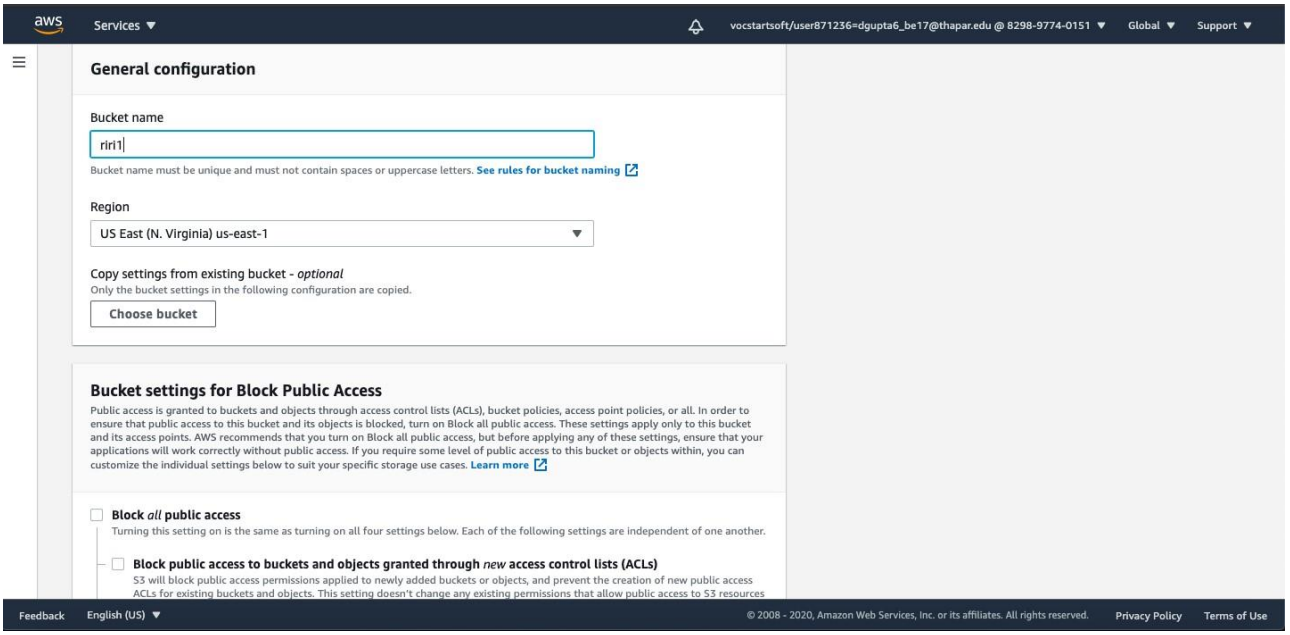
- If the content is already in the edge location with the lowest latency, CloudFront delivers it immediately.
- If the content is not in that edge location, CloudFront retrieves it from an origin that you've defined—such as an Amazon S3 bucket, a Media Package channel, or an HTTP server (for example, a web server) that you have identified as the source for the definitive version of your content.

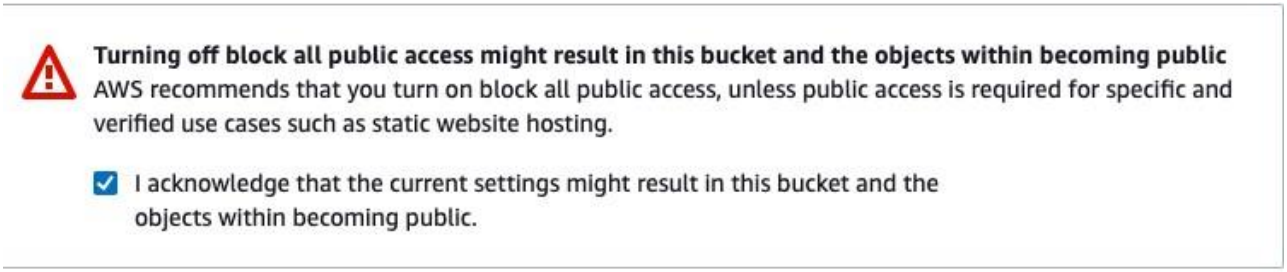
STEPS TO CREATE CLOUDFRONT DISTRIBUTION

1 Go to S3 services and CREATE Bucket.

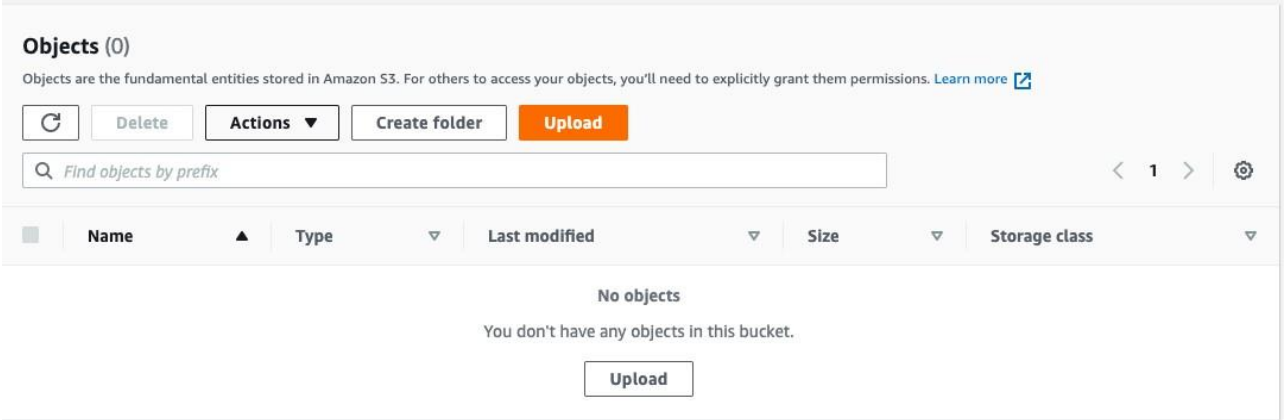


2 Provide the Bucket Name and uncheck the **Block all Public Access** and check the **Acknowledgement box**.

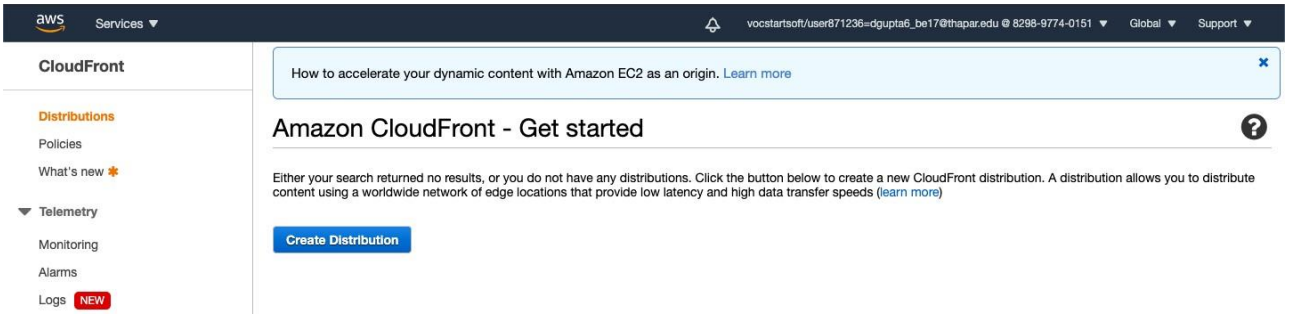




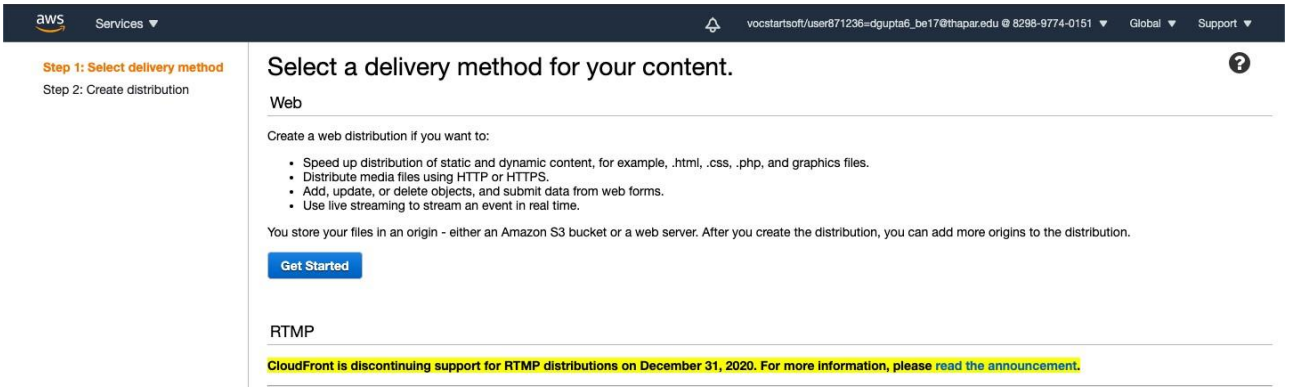
3 Upload the Image by clicking on the Upload button.



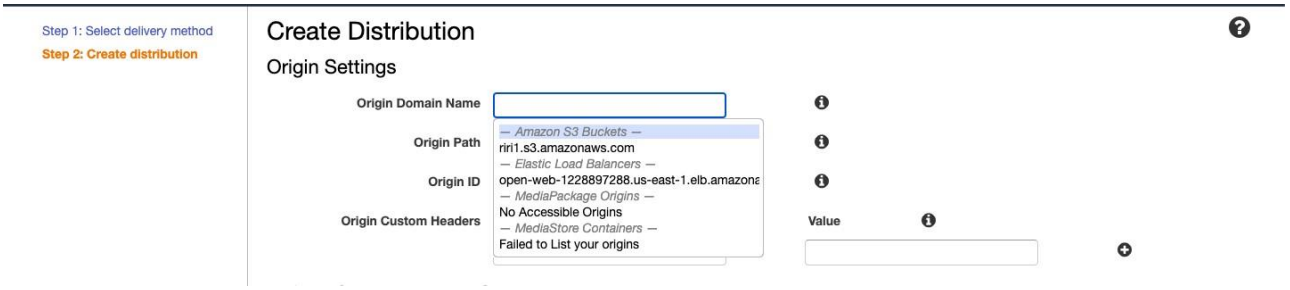
4 Now we have created our object using S3, now go to CloudFront and click on CREATE DISTRIBUTION.



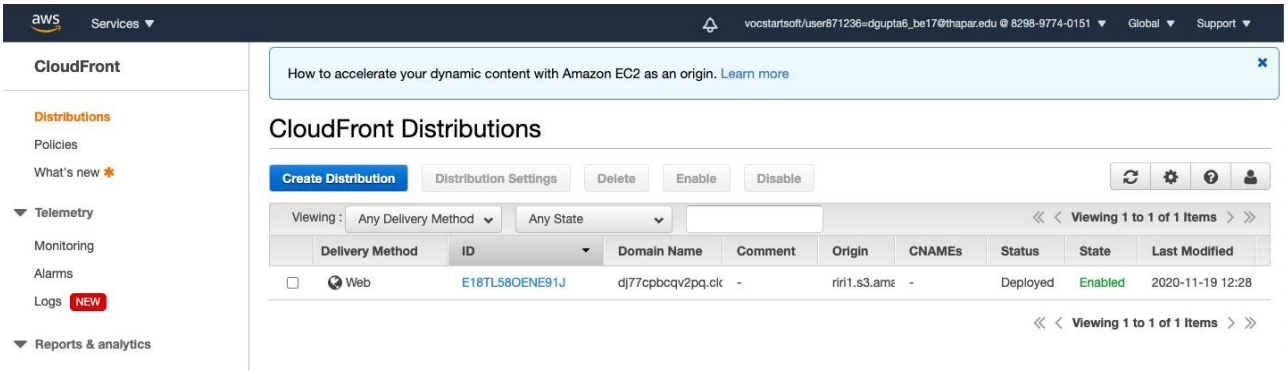
5 We will be creating the CloudFront using the WEB.



6 We will give Domain name that we have created earlier using S3 service.



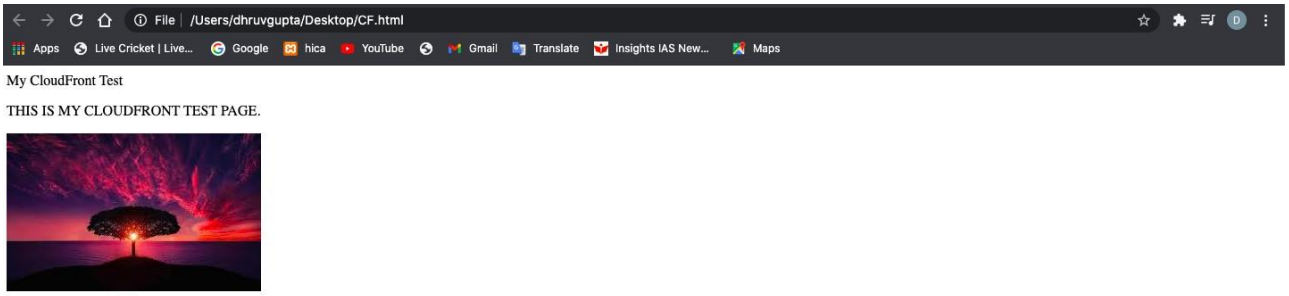
7 Create txt file and write the html content in it and instead of Domain name give the name Domain name that you have get after creating the CloudFront.



