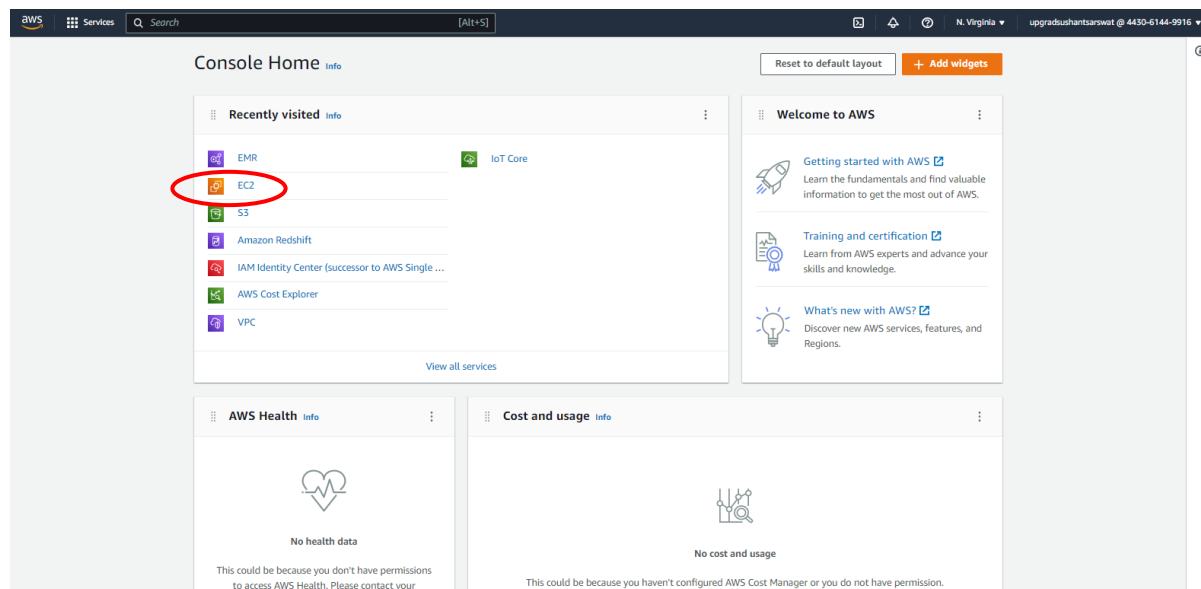


## CAPSTONE PROJECT INSTANT HEALTH ALERT SYSTEM

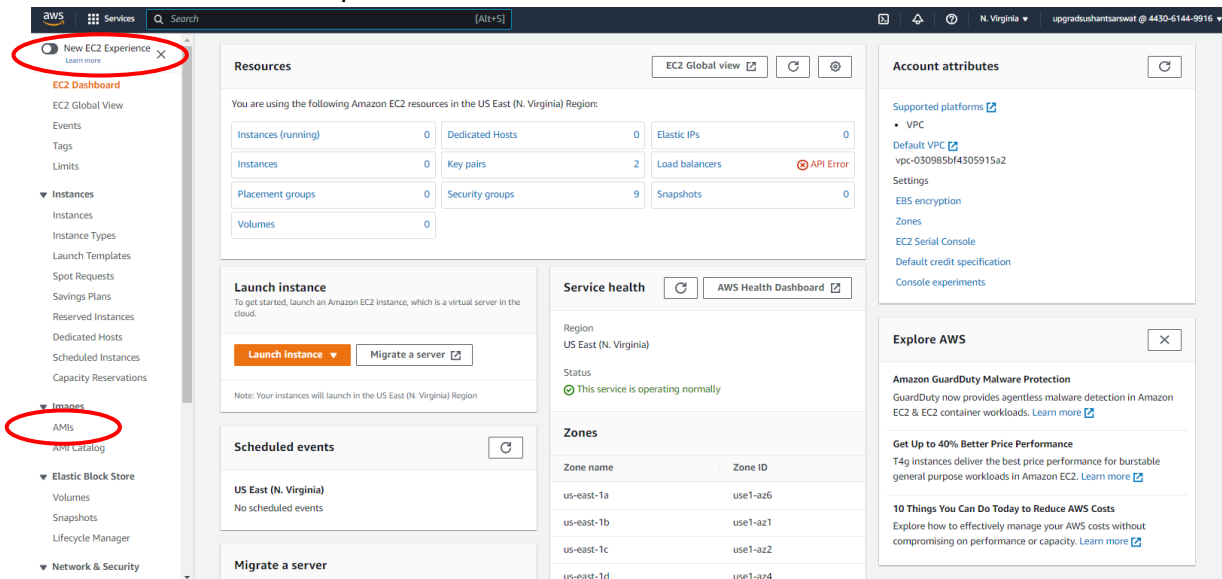
**SUBMITTED BY : Sushant Sarswat (C-38)**

### Creation and Configuration of Kafka Cluster

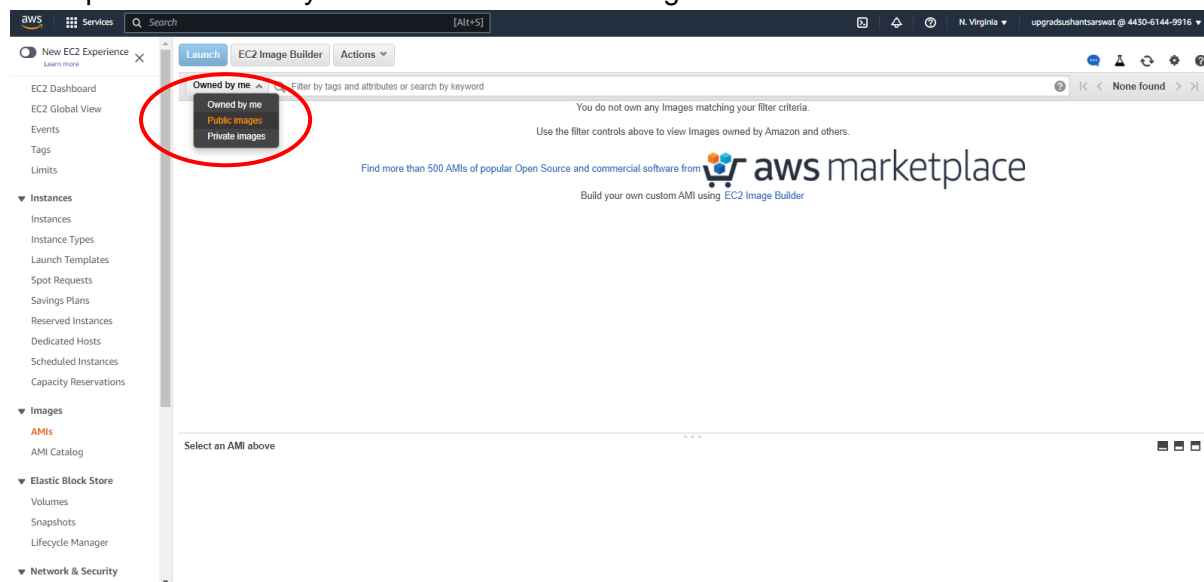
#### 1. Select EC2 instance from AWS services



