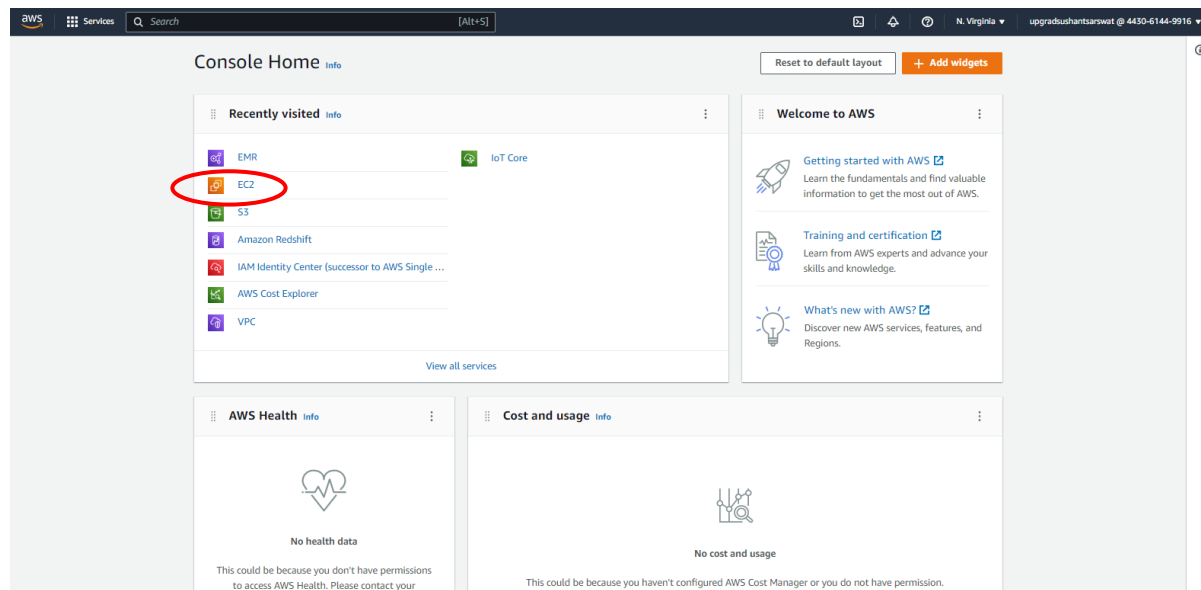


CAPSTONE PROJECT INSTANT HEALTH ALERT SYSTEM SUBMITTED BY : Sushant Sarswat (C-38)

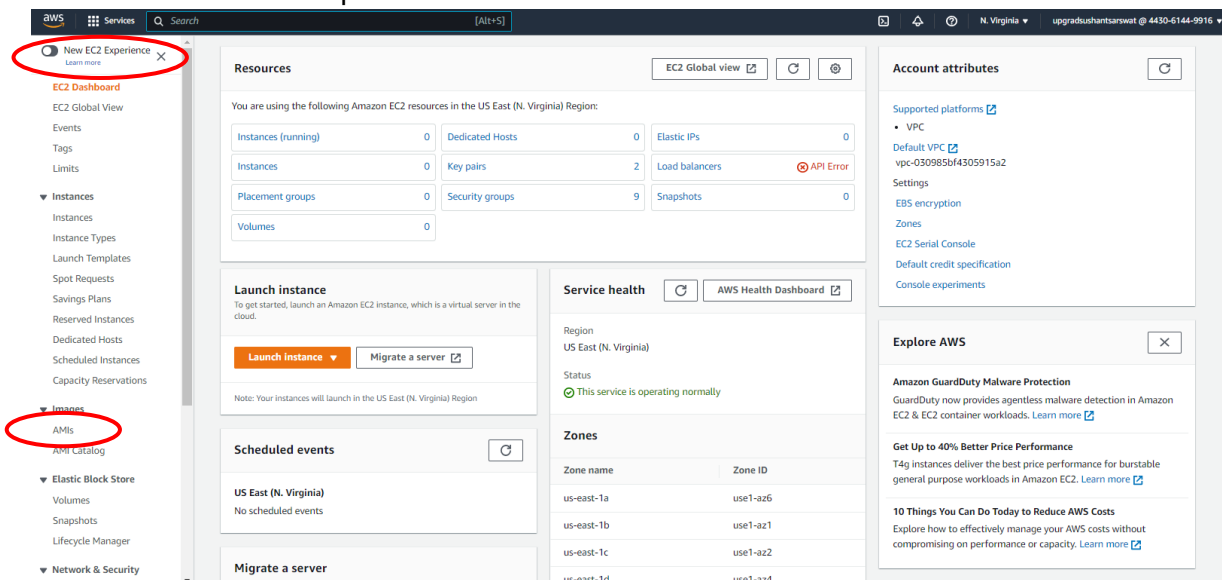
CREATION AND CONFIGURATION OF KAFKA CLUSTER.

The foremost requirement of the project is to create and configure a Kafka cluster with help of EC2 instance. The following steps illustrate the process of creating the Kafka cluster:-

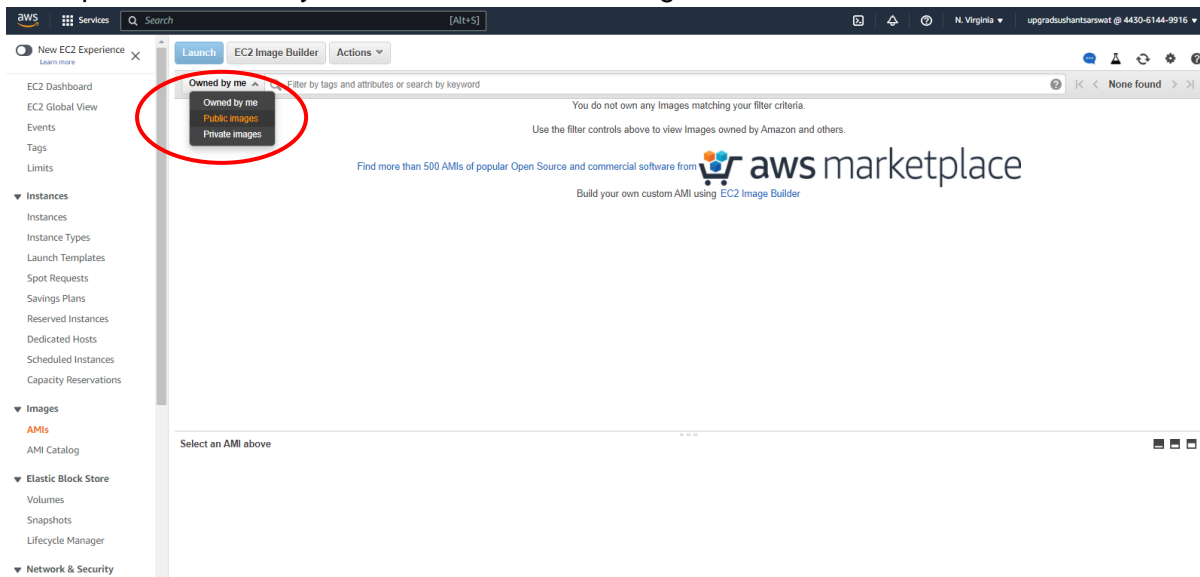
1. Select EC2 instance from AWS services



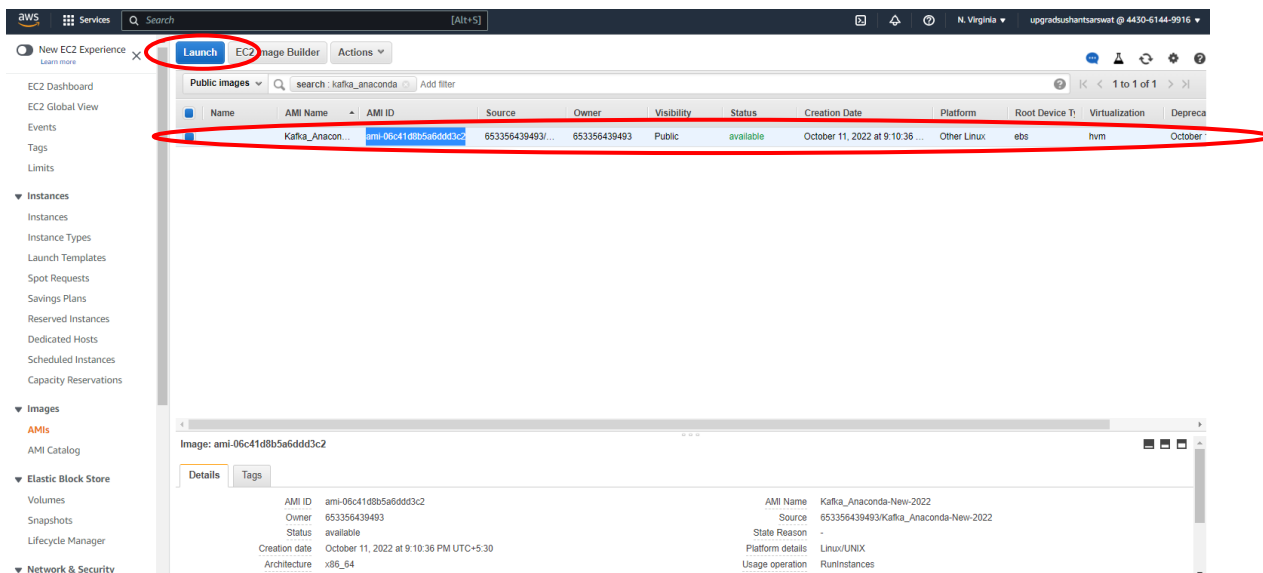
2. Turn off 'New EC2 Experience' and click on AMIs



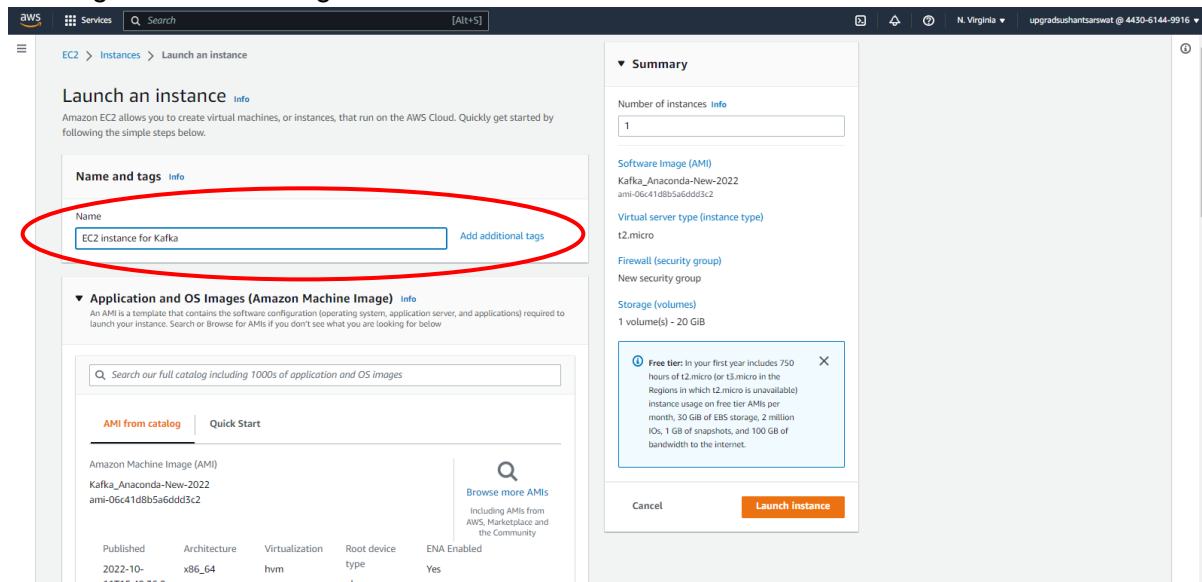
3. Dropdown 'Owned by Me' and select 'Public images'



4. Select AMI 'ami-06c41d8b5a6ddd3c2' and click 'Launch'



5. Assign Names and tags - 'EC2 instance for Kafka'



Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name
EC2 instance for Kafka [Add additional tags](#)

Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

AMI from catalog **Quick Start**

Amazon Machine Image (AMI)

Kafka_Anaconda-New-2022
ami-06c41db5a6dd3c2

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Published	Architecture	Virtualization	Root device	ENA Enabled
2022-10-11T15:40:36.0	x86_64	hvm	type	Yes

Summary

Number of instances [Info](#)
1

Software Image (AMI)
Kafka_Anaconda-New-2022
ami-06c41db5a6dd3c2

Virtual server type (instance type)
t2.micro

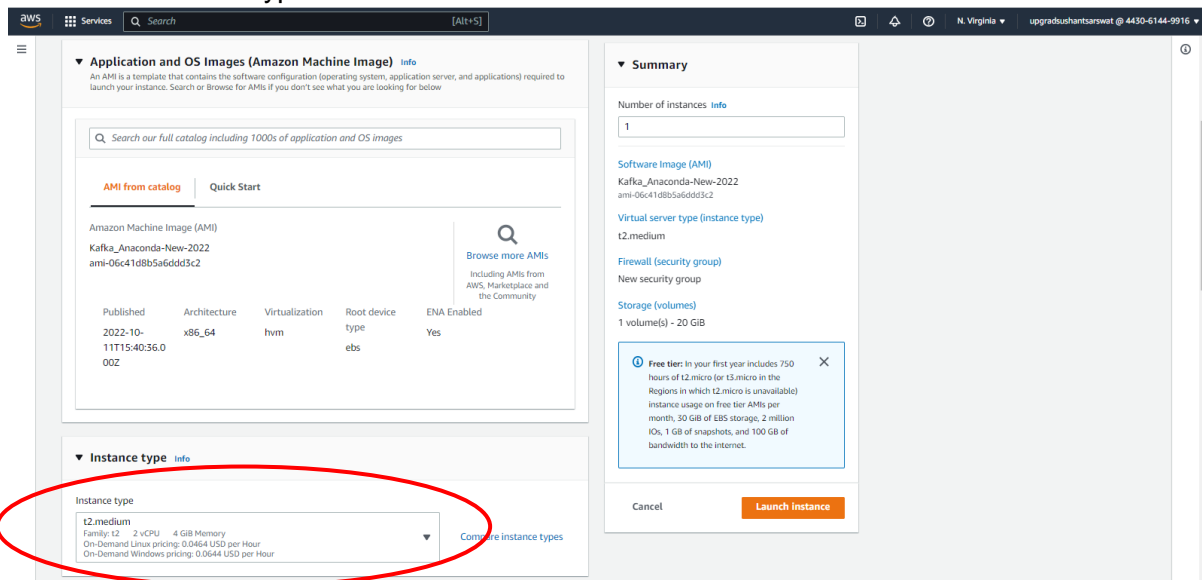
Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 20 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.

[Cancel](#) [Launch instance](#)

6. Select Instance Type 't2.medium'



Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

AMI from catalog **Quick Start**

Amazon Machine Image (AMI)

Kafka_Anaconda-New-2022
ami-06c41db5a6dd3c2

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Published	Architecture	Virtualization	Root device	ENA Enabled
2022-10-11T15:40:36.00Z	x86_64	hvm	type	Yes

Instance type [Info](#)

Instance type
t2.medium
Family: t2 2 vCPU 4 GiB Memory
On-Demand Linux pricing: 0.0464 USD per Hour
On-Demand Windows pricing: 0.0644 USD per Hour [Compare instance types](#)

Summary

Number of instances [Info](#)
1

Software Image (AMI)
Kafka_Anaconda-New-2022
ami-06c41db5a6dd3c2

Virtual server type (instance type)
t2.medium

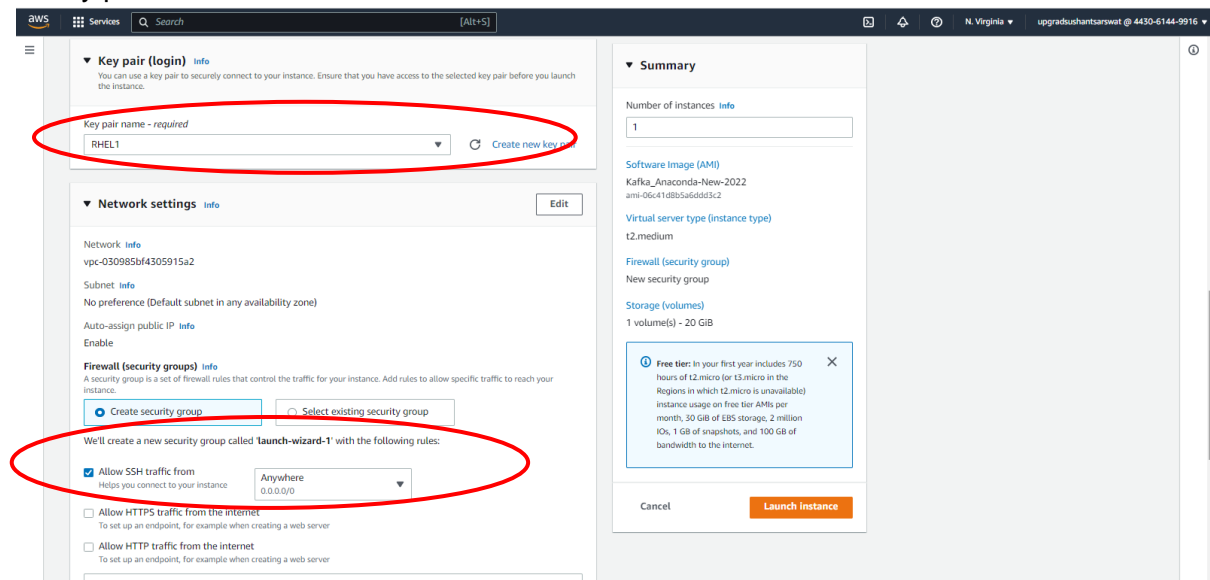
Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 20 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.