CF.txt

```
<html>
<head>My CloudFront Test</head>
<body>
<p>THIS IS MY CLOUDFRONT TEST PAGE.</p>
<p><img src="http://dj77cpbcqv2pq.cloudfront.net/download.jpeg" alt="my test image" /></p>
</body>
</html>
```

Results:



SMRITI PRIYA
101765006
ENC-7

Cloud Computing Lab-1

Part A

1. Write a Python program to print the following string in a specific format (see the output).
Twinkle, twinkle, little star,
How I wonder what you are!
Up above the world so high,
Like a diamond in the sky.
Twinkle, twinkle, little star,
How I wonder what you are

code:

```
print('Twinkle, twinkle, little star,')
print('How I wonder what you are!')
print('Up above the world so high,')
print('Like a diamond in the sky.')
print('Twinkle, twinkle, little star,)
print('How I wonder what you are')
```

output:

```
Twinkle, twinkle, little star,
How I wonder what you are!
Up above the world so high,
Like a diamond in the sky.
Twinkle, twinkle, little star,
How I wonder what you are
>>> sni
```

2. Write a Python program to display the current date and time.

code:

```
from datetime import datetime
now = datetime.now()
print("now =", now)
dt_string = now.strftime("%d/%m/%Y %H:%M:%S")
print("date and time =", dt_string)
```

output:

```
py
now = 2020-08-27 22:12:11.472483
date and time = 27/08/2020 22:12:11
>>>
```

SMRITI PRIYA
101765006
ENC-7

3. Write a Python program which accepts the radius of a circle from the user and compute the area.

code:

```
n=int(input())
print('area',str(3.14*n**2))
```

output:

```
py
5
area 78.5
>>>
```

4. Write a Python program which accepts the user's first and last name and print them in reverse order with a space between them.

code:

```
s=input('enter your name    ')
l=[i for i in s.split()]
for i in l[::-1]:
    print(i,end=' ')
print()
```

output:

```
 RESTART: C:/Users/hp/AppData/Local/Programs/Python/P
enter your name     Priyanshu Priyam Srivastava
Srivastava Priyam Priyanshu
>>>
```

5. Write a Python program to accept a filename from the user and print the extension of that.

code:

```
s=input('enter file name    ')
l=[i for i in s.split('.')]
print('extention ',l[-1])
```

output:

```
>>>
 RESTART: C:/Users/hp/AppData/Local/Prog
enter file name     word.txt
extention  txt
>>>
```

SMRITI PRIYA
101765006
ENC-7

6. Write a Python program to display the first and last colours from a list.

code:

```
s=input('enter the colors seperated by spaces   ')
l=[i for i in s.split()]
print('First colour ',l[0])
print('Last colour ',l[-1])
```

output:

```
 RESTART: C:/Users/hp/AppData/Local/Programs/Python/Python37-32/ofro
enter the colors seperated by spaces   red yellow blue pink orange
First colour  red
Last colour  orange
>>>
```

7. Write a Python program that accepts an integer (n) and computes the value of n+nn+nnn.

code:

```
n=input('enter the value ')
print('value of n+nn+nnn is ',end='')
l=int(n*2)
m=int(n*3)
n=int(n)
print(l+m+n)
```

output:

```
enter the value 5
value of n+nn+nnn is 615
>>>
```

8. Write a Python program to test whether a number is within 100 of 1000 or 2000.

code:

```
n=input('enter the value ')
s='NO'
for i in range(100 ,10001):
    if int(n)==i:
        s='YES'
        break
print(s)
```

SMRITI PRIYA
101765006
ENC-7

output:

```
 RESTART: C:/Users/hp/Appl
enter the value 5
NO
>>>
```

9. Write a Python program to find whether a given number (accept from the user) is even or odd, print out an appropriate message to the user.

code:

```
n=int(input('enter the value '))
if n%2==0:
   print(str(n)+' is even')
else:
   print(str(n)+' is odd')
```

output:

```
 RESTART: C:/Users/hp/Ap
enter the value 5
5 is odd
>>>
```

10. Write a Python program to count the number 4 in a given list.

code:

```
n=input('Enter the number')
print("Number of 4's present in "+str(n)+" is "+str(n.count('4')))
```

output:

```
 RESTART: C:/Users/hp/AppData/Local/Programs/
Enter the number54498444
Number of 4's present in 54498444 is 5
>>>
```

Part B

To Understand AWS Storage service (S3 and EBS) and summarize the learning in one page.

Amazon S3

SMRITI PRIYA
101765006
ENC-7
Amazon Simple Storage Service is storage for the Internet. It is designed to make web-scale computing easier for developers. Amazon S3 has a simple web services interface that you can use to store and retrieve any amount of data, at any time, from anywhere on the web. It gives any developer access to the same highly scalable, reliable, fast, inexpensive data storage infrastructure that Amazon uses to run its own global network of web sites. The service aims to maximize benefits of scale and to pass those benefits on to developers.

Following are some advantages:

1. Creating buckets: Create and name a bucket that stores data. Buckets are the fundamental containers in Amazon S3 for data storage

2. Storing data – Store an infinite amount of data in a bucket. Upload as many objects as you like into an Amazon S3 bucket. Each object can contain up to 5 TB of data. Each object is stored and retrieved using a unique developer-assigned key.

3. Downloading data – Download your data or enable others to do so. Download your data anytime you like, or allow others to do the same.

4. Permissions – Grant or deny access to others who want to upload or download data into your Amazon S3 bucket. Grant upload and download permissions to three types of users. Authentication mechanisms can help keep data secure from unauthorized access.

5. Standard interfaces – Use standards-based REST and SOAP interfaces designed to work with any internet-development toolkit.

Amazon EBS

Amazon EBS allows you to create storage volumes and attach them to Amazon EC2 instances. Once attached, you can create a file system on top of these volumes, run a database, or use them in any other way you would use block storage. Amazon EBS volumes are placed in a specific Availability Zone where they are automatically replicated to protect you from the failure of a single component. All EBS volume types offer durable snapshot capabilities and are designed for 99.999% availability.

Amazon EBS provides a range of options that allow you to optimize storage performance and cost for your workload. These options are divided into two major categories: SSD-backed storage for transactional workloads, such as databases and boot volumes (performance depends primarily on IOPS), and HDD-backed storage for throughput intensive workloads, such as MapReduce and log processing (performance depends primarily on MB/s).

SSD-backed volumes include the highest performance Provisioned IOPS SSD (io2 and io1) for latency-sensitive transactional workloads and General Purpose SSD (gp2) that balance price and performance for a wide variety of transactional data. HDD-backed volumes include Throughput Optimized HDD (st1) for frequently accessed, throughput intensive workloads and the lowest cost Cold HDD (sc1) for less frequently accessed data.

SMRITI PRIYA
101765006
ENC-7

Elastic Volumes is a feature of Amazon EBS that allows you to dynamically increase capacity, tune performance, and change the type of live volumes with no downtime or performance impact. This allows you to easily right-size your deployment and adapt to performance changes.

Cloud Computing Lab-2

1. Write a Python program to test whether a passed letter is a vowel or not.

code:

```
n=input('Enter the character')
s='aeiouAEIOU'
if n in s:
    print(str(n)+' is a vowel')
else:
    print(str(n)+' is not a vowel')
```

output:

```
Enter the characterA
A is a vowel
Enter the charactere
e is a vowel
Enter the characterP
P is not a vowel
Enter the characterz
z is not a vowel
>>>
```

2. Write a Python program to print out a set containing all the colours from color_list_1 which are not present in color_list_2.
color_list_1 = set(["White", "Black", "Red"])
color_list_2 = set(["Red", "Green"])

code:

```
color_list_1 = set(["White", "Black", "Red"])
color_list_2 = set(["Red", "Green"])
for i in color_list_2:
    if i not in color_list_1:
        print(i)
```

output:

```
>>>
 RESTART: C:/Users
Green
>>>
```

SMRITI PRIYA
101765006
ENC-7

3. Write a Python program to solve (x + y) * (x + y).

code:

```
x=int(input())
y=int(input())
print('(x+y)*(x+y) is ', end=' ')
print((x+y)**2)
```

output:

```
 RESTART: C:/Users/hp/App
2
3
(x+y)*(x+y) is  25
>>>
```

4. Write a Python program to calculate the sum of the digits in an integer.

code:

```
n=input()
l=[int(i) for i in n]
print('sum of digits of '+ str(n)+str(sum(l)))
```

output:

```
 RESTART: C:/Users/hp/AppData/Lo
457812
sum of digits of 45781227
>>>
```

5. Write a Python program to calculate the length of a string.

code:

```
n=input()

print('length of '+ str(n)+str(len(n)))
```

output:

```
 RESTART: C:/Users/hp/AppDat
564686745
length of 5646867459
>>>
```

6. Write a Python program to get a single string from two given strings, separated by a space and swap the first two characters of each string.
Sample String : 'abc', 'xyz'
Expected Result : 'xyc abz'

SMRITI PRIYA
101765006
ENC-7

7. Write a Python program to remove the nth index character from a nonempty string.

Code:

```
s=input('enter the string  ')
n=int(input('enter the index at which character is to be removed'))
ns=''
ns=s[:n]
ns+=s[n+1:]
print(ns)
```

Output:

```
enter the string        qwerttyuiop
enter the index at which character is to be removed6
        qwertyuiop
>>>
```

8. Write a Python script that takes input from the user and displays that input back in upper and lower cases

code:

```
s=input('enter the string  ')
print(s.upper())
```

output.

```
RESTART: C:/Users/hp/AppData/Local/Program
enter the string  fewufewiuefiohfrui
FEWUFEWIUEFIOHFRUI
>>>
```

9. Write a Python function to insert a string in the middle of a string.

code:

```
s=input('enter the string  ')
i=input('enter the string to be inserted in the middle')
mid=(len(s)+1)//2
ns=''
ns=s[:mid]
ns+=i
ns+=s[mid+1:]
print(ns)
```

SMRITI PRIYA
101765006
ENC-7

output:

```
enter the string  qwertyuiop
enter the string to be inserted in the middleasdf
qwertasdfuiop
>>>
```

10. Write a Python program to print the floating numbers upto 2 decimal places with a sign.

code:

```
a_float = input('Enter the Floating Point Number ')
a_float=float(a_float)
formatted_float = "{:.2f}".format(a_float)
print(formatted_float)
```

output:

```
Enter the Floating Point Number 3.1415278
3.14
>>>
```

Part B

To Understand AWS compute service (EC2 & lightsail) and summarize the learning in one page.

EC2

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

Amazon EC2 provides the following features:

1. Virtual computing environments, known as instances

2. Preconfigured templates for your instances, known as Amazon Machine Images (AMIs), that package the bits you need for your server (including the operating system and additional software)

3. Various configurations of CPU, memory, storage, and networking capacity for your instances, known as instance types

4. Secure login information for your instances using key pairs (AWS stores the public key, and you store the private key in a secure place)

5. Storage volumes for temporary data that's deleted when you stop or terminate your instance, known as instance store volumes

6. Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as Amazon EBS volumes

7. Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as Regions and Availability Zones

8. A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using security groups

9. Static IPv4 addresses for dynamic cloud computing, known as Elastic IP addresses

10.Metadata, known as tags, that you can create and assign to your Amazon EC2 resources

11.Virtual networks you can create that are logically isolated from the rest of the AWS cloud, and that you can optionally connect to your own network, known as virtual private clouds (VPCs)

Lightsail

AWS Lightsail is an entry-level AWS service, offering app developers access to a configurable virtual private server (VPS) and a suite of easy to use tools. Amazon Lightsail provides an easy, lightweight way for new cloud users to take advantage of AWS' cloud computing services.

Advantages:

1. Simplified UI: It runs through pre-installed software. There will be less confusion for users who are using it for the first time.

2. Familiarity: Developers and companies who use Amazon Lightsail for web hosting and application development will find it more alluring.

3. Affordability: The subscription is free for the first month and is $5/month for the cheapest tier.

4. Reliability: The need for replacement is rare as the services run through Amazon infrastructure and data centers.

AWS Lightsail is mostly used by first-time cloud users and developers looking to test out app prototypes, or by startups and small scale production studios for whom the limitations of the Lightsail environment are sufficient.

SMRITI PRIYA
101765006
ENC-7

Cloud Computing Lab-3

Part A

1. Write a Python program to sum all the items in a list

code:

```
l=[1,2,3,4,5,6443,3,68]
print(l)
print('sum',sum(l))
```

output:

```
 RESTART: C:/Users/hp/AppData/Local/Pro
[1, 2, 3, 4, 5, 6443, 3, 68]
sum 6529
>>>
```

2. Write a Python program to get the largest number from a list.

code:

```
l=[1,2,3,4,5,6443,3,68]
print('list',l)
print('largest number',sum(l))
```

output:

```
list [1, 2, 3, 4, 5, 6443, 3, 68]
largest number 6529
>>>
```

3. Write a Python program to get a list, sorted in increasing order by the last element in each tuple from a given list of non-empty tuples

code:

```
l=[1,2,3,4,5,64,8,3,68]
print('list',l)
print('sorted in increasing order',sum(l))
```

output:

```
list [1, 2, 3, 4, 5, 64, 8, 3, 68]
sorted in increasing order 158
>>>
```

4. Write a Python program to remove duplicates from a list

code:

```
l=[1,2,3,4,5,64,8,3,6,8,64,6]
print('list',l)
print('after removing duplicates ',set(l))
```

SMRITI PRIYA
101765006
ENC-7

output:

```
list [1, 2, 3, 4, 5, 64, 8, 3, 6, 8, 64, 6]
after removing duplicates  {64, 1, 2, 3, 4, 5, 6, 8}
>>>
```

5. Write a Python program to find the list of words that are longer than n from a given list of words

code:

```
s=input('enter the words in list ')
l=[i for i in s.split()]
n=int(input('enter th threshold '))
for i in l:
    if len(i)>n:
        print(i)
```

output:

```
 RESTART: C:/Users/hp/AppData/Local/Programs/Python/Python37-32
enter the words in list apple banana orange guava ,pineapple
enter th threshold 5
banana
orange
,pineapple
>>>
```

6. Write a Python program to get the difference between the two lists

code:

```
s=input('enter numbers in first list ')
t=input('enter numbers in second list ')
s=[int(i) for i in s.split()]
t=[int(i) for i in t.split()]
l=[s[i]-t[i] for i in range(len(s))]
print('difference between the lists are ',*l)
```

output:

```
enter numbers in first list 1 2 3 4
enter numbers in second list 5 2 1 3
difference between the lists are  -4 0 2 1
>>>
```

SMRITI PRIYA
101765006
ENC-7

7. Write a Python script to add a key to a dictionary
code:

```
dic1={'1':10, '2':20}
print(dic1)
i=input('enter the key in dict ')
v=input('enter the value of the key ')
dic1[i]=v
print(dic1)
```

output:

```
 RESTART: C:/Users/hp/AppData/Local/Pro(
{'1': 10, '2': 20}
enter the key in dict 5
enter the value of the key 40
{'1': 10, '2': 20, '5': '40'}
>>>
```

