

## EC2 INSTANCE SETUP FOR KAFKA

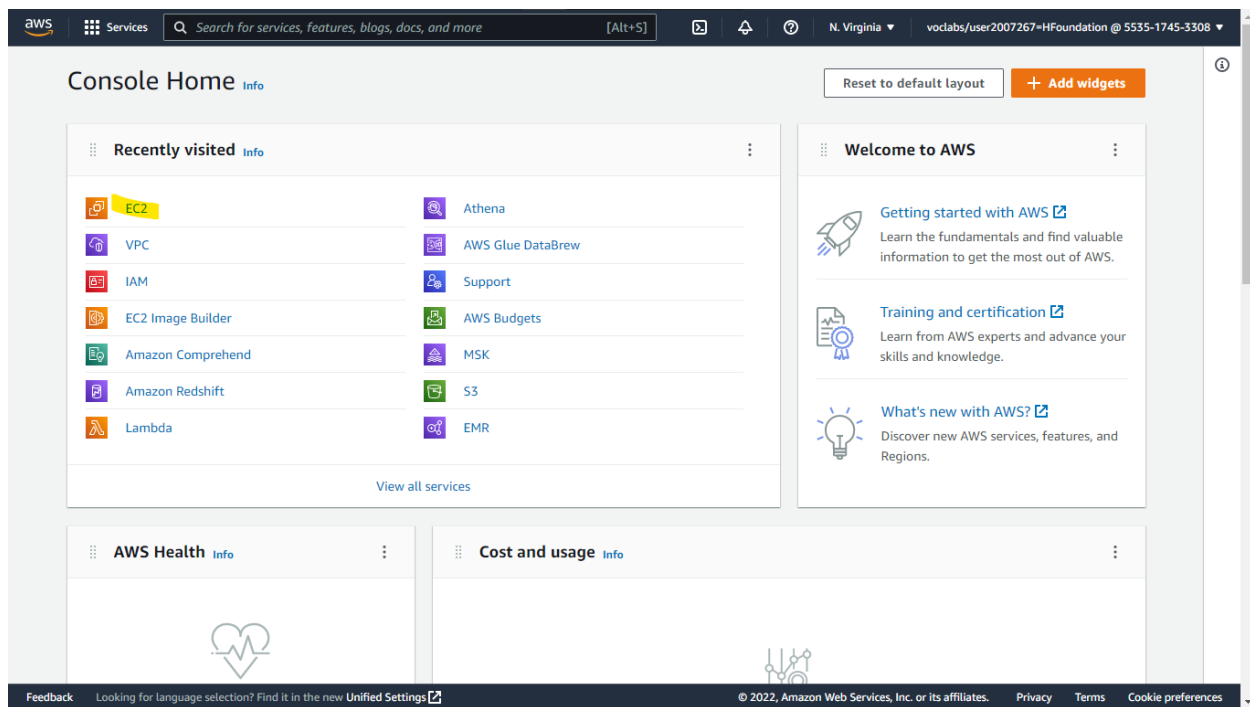
This document guides you through the setup of an EC2 instance which has been set up exclusively for Kafka. The following services have already been installed in the EC2 instance:

1. Zookeeper
2. Kafka
3. Anaconda (Jupyter Notebook)
4. Java 1.8

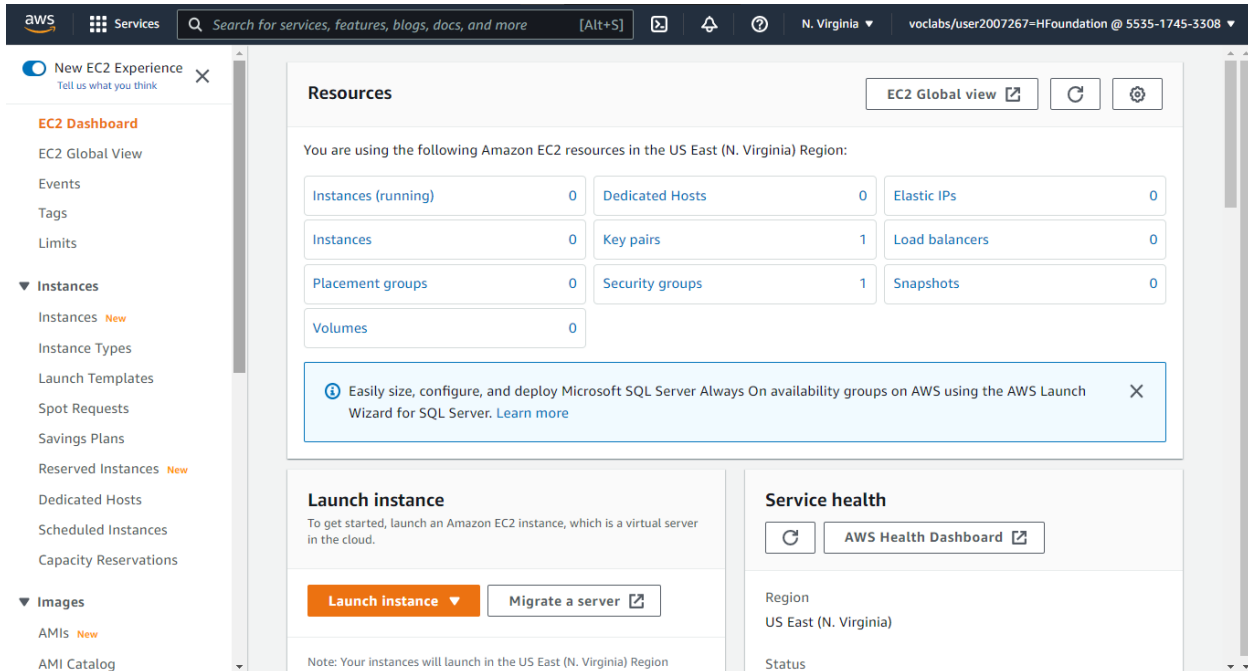
You don't need to install all the above services again. You just have to install OS image into your EC2 instance that's been published as a public AMI.

The steps to set up the EC2 instance are as follows:

- Go to the AWS Management Console and click on "All Services" and then click on "EC2" or search for the service in the search bar.



- Once you click on that, you will be redirected to a new page, shown below. Click on the “Launch Instance” button.