[Cancel](#) [Launch instance](#)

7. Key pair selection and SSH



Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key before you launch the instance.

Key pair name - required

RHEL1 [Create new key pair](#)

Network settings Info

[Edit](#)

Network Info

vpc-030985bf4305915a2

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group ☐ Select existing security group

We'll create a new security group called **launch-wizard-1** with the following rules:

☒ Allow SSH traffic from [Helps you connect to your instance](#) Anywhere 0.0.0.0/0

☐ Allow HTTPS traffic from the internet To set up an endpoint, for example when creating a web server

☐ Allow HTTP traffic from the internet To set up an endpoint, for example when creating a web server

Summary

Number of instances Info

1

Software Image (AMI)

Kafka_Anaconda-New-2022
ami-06c41db5a6d6d5c2

Virtual server type (instance type)

t2.medium

Firewall (security group)

New security group

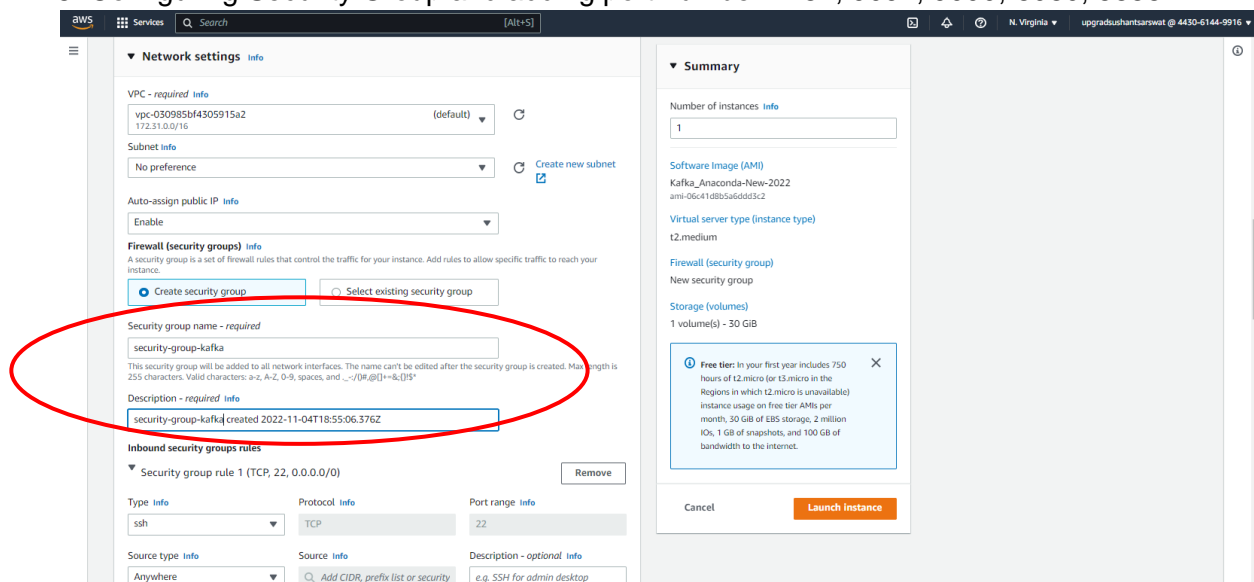
Storage (volumes)

1 volume(s) - 20 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.

[Cancel](#) [Launch instance](#)

8. Configuring Security Group and adding port number 2181, 9092, 9000, 8080, 8888



Network settings Info

[Info](#)

VPC - required Info

vpc-030985bf4305915a2 (default) [Create new VPC](#)

Subnet Info

No preference [Create new subnet](#)

Auto-assign public IP Info

Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group ☐ Select existing security group

Security group name - required

security-group-kafka

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-./@#&()*+=!~:;{}[]`^"@'

Description - required Info

security-group-kafka created 2022-11-04T18:55:06.376Z

Inbound security groups rules

Security group rule 1 (TCP, 22, 0.0.0.0/0) [Remove](#)

Type Info

ssh

Protocol Info

TCP

Port range Info

22

Source type Info

Anywhere

Source Info

[Add CIDR, prefix list or security](#)

Description - optional Info

e.g. SSH for admin desktop

Summary

Number of instances Info

1

Software Image (AMI)

Kafka_Anaconda-New-2022
ami-06c41db5a6d6d5c2

Virtual server type (instance type)

t2.medium

Firewall (security group)

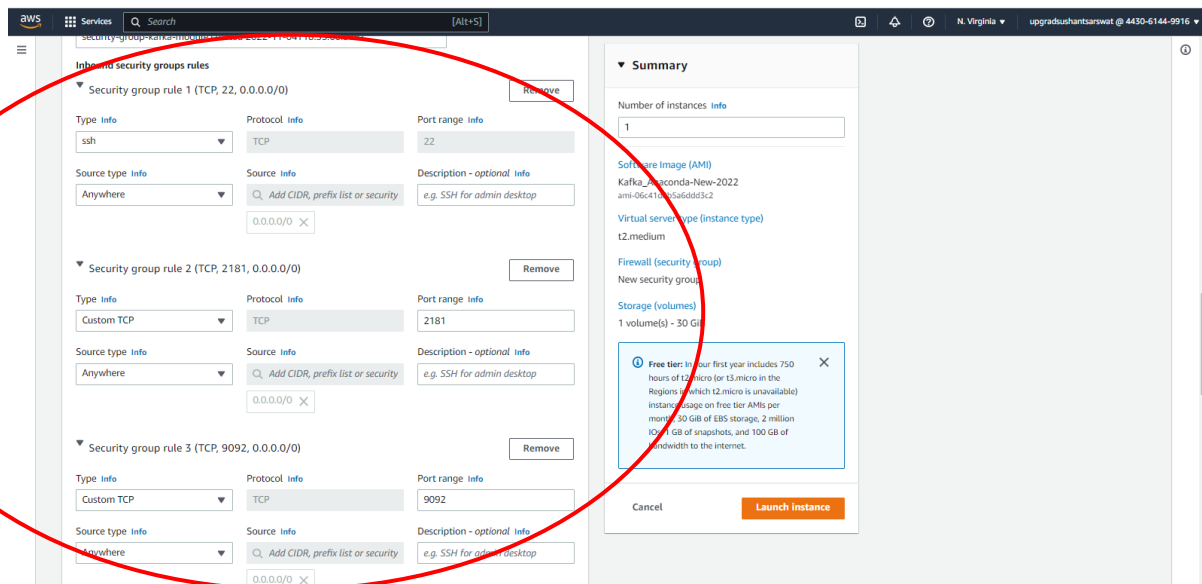
New security group

Storage (volumes)

1 volume(s) - 30 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.

[Cancel](#) [Launch instance](#)



Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type: **ssh** Protocol: **TCP** Port range: **22**

Source type: **Anywhere** Source: **0.0.0.0/0** Description - optional: **e.g. SSH for admin desktop**

Security group rule 2 (TCP, 2181, 0.0.0.0/0)

Type: **Custom TCP** Protocol: **TCP** Port range: **2181**

Source type: **Anywhere** Source: **0.0.0.0/0** Description - optional: **e.g. SSH for admin desktop**

Security group rule 3 (TCP, 9092, 0.0.0.0/0)

Type: **Custom TCP** Protocol: **TCP** Port range: **9092**

Source type: **Anywhere** Source: **0.0.0.0/0** Description - optional: **e.g. SSH for admin desktop**

Summary

Number of instances: **1**

Software Image (AMI): **Kafka_Anaconda-New-2022**

Virtual server type (instance type): **t2.medium**

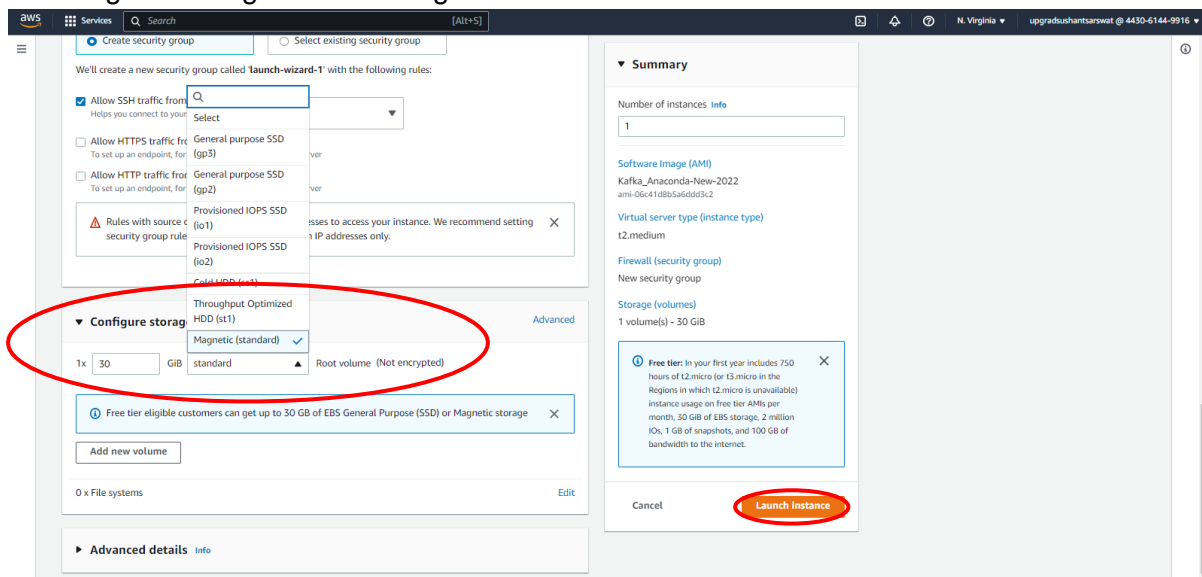
Firewall (security group): **New security group**

Storage (volumes): **1 volume(s) - 30 GiB**

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.

Launch Instance

9. Configure Storage - 30 GB Magnetic Standard and 'Launch instance'



Create security group

We'll create a new security group called **launch-wizard-1** with the following rules:

☒ Allow SSH traffic from **Anywhere**

☐ Allow HTTPS traffic from **Anywhere**

☐ Allow HTTP traffic from **Anywhere**

Rules with source security group rule

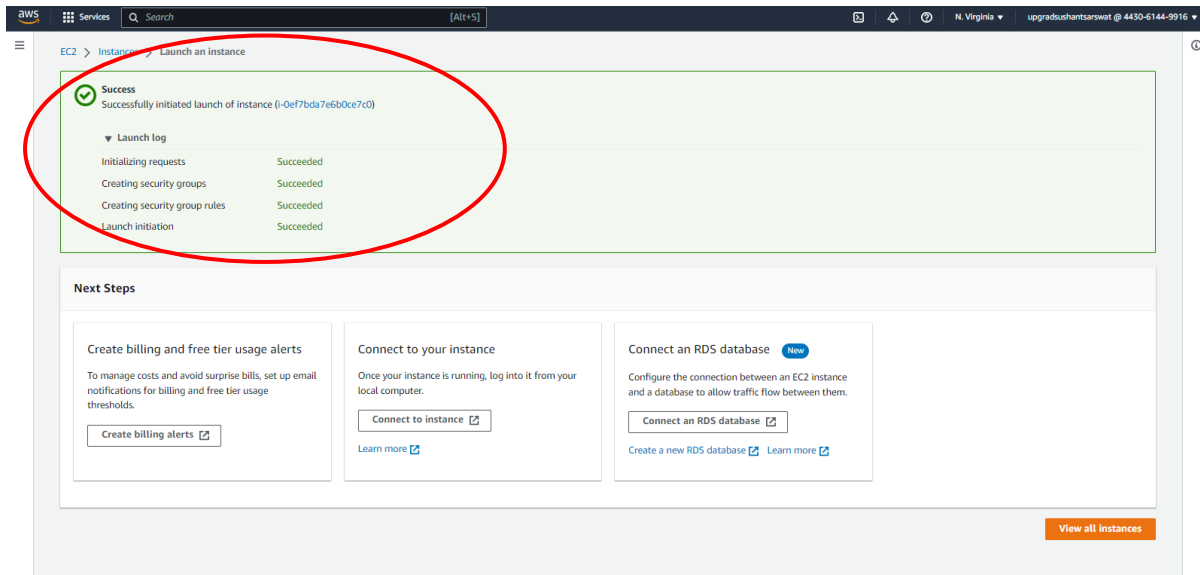
Configure storage

1x **30** GiB **Magnetic (standard)** Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GiB of EBS General Purpose (SSD) or Magnetic storage

Launch Instance

10. Successful Launch of instance - View all instances



Success
Successfully initiated launch of instance (i-0ef7bda7e6b0ce7c0)

Launch log

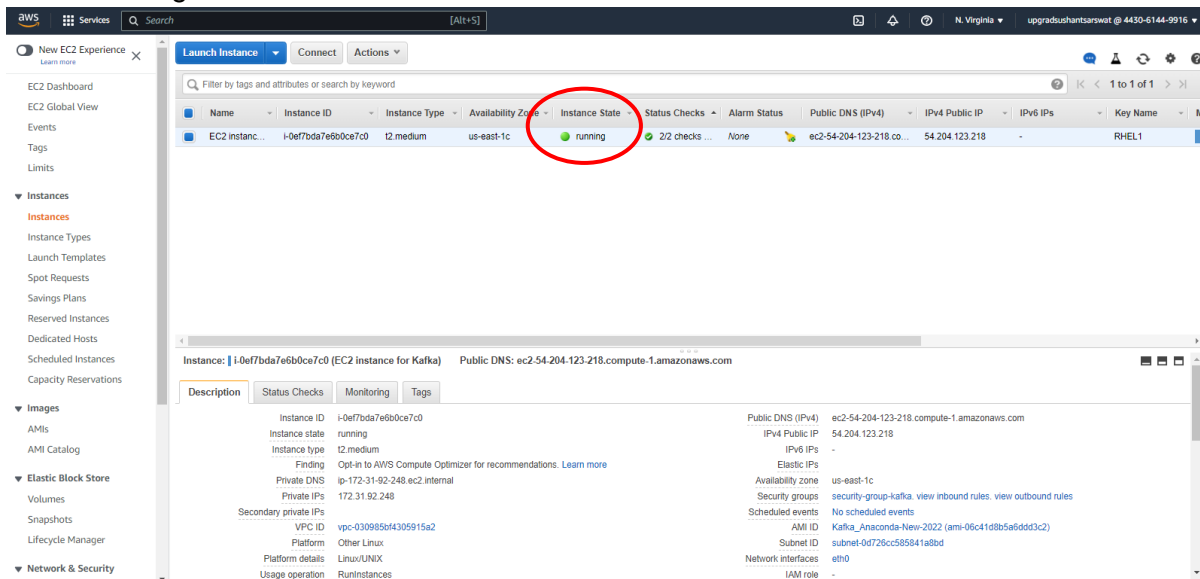
Event	Status
Initializing requests	Succeeded
Creating security groups	Succeeded
Creating security group rules	Succeeded
Launch initiation	Succeeded

Next Steps

- Create billing and free tier usage alerts
- Connect to your instance
- Connect an RDS database

[View all instances](#)