8. Write a Python script to concatenate following dictionaries to create a new one.
Sample Dictionary :
dic1={1:10, 2:20}
dic2={3:30, 4:40}
dic3={5:50,6:60}
Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
code:

```
from collections import Counter
dic1={1:10, 2:20}
print(dic1)
dic2={3:30, 4:40}
dic3={5:50,6:60}
print(dic2)
print(dic3)
dic1=Counter(dic1)
dic2=Counter(dic2)
dic3=Counter(dic3)
dic4=Counter()
dic4=dic1+dic2+dic3
print(dic4)
```

output:

```
 RESTART: C:/Users/hp/AppData/Local/Programs/Python/Python
{1: 10, 2: 20}
{3: 30, 4: 40}
{5: 50, 6: 60}
Counter({6: 60, 5: 50, 4: 40, 3: 30, 2: 20, 1: 10})
>>>
```

SMRITI PRIYA
101765006
ENC-7

9. Write a Python program to remove a key from a dictionary
code:

```
from collections import Counter
dic1={1:10, 2:20}
#print(dic1)
dic2={3:30, 4:40}
dic3={5:50,6:60}
#print(dic2)
#print(dic3)
dic1=Counter(dic1)
dic2=Counter(dic2)
dic3=Counter(dic3)
dic4=Counter()
dic4=dic1+dic2+dic3
print(dic4)
dic4=dict(dic4)
del(dic4[5])
print(dic4)
```

output:

```
>>
RESTART: C:/Users/hp/AppData/Local/Programs/Python/Python37-32/
Counter({6: 60, 5: 50, 4: 40, 3: 30, 2: 20, 1: 10})
{1: 10, 2: 20, 3: 30, 4: 40, 6: 60}
>>
```

10. Write a Python program to multiply all the items in a dictionary
code:

```
from collections import Counter
dic1={1:10, 2:20}
#print(dic1)
dic2={3:30, 4:40}
dic3={5:50,6:60}
#print(dic2)
#print(dic3)
dic1=Counter(dic1)
dic2=Counter(dic2)
dic3=Counter(dic3)
dic4=Counter()
dic4=dic1+dic2+dic3
print(dic4)
dic4=dict(dic4)
mul=1
```

SMRITI PRIYA
101765006
ENC-7

```
for i in dic4:
   mul*=dic4[i]
print(mul)
```

output:

```
>>>
 RESTART: C:/Users/hp/AppData/Loca
Counter({6: 60, 5: 50, 4: 40, 3: 3
720000000
```

Part B

To Understand AWS database service (Aurora, Redshift & DynamoDB) and summarize the learning in one page.

Aurora

Amazon Aurora (Aurora) is a fully managed relational database engine that's compatible with MySQL and PostgreSQL. Aurora includes a high-performance storage subsystem. Its MySQL- and PostgreSQL-compatible database engines are customized to take advantage of that fast distributed storage. The underlying storage grows automatically as needed, up to 64 tebibytes (TiB). Aurora also automates and standardizes database clustering and replication, which are typically among the most challenging aspects of database configuration and administration. Aurora is part of the managed database service Amazon Relational Database Service (Amazon RDS). Amazon RDS is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. If you are not already familiar with Amazon RDS

Redshift

Redshift is a fully managed, petabyte-scale data warehouse service in the cloud. It is also used to perform large scale database migrations.

Redshift's column-oriented database is designed to connect to SQL-based clients and business intelligence tools, making data available to users in real time. Based on PostgreSQL 8, Redshift delivers fast performance and efficient querying that help teams make sound business analyses and decisions.

Redshift is Amazon's analytics database, and is designed to crunch large amounts of data as a data warehouse. Those interested in Redshift should know that it consists of clusters of databases with dense storage nodes, and allows you to even run traditional relational databases in the cloud.

Amazon Redshift automatically handles many of the time-consuming tasks associated with managing your own data warehouse, including:

Setup: With Amazon Redshift, you simply create a data warehouse cluster, define your schema, and begin loading and querying your data. You don't have to manage provisioning, configuration or patching.

SMRITI PRIYA
101765006
ENC-7

Data Durability: Amazon Redshift replicates your data within your data warehouse cluster and continuously backs up your data to Amazon S3, which is designed for eleven nines of durability. Amazon Redshift mirrors each drive's data to other nodes within your cluster. If a drive fails, your queries will continue with a slight latency increase while Redshift rebuilds your drive from replicas. In case of node failure(s), Amazon Redshift automatically provisions new node(s) and begins restoring data from other drives within the cluster or from Amazon S3. It prioritizes restoring your most frequently queried data so your most frequently executed queries will become performant quickly.

Scaling: You can add or remove nodes from your Amazon Redshift data warehouse cluster with a single API call or via a few clicks in the AWS Management Console as your capacity and performance needs change. You can also schedule your scaling and resize operations by using the scheduler capability in Redshift.

Automatic Updates and Patching: Amazon Redshift automatically applies upgrades and patches your data warehouse so you can focus on your application and not on its administration.

Exabyte Scale Query Capability: Redshift Spectrum enables you to run queries against exabytes of data in Amazon S3. There is no loading or ETL required. Even if you don’t store any of your data in Amazon Redshift, you can still use Redshift Spectrum to query datasets as large as an exabyte in Amazon S3

DynamoDb

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. DynamoDB lets you offload the administrative burdens of operating and scaling a distributed database so that you don't have to worry about hardware provisioning, setup and configuration, replication, software patching, or cluster scaling. DynamoDB also offers encryption at rest, which eliminates the operational burden and complexity involved in protecting sensitive data.

DynamoDB provides on-demand backup capability. It allows you to create full backups of your tables for long-term retention and archival for regulatory compliance needs.DynamoDB allows you to delete expired items from tables automatically to help you reduce storage usage and the cost of storing data that is no longer relevant.

DynamoDB is a particularly good fit for the following use cases:

1. Applications with large amounts of data and strict latency requirements. As your amount of data scales, JOINs and advanced SQL operations can slow down your queries. With DynamoDB, your queries have predictable latency up to any size, including over 100 TBs!

2. Serverless applications using AWS Lambda. AWS Lambda provides auto-scaling, stateless, ephemeral compute in response to event triggers. DynamoDB is accessible via an HTTP API and performs authentication & authorization via IAM roles, making it a perfect fit for building Serverless applications.

3. Data sets with simple, known access patterns. If you're generating recommendations and serving them to users, DynamoDB's simple key-value access patterns make it a fast, reliable choice.

SMRITI PRIYA
101765006
ENC-7

Cloud Computing Lab-4

Part A

1. Write a Python program to create a tuple with different data types.

code:

```
tup1 = ('physics', 'chemistry', 1997, 2000)
tup2 = (1, 2, 3, 4, 5, 6, 7 )
print("tup1[0]:" , tup1[0])
print("tup2[1:5]: ", tup2[1:5])
```

output:

```
tup1[0]: physics
tup2[1:5]:  (2, 3, 4, 5)
>>>
```

2. Write a Python program to add an item in a tuple

code:

```
tup1 = ('physics', 'chemistry', 1997, 2000)
print('before updating',tup1)
#tup2 = (1, 2, 3, 4, 5, 6, 7 )
ad=input('enter the value to be updated to tuple ')
tup2=(ad,)
print(tup1+tup2)
```

output:

```
before updating ('physics', 'chemistry', 1997, 2000)
enter the value to be updated to tuple Food Technology
('physics', 'chemistry', 1997, 2000, 'Food Technology')
>>>
```

3. Write a Python program to convert a tuple to a string

code:

```
tup1 = ('physics', 'chemistry', 1997, 2000)
print('tuple',tup1)
#tup2 = (1, 2, 3, 4, 5, 6, 7 )
#ad=input('enter the value to be updated to tuple ')
#tup2=(ad,)
s=''
for i in tup1:
    s+=str(i)+' '
print('String ',s)
```

SMRITI PRIYA
101765006
ENC-7

output:

```
 RESTART: C:/Users/hp/AppData/Local/Programs/Py
tuple ('physics', 'chemistry', 1997, 2000)
String  physics chemistry 1997 2000
>>>
```

4. Write a Python program to get the 4th element and 4th element from last of a tuple.
code:

```
tup1 = ('physics', 'chemistry', 1997, 2000,'Food Technology','Astrophysics','String
Theory','infinite')
print('tuple',tup1)
#tup2 = (1, 2, 3, 4, 5, 6, 7 )
#ad=input('enter the value to be updated to tuple ')
#tup2=(ad,)
print('4th element is ',tup1[4])
print('4th element from last is ',tup1[-1])
```

output:

```
RESTART: C:/Users/hp/AppData/Local/Programs/Python/Python37-32/01101mmmmmkjcw.py
tuple ('physics', 'chemistry', 1997, 2000, 'Food Technology', 'Astrophysics', 'String Theory', 'infinite')
4th element is  Food Technology
4th element from last is  infinite
>>>
```

5. Write a Python program to find the repeated items of a tuple
code:

```
t=('A','BE','a','b','W','A','b','W')
print(t)
l=[]
for i in t:
    if t.count(i)>1 and i not in l:
        l.append(i)
print(*l)
```

output:

```
('A', 'BE', 'a', 'b', 'W', 'A', 'b', 'W')
A b W
>>>
```

6. Write a Python program to check whether an element exists within a tuple
code:

SMRITI PRIYA
101765006
ENC-7

```
t=('A','BE','a','b','W','A','b','W')
print(t)

i=input('enter the element to be checked ')
if i in t:
    s='Yes'
else:
    s='No'
print(s)
```

output:

```
('A', 'BE', 'a', 'b', 'W', 'A', 'b', 'W')
enter the element to be checked a
Yes
('A', 'BE', 'a', 'b', 'W', 'A', 'b', 'W')
enter the element to be checked z
No
>>>
```

7. Write a Python program to convert a list to a tuple

code:

```
t=['A','BE','a','b','W','A','b','W']
print('List',t)
ts=tuple(t)
print('tuple',ts)
```

output:

```
 RESTART: C:/Users/hp/AppData/Local/Programs/Python/Pyth
List ['A', 'BE', 'a', 'b', 'W', 'A', 'b', 'W']
tuple ('A', 'BE', 'a', 'b', 'W', 'A', 'b', 'W')
>>>
```

8. Write a Python program to slice a tuple

code:

```
tup1 = ('physics', 'chemistry', 1997, 2000)
tup2 = (1, 2, 3, 4, 5, 6, 7 )
print("tup1[0]:" , tup1[0])
print("tup2[1:5]: ", tup2[1:5])
```

output:

SMRITI PRIYA
101765006
ENC-7

```
tup1[0]: physics
tup2[1:5]:  (2, 3, 4, 5)
>>>
```

9. Write a Python program to find the length of a tuple
code:

```
t=['A','BE','a','b','W','A','b','W']
#print('List',t)
ts=tuple(t)
print('tuple',ts)
print('length ',len(ts))
```

output:

```
 RESTART: C:/Users/hp/AppData/Local/Programs/Python/
tuple ('A', 'BE', 'a', 'b', 'W', 'A', 'b', 'W')
length  8
>>>
```

10. Write a Python program to convert a list of tuples into a dictionary
code:

```
t=(('a',97),('b',98),('c',99),('d',100))
print('tuple',t)
d={}
for pair in t:
   d[pair[0]]=pair[1]
print('dict',d)
```

output:

```
tuple (('a', 97), ('b', 98), ('c', 99), ('d', 100))
dict {'a': 97, 'b': 98, 'c': 99, 'd': 100}
>>>
```

Part B

To Understand Map-Reduce Paradigm and summarize the learning in one page.

MapReduce is a programming framework that allows us to perform distributed and parallel processing on large data sets in a distributed environment.

MapReduce consists of two distinct tasks — Map and Reduce.

As the name MapReduce suggests, reducer phase takes place after the mapper phase has been completed.

SMRITI PRIYA
101765006
ENC-7
So, the first is the map job, where a block of data is read and processed to produce key-value pairs as intermediate outputs.

The output of a Mapper or map job (key-value pairs) is input to the Reducer.

The reducer receives the key-value pair from multiple map jobs.

Then, the reducer aggregates those intermediate data tuples (intermediate key-value pair) into a smaller set of tuples or key-value pairs which is the final output.

Advantages of MapReduce:

1. Parallel Processing

In MapReduce, we are dividing the job among multiple nodes and each node works with a part of the job simultaneously. So, MapReduce is based on Divide and Conquer paradigm which helps us to process the data using different machines. As the data is processed by multiple machines instead of a single machine in parallel, the time taken to process the data gets reduced by a tremendous amount

2. Data Locality

Instead of moving data to the processing unit, we are moving the processing unit to the data in the MapReduce Framework. In the traditional system, we used to bring data to the processing unit and process it. But, as the data grew and became very huge, bringing this huge amount of data to the processing unit posed the following issues:

a. Moving huge data to processing is costly and deteriorates the network performance.

b. Processing takes time as the data is processed by a single unit which becomes the bottleneck.

c. Master node can get over-burdened and may fail.

As data is distributed among multiple nodes where each node processes the part of the data residing on it, it has the following advantages

1. It is very cost effective to move the processing unit to the data.

2. The processing time is reduced as all the nodes are working with their part of the data in parallel.

3. Every node gets a part of the data to process and therefore, there is no chance of a node getting overburdened.

SMRITI PRIYA
101765006
ENC-7

Cloud Computing Lab-5

1. Create a text file named “INPUT” that comprises of some text. Using Mapper and Reducer program generate a KEY and COUNT matrix for the file.

code:

```
from functools import reduce
a=reduce(lambda output, current: output + [(current, ord(current))], 'abcde', [])
print('with reduce ',a)
print('dictionary ',dict(a))
```

output:

```
with reduce  [('a', 97), ('b', 98), ('c', 99), ('d', 100), ('e', 101)]
dictionary  {'a': 97, 'b': 98, 'c': 99, 'd': 100, 'e': 101}
>>>
```

2. XYZ.com is an online music website where users listen to various tracks, the data gets collected like shown below. Write a map reduce program to get following stats

1. Number of unique listeners
2. Number of times the track was shared with others
3. Number of times the track was listened to on the radio
4. Number of times the track was listened to in total
5. Number of times the track was skipped on the radio

The data is coming in log files and looks like as shown below.