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

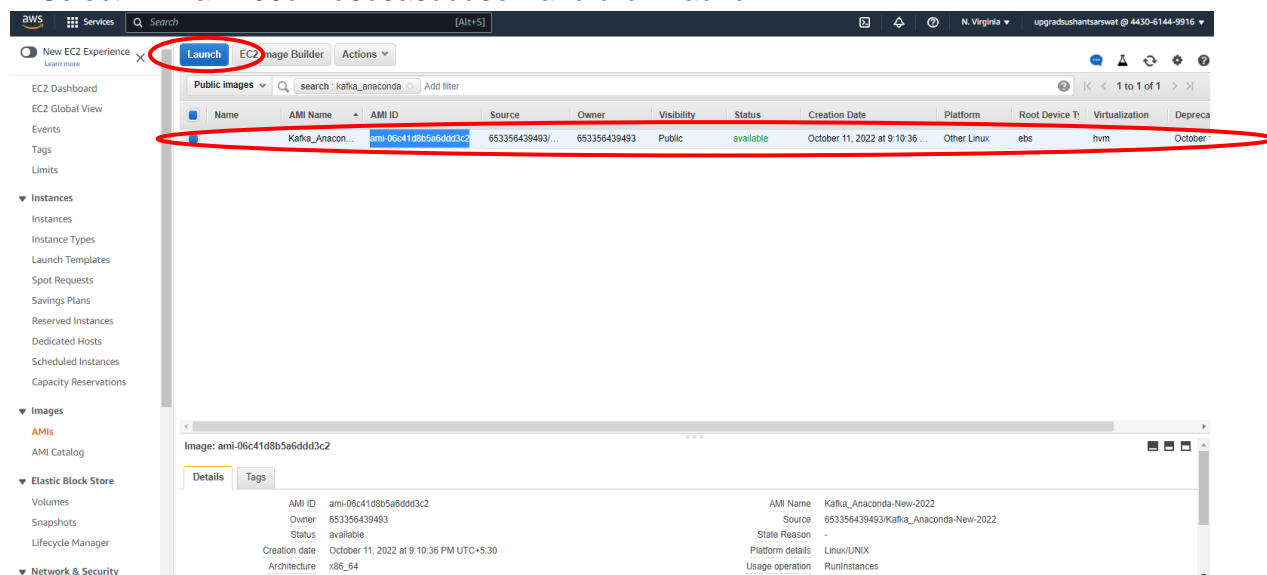


### 3. Dropdown 'Owned by Me' and select 'Public images'



The screenshot shows the AWS Management Console interface. On the left sidebar, the 'Images' section is expanded, and 'AMIs' is selected. The main content area shows the 'EC2 Image Builder' page. A dropdown menu is open, showing 'Owned by me', 'Public images', and 'Private images'. The 'Public images' option is highlighted. The search bar contains the text 'Filter by tags and attributes or search by keyword'. Below the search bar, there is a message: 'You do not own any Images matching your filter criteria. Use the filter controls above to view Images owned by Amazon and others.' The 'aws marketplace' logo is visible on the right side of the page.

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



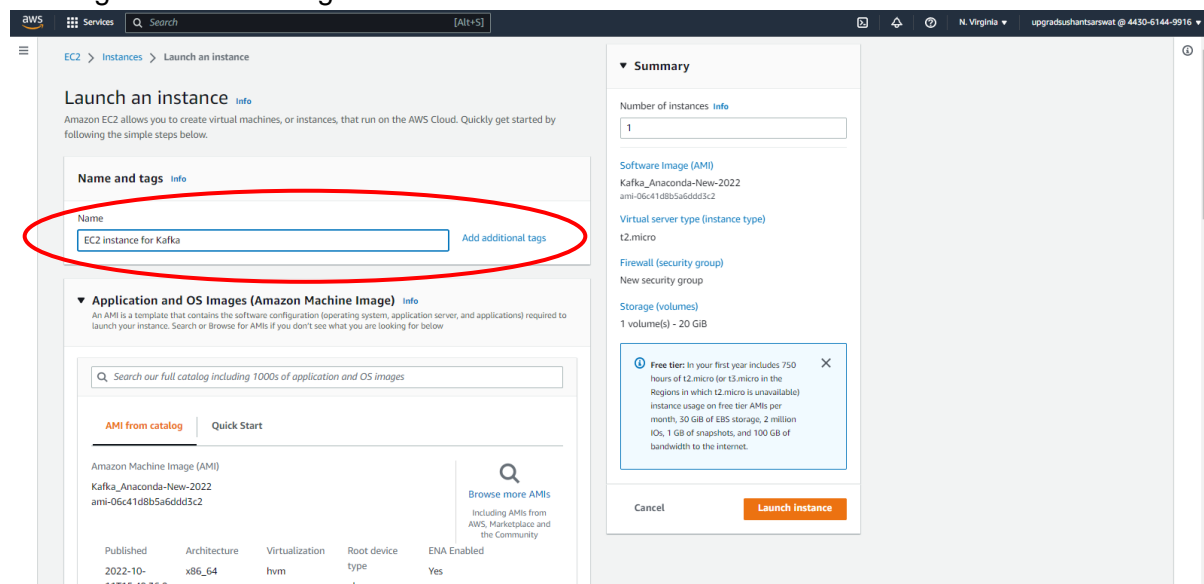
The screenshot shows the AWS Management Console interface. The 'Launch' button is highlighted in the top navigation bar. The main content area shows the 'Public images' page. A table lists the available AMIs. The AMI 'ami-06c41d8b5a6ddd3c2' is selected. The details for this AMI are shown below the table.

Name	AMI Name	AMI ID	Source	Owner	Visibility	Status	Creation Date	Platform	Root Device T1	Virtualization	Deprecate
Kafka_Anacon...	ami-06c41d8b5a6ddd3c2	ami-06c41d8b5a6ddd3c2	653356439493...	653356439493	Public	available	October 11, 2022 at 9:10:36 ...	Other Linux	ebs	hvm	October

Image: ami-06c41d8b5a6ddd3c2

Details	Tags
AMI ID	ami-06c41d8b5a6ddd3c2
Owner	653356439493
Status	available
Creation date	October 11, 2022 at 9:10:36 PM UTC+5:30
Architecture	x86_64
AMI Name	Kafka_Anaconda-New-2022
Source	653356439493/Kafka_Anaconda-New-2022
State Reason	-
Platform details	Linux/UNIX
Usage operation	RunInstances