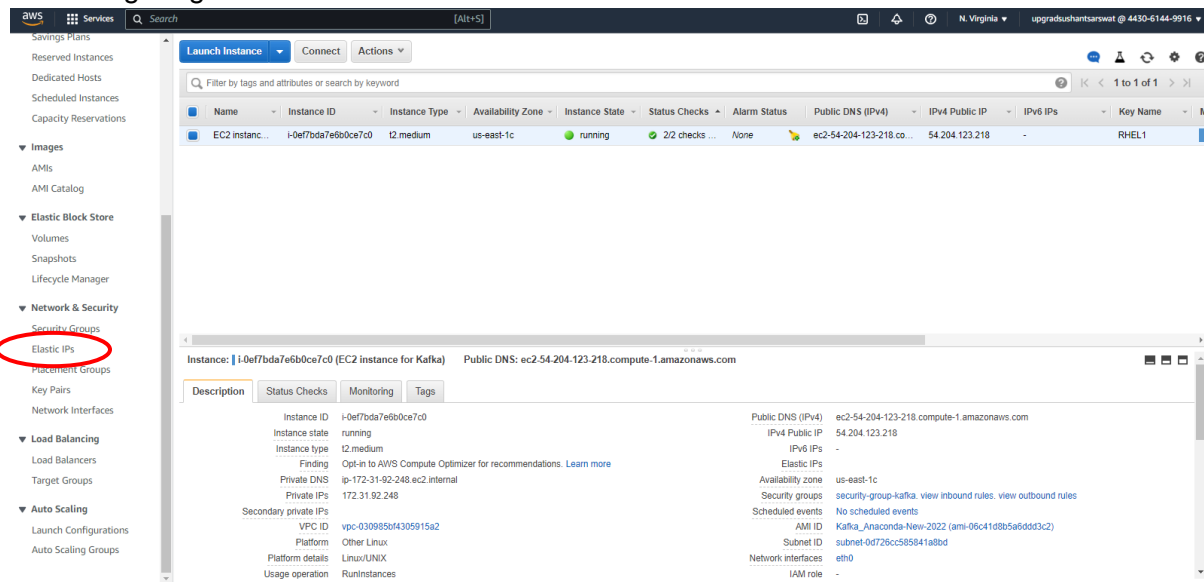
11. Monitoring 'Instance State'



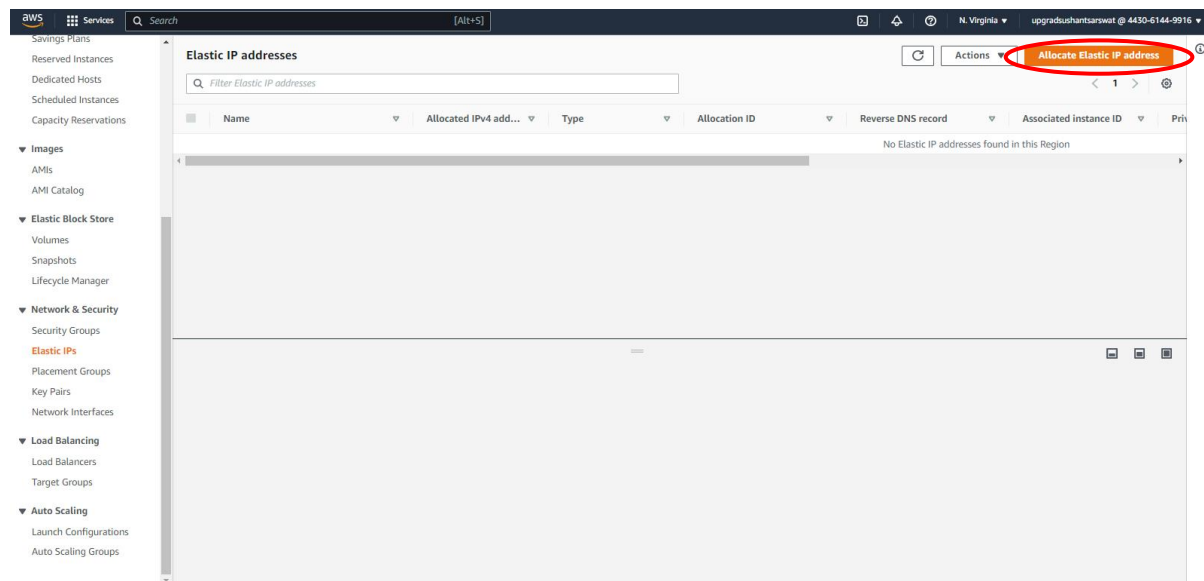
Instance: i-0ef7bda7e6b0ce7c0 (EC2 instance for Kafka) Public DNS: ec2-54-204-123-218.compute-1.amazonaws.com

Description	Status Checks	Monitoring	Tags
Instance ID	i-0ef7bda7e6b0ce7c0		
Instance state	running		
Instance type	t2.medium		
Finding	Opt-in to AWS Compute Optimizer for recommendations. Learn more		
Private DNS	ip-172-31-92-248.ec2.internal		
Private IPs	172.31.92.248		
Secondary private IPs			
VPC ID	vpc-0309856f4305915a2		
Platform	Other Linux		
Platform details	Linux/UNIX		
Usage operation	RunInstances		

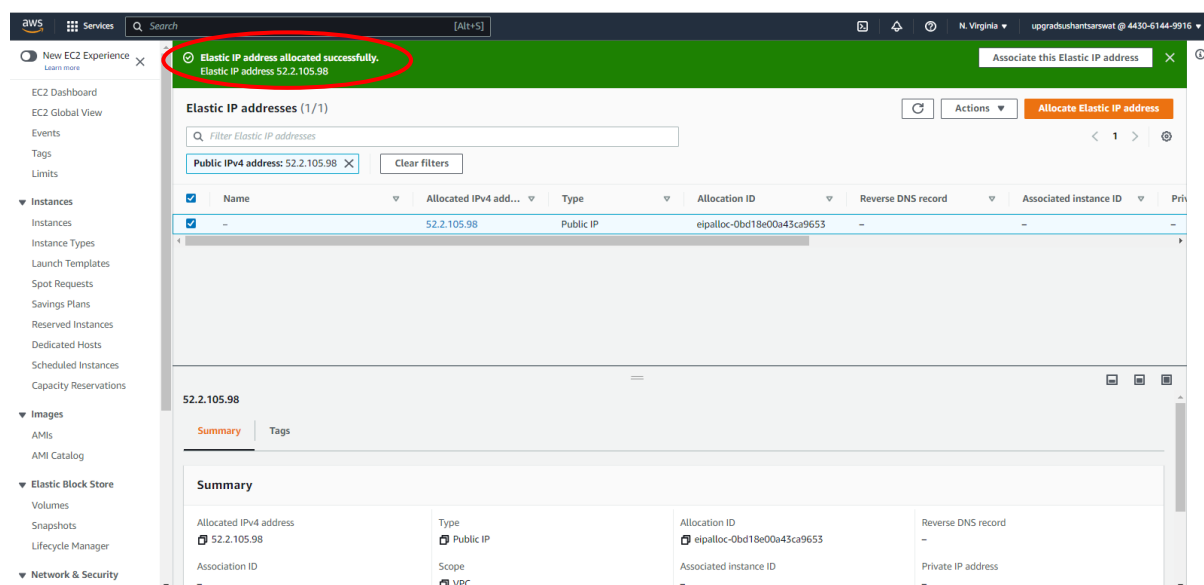
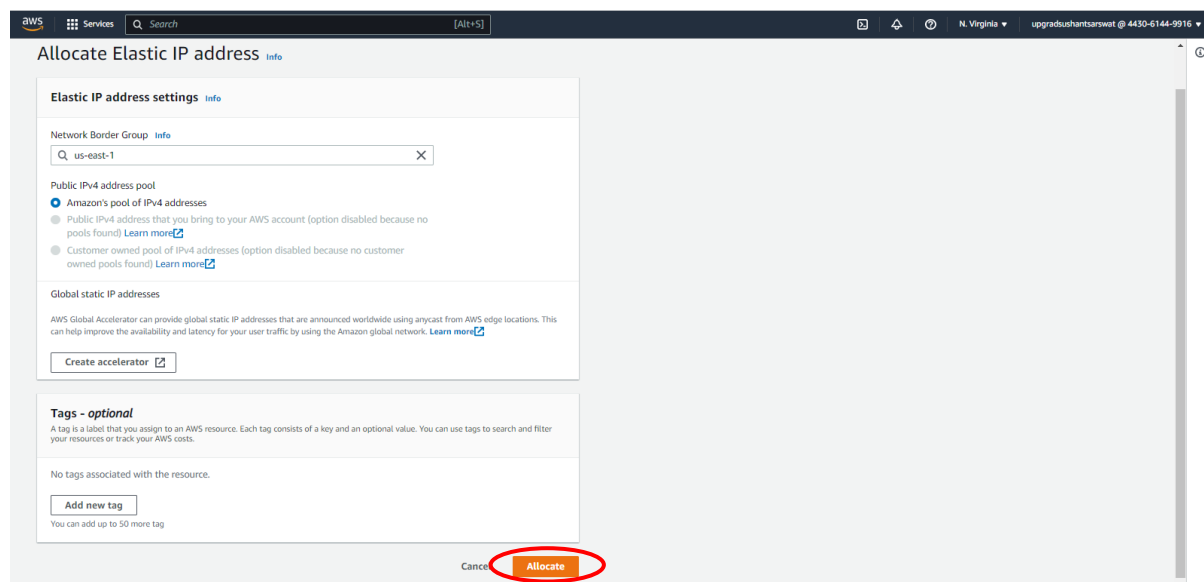
12. Configuring Elastic IP



The screenshot shows the AWS Management Console interface. On the left sidebar, under the 'Network & Security' section, 'Elastic IPs' is highlighted with a red circle. The main content area displays a table of EC2 instances. One instance, 'EC2 instance...', is selected. Below the table, the 'Description' tab is active, showing details for the instance 'i-0e77bda7e6b0ce7c0'. The 'Public DNS (IPv4)' field shows 'ec2-54-204-123-218.compute-1.amazonaws.com'. The 'Elastic IPs' field shows 'us-east-1c'. The 'Availability zone' is 'us-east-1c'. The 'Security groups' are 'security-group-kafka, view inbound rules, view outbound rules'. The 'Scheduled events' are 'No scheduled events'. The 'AMI ID' is 'Kafka_Anaconda-New-2022 (ami-06c41d8b5a6dd3c2)'. The 'Subnet ID' is 'subnet-0d728cc585841a1bd'. The 'Network interfaces' are 'eth0'. The 'IAM role' is '-'. The 'Public DNS (IPv4)' field is also circled in red.



The screenshot shows the AWS Management Console interface. On the left sidebar, under the 'Network & Security' section, 'Elastic IPs' is highlighted with a red circle. The main content area displays the 'Elastic IP addresses' page. The 'Actions' button is circled in red, and the 'Allocate Elastic IP address' option is highlighted with a red circle. The table below shows no Elastic IP addresses found in this Region.



13. Associating Elastic IP address with instance

Associate Elastic IP address

Choose the instance or network interface to associate to this Elastic IP address (52.2.105.98)

Elastic IP address: 52.2.105.98

Resource type
Choose the type of resource with which to associate the Elastic IP address.

☒ Instance
☐ Network interface

Instance
Choose an instance
i-0ef7bda7e6b0ce7c0 (EC2 instance for Kafka) - running

The private IP address with which to associate the Elastic IP address.
Choose a private IP address

Reassociation
Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.
☐ Allow this Elastic IP address to be reassociated

Cancel Associate

Associate Elastic IP address

Choose the instance or network interface to associate to this Elastic IP address (52.2.105.98)

Elastic IP address: 52.2.105.98

Resource type
Choose the type of resource with which to associate the Elastic IP address.

☒ Instance
☐ Network interface

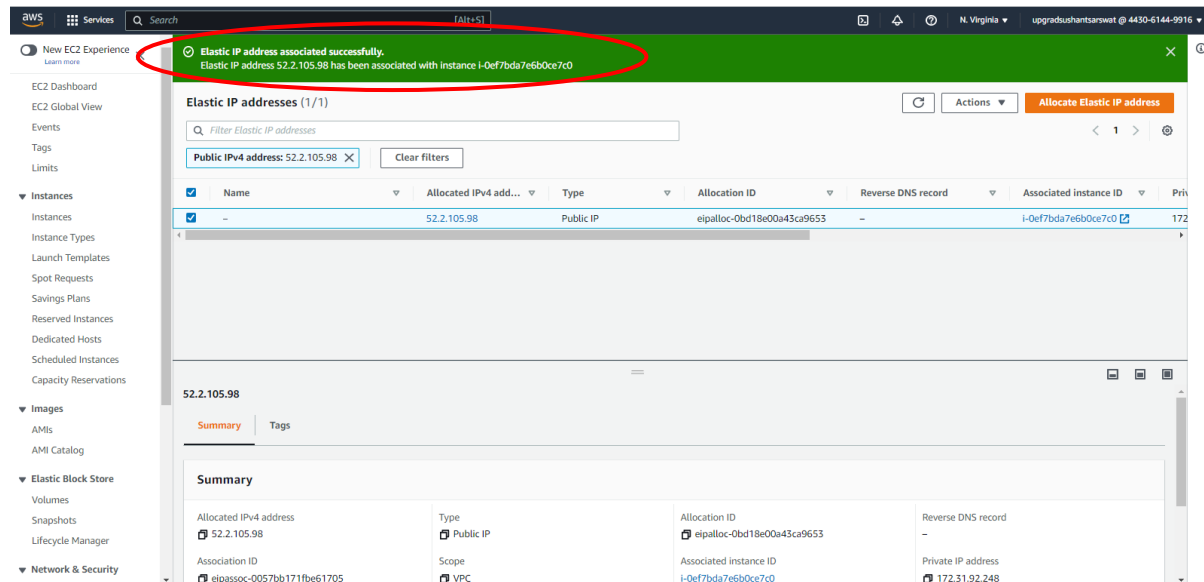
Instance
i-0ef7bda7e6b0ce7c0

Private IP address
The private IP address with which to associate the Elastic IP address.
Choose a private IP address
172.31.92.248

Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.
☐ Allow this Elastic IP address to be reassociated

Cancel Associate

14. Elastic IP address associated successfully



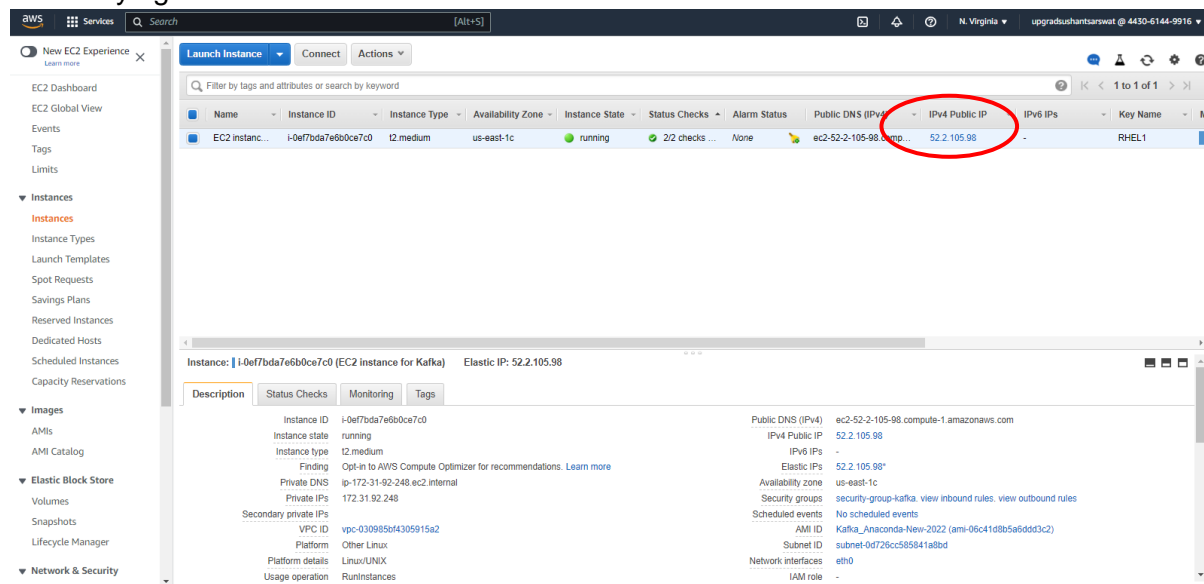
The screenshot shows the AWS Elastic IP console. A green notification banner at the top states: "Elastic IP address associated successfully. Elastic IP address 52.2.105.98 has been associated with instance i-0ef7bda7e6b0ce7c0". Below this, the "Elastic IP addresses (1/1)" section shows a table with one entry:

Name	Allocated IPv4 address	Type	Allocation ID	Reverse DNS record	Associated instance ID	Private IP address
-	52.2.105.98	Public IP	eipalloc-0bd18e00a43ca9653	-	i-0ef7bda7e6b0ce7c0	172.31.92.248

The "Summary" section for the Elastic IP 52.2.105.98 provides further details:

- Allocated IPv4 address:** 52.2.105.98
- Type:** Public IP
- Allocation ID:** eipalloc-0bd18e00a43ca9653
- Reverse DNS record:** -
- Association ID:** eipassoc-0057bb171f6e1705
- Scope:** VPC
- Associated instance ID:** i-0ef7bda7e6b0ce7c0
- Private IP address:** 172.31.92.248

15. Verifying association of Elastic IP address with EC2 instance



The screenshot shows the AWS EC2 console. The "Instances" list shows an EC2 instance with the following details:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name
EC2 instanc...	i-0ef7bda7e6b0ce7c0	t2.medium	us-east-1c	running	2/2 checks ...	None	ec2-52-2-105-98.compute-1.amazonaws.com	52.2.105.98	-	RHEL1

The "Instance: i-0ef7bda7e6b0ce7c0 (EC2 instance for Kafka) Elastic IP: 52.2.105.98" details are shown below. The "Description" tab is active, displaying the following information:

- Instance ID:** i-0ef7bda7e6b0ce7c0
- Instance state:** running
- Instance type:** t2.medium
- Finding:** Opt-in to AWS Compute Optimizer for recommendations. [Learn more](#)
- Private DNS:** ip-172.31-92-248.ec2.internal
- Private IPs:** 172.31.92.248
- Secondary private IPs:** -
- VPC ID:** vpc-030985b4305915a2
- Platform:** Other Linux
- Platform details:** Linux/UNIX
- Usage operation:** RunInstances
- Public DNS (IPv4):** ec2-52-2-105-98.compute-1.amazonaws.com
- IPv4 Public IP:** 52.2.105.98
- IPv6 IPs:** -
- Elastic IPs:** 52.2.105.98*
- Availability zone:** us-east-1c
- Security groups:** security-group-kafka, view inbound rules, view outbound rules
- Scheduled events:** No scheduled events
- AMI ID:** Kafka_Anaconda-New-2022 (ami-06c41d8b5a6dd3c2)
- Subnet ID:** subnet-0d72c5858541a8bd
- Network interfaces:** eth0
- IAM role:** -

16. Login to EC2 instance

```
ec2-user@ip-172-31-92-248:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
Last login: Sat Sep 19 05:18:08 2020 from 103.212.145.122  
  
  _| _|_ )  
  _| ( _ /  Amazon Linux 2 AMI  
  __| \__|__|  
  
https://aws.amazon.com/amazon-linux-2/  
64 package(s) needed for security, out of 117 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-92-248 ~]$
```

17. Updating EC2 instance using 'sudo yum update'

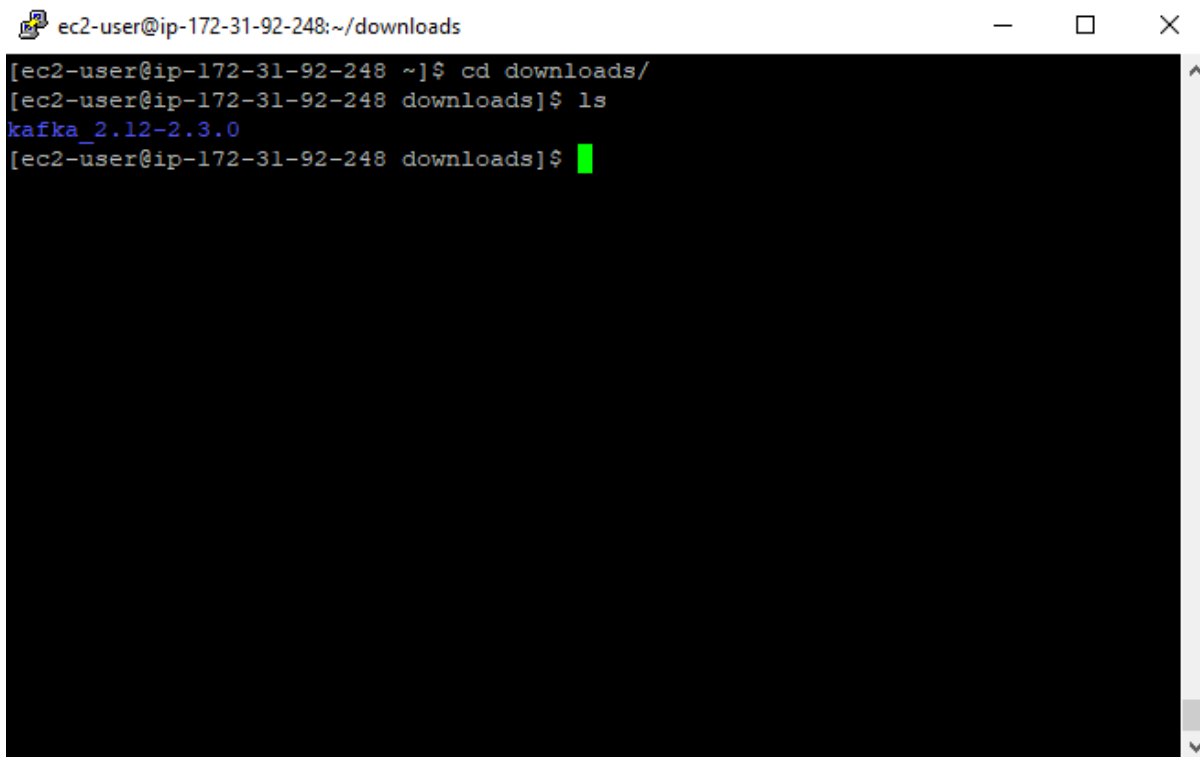
```
rpm.x86_64 0:4.11.3-48.amzn2.0.2
rpm-build-libs.x86_64 0:4.11.3-48.amzn2.0.2
rpm-libs.x86_64 0:4.11.3-48.amzn2.0.2
rpm-plugin-systemd-inhibit.x86_64 0:4.11.3-48.amzn2.0.2
selinux-policy.noarch 0:3.13.1-192.amzn2.6.8
selinux-policy-targeted.noarch 0:3.13.1-192.amzn2.6.8
sysctl-defaults.noarch 0:1.0-3.amzn2
system-release.x86_64 1:2-14.amzn2
systemtap-runtime.x86_64 0:4.5-1.amzn2.0.1
tzdata.noarch 0:2022e-1.amzn2.0.1
update-motd.noarch 0:1.1.2-2.amzn2.0.2
util-linux.x86_64 0:2.30.2-2.amzn2.0.7
vim-common.x86_64 2:9.0.475-1.amzn2.0.1
vim-enhanced.x86_64 2:9.0.475-1.amzn2.0.1
vim-filesystem.noarch 2:9.0.475-1.amzn2.0.1
vim-minimal.x86_64 2:9.0.475-1.amzn2.0.1
yum.noarch 0:3.4.3-158.amzn2.0.6
zlib.x86_64 0:1.2.7-19.amzn2.0.2

Replaced:
grub2.x86_64 1:2.02-35.amzn2.0.4      grub2-tools.x86_64 1:2.02-35.amzn2.0.4

Complete!
[ec2-user@ip-172-31-92-248 ~]$
```

18. Listing files available in downloads folder

```
cd downloads/  
ls
```

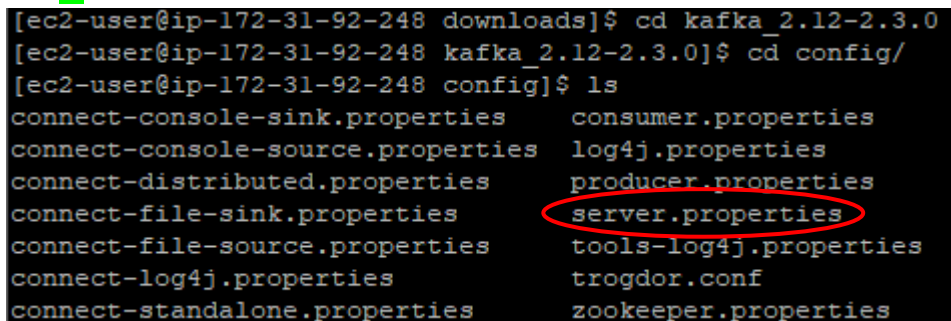


A terminal window titled "ec2-user@ip-172-31-92-248: ~/downloads" with standard window controls. The terminal shows the following commands and output:

```
[ec2-user@ip-172-31-92-248 ~]$ cd downloads/  
[ec2-user@ip-172-31-92-248 downloads]$ ls  
kafka_2.12-2.3.0  
[ec2-user@ip-172-31-92-248 downloads]$
```

19. Changing path to kafka_2.12-2.3.0/config and modifying server.properties

```
cd kafka_2.12-2.3.0  
cd config/  
ls
```



A terminal window showing the following commands and output:

```
[ec2-user@ip-172-31-92-248 downloads]$ cd kafka_2.12-2.3.0  
[ec2-user@ip-172-31-92-248 kafka_2.12-2.3.0]$ cd config/  
[ec2-user@ip-172-31-92-248 config]$ ls  
connect-console-sink.properties    consumer.properties  
connect-console-source.properties  log4j.properties  
connect-distributed.properties     producer.properties  
connect-file-sink.properties       server.properties  
connect-file-source.properties     tools-log4j.properties  
connect-log4j.properties           trogdor.conf  
connect-standalone.properties     zookeeper.properties
```

20. Type **vi server.properties** and press 'I' for insert mode. Remove the '#' from the following line and insert the elastic IP address as displayed. Post changes press 'Esc' and type **:wq!** to save changes

```
[ec2-user@ip-172-31-92-248 config]$ vi server.properties
#advertised.listeners=PLAINTEXT://your.host.name:9092
advertised.listeners=PLAINTEXT://52.2.105.98:9092
```

21. Verify Installation

(a) Zookeeper

Go to the Kafka directory using the **cd kafka_2.12-2.3.0/** command and then start the Zookeeper server using the **bin/zookeeper-server-start.sh config/zookeeper.properties** command

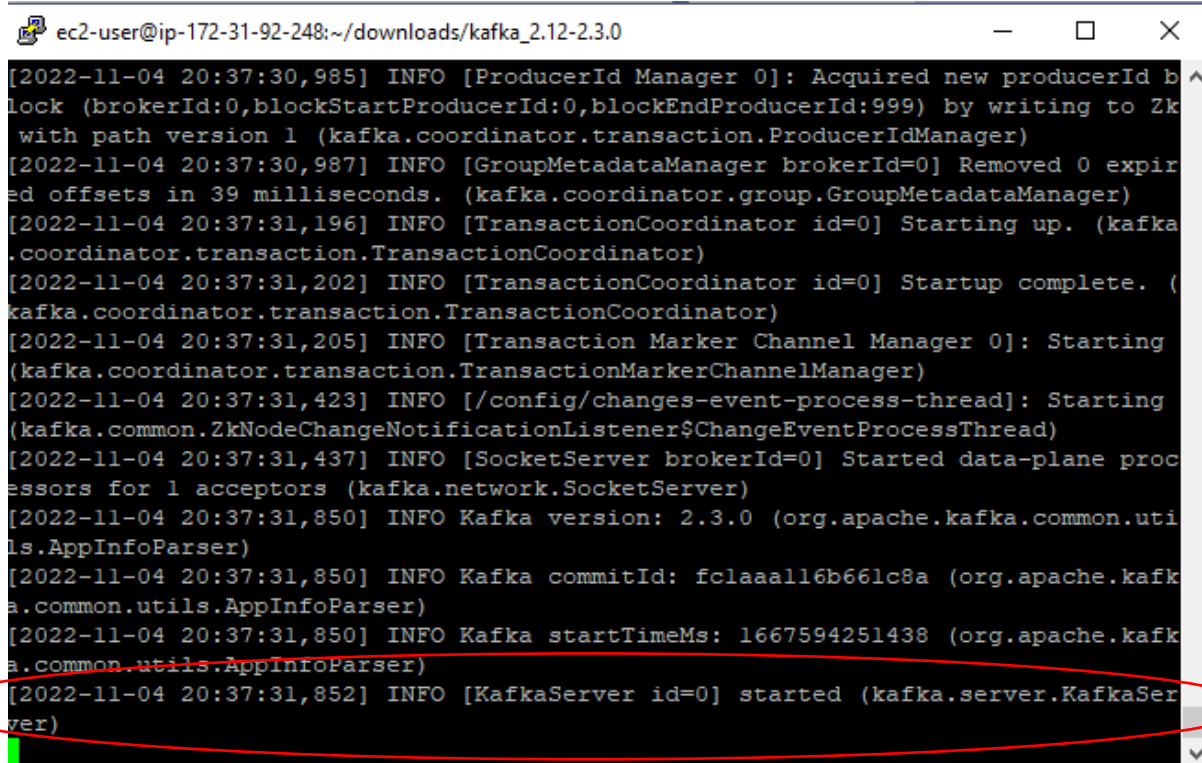
```
[ec2-user@ip-172-31-92-248 kafka_2.12-2.3.0]$ bin/zookeeper-server-start.sh config/zookeeper.properties
[2022-11-04 20:27:45,623] INFO Server environment:os.name=Linux (org.apache.zookeeper.server.ZooKeeperServer)
[2022-11-04 20:27:45,623] INFO Server environment:os.arch=amd64 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-11-04 20:27:45,623] INFO Server environment:os.version=4.14.193-149.317.amzn2.x86_64 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-11-04 20:27:45,623] INFO Server environment:user.name=ec2-user (org.apache.zookeeper.server.ZooKeeperServer)
[2022-11-04 20:27:45,623] INFO Server environment:user.home=/home/ec2-user (org.apache.zookeeper.server.ZooKeeperServer)
[2022-11-04 20:27:45,623] INFO Server environment:user.dir=/home/ec2-user/downloads/kafka_2.12-2.3.0 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-11-04 20:27:45,644] INFO tickTime set to 3000 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-11-04 20:27:45,644] INFO minSessionTimeout set to -1 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-11-04 20:27:45,644] INFO maxSessionTimeout set to -1 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-11-04 20:27:45,966] INFO Using org.apache.zookeeper.server.NIOServerCnxnFactory as server connection factory (org.apache.zookeeper.server.ServerCnxnFactory)
[2022-11-04 20:27:46,226] INFO binding to port 0.0.0.0/0.0.0.0:2181 (org.apache.zookeeper.server.NIOServerCnxnFactory)
```

(b) Starting Kafka server (Do this with Zookeeper server running)

```
cd downloads/kafka_2.12-2.3.0
```

```
bin/kafka-server-start.sh config/server.properties
```

```
[ec2-user@ip-172-31-92-248 ~]$ cd downloads/kafka_2.12-2.3.0
[ec2-user@ip-172-31-92-248 kafka_2.12-2.3.0]$ bin/kafka-server-start.sh config/s
server.properties
```



```
ec2-user@ip-172-31-92-248:~/downloads/kafka_2.12-2.3.0
[2022-11-04 20:37:30,985] INFO [ProducerId Manager 0]: Acquired new producerId b
lock (brokerId:0,blockStartProducerId:0,blockEndProducerId:999) by writing to Zk
with path version 1 (kafka.coordinator.transaction.ProducerIdManager)
[2022-11-04 20:37:30,987] INFO [GroupMetadataManager brokerId=0] Removed 0 expir
ed offsets in 39 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2022-11-04 20:37:31,196] INFO [TransactionCoordinator id=0] Starting up. (kafka
.coordinator.transaction.TransactionCoordinator)
[2022-11-04 20:37:31,202] INFO [TransactionCoordinator id=0] Startup complete. (
kafka.coordinator.transaction.TransactionCoordinator)
[2022-11-04 20:37:31,205] INFO [Transaction Marker Channel Manager 0]: Starting
(kafka.coordinator.transaction.TransactionMarkerChannelManager)
[2022-11-04 20:37:31,423] INFO [/config/changes-event-process-thread]: Starting
(kafka.common.ZkNodeChangeNotificationListener$ChangeEventProcessThread)
[2022-11-04 20:37:31,437] INFO [SocketServer brokerId=0] Started data-plane proc
essors for 1 acceptors (kafka.network.SocketServer)
[2022-11-04 20:37:31,850] INFO Kafka version: 2.3.0 (org.apache.kafka.common.uti
ls.AppInfoParser)
[2022-11-04 20:37:31,850] INFO Kafka commitId: fclaaall6b661c8a (org.apache.kafk
a.common.utils.AppInfoParser)
[2022-11-04 20:37:31,850] INFO Kafka startTimeMs: 1667594251438 (org.apache.kafk
a.common.utils.AppInfoParser)
[2022-11-04 20:37:31,852] INFO [KafkaServer id=0] started (kafka.server.KafkaSer
ver)
```