UserId|TrackId|Shared|Radio|Skip
111115|222|0|1|0
111113|225|1|0|0
111117|223|0|1|1
111115|225|1|0|0

code:

output:

3. Five persons A, B, C, D, E have a friendship relation as shown below.

A -> B C D
B -> A C D E
C -> A B D E
D -> A B C E
E -> B C D

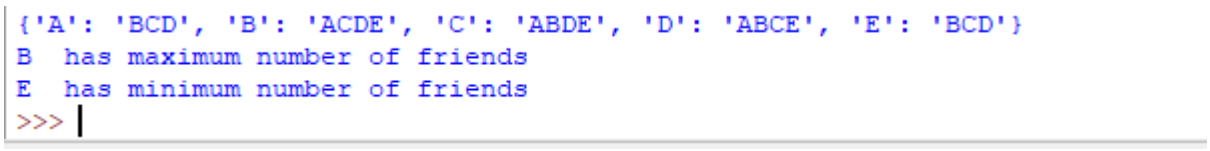
Identify who is having maximum friends, minimum friends and no friends.

code:

SMRITI PRIYA
101765006
ENC-7

```
d={'A':'BCD','B':'ACDE','C':'ABDE','D':'ABCE','E':'BCD'}
print(d)
maxx='A'
minn='A'
for i in d:
   if len(d[i])>len(d[maxx]):
      maxx=i
   if len(d[i])<len(d[maxx]):
      minn=i
print(maxx,' has maximum number of friends')
print(minn,' has minimum number of friends ')
```

output:



Experiment 6

To Launch an AWS EC2 instance and connect to it using PuTTY.

Elaboration of the terms used:-

AWS EC2 Instance:-

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides with complete control of your computing resources and lets you run on Amazon's proven computing environment. There are 5 types of instances:- General purpose, Compute Optimized, Memory optimized, Accelerated Computing and Storage Optimized. We will be using General purpose instance as it provides a balance of compute, memory and networking resources, and can be used for a variety of diverse workloads. These instances are ideal for applications that use these resources in equal proportions such as web servers and code repositories.

PuTTY:-

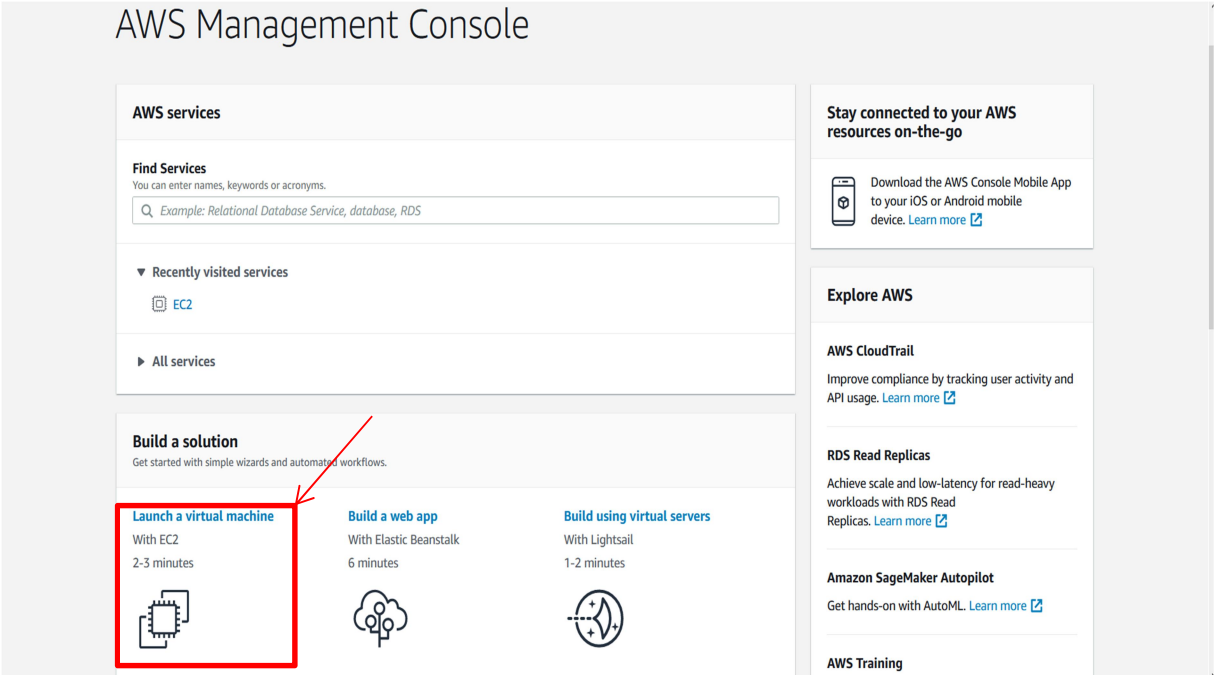
PuTTY is a free and open-source terminal emulator, serial console and network file transfer application. It supports several network protocols, including SCP, SSH, Telnet, rlogin, and raw socket connection. It can also connect to a serial port. PuTTY supports many variations on the secure remote terminal, and provides user control over the SSH encryption key and protocol version, alternate ciphers such as AES, 3DES, RC4, Blowfish, DES, and Public-key authentication. PuTTY uses own format of key files – PPK.

SSH:-

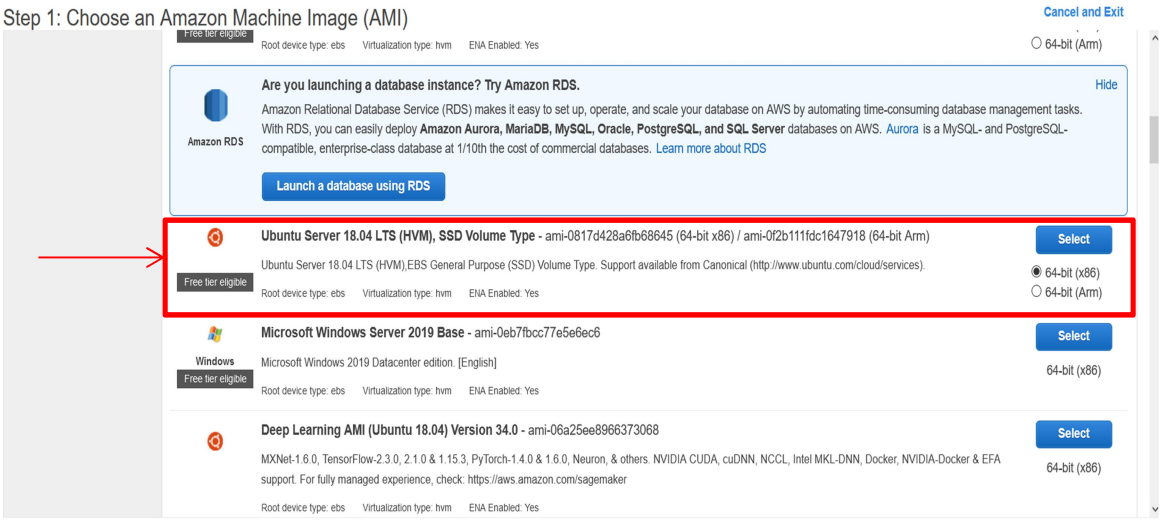
Secure Shell is a cryptographic network protocol for operating network services securely over an unsecured network. Typical applications include remote command-line, login, and remote command execution, but any network service can be secured with SSH. The SSH protocol uses encryption to secure the connection between a client and a server. All user authentication, commands, output, and file transfers are encrypted to protect against attacks in the network.

Steps to launch and connect to EC2 instance:-

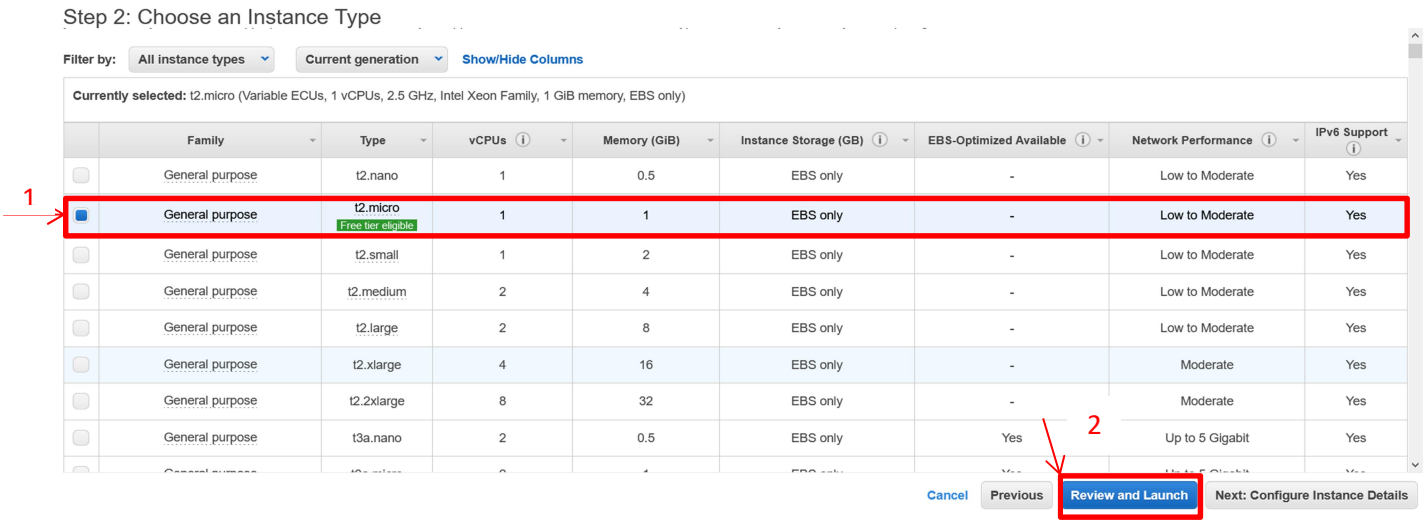
- Go to AWS educate and open AWS Management console. Now, click on Launch a Virtual Machine with EC2.



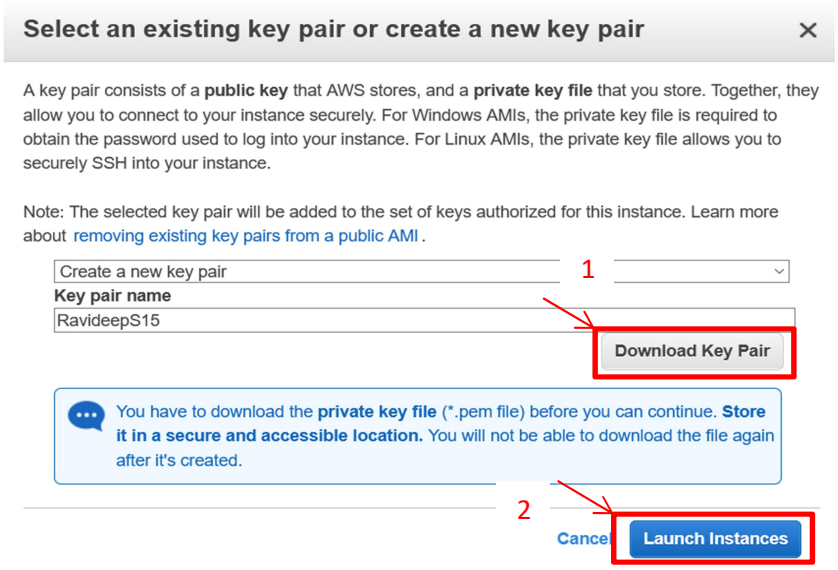
- Now, choose an Amazon machine language(AMI). It is a template that contains the software configuration (operating system, application server, and applications) required to launch instance. Here we choose **Ubuntu Server 18.04 LTS**



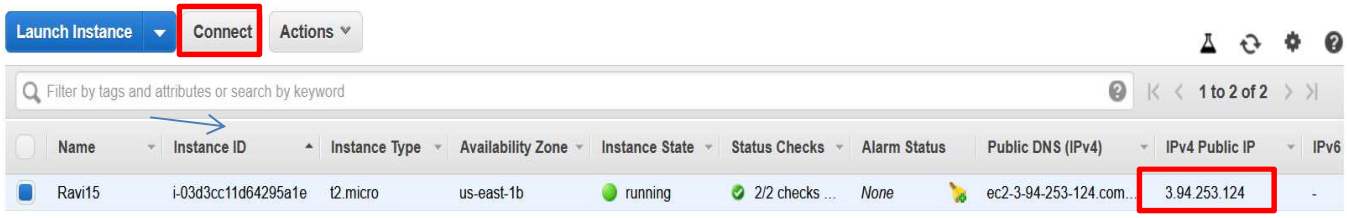
- Choose the Instance type that you require according to your need. Here we select general purpose t2.micro instance which consists of 1 CPU with 1GiB Memory and EBS instance storage. Then click Review and Launch. After reviewing, Launch the instance.