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



**Launch an instance**

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**

Name:  Add additional tags

**Application and OS Images (Amazon Machine Image)**

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-06c41d8b5a6dd3c2

Browse more AMIs  
Including AMIs from AWS, Marketplace and the Community

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

**Summary**

Number of instances: 1

Software Image (AMI): Kafka\_Anaconda-New-2022  
ami-06c41d8b5a6dd3c2

Virtual server type (instance type): t2.micro

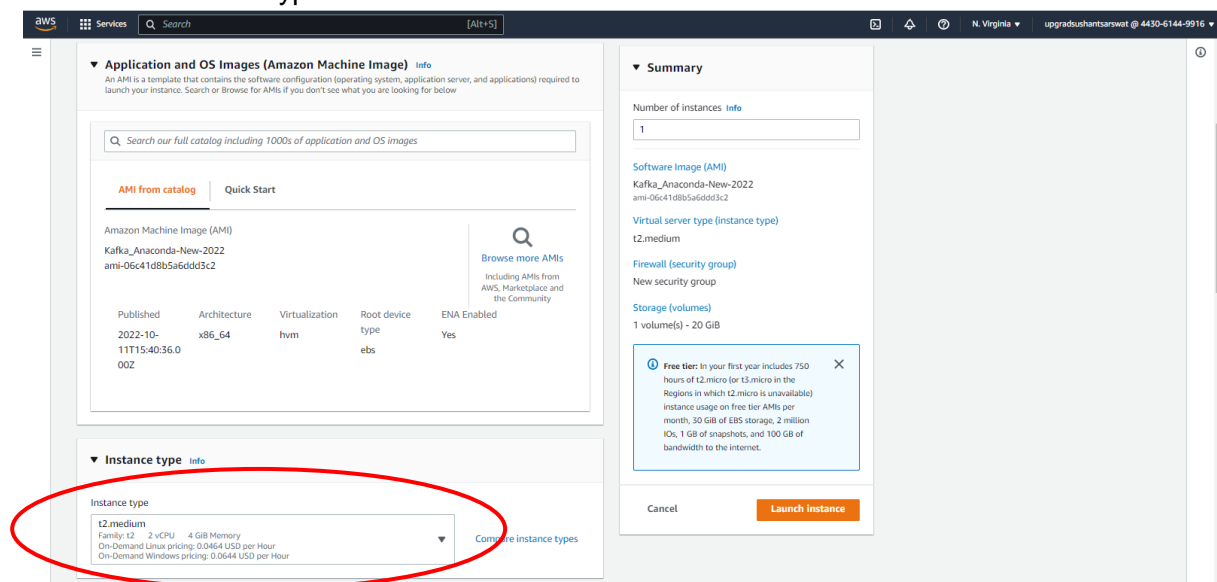
Firewall (security group): New security group

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)**

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-06c41d8b5a6dd3c2

Browse more AMIs  
Including AMIs from AWS, Marketplace and the Community

Published	Architecture	Virtualization	Root device type	ENA Enabled
2022-10-11T15:40:36.002	x86_64	hvm	ebs	Yes

**Instance type**

Instance type:  Compare instance types

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

**Summary**

Number of instances: 1

Software Image (AMI): Kafka\_Anaconda-New-2022  
ami-06c41d8b5a6dd3c2

Virtual server type (instance type): t2.medium

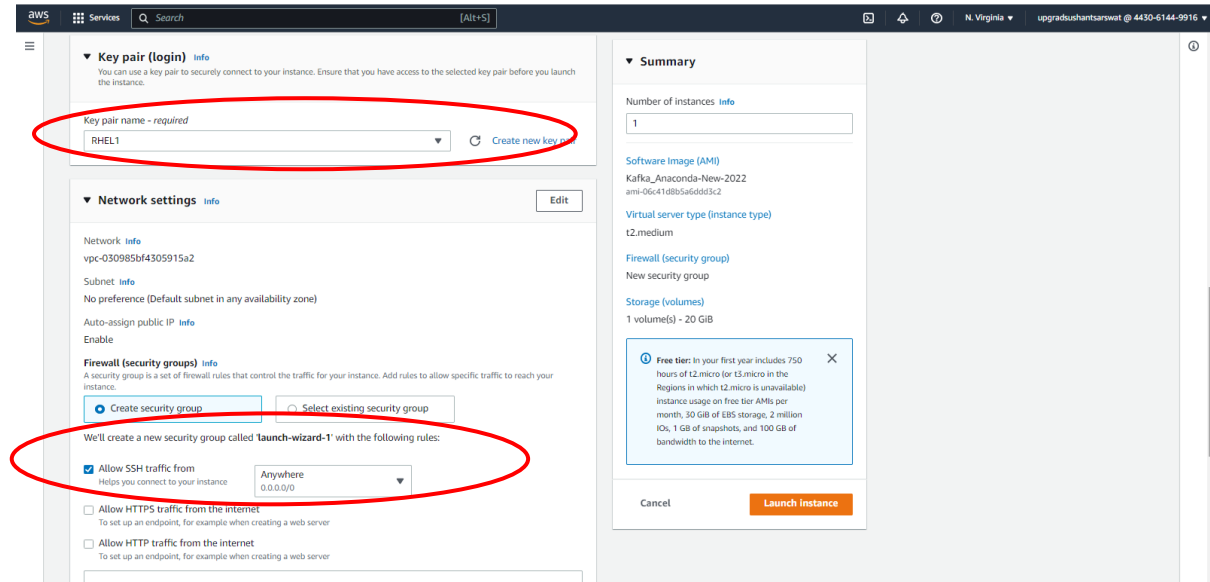
Firewall (security group): New security group

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.