22. STATEMENT TO CREATE TOPICS

To create topic in kafka server ,the command used is

```
bin/kafka-topics.sh --create --bootstrap-server localhost:9092 --replication-factor 1 --
partitions 1 --topic PatientInformation
```

23. STATEMENT TO LIST TOPICS

```
bin/kafka-topics.sh --list --bootstrap-server localhost:9092
```

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

24. Install the mysql-connector using the following command:

```
pip install mysql-connector-python-rf
```

```
[ec2-user@ip-172-31-92-248 kafka_2.12-2.3.0]$ pip install mysql-connector-python-rf
```

```
ec2-user@ip-172-31-92-248:~/downloads/kafka_2.12-2.3.0
[ec2-user@ip-172-31-92-248 kafka_2.12-2.3.0]$ pip install mysql-connector-python-rf
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 is no longer maintained. pip 21.0 will drop support for Python 2.7 in January 2021. More details about Python 2 support in pip can be found at https://pip.pypa.io/en/latest/development/release-process/#python-2-support pip 21.0 will remove support for this functionality.
Collecting mysql-connector-python-rf
  Downloading mysql-connector-python-rf-2.2.2.tar.gz (11.9 MB)
    | 11.9 MB 1.0 MB/s
Building wheels for collected packages: mysql-connector-python-rf
  Building wheel for mysql-connector-python-rf (setup.py) ... done
  Created wheel for mysql-connector-python-rf: filename=mysql_connector_python_rf-2.2.2-cp27-cp27mu-linux_x86_64.whl size=243519 sha256=6e8b2da3989574cd47b0de18d39f6743f0c007e0ff80d48cabd52fb80bf33374
  Stored in directory: /home/ec2-user/.cache/pip/wheels/3b/b5/d4/5d0e3338625186ab2fbf75908b58178b859aa8elfdl291a0fa
Successfully built mysql-connector-python-rf
Installing collected packages: mysql-connector-python-rf
Successfully installed mysql-connector-python-rf-2.2.2
WARNING: You are using pip version 20.2.3; however, version 20.3.4 is available. You should consider upgrading via the '/home/ec2-user/anaconda2/bin/python -m pip install --upgrade pip' command.
[ec2-user@ip-172-31-92-248 kafka_2.12-2.3.0]$
```

25. Install kafka using the following command:

```
pip install kafka
```

```
[ec2-user@ip-172-31-92-248 kafka_2.12-2.3.0]$ pip install kafka
[ec2-user@ip-172-31-92-248 kafka_2.12-2.3.0]$ pip install kafka
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 is no longer maintained. pip 21.0 will drop support for Python 2.7 in January 2021. More details about Python 2 support in pip can be found at https://pip.pypa.io/en/latest/development/release-process/#python-2-support pip 21.0 will remove support for this functionality.
Collecting kafka
  Downloading kafka-1.3.5-py2.py3-none-any.whl (207 kB)
    | 207 kB 33.3 MB/s
Installing collected packages: kafka
Successfully installed kafka-1.3.5
WARNING: You are using pip version 20.2.3; however, version 20.3.4 is available. You should consider upgrading via the '/home/ec2-user/anaconda2/bin/python -m pip install --upgrade pip' command.
[ec2-user@ip-172-31-92-248 kafka_2.12-2.3.0]$
```


26. Start producer using the following command from /home/ec2-user:

python kafka_produce_patient_vitals.py

```
[ec2-user@ip-172-31-92-248 ~]$ python kafka_produce_patient_vitals.py
1,74,202
2,68,173
3,71,152
4,72,166
5,68,171
1,70,189
2,72,173
3,68,178
4,71,152
5,73,166
1,74,185
2,67,177
3,66,158
4,71,177
5,66,155
1,71,220
2,67,161
3,67,174
4,67,157
```

27. Prior to running consumer, execute the following commands as displayed in screenshot:

pip install pyspark

python -m pip install findspark

```
[hadoop@ip-172-31-47-57 ~]$ pip install pyspark
Defaulting to user installation because normal site-packages is not writeable
Collecting pyspark
  Downloading pyspark-3.3.1.tar.gz (281.4 MB)
    | 281.4 MB 32 KB/s
Collecting py4j==0.10.9.5
  Downloading py4j-0.10.9.5-py2.py3-none-any.whl (199 kB)
    | 199 kB 49.1 MB/s
Using legacy 'setup.py install' for pyspark, since package 'wheel' is not installed.
Installing collected packages: py4j, pyspark
  Running setup.py install for pyspark ... done
ERROR: After October 2020 you may experience errors when installing or updating packages. This is because pip will change the way that it resolves dependency conflicts.
We recommend you use --use-feature=2020-resolver to test your packages with the new resolver before it becomes the default.

python3?-sagemaker-pyspark 1.4.1 requires pyspark==2.3.4, but you'll have pyspark 3.3.1 which is incompatible.
Successfully installed py4j-0.10.9.5 pyspark-3.3.1
[hadoop@ip-172-31-47-57 ~]$ python -m pip install findspark
Defaulting to user installation because normal site-packages is not writeable
Collecting findspark
  Downloading findspark-2.0.1-py2.py3-none-any.whl (4.4 kB)
Installing collected packages: findspark
Successfully installed findspark-2.0.1
```

wget https://ds-spark-sql-kafka-jar.s3.amazonaws.com/spark-sql-kafka-0-10_2.11-2.3.0.jar

**spark-submit --jars spark-sql-kafka-0-10_2.11-2.3.0.jar
kafka_spark_patient_vitals.py 52.2.105.98 9092 PatientInformation**

```
[hadoop@ip-172-31-47-57 ~]$ wget https://ds-spark-sql-kafka-jar.s3.amazonaws.com/spark-sql-kafka-0-10_2.11-2.3.0.jar
--2022-11-13 20:19:38-- https://ds-spark-sql-kafka-jar.s3.amazonaws.com/spark-sql-kafka-0-10_2.11-2.3.0.jar
Resolving ds-spark-sql-kafka-jar.s3.amazonaws.com (ds-spark-sql-kafka-jar.s3.amazonaws.com)... 52.216.97.203
Connecting to ds-spark-sql-kafka-jar.s3.amazonaws.com (ds-spark-sql-kafka-jar.s3.amazonaws.com)|52.216.97.203|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 406313 (397K) [binary/octet-stream]
Saving to: 'spark-sql-kafka-0-10_2.11-2.3.0.jar'

100%[=====>] 406,313 --.-K/s in 0.01s

2022-11-13 20:19:38 (32.1 MB/s) - 'spark-sql-kafka-0-10_2.11-2.3.0.jar' saved [406313/406313]

[hadoop@ip-172-31-47-57 ~]$ ls
kafka_spark_patient_vitals.py      threshold.txt
spark-sql-kafka-0-10_2.11-2.3.0.jar
[hadoop@ip-172-31-47-57 ~]$ spark-submit --jars spark-sql-kafka-0-10_2.11-2.3.0.jar kafka_spark_patient_vitals.py 52.2.105.98 9092 PatientInformation
```

28. We will witness the following output on the terminal:

```
Batch: 16
-----+-----+-----+-----+
|CustomerID|HeartBeatRate|BloodPressure|Timestamp|
|-----+-----+-----+-----+
|3|167|163|2022-11-25 19:32:19.365|
|-----+-----+-----+-----+

Batch: 17
-----+-----+-----+-----+
|CustomerID|HeartBeatRate|BloodPressure|Timestamp|
|-----+-----+-----+-----+
|4|166|177|2022-11-25 19:32:20.367|
|-----+-----+-----+-----+

Batch: 18
-----+-----+-----+-----+
|CustomerID|HeartBeatRate|BloodPressure|Timestamp|
|-----+-----+-----+-----+
|5|168|158|2022-11-25 19:32:21.368|
|-----+-----+-----+-----+

Batch: 19
-----+-----+-----+-----+
|CustomerID|HeartBeatRate|BloodPressure|Timestamp|
|-----+-----+-----+-----+
|1|172|187|2022-11-25 19:32:22.369|
|-----+-----+-----+-----+

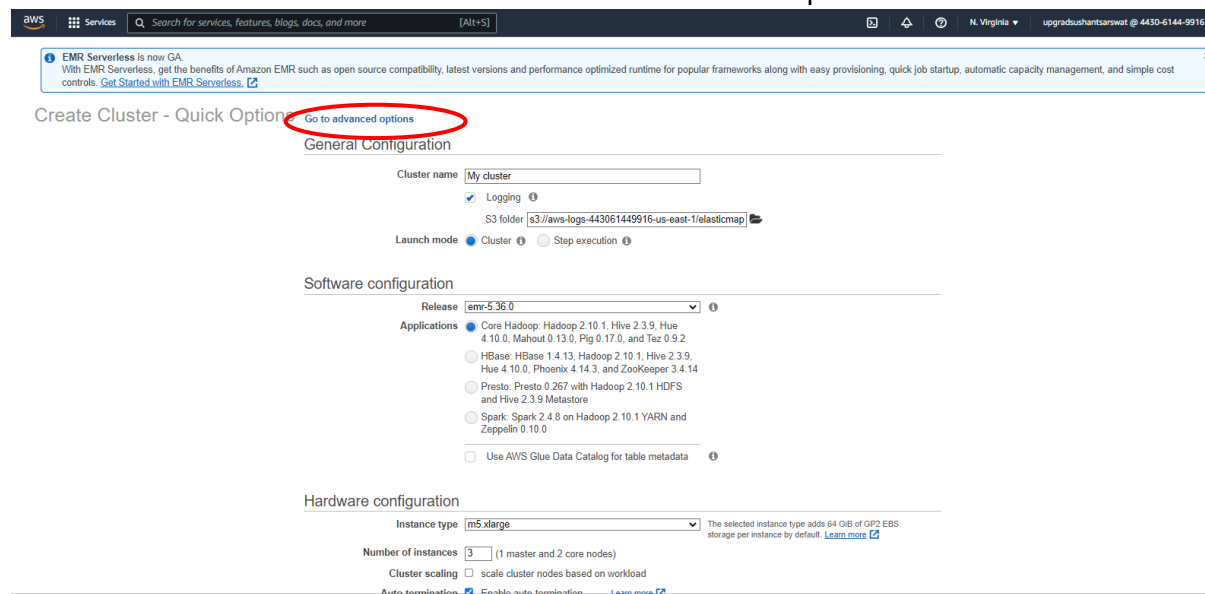
Batch: 20
-----+-----+-----+-----+
|CustomerID|HeartBeatRate|BloodPressure|Timestamp|
|-----+-----+-----+-----+
|2|173|166|2022-11-25 19:32:23.371|
|-----+-----+-----+-----+

Batch: 21
-----+-----+-----+-----+
|CustomerID|HeartBeatRate|BloodPressure|Timestamp|
|-----+-----+-----+-----+
|3|168|152|2022-11-25 19:32:24.372|
|-----+-----+-----+-----+
```

CREATION AND CONFIGURATION OF HIVE CLUSTER

Following the creation of the Kafka cluster, the next step is to set up EMR cluster for Hive and other software. The following steps illustrate the process of creating the EMR cluster:-

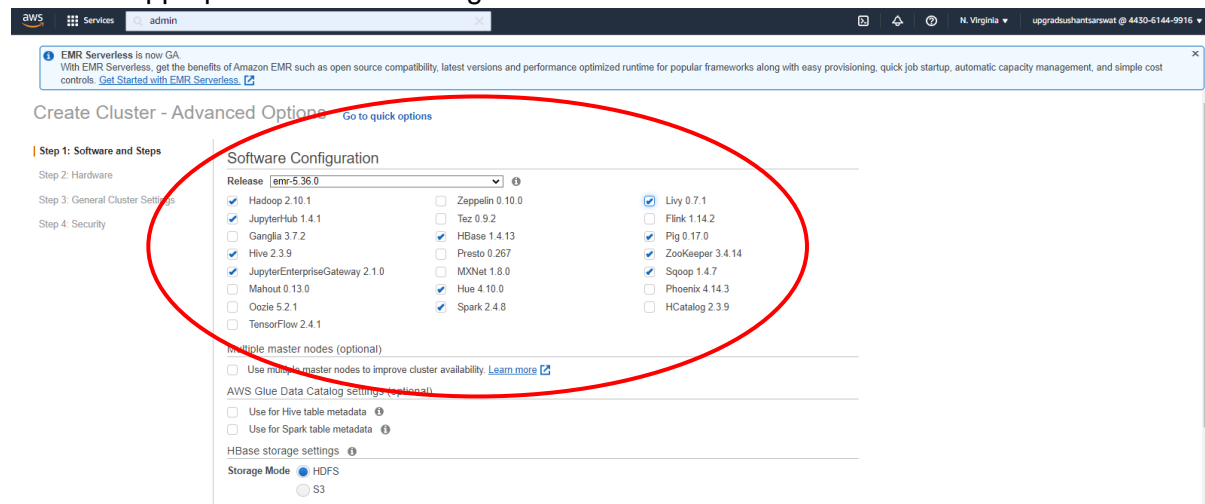
1. Select EMR in AWS services and Select 'Go to advanced options'



The screenshot shows the AWS EMR console 'Create Cluster - Quick Options' page. The 'Go to advanced options' link is circled in red. The page displays the following configuration options:

- General Configuration:**
 - Cluster name: My cluster
 - Logging: ☒ Logging
 - S3 folder: s3://aws-logs-44306144916-us-east-1/elasticmap
 - Launch mode: ☒ Cluster ☐ Step execution
- Software configuration:**
 - Release: emr-5.36.0
 - Applications:
 - ☒ Core Hadoop: Hadoop 2.10.1, Hive 2.3.9, Hue 4.10.0, Mahout 0.13.0, Pig 0.17.0, and Tez 0.9.2
 - ☐ HBase: HBase 1.4.13, Hadoop 2.10.1, Hive 2.3.9, Hue 4.10.0, Phoenix 4.14.3, and ZooKeeper 3.4.14
 - ☐ Presto: Presto 0.267 with Hadoop 2.10.1 HDFS and Hive 2.3.9 Metastore
 - ☐ Spark: Spark 2.4.8 on Hadoop 2.10.1 YARN and Zeppelin 0.10.0
 - ☐ Use AWS Glue Data Catalog for table metadata
- Hardware configuration:**
 - Instance type: m5.xlarge
 - Number of instances: 3 (1 master and 2 core nodes)
 - Cluster scaling: ☐ scale cluster nodes based on workload
 - Auto-termination: ☒ Enable auto-termination

2. Select appropriate software configuration



The screenshot shows the AWS EMR console 'Create Cluster - Advanced Options' page. The 'Software Configuration' section is circled in red. The page displays the following configuration options:

- Step 1: Software and Steps:**
 - Release: emr-5.36.0
 - Software Configuration:
 - ☒ Hadoop 2.10.1
 - ☒ JupyterHub 1.4.1
 - ☐ Ganglia 3.7.2
 - ☒ Hive 2.3.9
 - ☒ JupyterEnterpriseGateway 2.1.0
 - ☐ Mahout 0.13.0
 - ☐ Oozie 5.2.1
 - ☐ TensorFlow 2.4.1
 - ☐ Zeppelin 0.10.0
 - ☐ Tez 0.9.2
 - ☒ HBase 1.4.13
 - ☐ Presto 0.267
 - ☐ MXNet 1.8.0
 - ☒ Hue 4.10.0
 - ☒ Spark 2.4.8
 - ☒ Livy 0.7.1
 - ☐ Flink 1.14.2
 - ☒ Pig 0.17.0
 - ☒ ZooKeeper 3.4.14
 - ☒ Sqoop 1.4.7
 - ☐ Phoenix 4.14.3
 - ☐ HCatalog 2.3.9
 - ☐ Multiple master nodes (optional)
 - ☐ Use multiple master nodes to improve cluster availability
 - AWS Glue Data Catalog settings (optional):**
 - ☐ Use for Hive table metadata
 - ☐ Use for Spark table metadata
 - HBase storage settings:**
 - Storage Mode: ☒ HDFS ☐ S3

3. Machine Selection

Instance types

<input type="radio"/>	m3.2xlarge	8	30	160 SSD
<input type="radio"/>	m4.large	2	8	EBS only
<input checked="" type="radio"/>	m4.xlarge	4	16	EBS only
<input type="radio"/>	m4.2xlarge	8	32	EBS only
<input type="radio"/>	m4.4xlarge	16	64	EBS only
<input type="radio"/>	m4.10xlarge	40	160	EBS only
<input type="radio"/>	m4.16xlarge	64	256	EBS only
<input type="radio"/>	m5.xlarge	4	16	EBS only
<input type="radio"/>	m5.2xlarge	8	32	EBS only
<input type="radio"/>	m5.4xlarge	16	64	EBS only
<input type="radio"/>	m5.8xlarge	32	128	EBS only

Cancel

Save

4. Cluster Nodes and instances

Cluster Nodes and Instances

Choose the instance type, number of instances, and a purchasing option. [Learn more about instance purchasing options](#)

Console options for automatic scaling have changed. [Learn more](#)

Node type	Instance type	Instance count	Purchasing option
Master Master - 1	m4.xlarge 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 64 GiB Add configuration settings	1 Instances	<input checked="" type="radio"/> On-demand <input type="radio"/> Spot Use on-demand as max price
Core Core - 2	m4.xlarge 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 64 GiB Add configuration settings	2 Instances	<input checked="" type="radio"/> On-demand <input type="radio"/> Spot Use on-demand as max price
Task Task - 3	m5.xlarge 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	0 Instances	<input checked="" type="radio"/> On-demand <input type="radio"/> Spot Use on-demand as max price

Create Cluster - Advanced Options [Go to quick options](#)

Step 1: Software and Steps

Step 2: Hardware

Step 3: General Cluster Settings

Step 4: Security

General Options

Cluster name

☒ Logging

S3 folder

☐ Log encryption

☒ Debugging

☐ Termination protection

5. Security Options

Create Cluster - Advanced Options

[Go to quick options](#)

[Step 1: Software and Steps](#)

[Step 2: Hardware](#)

[Step 3: General Cluster Settings](#)

Step 4: Security

Security Options

EC2 key pair **RHEL1** ⓘ

☒ Cluster visible to all IAM users in account ⓘ

Permissions ⓘ

☒ Default ☐ Custom

Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.