- On launching, create a new key pair. This key pair will allow you to connect to the instance securely. Type a key pair name and download the key pair. Keep it in a safe and accessible location. After this click on Launch Instance

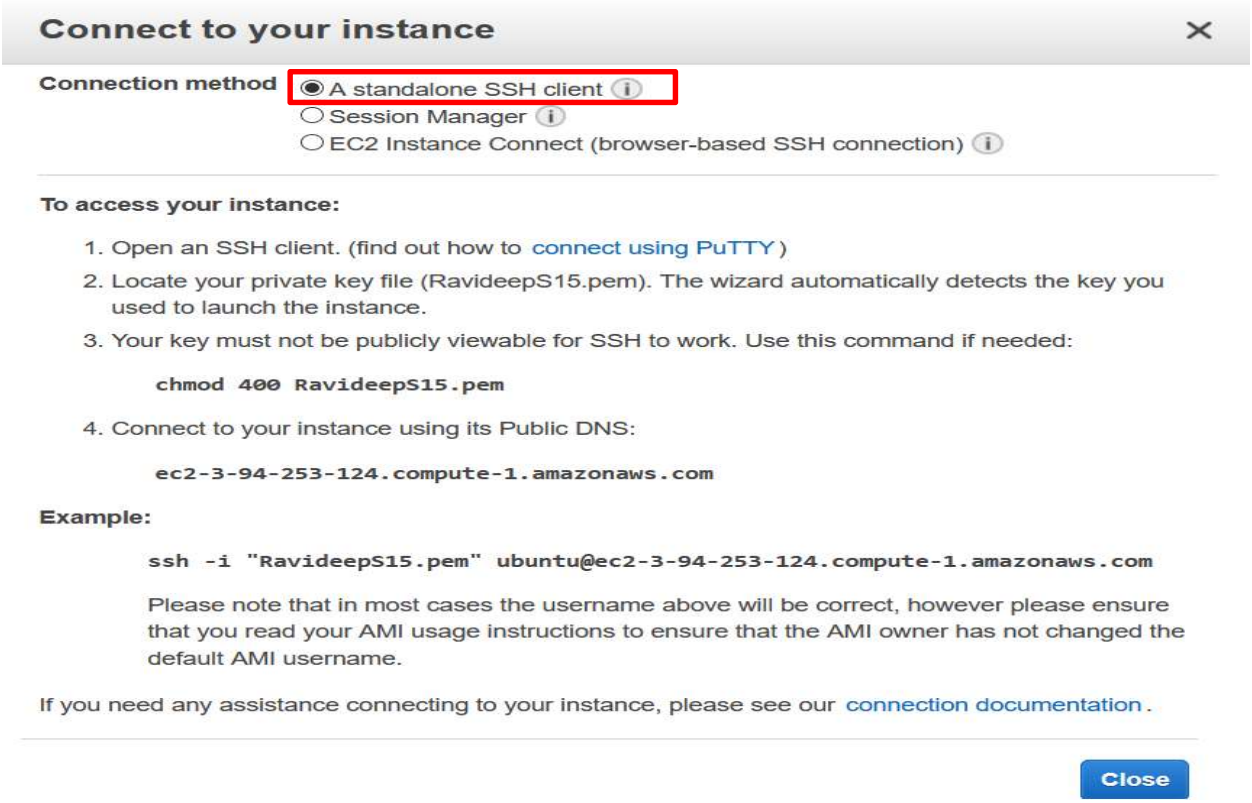


- Click on view instance. You would be able to see the instance that you created as shown below

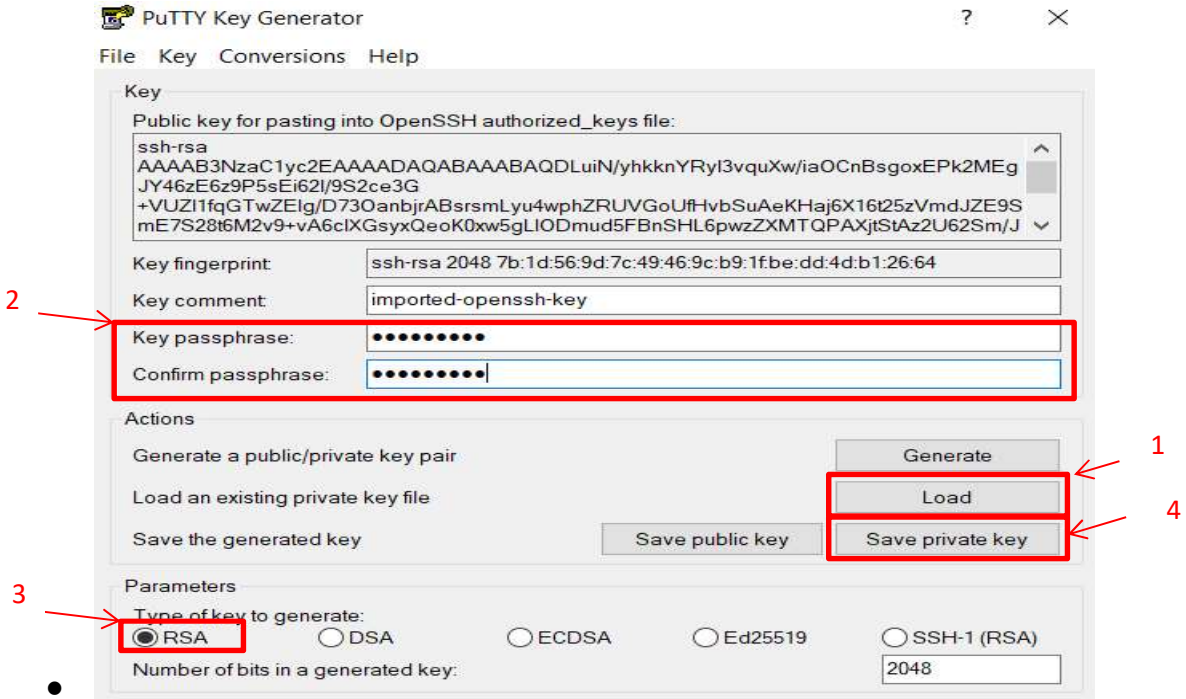


Note the value of IPv4 public IP as it will be used to connect to the instance. After this click on Connect.

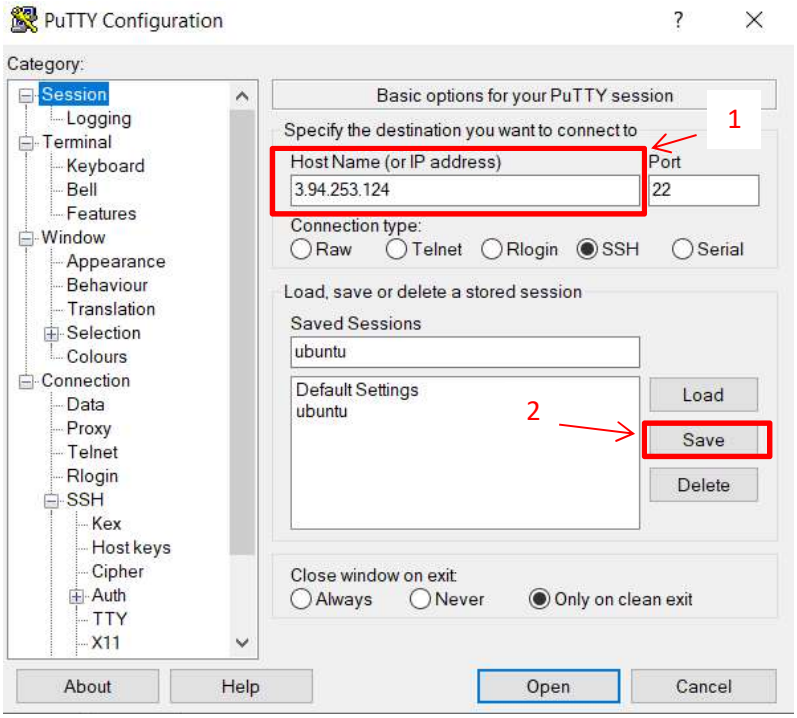
Now choose the connection method as standalone SSH client.



- Now download PuTTY from the internet and install it. After installation open PuTTYgen to generate a key. After opening, select the private key file that you downloaded before and load it. Also put a key passphrase. This is a password that you would need to enter for accessing the instance you created. After this select parameter as RSA and click on save private key and download the key.

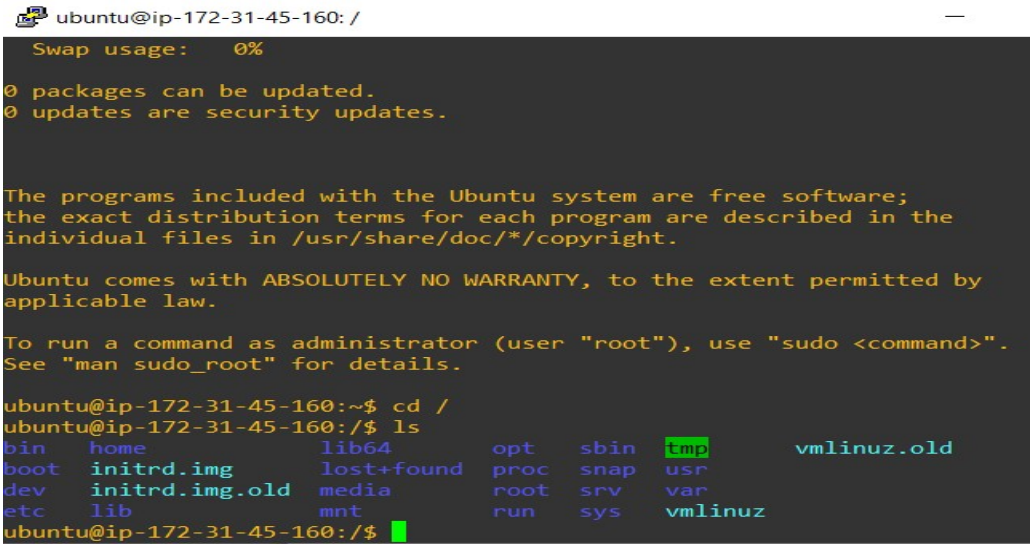


- Now open PuTTY. Under sessions category, put host name as the IPv4 Public address that you noted in the beginning and save the session by giving a name (Ubuntu here).



Under Auth section, browse the putty file that you generated thorugh PuTTYgen in the previous step. Click open. A command window will open. Type the name of the session in login as and password in passphrase section.

After the correct input of username and password, the command line of the Ubuntu operating system will open



Experiment 7

To create an AWS S3 bucket and load an object into it. Create a backup bucket.

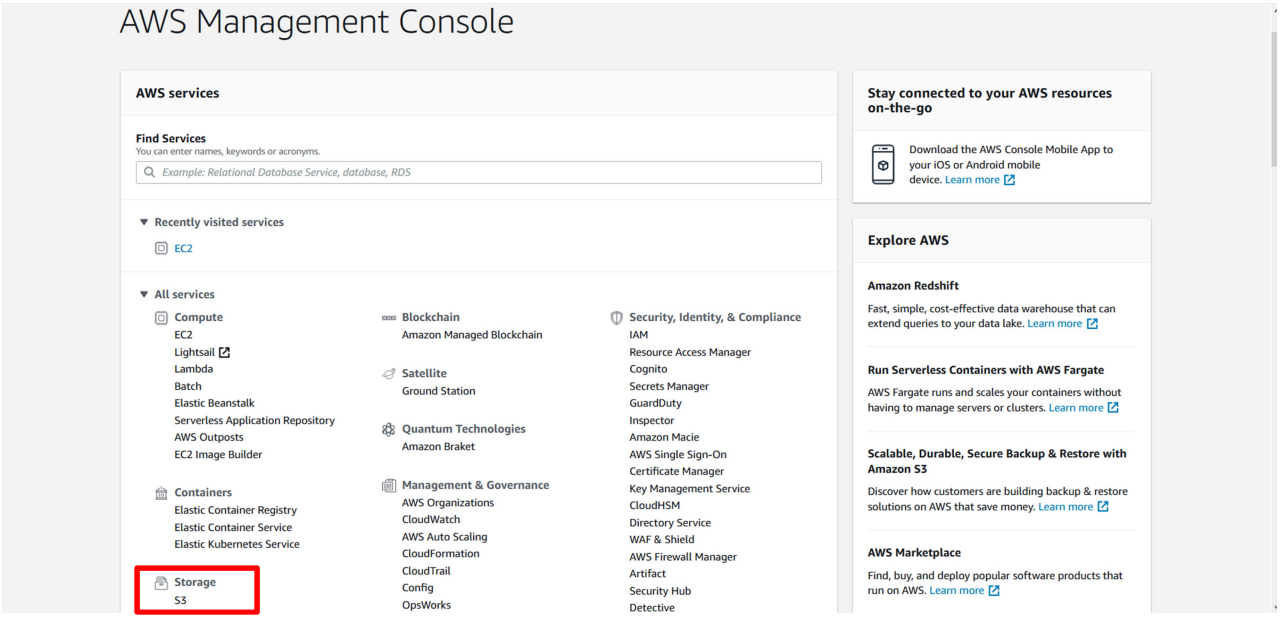
Elaboration of the terms used:-

AWS S3:-

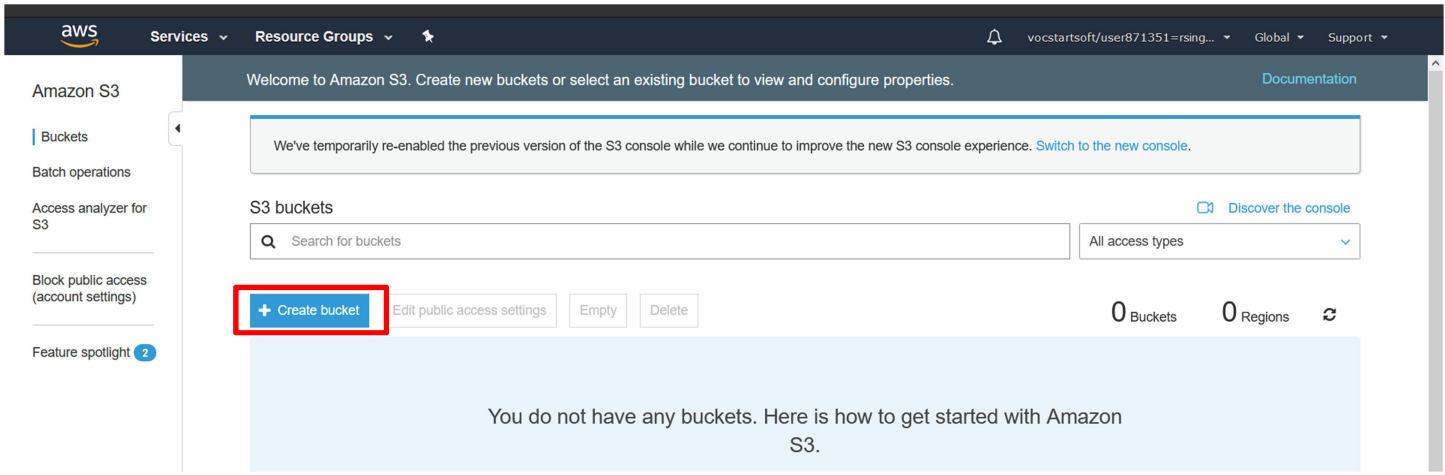
Amazon S3 or Amazon Simple Storage Service is a service offered by Amazon Web Services that provides object storage through a web service interface. It is an object storage service that offers industry-leading scalability, data availability, security, and performance. Amazon S3 uses the same scalable storage infrastructure that Amazon.com uses to run its global e-commerce network. Customers of all sizes and industries can use it to store and protect any amount of data for a range of use cases, such as websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics. Amazon S3 provides easy-to-use management features so you can organize your data and configure finely-tuned access controls to meet your specific business, organizational, and compliance requirements.