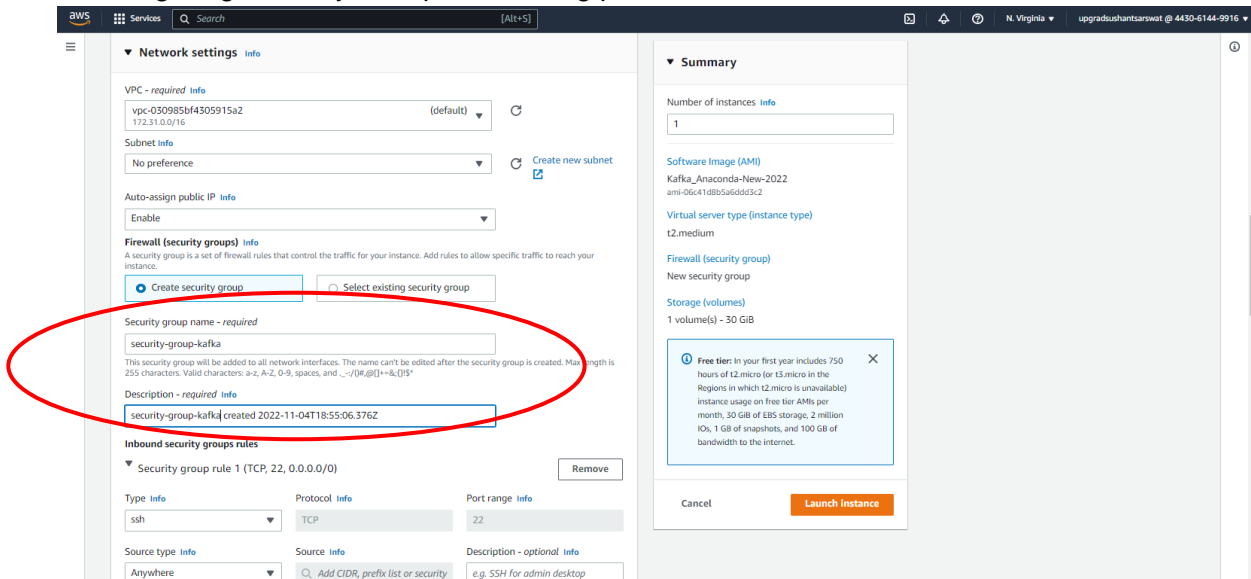
Cancel Launch instance

## 7. Key pair selection and SSH

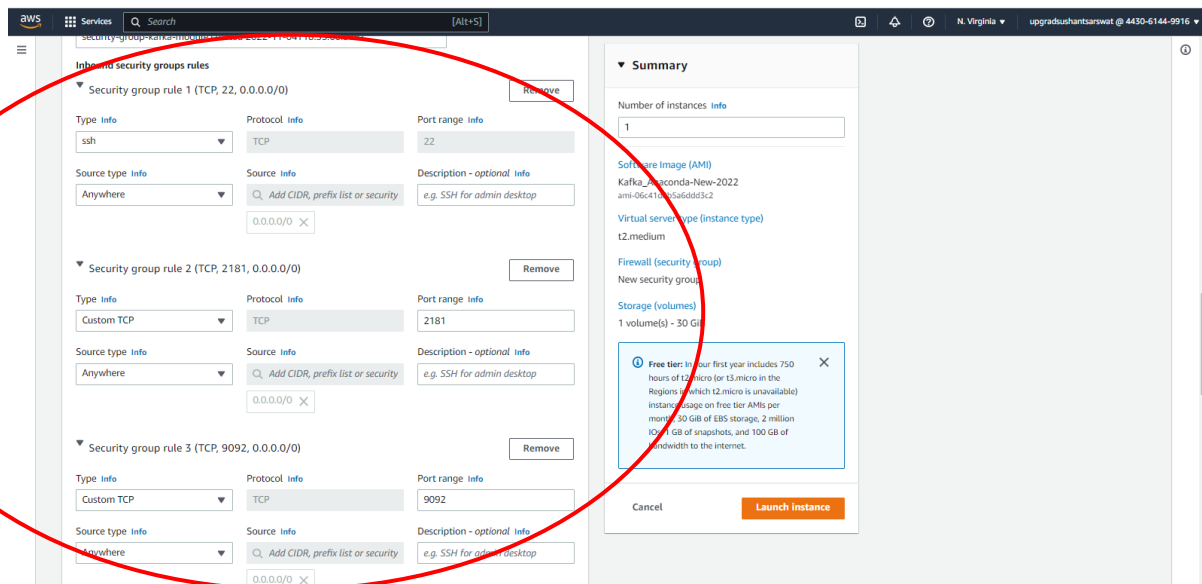


The screenshot shows the AWS Management Console interface for creating an EC2 instance. The 'Key pair (login)' section is highlighted with a red circle, showing the 'Key pair name' dropdown set to 'RHEL1'. The 'Network settings' section is also highlighted with a red circle, showing the 'Firewall (security groups)' section with the 'Create security group' radio button selected. Below this, the 'We'll create a new security group called launch-wizard-1 with the following rules:' section is highlighted, showing the 'Allow SSH traffic from' checkbox checked and the 'Anywhere' dropdown selected. The 'Summary' section on the right shows the instance configuration, including the number of instances (1), software image (Kafka\_Anaconda-New-2022), virtual server type (t2.medium), firewall (New security group), and storage (1 volume(s) - 20 GiB).

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



The screenshot shows the AWS Management Console interface for creating an EC2 instance. The 'Network settings' section is highlighted with a red circle, showing the 'VPC' dropdown set to 'vpc-030985bf4305915a2' and the 'Subnet' dropdown set to 'No preference'. The 'Firewall (security groups)' section is also highlighted with a red circle, showing the 'Create security group' radio button selected. Below this, the 'Security group name' field is set to 'security-group-kafka' and the 'Description' field is set to 'security-group-kafka created 2022-11-04T18:55:06.376Z'. The 'Inbound security groups rules' section shows a rule for 'SSH' traffic on port '22' from 'Anywhere'. The 'Summary' section on the right shows the instance configuration, including the number of instances (1), software image (Kafka\_Anaconda-New-2022), virtual server type (t2.medium), firewall (New security group), and storage (1 volume(s) - 30 GiB).



**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**

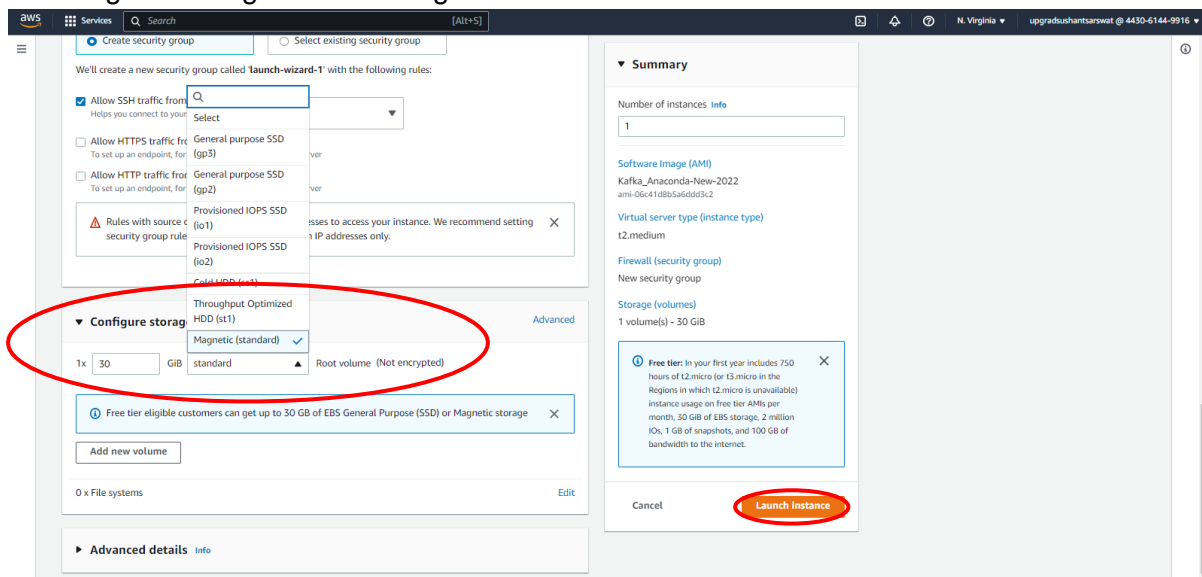
Firewall (security group): **New security group**

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**

**Advanced details**

**Summary**

Number of instances: **1**

Software Image (AMI): **Kafka\_Anaconda-New-2022**

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

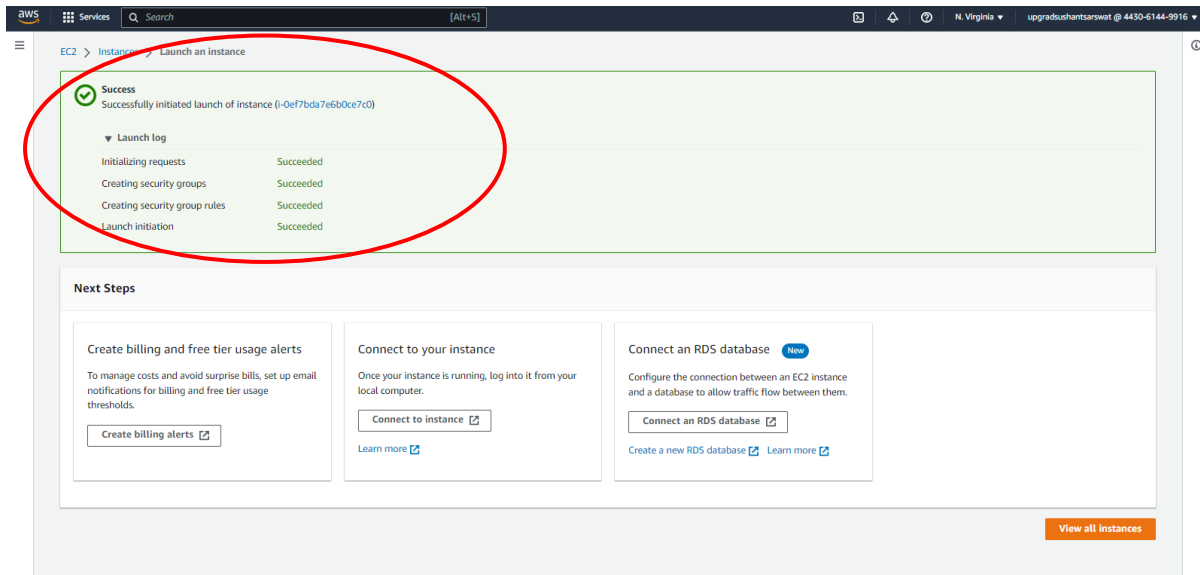
Firewall (security group): **New security group**