EMR role [EMR_DefaultRole](#) ☐ Use EMR_DefaultRole_V2 ⓘ

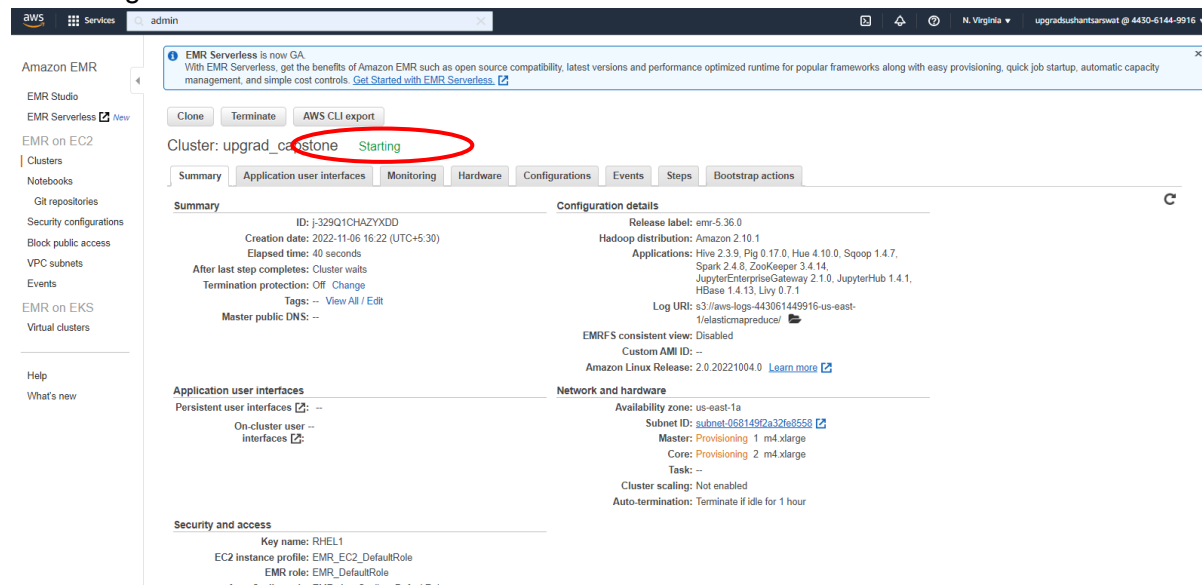
EC2 instance profile [EMR_EC2_DefaultRole](#) ⓘ

Auto Scaling role [EMR_AutoScaling_DefaultRole](#) ⓘ

▶ Security Configuration

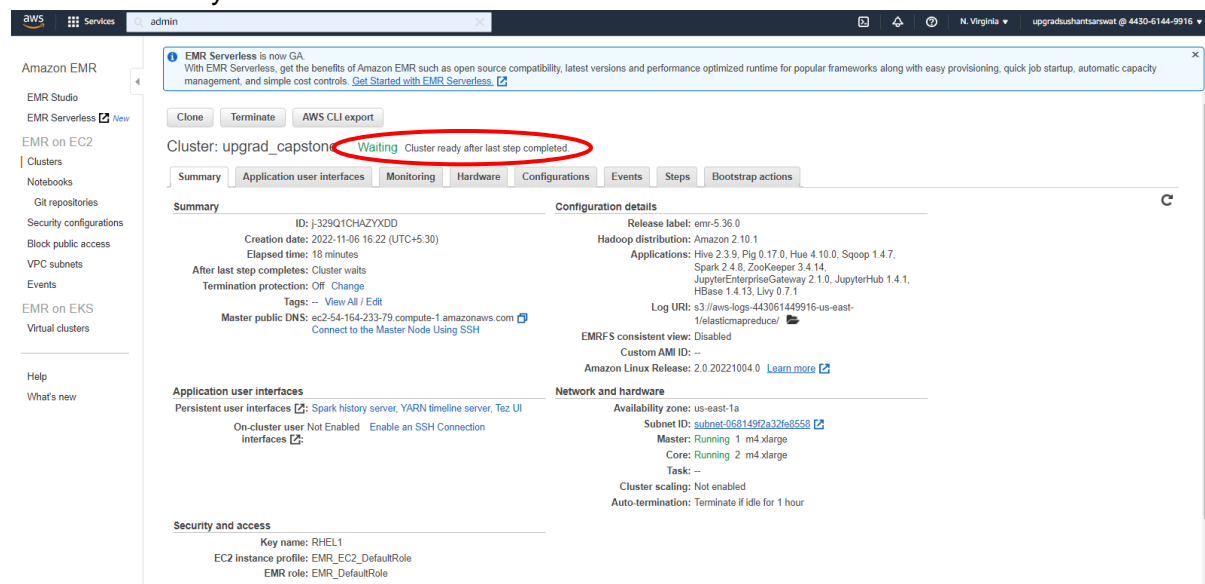
▶ EC2 security groups

6. Starting the cluster



The screenshot shows the AWS Management Console interface for an Amazon EMR cluster. The cluster name is 'upgrad_capstone' and its status is 'Starting', which is circled in red. The console displays various tabs for the cluster, including Summary, Application user interfaces, Monitoring, Hardware, Configurations, Events, Steps, and Bootstrap actions. The Summary tab is selected, showing details such as ID, Creation date, Elapsed time, and Configuration details. The Configuration details section lists the Release label, Hadoop distribution, Applications, Log URI, EMRFS consistent view, Custom AMI ID, Amazon Linux Release, Network and hardware, and Security access.

7. Cluster Ready



8. STATEMENT TO BUILD AN EXTERNAL HIVE TABLE FOR THRESHOLD DATA AND VIEW THRESHOLD DATA

(a) Create a text file in HDFS to load threshold values data by logging into Hadoop and opening a text file by using **vi threshold.txt** command

```
[hadoop@ip-172-31-46-73 ~]$ vi threshold.txt
```

(b) Insert the following lines in the threshold.txt file

```
1|heartBeat|0|40|0|69|1|LowerHeartRatethanNormal|
2|heartBeat|0|40|70|78|0|Normal|
3|heartBeat|0|40|79|9999|1|HigherHeartRatethanNormal|
4|bp|0|40|0|160|1|LowerBPthanNormal|
5|bp|0|40|161|220|0|Normal|
6|bp|0|40|221|9999|1|HigherBPthanNormal|
7|heartBeat|41|100|0|65|1|LowHeartRatethanNormal|
8|heartBeat|41|100|66|73|0|Normal|
9|heartBeat|41|100|74|9999|1|HigherHeartRatethanNormal|
10|bp|41|100|0|150|1|LowBPthanNormal|
11|bp|41|100|151|180|0|Normal|
12|bp|41|100|181|9999|1|HigherBPthanNormal|
```

Save the file by clicking **escape** key and save the file using **:wq!**

```
1|heartBeat|0|40|0|69|1|LowerHeartRatethanNormal
2|heartBeat|0|40|70|78|0|Normal
3|heartBeat|0|40|79|9999|1|HigherHeartRatethanNormal
4|bp|0|40|0|160|1|LowBPthanNormal
5|bp|0|40|161|220|0|Normal
6|bp|0|40|221|9999|1|HigherBPthanNormal
7|heartBeat|41|100|0|65|1|LowHeartRatethanNormal
8|heartBeat|41|100|66|73|0|Normal
9|heartBeat|41|100|74|9999|1|HigherHeartRatethanNormal
10|bp|41|100|0|150|1|LowBPthanNormal
11|bp|41|100|151|180|0|Normal
12|bp|41|100|181|9999|1|HigherBPthanNormal

wq!
```

(c) Log into hive

```
[hadoop@ip-172-31-46-73 ~]$ hive
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.
properties Async: false
hive>
```

(d) Create database using command **create database if not exists Patient_db;**

```
hive> Create database if not exists Patient_db;  
OK  
Time taken: 0.845 seconds
```

(e) Use database by using command `use Patient_db;`

```
hive> Use Patient_db;
OK
Time taken: 0.078 seconds
```

(f) Create external hive table for threshold values using the following command

```
create external table if not exists ThresholdValueTable(
Key int,
Attribute varchar(20),
Low_Age_Limit int,
High_Age_Limit int,
Low_Range_Value int,
High_Range_Value int,
```

Alert_Flag int,
Alert_Message varchar(255))
row format delimited fields terminated by '|'
lines terminated by '\n' stored as textfile;

```
hive> create external table if not exists ThresholdValueTable(
  > Key int,
  > Attribute varchar(20),
  > Low_Age_Limit int,
  > High_Age_Limit int,
  > Low_Range_Value int,
  > High_Range_Value int,
  > Alert_Flag int,
  > Alert_Message varchar(255))
  > row format delimited fields terminated by '|'
  > lines terminated by '\n' stored as textfile;
OK
Time taken: 0.783 seconds
```

(g) Load data from threshold.txt into the table ThresholdValueTable using the following command

load data local inpath 'threshold.txt' into table ThresholdValueTable;

```
hive> load data local inpath 'threshold.txt' into table ThresholdValueTable;
Loading data to table patient_db.thresholdvaluetable
OK
Time taken: 1.195 seconds
```

(h) Verify that the values have been loaded properly into the table by using the following command

select * from thresholdvaluetable;

```
hive> select * from thresholdvaluetable;
OK
1      heartBeat      0      40      0      69      1      LowerHeartRatethanNormal
2      heartBeat      0      40      70      78      0      Normal
3      heartBeat      0      40      79      9999     1      HigherHeartRatethanNormal
4      bp      0      40      0      160     1      LowBPthanNormal
5      bp      0      40      161     220     0      Normal
6      bp      0      40      221     9999     1      HigherBPthanNormal
7      heartBeat      41     100     0      65      1      LowHeartRatethanNormal
8      heartBeat      41     100     66      73      0      Normal
9      heartBeat      41     100     74      9999     1      HigherHeartRatethanNormal
10     bp      41     100     0      150     1      LowBPthanNormal
11     bp      41     100     151     180     0      Normal
12     bp      41     100     181     9999     1      HigherBPthanNormal
Time taken: 0.216 seconds, Fetched: 12 row(s)
```


STATEMENTS TO CREATE HBASE TABLE ON HIVE

1. From EMR, use the below command to go to hbase shell

Hbase shell

```
[hadoop@ip-172-31-38-49 ~]$ hbase shell
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
Version 1.4.13, rUnknown, Wed Jun  8 00:30:30 UTC 2022
hbase(main):001:0>
```

2. Create HBASE threshold table and insert data as described in the following steps:-

(a) We can create hbase table using the following command:

create 'Threshold_Reference_Table', 'attribute', 'limit', 'alert'

```
hbase(main):001:0> create 'Threshold_Reference_Table', 'attribute', 'limit', 'alert'
0 row(s) in 1.7150 seconds

=> Hbase::Table - Threshold_Reference_Table
hbase(main):002:0>
```

2. We can describe the table column field using the following command:

describe 'Threshold_Reference_Table'

```
hbase(main):002:0> describe 'Threshold_Reference_Table'
Table Threshold_Reference_Table is ENABLED
Threshold_Reference_Table
COLUMN FAMILIES DESCRIPTION
(NAME => 'alert', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0')
(NAME => 'attribute', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0')
(NAME => 'limit', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0')
3 row(s) in 0.0850 seconds
hbase(main):003:0>
```