Steps to create bucket and load object into it:-

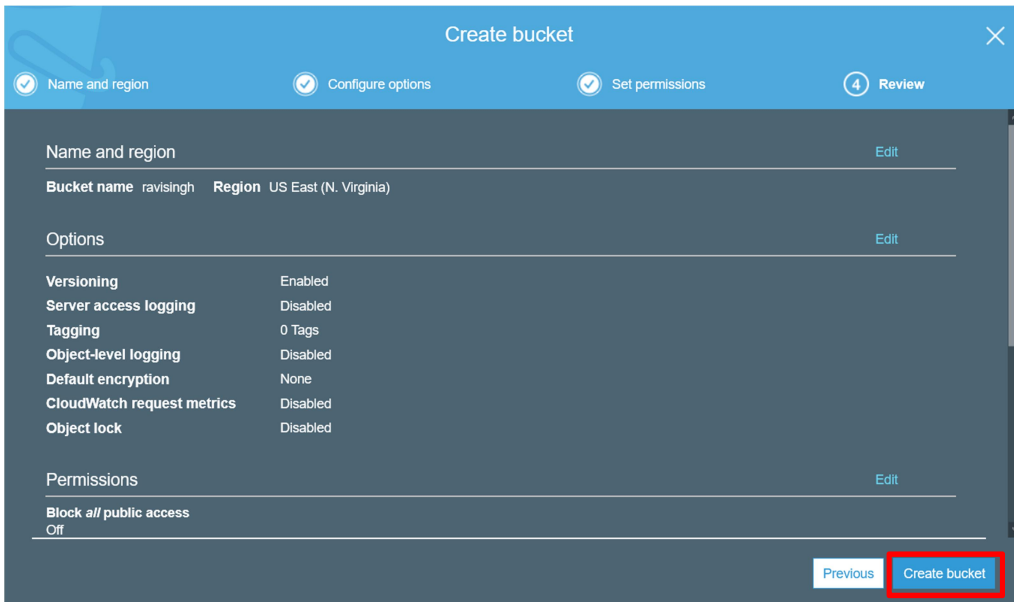
1. Go to AWS educate and open AWS Management console. Now, go to all services and click on S3 under Storage category.



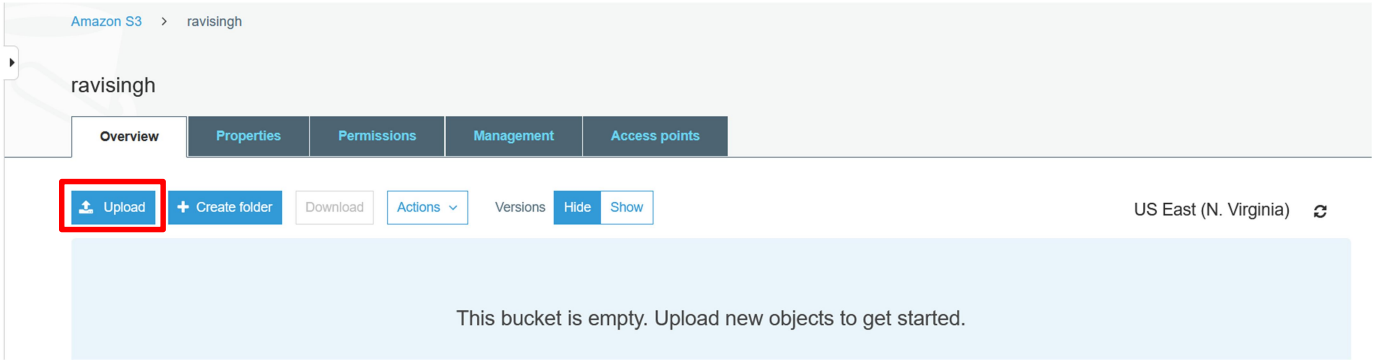
2. Now, click on create bucket option in Amazon S3



3. Now type a unique bucket name and check the necessary permissions required. Review the setting and after that click on Create Bucket.

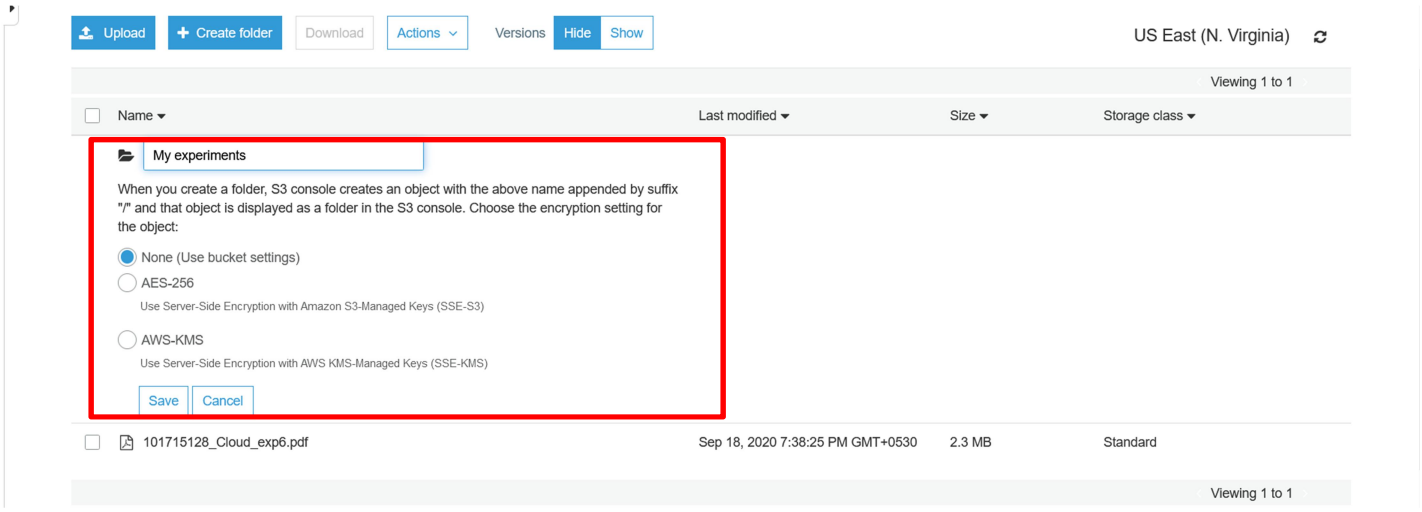


4. You will be able to see the created bucket on the list on buckets. Click on the name of bucket and further click on Upload to upload on object in the bucket



5. Browse the file to be uploaded and click on Add file. Set necessary permissions and properties required. Review the settings and click upload.

Now create a folder in this bucket (eg. My experiments). Use the bucket settings and click save. Now right click on the file and move the file to this newly created folder.



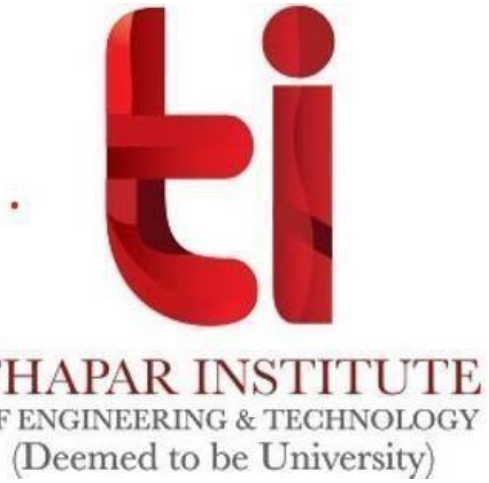
6. Create a backup folder (eg. Backupravisingh) in AWS S3. Go to the previous bucket and go to the file. Right click on the file and select copy. Select the destination as the backup folder and click choose. Your backup will be created.

A Practical Activity Report For
Cloud Computing

Submitted By:

101765006 Smriti(ENC-7)

Submitted To:
Dr. Gitanjali Chandwani



Depatment of Electronics and Communication Engineering

THAPAR INSTITUTE OF ENGINEERING & TECHNOLOGY, (DEEMED TO BE UNIVERSITY), PATIALA, PUNJAB

Experiment 8

Aim: To Build Amazon Virtual Private Cloud (Amazon VPC) Using the VPC Wizard.

Introduction:

Amazon VPC and Its Types:

AWS provides a lot of services; these services are sufficient to run your architecture. The backbone for the security of this architecture is VPC (Virtual Private Cloud). VPC is basically a private cloud in the AWS environment that helps you to use all the services by AWS in your defined private space. You have control over the virtual network and you can also restrict the incoming traffic using security groups.

Overall, VPC helps you to secure your environment and give you a complete authority of incoming traffic. There are two types of VPCs, Default VPC that is by default created by Amazon and Non-Default VPC that is created by you to suffice your security needs.

Route Tables:

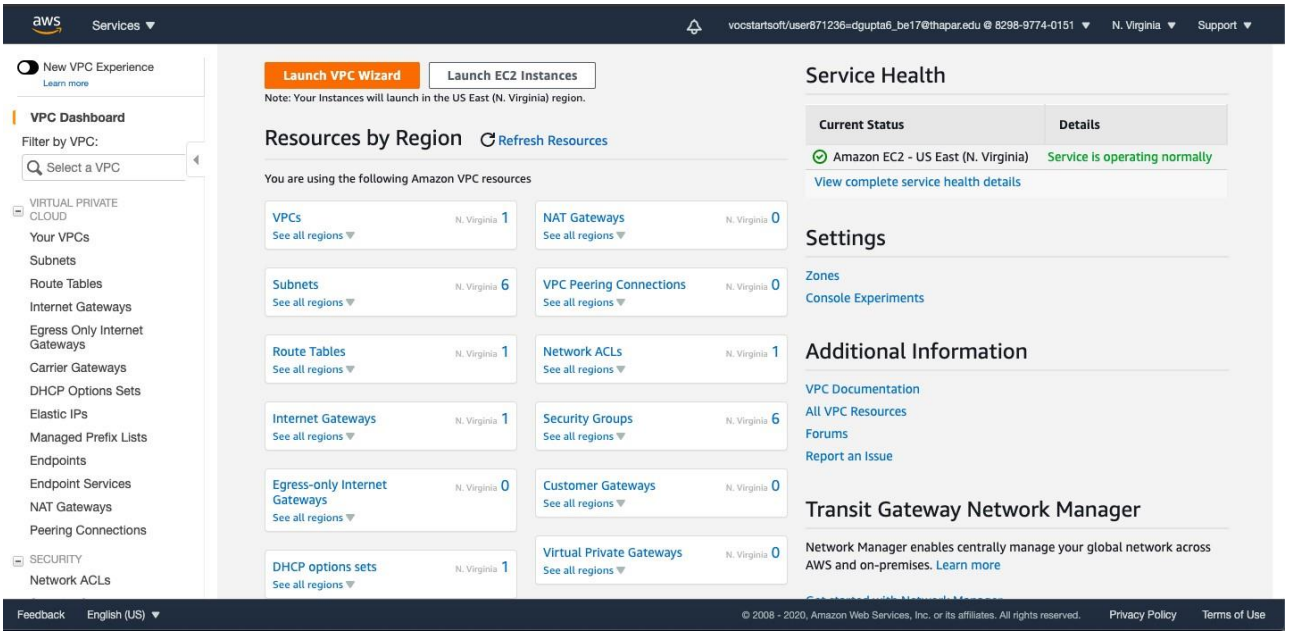
Route table can be understood as a table that contains rules for routing traffic within and outside a subnet. The route table is also used to add Internet Gateway to the subnet. There can be multiple route tables in a VPC.

Internet Gateway:

Internet Gateway is a very important component that allows your instance to connect to the internet. It allows the user to make the subnet pubic by providing a route to the internet. With the help of Internet Gateway, an instance can access the internet and the resources outside instance can access the instance.

STEPS TO BIULD VPC:

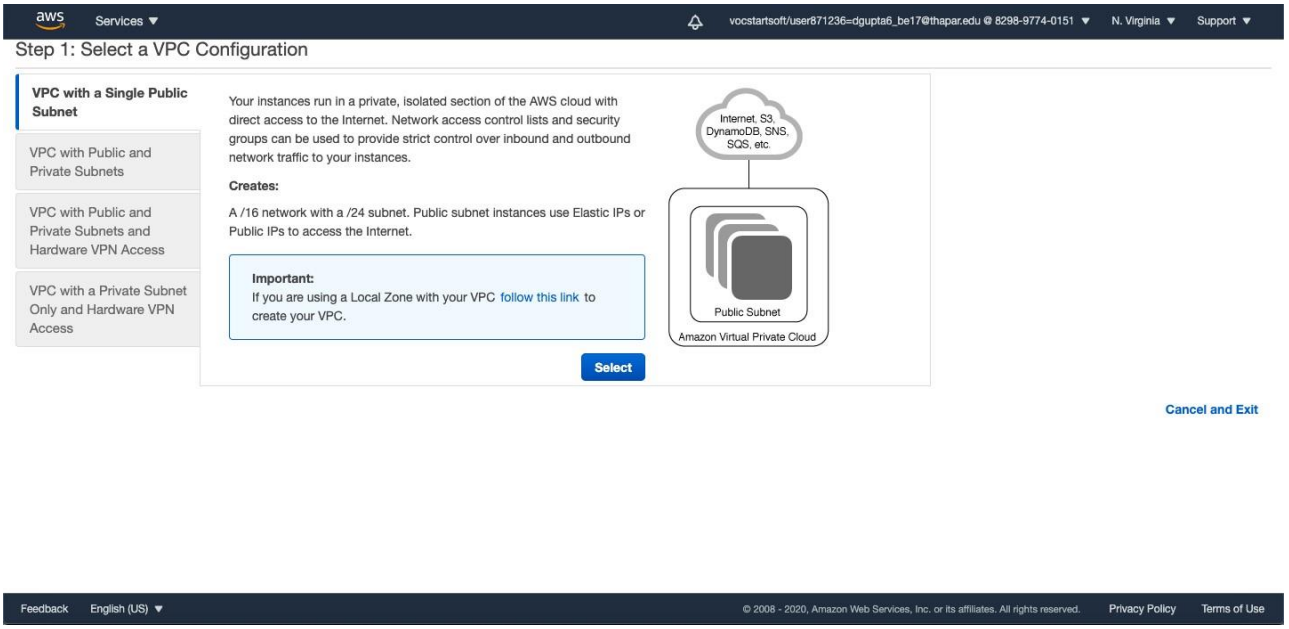
1 Navigate to the VPC Dashboard. Here you will see a "**Launch VPC Wizard**" click on it.