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**

## 10. Successful Launch of instance - View all instances



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

**Launch log**

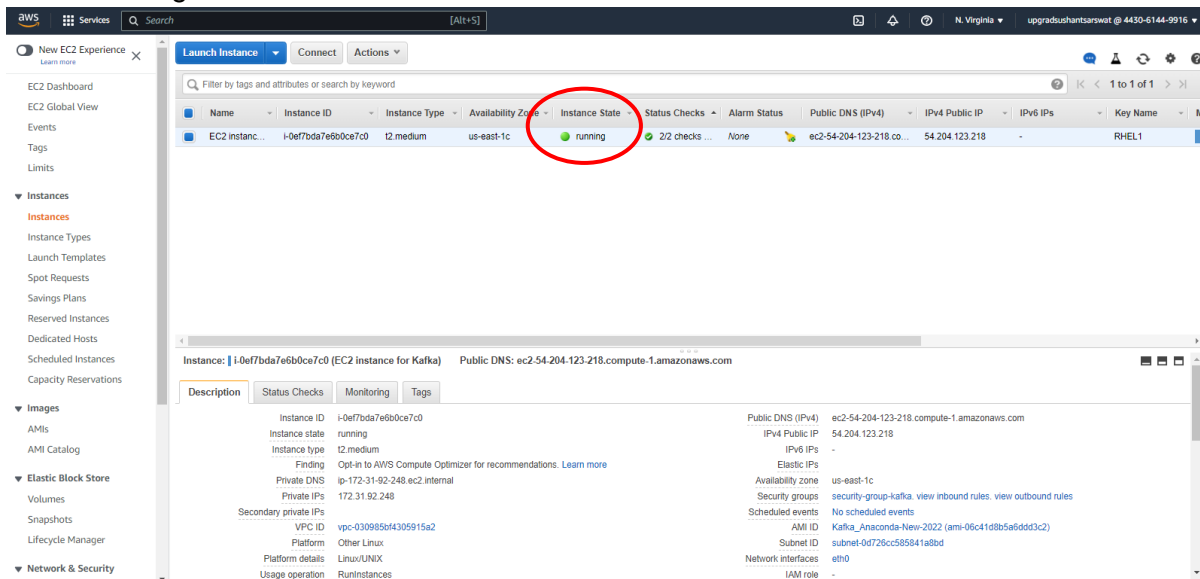
Step	Status
Initializing requests	Succeeded
Creating security groups	Succeeded
Creating security group rules	Succeeded
Launch initiation	Succeeded

**Next Steps**

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

[View all instances](#)

## 11. Monitoring 'Instance State'

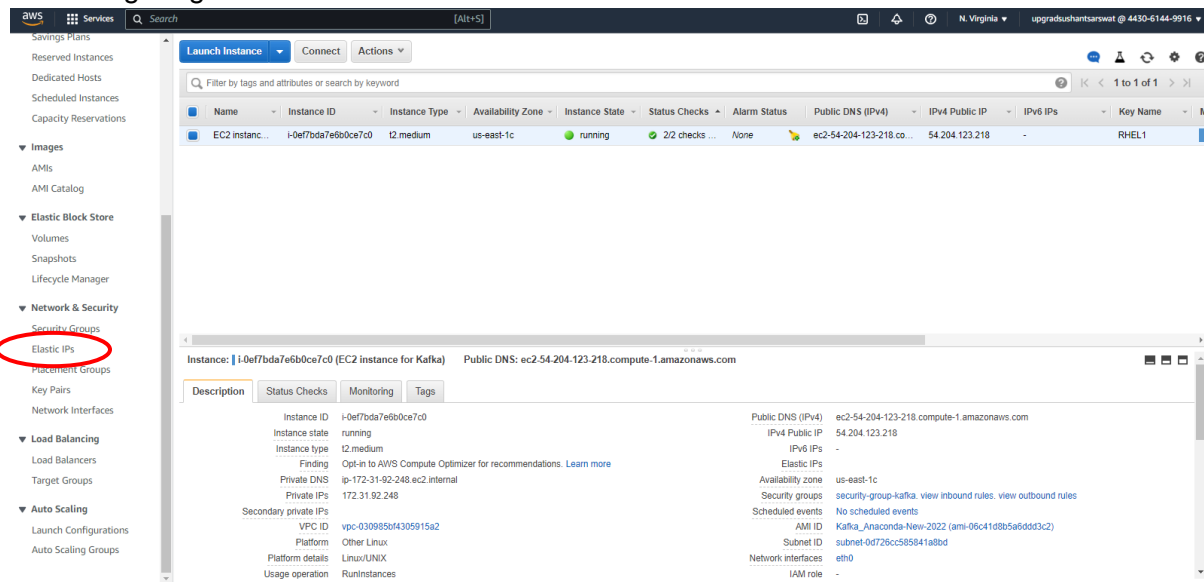


Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name
EC2 Instance...	i-0ef7bda7e6b0ce7c0	t2.medium	us-east-1c	running	2/2 checks ...	None	ec2-54-204-123-218.co...	54.204.123.218	-	RHEL1