3. We can insert the threshold data using put command for 12 records as follows:

```
put 'Threshold_Reference_Table','row1','attribute:attribute','value=heartBeat'
put 'Threshold_Reference_Table','row2','attribute:attribute','value=heartBeat'
put 'Threshold_Reference_Table','row3','attribute:attribute','value=heartBeat'
put 'Threshold_Reference_Table','row4','attribute:attribute','value=bp'
put 'Threshold_Reference_Table','row5','attribute:attribute','value=bp'
put 'Threshold_Reference_Table','row6','attribute:attribute','value=bp'
put 'Threshold_Reference_Table','row7','attribute:attribute','value=heartBeat'
put 'Threshold_Reference_Table','row8','attribute:attribute','value=heartBeat'
put 'Threshold_Reference_Table','row9','attribute:attribute','value=heartBeat'
put 'Threshold_Reference_Table','row10','attribute:attribute','value=bp'
put 'Threshold_Reference_Table','row11','attribute:attribute','value=bp'
```

```
put 'Threshold_Reference_Table','row12','attribute:attribute','value=bp'
```

```
put 'Threshold_Reference_Table','row1','limit:low_age_limit','value=0'
put 'Threshold_Reference_Table','row1','limit:high_age_limit','value=40'
put 'Threshold_Reference_Table','row1','limit:low_value','value=0'
put 'Threshold_Reference_Table','row1','limit:high_value','value=69'
```

```
put 'Threshold_Reference_Table','row2','limit:low_age_limit','value=0'
put 'Threshold_Reference_Table','row2','limit:high_age_limit','value=40'
put 'Threshold_Reference_Table','row2','limit:low_value','value=70'
put 'Threshold_Reference_Table','row2','limit:high_value','value=78'
```

```
put 'Threshold_Reference_Table','row3','limit:low_age_limit','value=0'
put 'Threshold_Reference_Table','row3','limit:high_age_limit','value=40'
put 'Threshold_Reference_Table','row3','limit:low_value','value=79'
put 'Threshold_Reference_Table','row3','limit:high_value','value=9999'
```

```
put 'Threshold_Reference_Table','row4','limit:low_age_limit','value=0'
put 'Threshold_Reference_Table','row4','limit:high_age_limit','value=40'
put 'Threshold_Reference_Table','row4','limit:low_value','value=0'
put 'Threshold_Reference_Table','row4','limit:high_value','value=160'
```

```
put 'Threshold_Reference_Table','row5','limit:low_age_limit','value=0'
put 'Threshold_Reference_Table','row5','limit:high_age_limit','value=40'
put 'Threshold_Reference_Table','row5','limit:low_value','value=161'
put 'Threshold_Reference_Table','row5','limit:high_value','value=220'
```

```
put 'Threshold_Reference_Table','row6','limit:low_age_limit','value=0'
put 'Threshold_Reference_Table','row6','limit:high_age_limit','value=40'
put 'Threshold_Reference_Table','row6','limit:low_value','value=221'
put 'Threshold_Reference_Table','row6','limit:high_value','value=9999'
```

```
put 'Threshold_Reference_Table','row7','limit:low_age_limit','value=41'
put 'Threshold_Reference_Table','row7','limit:high_age_limit','value=100'
put 'Threshold_Reference_Table','row7','limit:low_value','value=0'
put 'Threshold_Reference_Table','row7','limit:high_value','value=65'
```

```
put 'Threshold_Reference_Table','row8','limit:low_age_limit','value=41'
put 'Threshold_Reference_Table','row8','limit:high_age_limit','value=100'
put 'Threshold_Reference_Table','row8','limit:low_value','value=66'
put 'Threshold_Reference_Table','row8','limit:high_value','value=73'
```

```
put 'Threshold_Reference_Table','row9','limit:low_age_limit','value=41'
put 'Threshold_Reference_Table','row9','limit:high_age_limit','value=100'
put 'Threshold_Reference_Table','row9','limit:low_value','value=74'
put 'Threshold_Reference_Table','row9','limit:high_value','value=9999'
```

```
put 'Threshold_Reference_Table','row10','limit:low_age_limit','value=41'
put 'Threshold_Reference_Table','row10','limit:high_age_limit','value=100'
put 'Threshold_Reference_Table','row10','limit:low_value','value=0'
put 'Threshold_Reference_Table','row10','limit:high_value','value=150'
```

```
put 'Threshold_Reference_Table','row11','limit:low_age_limit','value=41'
put 'Threshold_Reference_Table','row11','limit:high_age_limit','value=100'
put 'Threshold_Reference_Table','row11','limit:low_value','value=151'
put 'Threshold_Reference_Table','row11','limit:high_value','value=180'
```

```
put 'Threshold_Reference_Table','row12','limit:low_age_limit','value=41'
put 'Threshold_Reference_Table','row12','limit:high_age_limit','value=100'
put 'Threshold_Reference_Table','row12','limit:low_value','value=181'
put 'Threshold_Reference_Table','row12','limit:high_value','value=9999'
```

```
put 'Threshold_Reference_Table','row1','alert:alert_flag','value=1'
put 'Threshold_Reference_Table','row2','alert:alert_flag','value=0'
put 'Threshold_Reference_Table','row3','alert:alert_flag','value=1'
put 'Threshold_Reference_Table','row4','alert:alert_flag','value=1'
put 'Threshold_Reference_Table','row5','alert:alert_flag','value=0'
put 'Threshold_Reference_Table','row6','alert:alert_flag','value=1'
put 'Threshold_Reference_Table','row7','alert:alert_flag','value=1'
put 'Threshold_Reference_Table','row8','alert:alert_flag','value=0'
put 'Threshold_Reference_Table','row9','alert:alert_flag','value=1'
put 'Threshold_Reference_Table','row10','alert:alert_flag','value=1'
put 'Threshold_Reference_Table','row11','alert:alert_flag','value=0'
put 'Threshold_Reference_Table','row12','alert:alert_flag','value=1'
```

```
put 'Threshold_Reference_Table','row1','alert:alert_message','value=Low Heart Rate than Normal'
put 'Threshold_Reference_Table','row2','alert:alert_message','value=Normal'
put 'Threshold_Reference_Table','row3','alert:alert_message','value=Higher Heart Rate than Normal'
put 'Threshold_Reference_Table','row4','alert:alert_message','value=Low BP than Normal'
put 'Threshold_Reference_Table','row5','alert:alert_message','value=Normal'
put 'Threshold_Reference_Table','row6','alert:alert_message','value=Higher BP than Normal'
put 'Threshold_Reference_Table','row7','alert:alert_message','value=Low Heart Rate than Normal'
put 'Threshold_Reference_Table','row8','alert:alert_message','value=Normal'
put 'Threshold_Reference_Table','row9','alert:alert_message','value=Higher Heart Rate than Normal'
put 'Threshold_Reference_Table','row10','alert:alert_message','value=Low BP than Normal'
put 'Threshold_Reference_Table','row11','alert:alert_message','value=Normal'
put 'Threshold_Reference_Table','row12','alert:alert_message','value=Higher BP than Normal'
```

```
hbase(main):124:0> put 'Threshold_Reference_Table','row8','alert:alert_flag','value=0'
0 row(s) in 0.0030 seconds

hbase(main):125:0> put 'Threshold_Reference_Table','row9','alert:alert_flag','value=1'
0 row(s) in 0.0020 seconds

hbase(main):126:0> put 'Threshold_Reference_Table','row10','alert:alert_flag','value=1'
0 row(s) in 0.0030 seconds

hbase(main):127:0> put 'Threshold_Reference_Table','row11','alert:alert_flag','value=0'
0 row(s) in 0.0030 seconds

hbase(main):128:0> put 'Threshold_Reference_Table','row12','alert:alert_flag','value=1'
0 row(s) in 0.0030 seconds

hbase(main):129:0>
hbase(main):130:0> put 'Threshold_Reference_Table','row1','alert:alert_message','value=Low Heart Rate than Normal'
0 row(s) in 0.0030 seconds

hbase(main):131:0> put 'Threshold_Reference_Table','row2','alert:alert_message','value=Normal'
0 row(s) in 0.0030 seconds

hbase(main):132:0> put 'Threshold_Reference_Table','row3','alert:alert_message','value=Higher Heart Rate than Normal'
0 row(s) in 0.0030 seconds

hbase(main):133:0> put 'Threshold_Reference_Table','row4','alert:alert_message','value=Low BP than Normal'
0 row(s) in 0.0030 seconds

hbase(main):134:0> put 'Threshold_Reference_Table','row5','alert:alert_message','value=Normal'
0 row(s) in 0.0030 seconds

hbase(main):135:0> put 'Threshold_Reference_Table','row6','alert:alert_message','value=Higher BP than Normal'
0 row(s) in 0.0040 seconds

hbase(main):136:0> put 'Threshold_Reference_Table','row7','alert:alert_message','value=Low Heart Rate than Normal'
0 row(s) in 0.0020 seconds

hbase(main):137:0> put 'Threshold_Reference_Table','row8','alert:alert_message','value=Normal'
0 row(s) in 0.0030 seconds

hbase(main):138:0> put 'Threshold_Reference_Table','row9','alert:alert_message','value=Higher Heart Rate than Normal'
0 row(s) in 0.0030 seconds

hbase(main):139:0> put 'Threshold_Reference_Table','row10','alert:alert_message','value=Low BP than Normal'
0 row(s) in 0.0040 seconds

hbase(main):140:0> put 'Threshold_Reference_Table','row11','alert:alert_message','value=Normal'
0 row(s) in 0.0030 seconds

hbase(main):141:0> put 'Threshold_Reference_Table','row12','alert:alert_message','value=Higher BP than Normal'
0 row(s) in 0.0030 seconds

hbase(main):142:0> █
```

4. We can view the table using the following command:

```
t = get_table 'Threshold_Reference_Table'
```

```
hbase(main):001:0> t = get_table 'Threshold_Reference_Table'
0 row(s) in 0.0470 seconds

=> Hbase::Table - Threshold_Reference_Table
```

```
t.scan
```

```
hbase(main):002:0> t.scan
ROW                                COLUMN=CELL
row1                                column=alert:alert_flag, timestamp=1669405739600, value=value=1
row1                                column=alert:alert_message, timestamp=1669405739786, value=value=Low Heart Rate than Normal
row1                                column=attribute:attribute, timestamp=1669405598094, value=value=heartBeat
row1                                column=limit:high_age_limit, timestamp=1669405598771, value=value=40
row1                                column=limit:high_value, timestamp=1669405598841, value=value=69
row1                                column=limit:low_age_limit, timestamp=1669405598740, value=value=0
row1                                column=limit:low_value, timestamp=1669405598816, value=value=0
row10                               column=alert:alert_flag, timestamp=1669405739738, value=value=1
row10                               column=alert:alert_message, timestamp=1669405739925, value=value=Low BP than Normal
row10                               column=attribute:attribute, timestamp=1669405598583, value=value=bp
row10                               column=limit:high_age_limit, timestamp=1669405739401, value=value=100
row10                               column=limit:high_value, timestamp=1669405739453, value=value=150
row10                               column=limit:low_value, timestamp=1669405739436, value=value=0
row11                               column=alert:alert_flag, timestamp=1669405739751, value=value=0
row11                               column=alert:alert_message, timestamp=1669405739938, value=value=Normal
row11                               column=attribute:attribute, timestamp=1669405598665, value=value=bp
row11                               column=limit:high_age_limit, timestamp=1669405739488, value=value=100
row11                               column=limit:high_value, timestamp=1669405739517, value=value=180
row11                               column=limit:low_age_limit, timestamp=1669405739475, value=value=41
row11                               column=limit:low_value, timestamp=1669405739502, value=value=151
row12                               column=alert:alert_flag, timestamp=1669405739765, value=value=1
row12                               column=alert:alert_message, timestamp=1669405739952, value=value=Higher BP than Normal
row12                               column=attribute:attribute, timestamp=1669405598699, value=value=bp
row12                               column=limit:high_age_limit, timestamp=1669405739551, value=value=100
row12                               column=limit:high_value, timestamp=1669405739579, value=value=9999
row12                               column=limit:low_age_limit, timestamp=1669405739537, value=value=41
row12                               column=limit:low_value, timestamp=1669405739566, value=value=181
row2                                column=alert:alert_flag, timestamp=1669405739612, value=value=0
row2                                column=alert:alert_message, timestamp=1669405739803, value=value=Normal
row2                                column=attribute:attribute, timestamp=1669405598258, value=value=heartBeat
row2                                column=limit:high_age_limit, timestamp=1669405598909, value=value=40
row2                                column=limit:high_value, timestamp=1669405598975, value=value=78
row2                                column=limit:low_age_limit, timestamp=1669405598873, value=value=0
row2                                column=limit:low_value, timestamp=1669405598946, value=value=70
row3                                column=alert:alert_flag, timestamp=1669405739625, value=value=1
row3                                column=alert:alert_message, timestamp=1669405739828, value=value=Higher Heart Rate than Normal
row3                                column=attribute:attribute, timestamp=1669405598316, value=value=heartBeat
row3                                column=limit:high_age_limit, timestamp=1669405599067, value=value=40
row3                                column=limit:high_value, timestamp=1669405599130, value=value=9999
row3                                column=limit:low_age_limit, timestamp=1669405599008, value=value=0
row3                                column=limit:low_value, timestamp=1669405599082, value=value=79
row4                                column=alert:alert_flag, timestamp=1669405739638, value=value=1
row4                                column=alert:alert_message, timestamp=1669405739840, value=value=Low BP than Normal
row4                                column=attribute:attribute, timestamp=1669405598376, value=value=bp
row4                                column=limit:high_age_limit, timestamp=1669405599206, value=value=40
row4                                column=limit:high_value, timestamp=1669405599274, value=value=160
row4                                column=limit:low_age_limit, timestamp=1669405599162, value=value=0
row4                                column=limit:low_value, timestamp=1669405599241, value=value=0
row5                                column=alert:alert_flag, timestamp=1669405739658, value=value=0
row5                                column=alert:alert_message, timestamp=1669405739853, value=value=Normal
row5                                column=attribute:attribute, timestamp=1669405598409, value=value=bp
```

STATEMENTS TO CREATE EXTERNAL HIVE TABLE FOR PATIENTS VITAL INFORMATION

1. We can create external table for patient vital information using the following command:

```
hive> CREATE EXTERNAL TABLE PatientInfoTable(
CustomerId int,
BP int,
HeartBeat int,
Message_time timestamp)
STORED AS PARQUET;
```

```
hive> CREATE EXTERNAL TABLE PatientInfoTable (
> CustomerId int,
> BP int,
> HeartBeat int,
> Message_time timestamp)
> STORED AS PARQUET;
OK
Time taken: 0.109 seconds
```

2. For loading data from parquet files into hive table, we can use the following command:

```
load data inpath 'PatientInformation' into table PatientInfoTable;
```

```
hive> load data inpath 'PatientInfo' into table PatientInfoTable;
Loading data to table patient_db.patientinfotable
OK
Time taken: 0.432 seconds
```

3. In order to view the data in PatientInfoTable, we can use the following command:

```
hive> select * from PatientInfoTable;
```

```
hive> select * from PatientInfoTable;
OK
```

4. Data Retrieved Screenshot:

158	3	66
176	4	71
168	5	67
184	1	73
160	2	73
156	3	67
154	4	66
172	5	66
212	1	75
165	2	67
154	3	68
178	4	71
177	5	69
174	1	73
178	2	69
176	3	70

STATEMENTS TO RETRIEVE THE PATIENT'S CONTACT DETAILS USING SQOOP

1. Prior to executing sqoop job, execute the following commands:

```
wget https://de-mysql-connector.s3.amazonaws.com/mysql-connector-java-8.0.25.tar.gz
```

```
[root@ip-172-31-39-26 ~]# wget https://de-mysql-connector.s3.amazonaws.com/mysql-connector-java-8.0.25.tar.gz
--2022-11-27 20:27:39-- https://de-mysql-connector.s3.amazonaws.com/mysql-connector-java-8.0.25.tar.gz
Resolving de-mysql-connector.s3.amazonaws.com (de-mysql-connector.s3.amazonaws.com)... 52.217.233.9, 54.231.172.105, 54.231.229.65, ...
Connecting to de-mysql-connector.s3.amazonaws.com (de-mysql-connector.s3.amazonaws.com) [52.217.233.9]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4079310 (3.9M) [application/x-gzip]
Saving to: 'mysql-connector-java-8.0.25.tar.gz'
100%[=====] 4,079,310 --.-K/s in 0.05s
2022-11-27 20:27:39 (71.7 MB/s) - 'mysql-connector-java-8.0.25.tar.gz' saved [4079310/4079310]
```

```
tar -xvf mysql-connector-java-8.0.25.tar.gz
```

```
[root@ip-172-31-39-26 ~]# tar -xvf mysql-connector-java-8.0.25.tar.gz
```

```
cd mysql-connector-java-8.0.25
```

```
sudo cp mysql-connector-java-8.0.25.jar /usr/lib/sqoop/lib/
```

```
[root@ip-172-31-39-26 ~]# cd mysql-connector-java-8.0.25
[root@ip-172-31-39-26 mysql-connector-java-8.0.25]# sudo cp mysql-connector-java-8.0.25.jar /usr/lib/sqoop/lib/
```

SQOOP COMMAND

2. The following command can be used for executing Sqoop Job:

```
sqoop import \
--connect jdbc:mysql://upgraddetest.cyaieic9bmnf.us-east-1.rds.amazonaws.com/testdatabase \
--username student \
--password STUDENT123 \
--table patients_vital_info \
--direct \
-m 1 \
--hive-import \
--create-hive-table \
--hive-table Patients_Contact_Info \
--fields-terminated-by ';' \
```

```
[hadoop@ip-172-31-39-26 ~]$ sqoop import --connect jdbc:mysql://upgradtest.cyaie1c9bmhf.us-east-1.rds.amazonaws.com/testdatabase --username student --password STUDENT123 --table patients_informat
ion --direct --h1 --hive-import --create-hive-table --hive-table PatientContactInfo --fields-terminated-by ';'
Warning: /usr/lib/sqoop/.accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-log4j12-1.7.35.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/hive/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/faq.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
22/11/27 20:38:02 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
22/11/27 20:38:02 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
22/11/27 20:38:02 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
22/11/27 20:38:02 INFO tool.CodeGenTool: Beginning code generation
Loading class 'com.mysql.jdbc.Driver'. This is deprecated. The new driver class is 'com.mysql.cj.jdbc.Driver'. The driver is automatically registered via the SPI and manual loading of the driver cl
ass is generally unnecessary.
22/11/27 20:38:03 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM 'patients_information' AS t LIMIT 1
22/11/27 20:38:03 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/sqoop-hadoop/compile/5209fbc347517cb7d8102a8bd2433a40/patients_information.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
22/11/27 20:38:07 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-hadoop/compile/5209fbc347517cb7d8102a8bd2433a40/patients_information.jar
22/11/27 20:38:07 INFO manager.DirectMySQLManager: Beginning mysqldump fast path import
22/11/27 20:38:07 INFO mapreduce.ImportJobBase: Beginning import of patients information
22/11/27 20:38:07 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
22/11/27 20:38:08 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
22/11/27 20:38:08 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-39-26.ec2.internal/172.31.39.26:8032
22/11/27 20:38:08 INFO client.AHSProxy: Connecting to Application History server at ip-172-31-39-26.ec2.internal/172.31.39.26:10200
22/11/27 20:38:10 INFO db.DBInputFormat: Using read committed transaction isolation
22/11/27 20:38:10 INFO mapreduce.JobSubmitter: number of splits:1
22/11/27 20:38:11 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1669577452461_0004
22/11/27 20:38:11 INFO Conf.Configuration: resource-types.xml not found
22/11/27 20:38:11 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
22/11/27 20:38:11 INFO resource.ResourceUtils: Adding resource type - name = memory-mb, units = Mi, type = COUNTABLE
22/11/27 20:38:11 INFO resource.ResourceUtils: Adding resource type - name = vcores, units = , type = COUNTABLE
22/11/27 20:38:11 INFO impl.YarnClientImpl: Submitted application application_1669577452461_0004
22/11/27 20:38:11 INFO mapreduce.Job: The url to track the job: http://ip-172-31-39-26.ec2.internal:20888/proxy/application_1669577452461_0004/
22/11/27 20:38:11 INFO mapreduce.Job: Running job: job_1669577452461_0004
22/11/27 20:38:18 INFO mapreduce.Job: Job job_1669577452461_0004 running in uber mode : false
22/11/27 20:38:18 INFO mapreduce.Job: map 0% reduce 0%
22/11/27 20:38:24 INFO mapreduce.Job: map 100% reduce 0%
22/11/27 20:38:25 INFO mapreduce.Job: Job job_1669577452461_0004 completed successfully
```