2 Navigate to the VPC Dashboard. Here you will see a "**Launch VPC Wizard**" click on it. This is the "*VPC creation*" wizard. Here you can find 4 different options:

1. VPC with Single Public Subnet, the one we are going to choose.
2. VPC with Public and Private Subnets.
3. VPC with Public and Private Subnets and Hardware VPN Access.
4. VPC with a Private Subnet only and Hardware VPN Access.

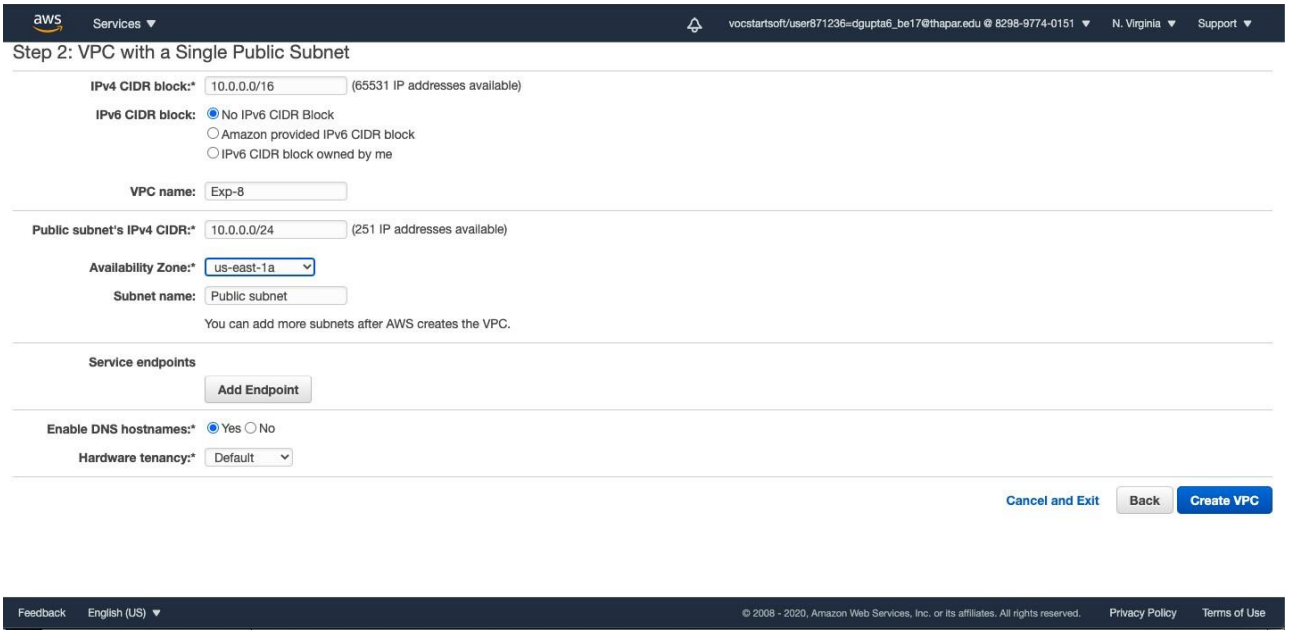
So, let's start by creating a VPC with a single public subnet. Click on "*Select* ".



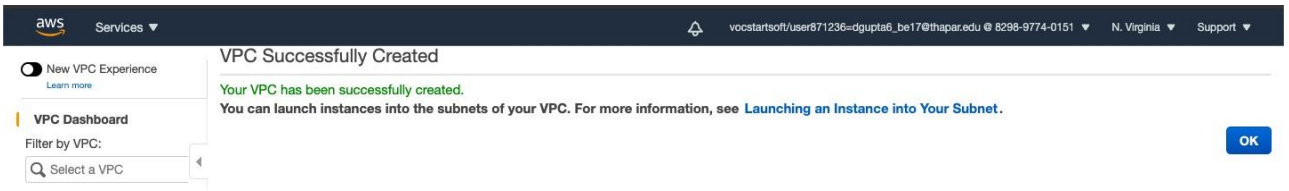
3 Here you will have to mention a few details for creating your VPC.

- The IPv4 CIDR block VPC Name
- Public Subnet's IPv4 CIDR
- Availability Zone where you want your VPC to be created
- Subnet name
- Hardware tenancy

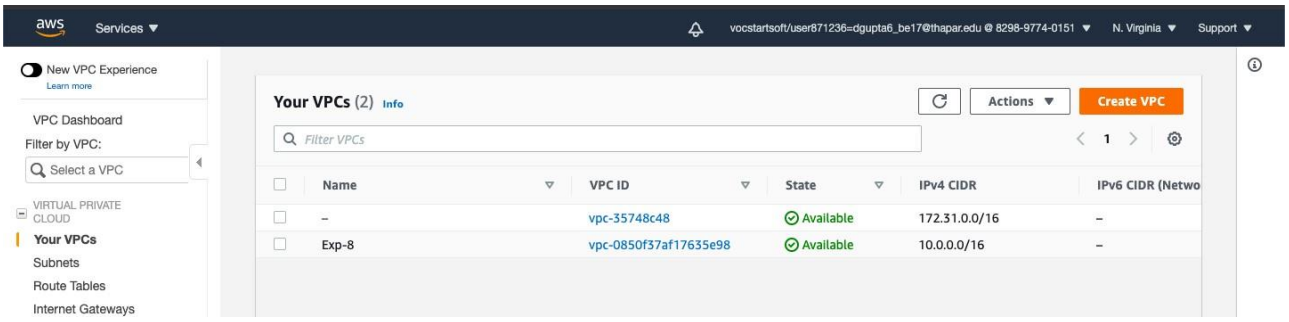
After mentioning all the details, click on "*Create VPC*".



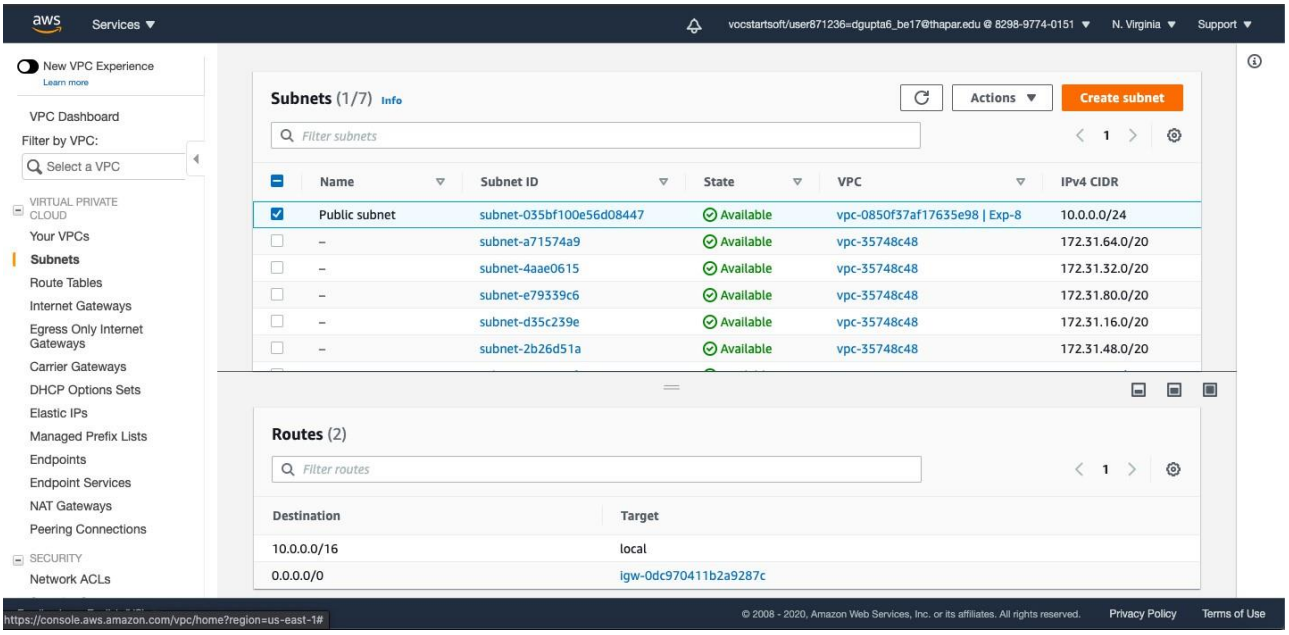
4 You will get a message saying "*Your VPC has been successfully created* ". Click on "*OK* ".



5 In the "Your VPC" section, you can see that there is a new VPC named "*Exp-8*" created.



6 Now let’s verify the public subnet. You can see that a subnet named “Public Subnet” is created. This subnet has a route table attached which consists of local and public access with an Internet Gateway.



Experiment 9

Aim: To Create a DynamoDB table and perform both query and scan searches of the table

Introduction:

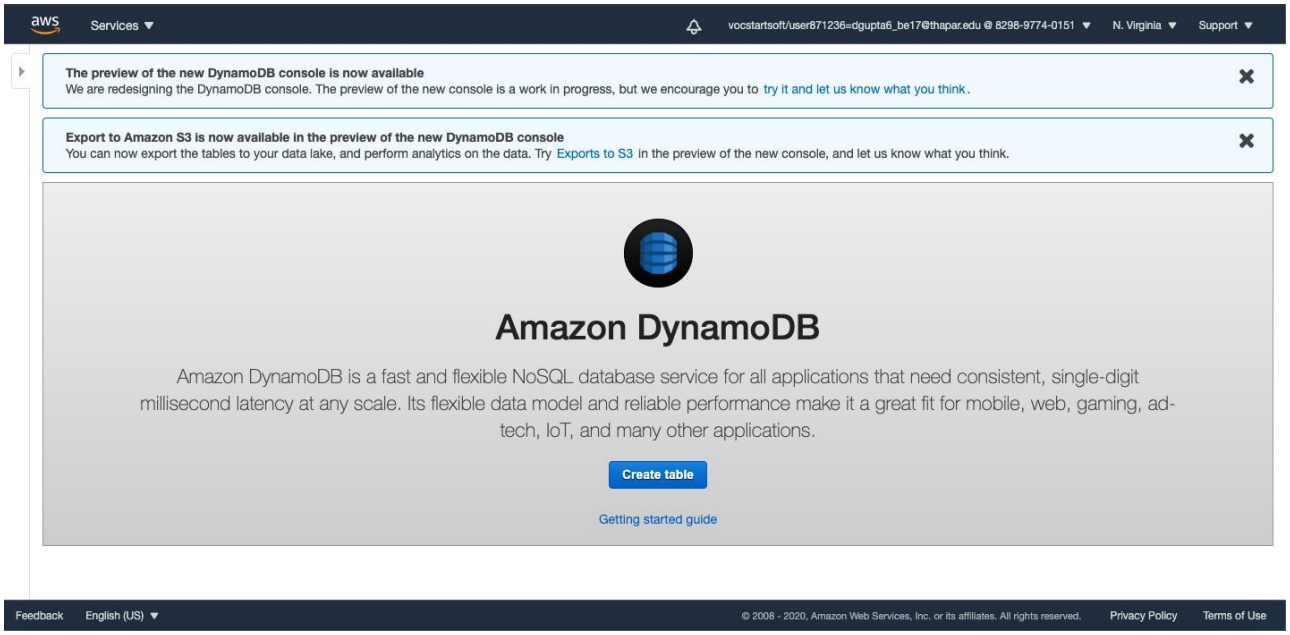
Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. DynamoDB lets you offload the administrative burdens of operating and scaling a distributed database so that you don't have to worry about hardware provisioning, setup and configuration, replication, software patching, or cluster scaling.
DynamoDB also offers encryption at rest, which eliminates the operational burden and complexity involved in protecting sensitive data. For more information, see DynamoDB Encryption at Rest.