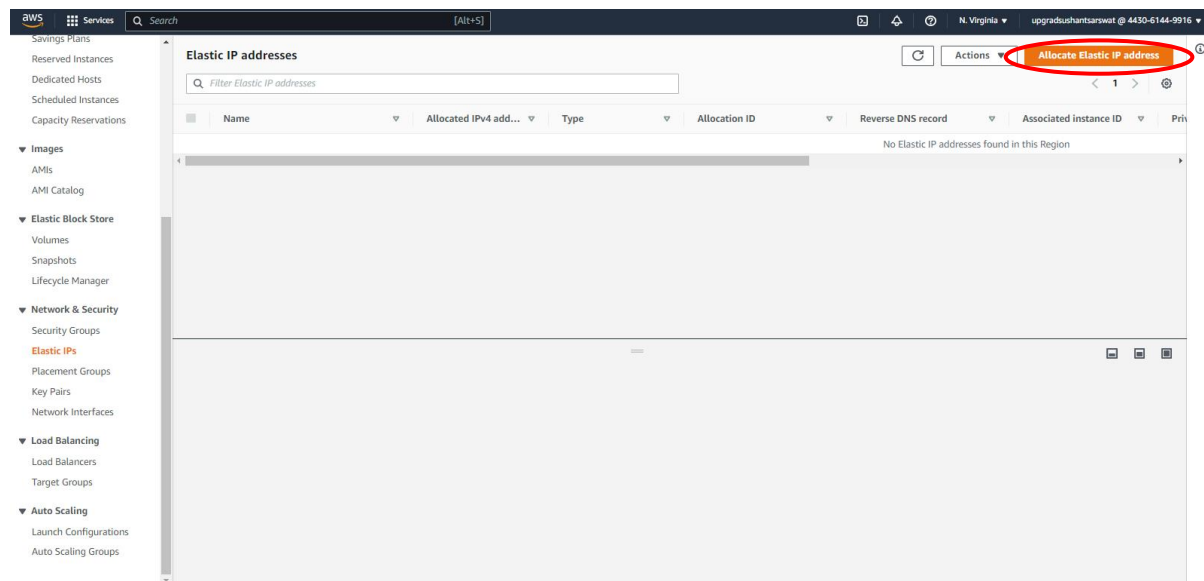
**Instance: i-0ef7bda7e6b0ce7c0 (EC2 instance for Kafka)** Public DNS: ec2-54-204-123-218.compute-1.amazonaws.com

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. <a href="#">Learn more</a>		
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		
Public DNS (IPv4)	ec2-54-204-123-218.compute-1.amazonaws.com		
IPv4 Public IP	54.204.123.218		
IPv6 IPs	-		
Elastic IPs			
Availability zone	us-east-1c		
Security groups	security-group-kafka. <a href="#">view inbound rules</a> . <a href="#">view outbound rules</a>		
Scheduled events	No scheduled events		
AMI ID	Kafka_Anaconda-New-2022 (ami-06c41d8b5a6dd3c2)		
Subnet ID	subnet-0d726cc585841a8bd		
Network interfaces	eth0		
IAM role	-		

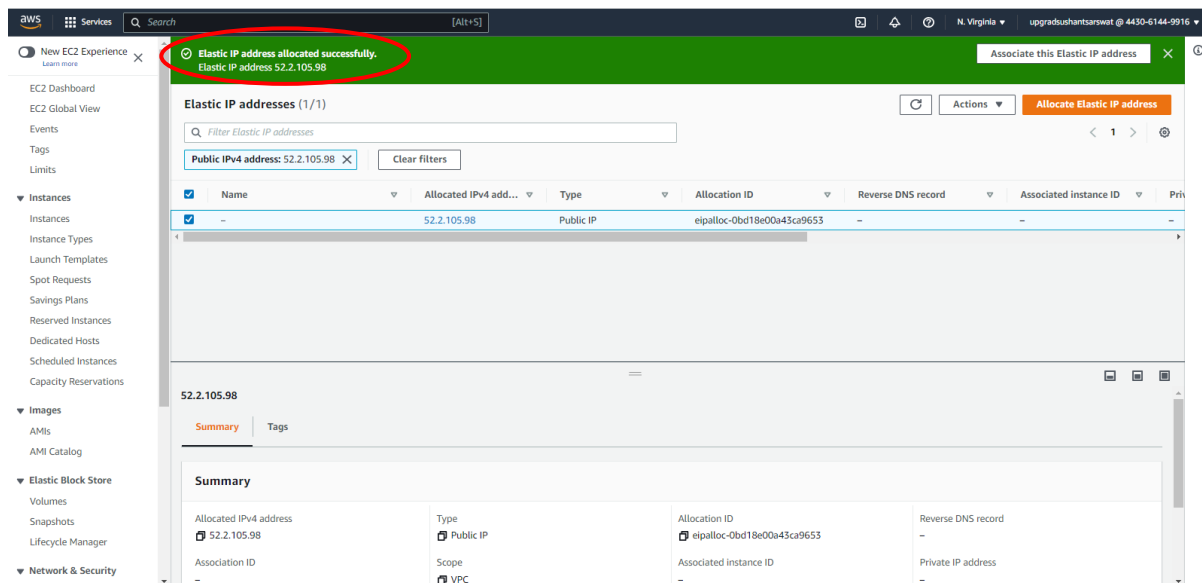
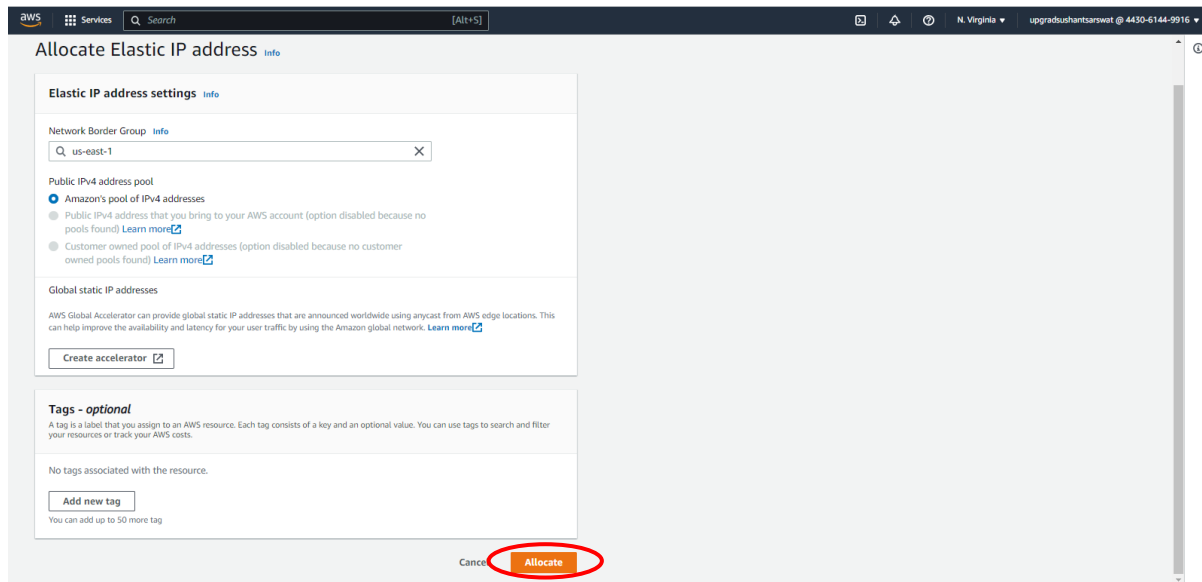
## 12. Configuring Elastic IP



The screenshot shows the AWS Management Console interface. On the left sidebar, under 'Network & Security', '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-0ef7bda7e6b0ce7c0'. The 'Public DNS (IPv4)' field shows 'ec2-54-204-123-218.compute-1.amazonaws.com'. The 'Elastic IPs' section shows 'us-east-1c' and 'security-group-kafka'. The 'IAM role' is listed as '-'. The 'Actions' button in the top right corner is highlighted with a red circle.