SNAPSHOT OF DATA COLLECTED

```
[hadoop@ip-172-31-39-26 ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
hive> select * from patientcontactinfo
> ;
OK
1      Alex S   XDC test Address      8982739282      1      23      NULL
2      Sammy A New Building Address  2382739282      2      45      NULL
3      Karan C  Aws Address      8923739282      3      56      NULL
4      Dara M   India Address   2182739282      4      67      NULL
5      Pam      ABC test Address  4982739282      5      72      NULL
Time taken: 2.337 seconds, Fetched: 5 row(s)
```


STATEMENTS TO READ DATA FROM HDFS AND COMPARE WITH HBase TABLE

Execute the spark streaming application `kafka_spark_generate_alerts.py` using following command:

```
python kafka_spark_generate_alerts.py
```

STATEMENTS TO EXECUTE THE CONSUMER APPLICATION FOR SENDING ALERTS

The following steps are to be executed :-

1. From ec2-user navigate to kafka folder using **cd** command as described earlier.
2. Start the zookeeper by using the following command:

```
bin/zookeeper-server-start.sh config/zookeeper.properties
```

3. Start the kafka Server using the below command:

```
bin/kafka-server-start.sh config/server.properties
```

STATEMENT TO CREATE TOPICS

1. Use the following command to create topic in kafka server:

```
bin/kafka-topics.sh --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1 --topic PatientHealthNotification
```

2. List the topics in kafka server using following command:

```
bin/kafka-topics.sh --list --bootstrap-server localhost:9092
```

3. Run the following command to start the producer:

```
bin/kafka-console-producer.sh --broker-list localhost:9092 --topic PatientHealthNotification
```

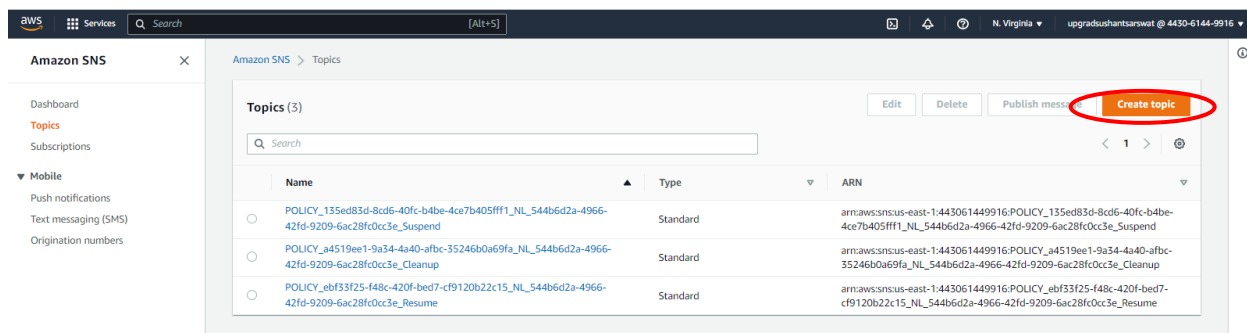
4. Open another terminal in putty and run the consumer program named `kafka_consume_alerts.py` using the following command:

```
Python kafka_consume_alerts.py
```

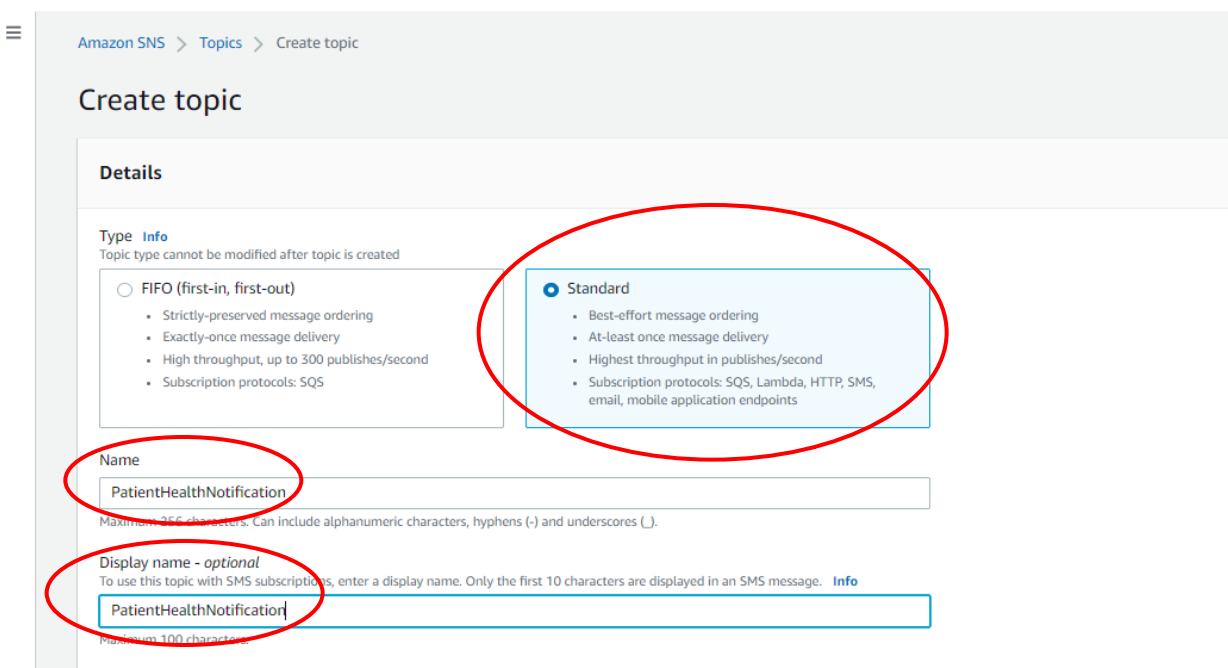
PROCEDURE TO CONFIGURE SNS SERVICE

The following steps are to be followed to execute the SNS service on AWS:

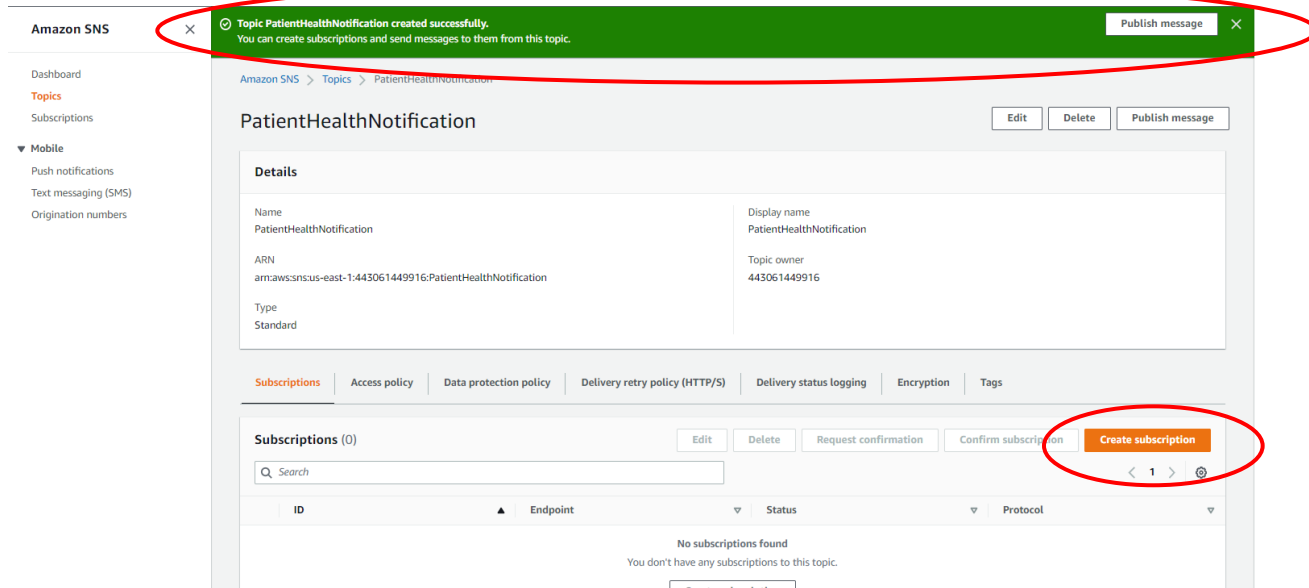
1. Open Amazon SNS through Services and click on Create Topic



2. Choose Type as Standard and enter Topic Name and Display Name

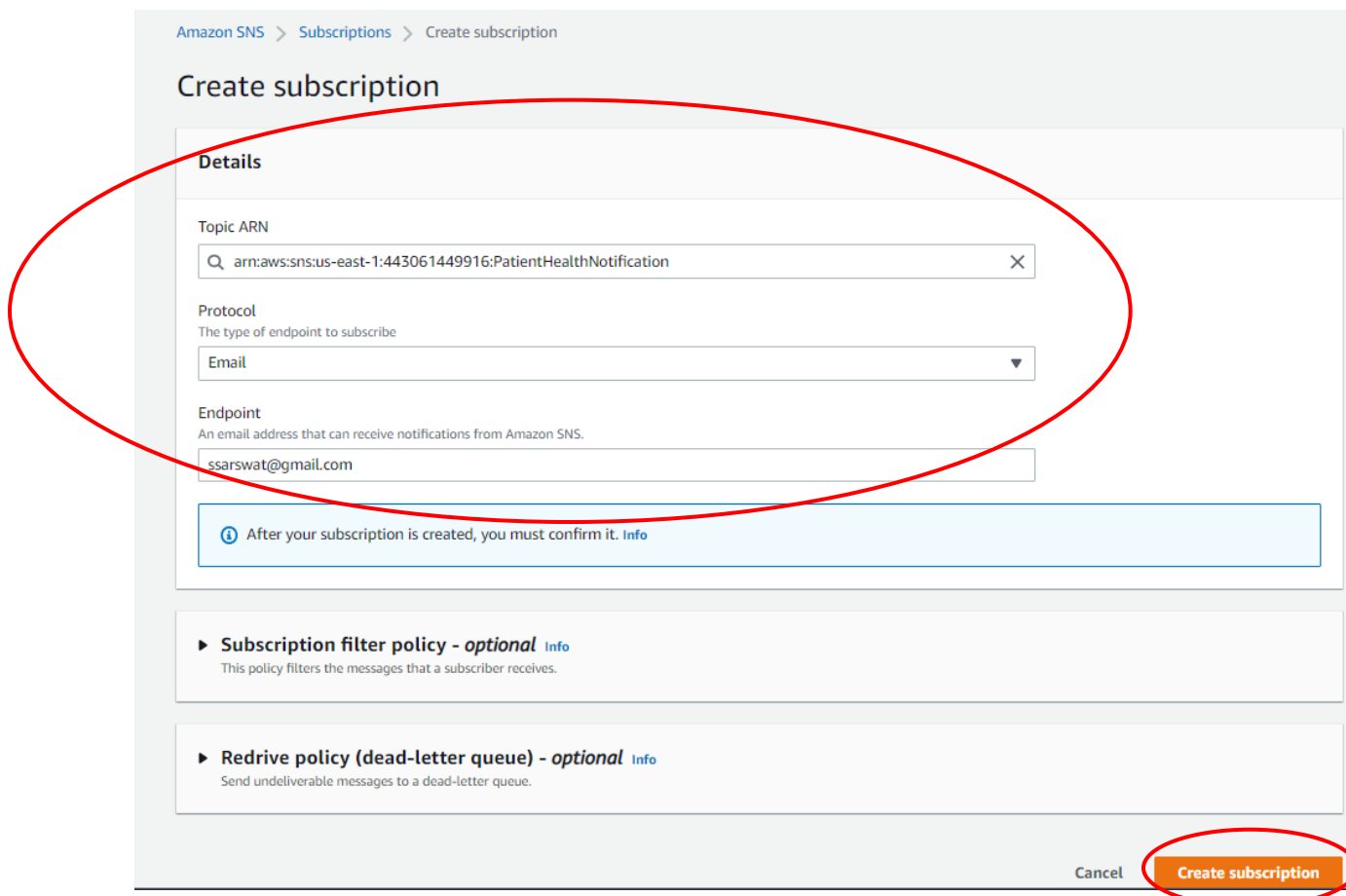


3. Check for notification for successful creation of topic and click on Create Subscription



The screenshot shows the Amazon SNS console. A green notification banner at the top states: "Topic PatientHealthNotification created successfully. You can create subscriptions and send messages to them from this topic." Below this, the "PatientHealthNotification" topic details are displayed. In the "Subscriptions" section, the "Create subscription" button is highlighted with a red circle.

4. Fill/choose the necessary details in Create Subscription and click on Create Subscription.

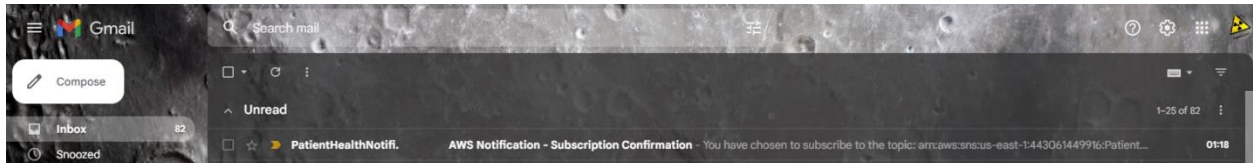


The screenshot shows the "Create subscription" form in the Amazon SNS console. The "Details" section is highlighted with a red circle and contains the following fields:

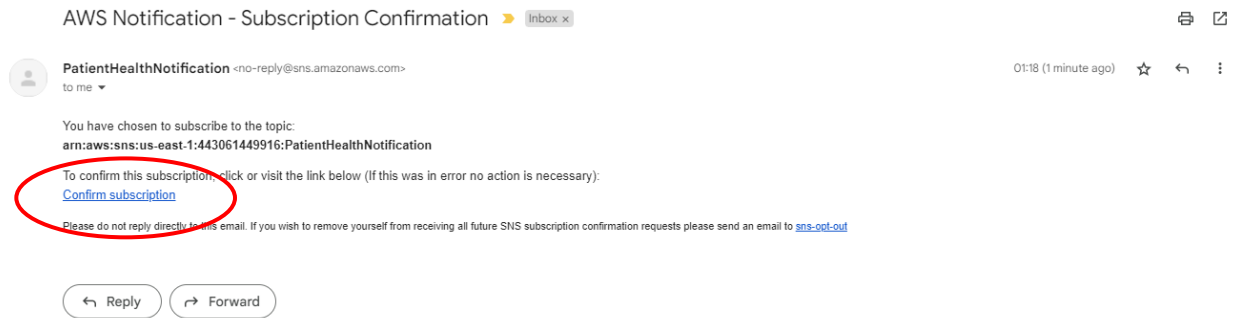
- Topic ARN:** A text input field containing "arn:aws:sns:us-east-1:443061449916:PatientHealthNotification".
- Protocol:** A dropdown menu with "Email" selected.
- Endpoint:** A text input field containing "ssarswat@gmail.com".

Below the "Details" section, there is a blue information box stating: "After your subscription is created, you must confirm it. Info". At the bottom of the form, the "Create subscription" button is highlighted with a red circle.

5. Open the email entered in the details and check for AWS Notification



6. Open the mail and confirm the subscription



7. Check for subscription confirmation message



Simple Notification Service

Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:

arn:aws:sns:us-east-1:443061449916:PatientHealthNotification:94633bb8-fc16-4fc0-9d38-d50ced5a55e0

If it was not your intention to subscribe, [click here to unsubscribe](#).