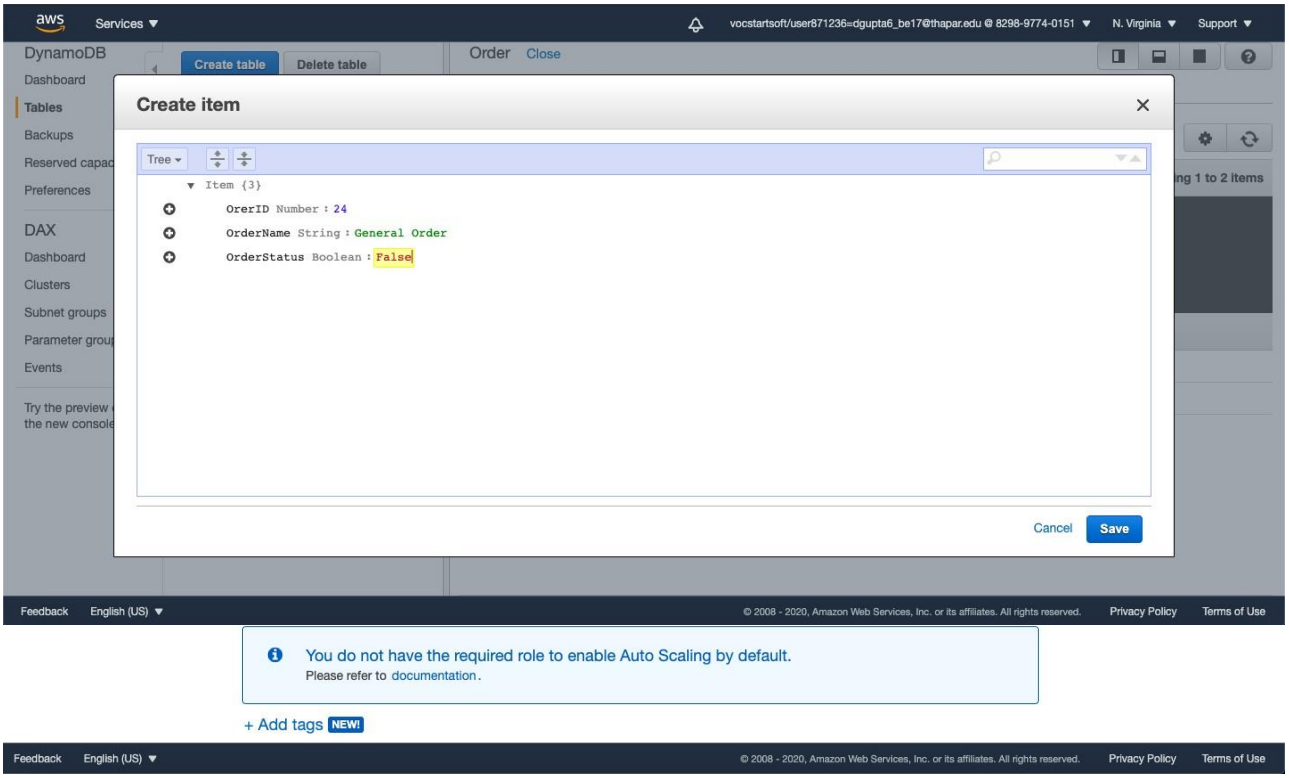
With DynamoDB, you can create database tables that can store and retrieve any amount of data and serve any level of request traffic. You can scale up or scale down your tables' throughput capacity without downtime or performance degradation. You can use the AWS Management Console to monitor resource utilization and performance metrics.

Steps to Create a Dynamo DB Table:

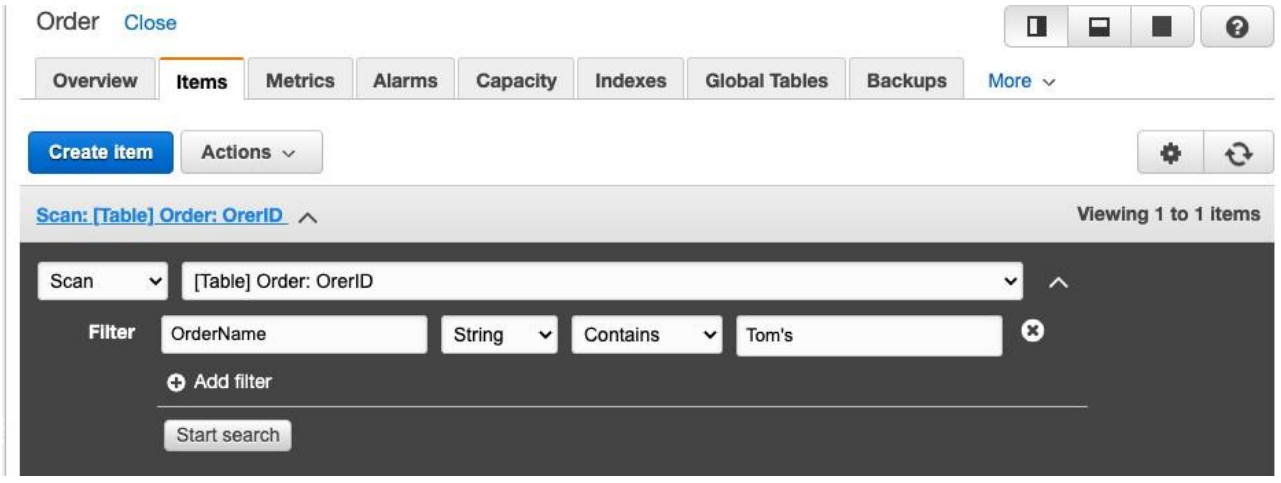
1 Go to Amazon Dynamo DB

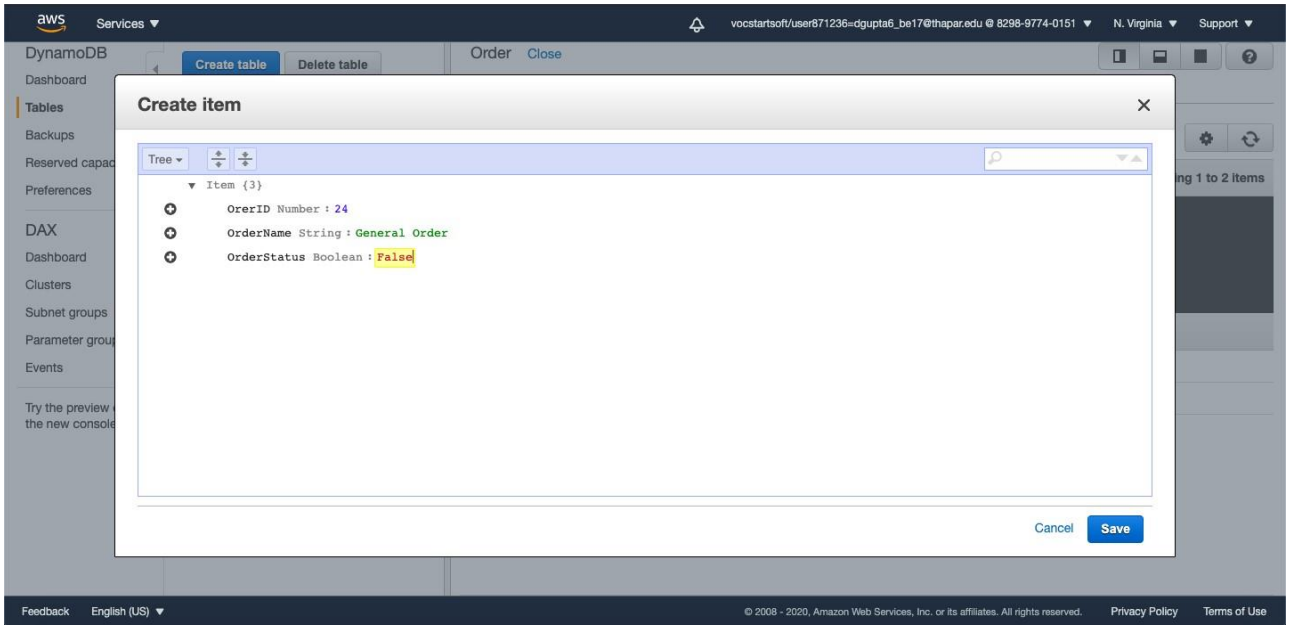
2 Click on CREATE TABLE



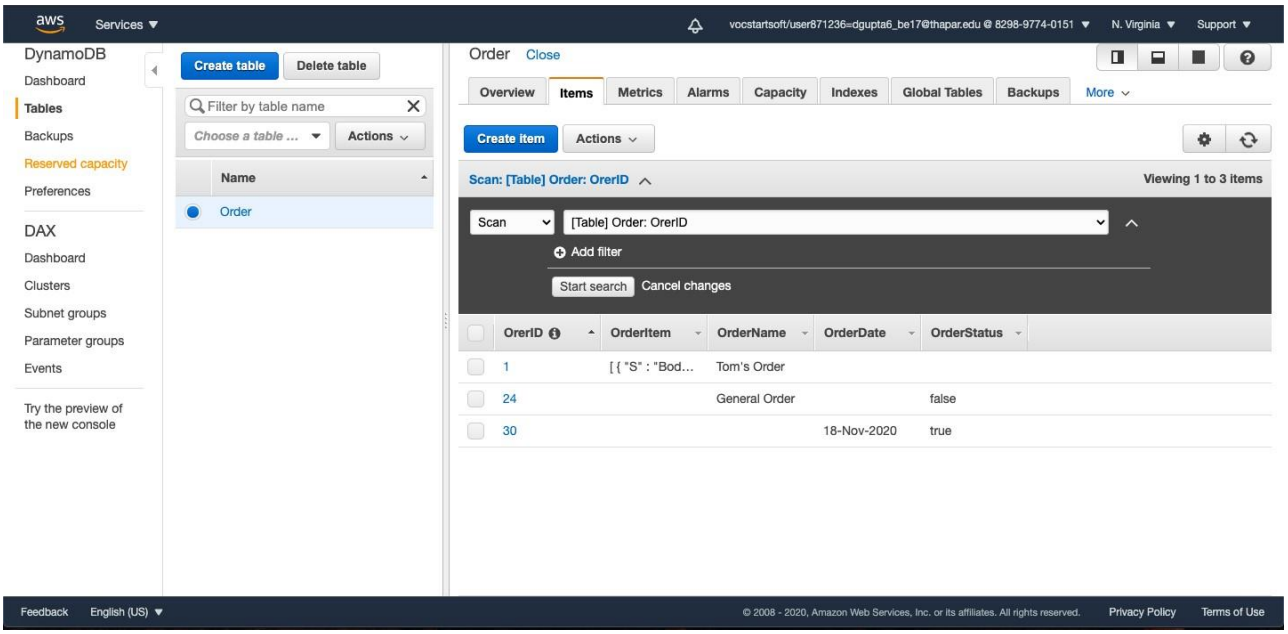


3 Click on CREATE ITEM to add items in the table that we have created.





4 The Table is created and ITEMS are added into the Table.

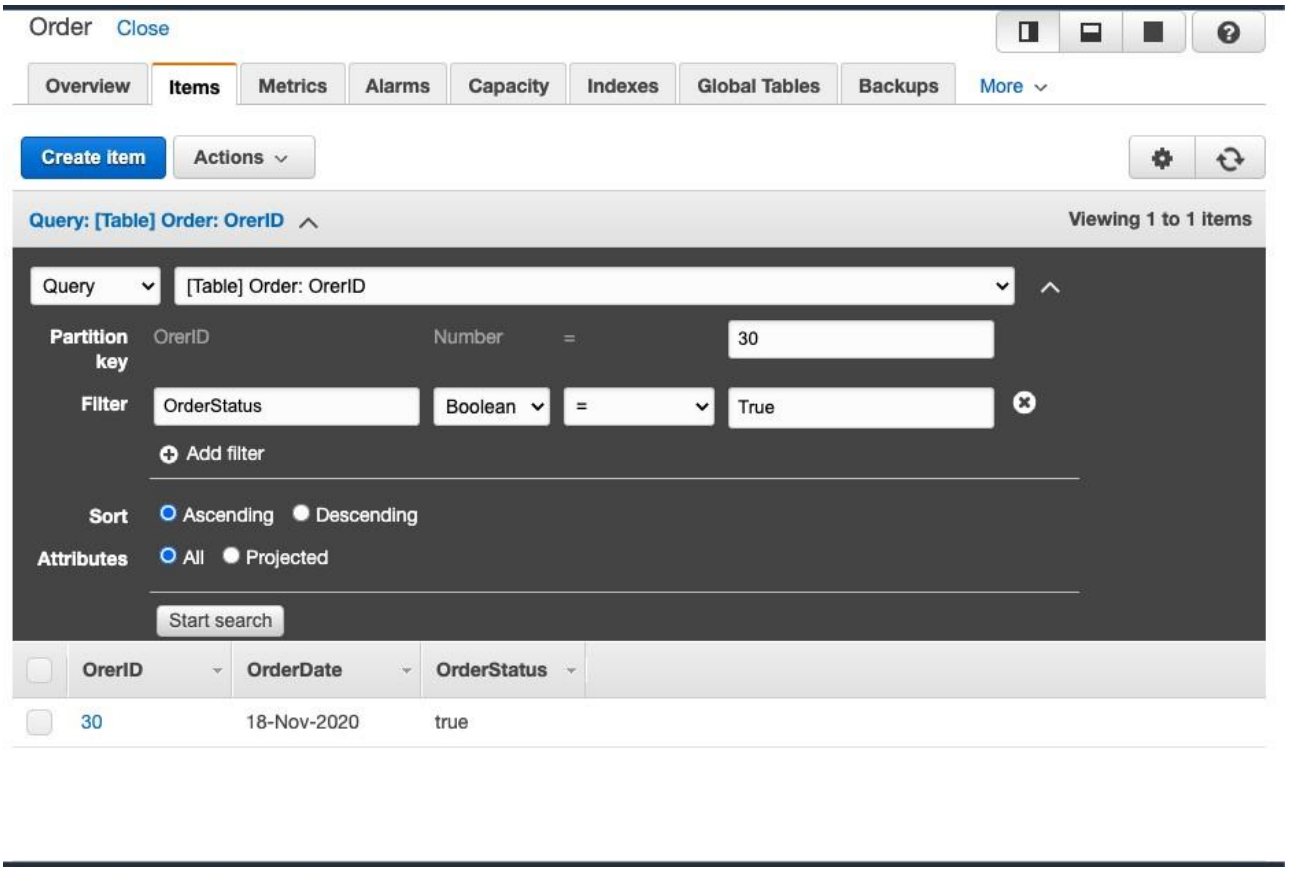


5 Perform SCAN and QUERY operation on the Table we created.

a. SCAN


Order
Close
Overview
Items
Metrics
Alarms
Capacity
Indexes
Global Tables
Backups
More
Create item
Actions
Scan: [Table] Order: OrerID
Viewing 1 to 1 items
Scan
[Table] Order: OrerID
Filter
OrderName
String
Contains
Tom's
Add filter
Start search
OrerID
OrderItem
OrderName
1
[{ "S" : "Bodywash" }, { "S" : "Toilet\npaper\n" }]
Tom's Order

b. QUERY


Order
Close
Overview
Items
Metrics
Alarms
Capacity
Indexes
Global Tables
Backups
More
Create item
Actions
Query: [Table] Order: OrerID
Viewing 1 to 1 items
Query
[Table] Order: OrerID
Partition key
OrerID
Number
=
30
Filter
OrderStatus
Boolean
=
True
Add filter
Sort
Ascending
Descending
Attributes
All
Projected
Start search
OrerID
OrderDate
OrderStatus
30
18-Nov-2020
true