The screenshot shows the 'Elastic IP addresses' page in the AWS Management Console. The 'Actions' button in the top right corner is highlighted with a red circle, and the 'Allocate Elastic IP address' option is selected. The main content area shows a table with columns: Name, Allocated IPv4 address, Type, Allocation ID, Reverse DNS record, and Associated instance ID. The table is currently empty, displaying '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

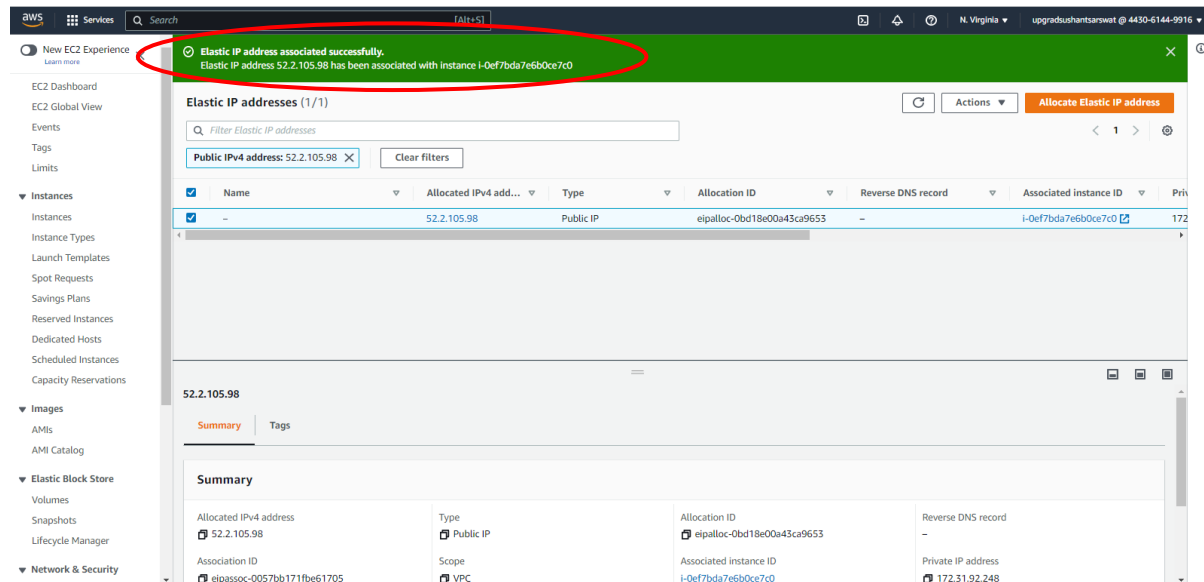
**Instance**  
Choose an instance  
i-0ef7bda7e6b0ce7c0 (EC2 instance for Kafka) - running

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

**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

**Associate**

## 14. Elastic IP address associated successfully



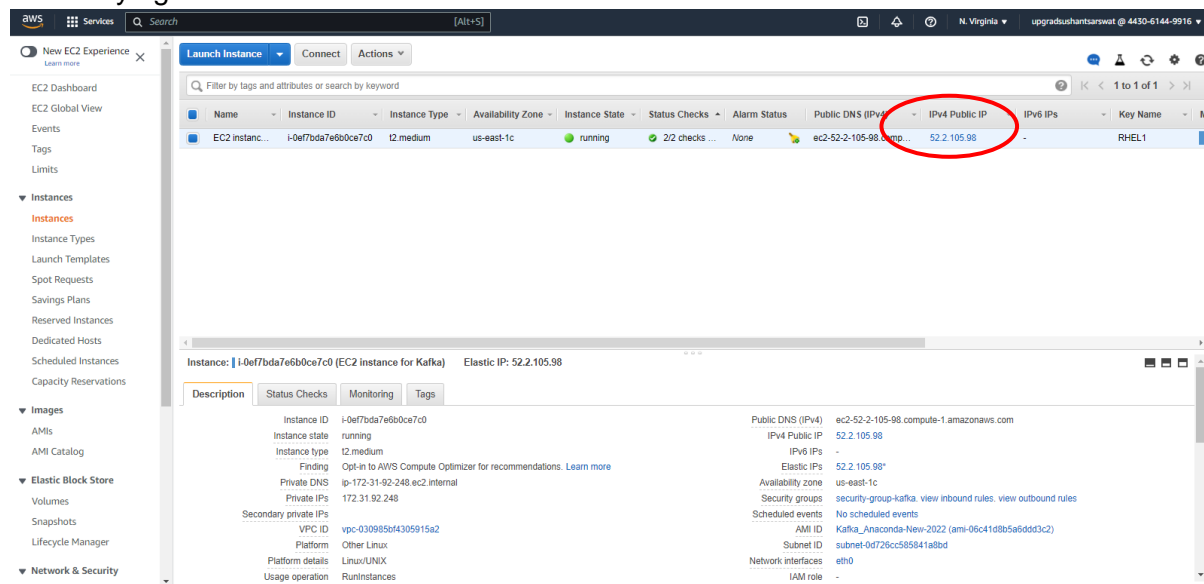
The screenshot shows the AWS Elastic IP console. A green notification banner at the top states: "Elastic IP address associated successfully. Elastic IP address 52.2.105.98 has been associated with instance i-0ef7bda7e6b0ce7c0". Below this, the "Elastic IP addresses (1/1)" table shows one entry:

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 this Elastic IP provides more 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 table shows one instance:

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

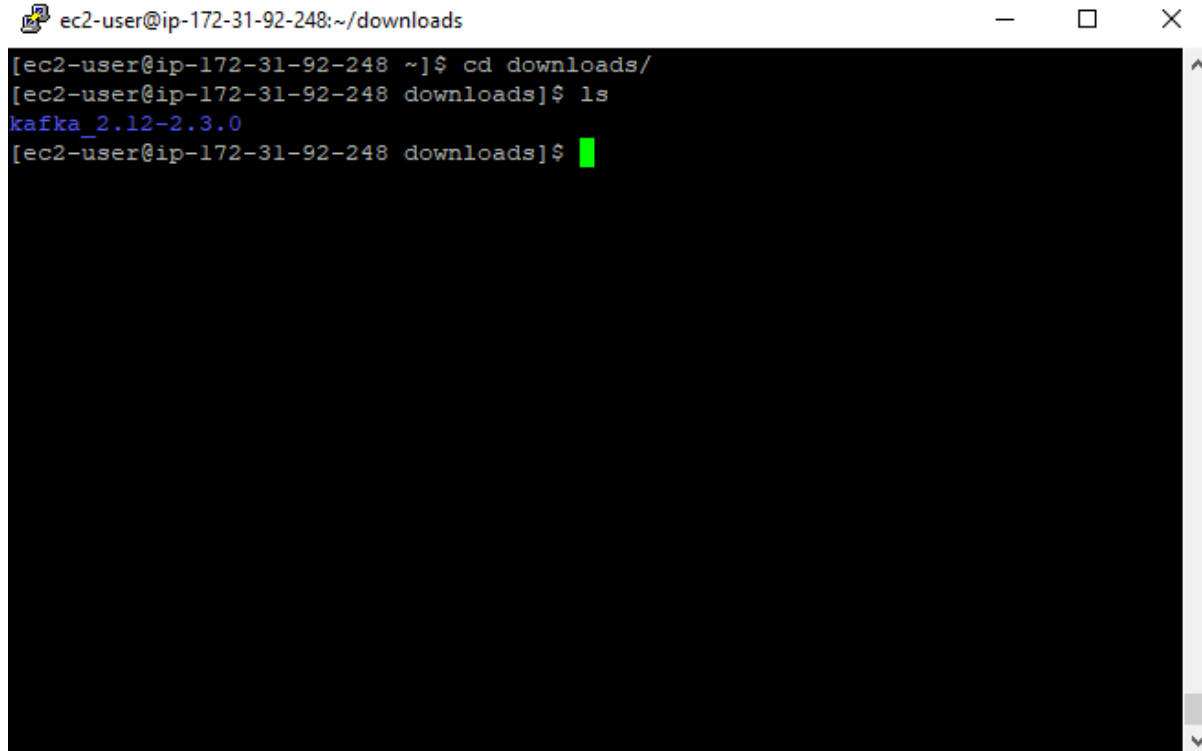
```
ec2-user@ip-172-31-92-248:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
Last login: Sat Sep 19 05:18:08 2020 from 103.212.145.122  
  
  _ | _ | _ )  
  _ | ( _ | /  Amazon Linux 2 AMI  
  _ | \ _ | _ |  
  
https://aws.amazon.com/amazon-linux-2/  
64 package(s) 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'

```
ec2-user@ip-172-31-92-248:~  
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-filessystem.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

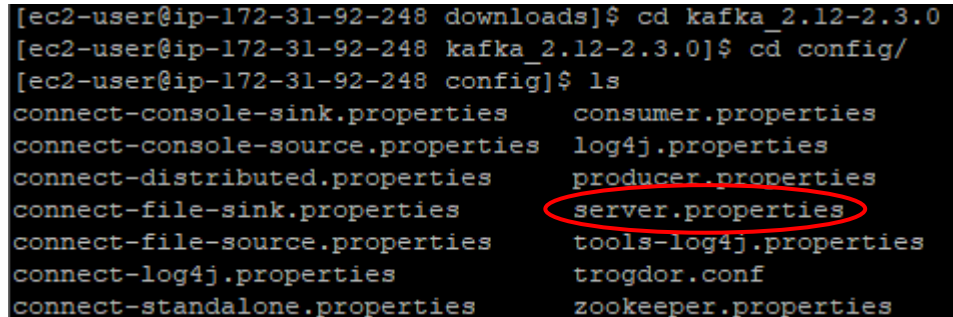
```
cd downloads/  
ls
```



```
ec2-user@ip-172-31-92-248: ~/downloads  
[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
```



```
[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

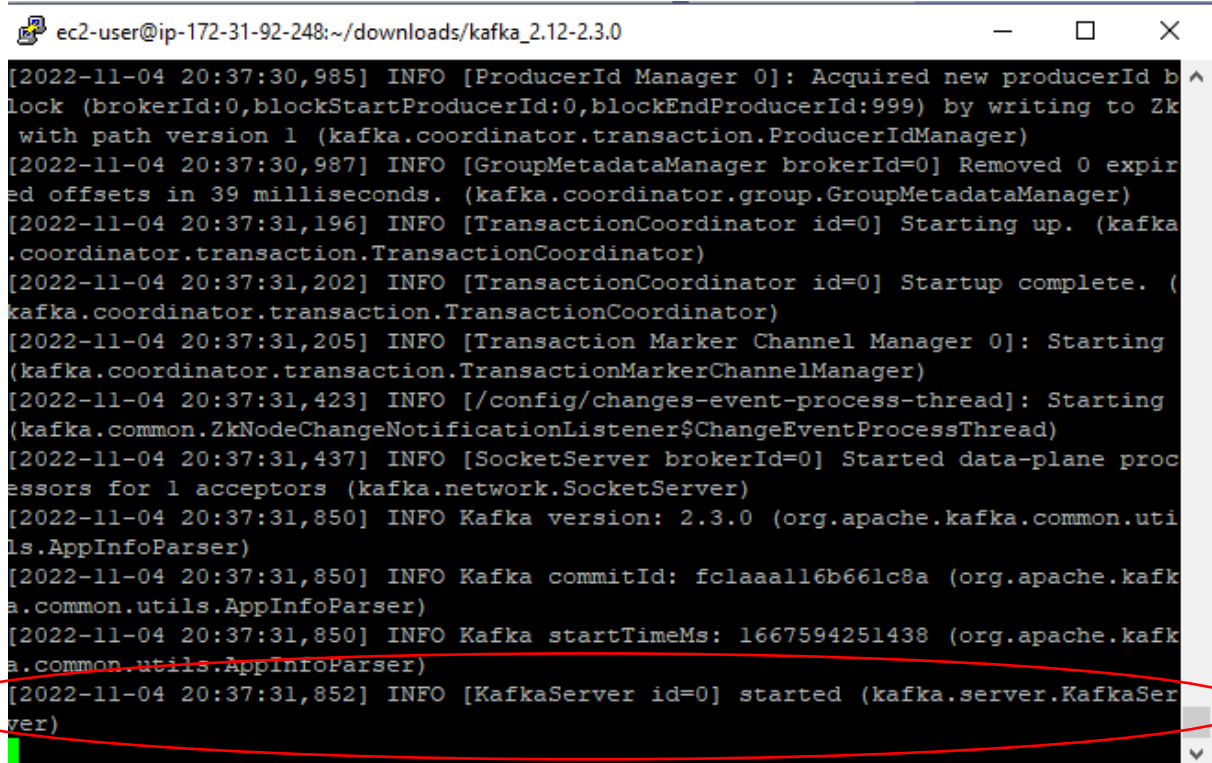
```
[ec2-user@ip-172-31-92-248 kafka_2.12-2.3.0]$ bin/zookeeper-server-start.sh config/zookeeper.properties
[2022-11-04 20:27:45,623] INFO Server environment:os.name=Linux (org.apache.zookeeper.server.ZooKeeperServer)
[2022-11-04 20:27:45,623] INFO Server environment:os.arch=amd64 (org.apache.zookeeper.server.ZooKeeperServer)
[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
```

```

ec2-user@ip-172-31-92-248:~/downloads/kafka_2.12-2.3.0
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Fri Nov  4 20:39:27 2022 from 103.122.62.29

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

https://aws.amazon.com/amazon-linux-2/
[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]$ cd kafka_2.12-2.3.0
[ec2-user@ip-172-31-92-248 kafka_2.12-2.3.0]$ bin/kafka-topics.sh --list --bootstrap-server localhost:9092

[ec2-user@ip-172-31-92-248 kafka_2.12-2.3.0]$ bin/kafka-topics.sh --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1 --topic PatientInformation

```

## 23. STATEMENT TO LIST TOPICS

To list the topics in kafka server ,the command used is :

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

```
ec2-user@ip-172-31-92-248:~/downloads/kafka_2.12-2.3.0
```

```
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Fri Nov  4 20:39:27 2022 from 103.122.62.29

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

https://aws.amazon.com/amazon-linux-2/
[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]$ cd kafka_2.12-2.3.0
[ec2-user@ip-172-31-92-248 kafka_2.12-2.3.0]$ bin/kafka-topics.sh --list --bootstrap-server localhost:9092

[ec2-user@ip-172-31-92-248 kafka_2.12-2.3.0]$ bin/kafka-topics.sh --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1 --topic PatientInformation
[ec2-user@ip-172-31-92-248 kafka_2.12-2.3.0]$ bin/kafka-topics.sh --list --bootstrap-server localhost:9092
PatientInformation
[ec2-user@ip-172-31-92-248 kafka_2.12-2.3.0]$
```



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
```