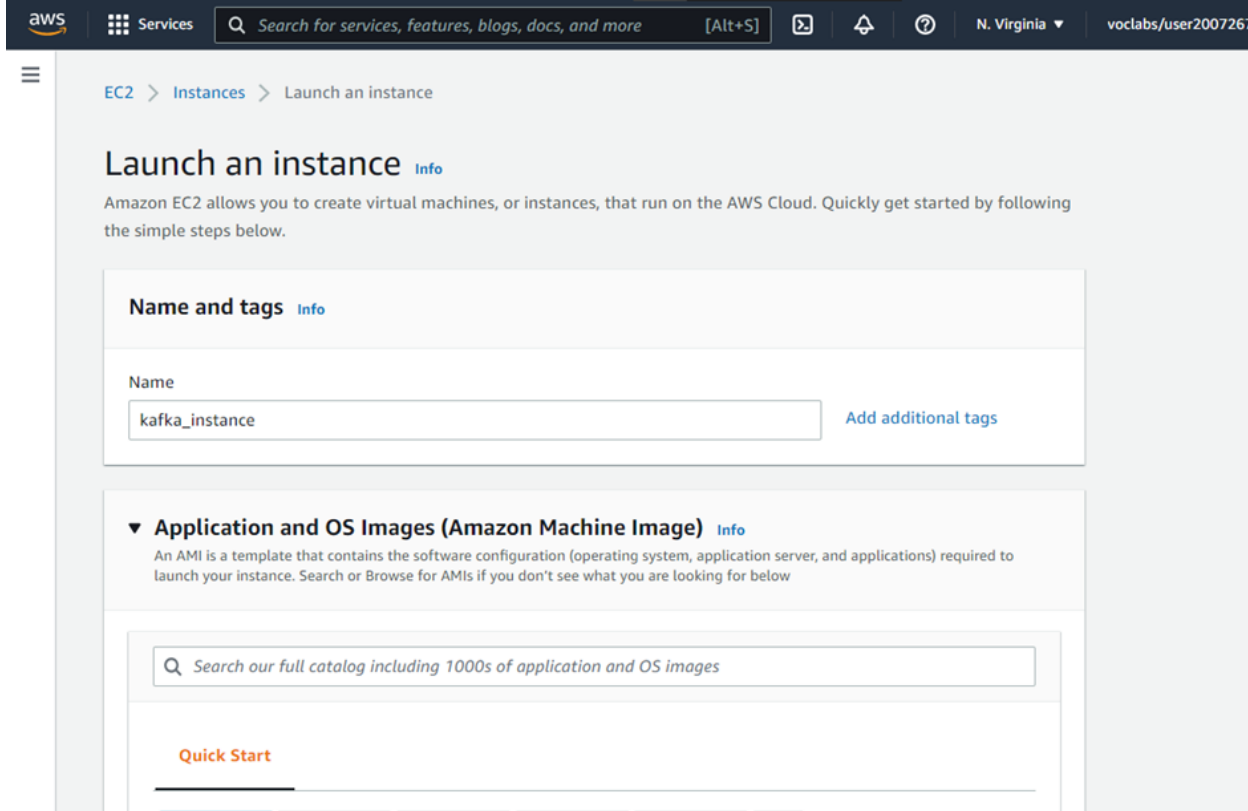


The screenshot shows the AWS Management Console interface. On the left is a navigation sidebar with options like 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', 'Instances', 'Images', 'AMIs', and 'AMI Catalog'. The main content area is titled 'Resources' and shows a summary of EC2 resources in the 'US East (N. Virginia) Region'. The summary includes:

Resource	Count
Instances (running)	0
Dedicated Hosts	0
Elastic IPs	0
Instances	0
Key pairs	1
Load balancers	0
Placement groups	0
Security groups	1
Snapshots	0
Volumes	0

Below the summary, there is a 'Launch instance' section with a 'Launch instance' button and a 'Migrate a server' link. A 'Service health' section shows the 'AWS Health Dashboard' link. A notification banner at the top right says: 'Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more'.

- You'll be redirected the EC2 instance launch page.



The screenshot shows the 'Launch an instance' page in the AWS Management Console. The page has a breadcrumb trail: 'EC2 > Instances > Launch an instance'. The main heading is 'Launch an instance' with an 'Info' link. Below the heading is a paragraph: '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.'

The first section is 'Name and tags' with an 'Info' link. It contains a 'Name' label and a text input field with the value 'kafka\_instance'. To the right of the input field is a link 'Add additional tags'.

The second section is 'Application and OS Images (Amazon Machine Image)' with an 'Info' link. It contains a paragraph: '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'. Below this paragraph is a search bar with the placeholder text: 'Search our full catalog including 1000s of application and OS images'. At the bottom of the page, there is a 'Quick Start' section.

- In the Application and OS Images section, you will need to select the Image to be used for the EC2 instance.  
Click on “**Browse more AMIs**”.

### ▼ 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

#### Quick Start

Amazon Linux  
aws

macOS  
Mac


Ubuntu  
ubuntu

Windows  
Microsoft

Red Hat  
Red Hat

⋮

➤

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

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type  
ami-026b57f3c383c2eec (64-bit (x86)) / ami-0636eac5d73e0e5d7 (64-bit (Arm))  
Virtualization: hvm    ENA enabled: true    Root device type: ebs

Free tier eligible ▼

Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20220912.1 x86\_64 HVM gp2

Architecture	AMI ID	
64-bit (x86) ▼	ami-026b57f3c383c2eec	Verified provider

- In the search box that appears at the top, copy and paste the following AMI id and press enter: **ami-06c41d8b5a6ddd3c2**  
In the “Community AMIs” tab, you will find the AMI with the following AMI Name: **Kafka\_Anaconda-New-2022**  
Click on the “**Select**” button to choose the image.

## Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

X
▼

**Quickstart AMIs (0)**  
Commonly used AMIs

**My AMIs (0)**  
Created by me

**AWS Marketplace AMIs (6181)**  
AWS & trusted third-party AMIs

**Community AMIs (1)**  
Published by anyone

**Refine results**  
  

Clear all filters

▼ Operating system

▼ Linux/Unix

☐ All Linux/Unix  
☐ Amazon Linux  
☐ CentOS  
☐ Debian  
☐ Fedora  
☐ Gentoo  
☐ macOS  
☐ openSUSE

ami-06c41d8b5a6ddd3c2 (1 filtered, 1 unfiltered)

**Community AMIs**  
 Community AMIs contain all AMIs that are public, therefore anyone can publish an AMI and it will show in this catalog. This catalog can also contain paid products. When using community AMIs it is best practice to ensure you know and trust the publisher before launching an AMI.

**Kafka\_Anaconda-New-2022**  
 ami-06c41d8b5a6ddd3c2  
 Kafka\_Anaconda-New-2022  
Platform: Other Linux Architecture: x86\_64 Owner: 653356439493 Publish date: 2022-10-11 Root device type: ebs Virtualization: hvm ENA enabled: Yes

Select

The following results for "ami-06c41d8b5a6ddd3c2" were found in other categories  

- 6181 results in AWS Marketplace AMIs  
AWS Marketplace AMIs are AMIs that are published by AWS & trusted third-parties

- Once you hit "Select" you should get the screen shown below:

▼ **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

**AMI from catalog**

**Quick Start**

**Amazon Machine Image (AMI)**  
**Kafka\_Anaconda-New-2022**  
**ami-06c41d8b5a6ddd3c2**

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

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

- Select the **General Purpose t2.medium** type EC2 instance, as shown in the image below. Using any other instance type, such as t2.micro which has 1 vCPU and 1 GB memory, may lead to memory errors while working with Kafka. You also need to select the Key Pair to login to the instance via SSH.

**Description**  
Amazon Linux 2 Kernel 5.10 AMI 2.0.20220912.1 x86\_64 HVM gp2

**Architecture**  
64-bit (x86)

**AMI ID**  
ami-026b57f3c383c2eec Verified provider

**▼ 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](#)

**▼ 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 pair before you launch the instance.

**Key pair name - required**  
 [Create new key pair](#)

**▼ Summary**

**Number of instances** [Info](#)

**Software Image (AMI)**  
Amazon Linux 2 Kernel 5.10 AMI...[read more](#)  
ami-026b57f3c383c2eec

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

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

**Storage (volumes)**  
1 volume(s) - 8 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 IOs, 1 GB of snapshots, and

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

- In the “Network Settings” section, go with the default security group. Make sure that the option “Allow SSH traffic from Anywhere” is ticked. This will ensure that you’re able to SSH into the instance from your SSH client. The settings for this will be modified in the later steps.

**▼ Network settings** [Info](#) [Edit](#)

**Network** [Info](#)  
vpc-0823844c6db5187b2

**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

☐ 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

**Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.**

**▼ Summary**

**Number of instances** [Info](#)

**Software Image (AMI)**  
Amazon Linux 2 Kernel 5.10 AMI...[read more](#)  
ami-026b57f3c383c2eec

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

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

**Storage (volumes)**  
1 volume(s) - 8 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 IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

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

- In the “Configure storage” settings, you need to enter the volume size as **30 GiB** and volume type as **standard (magnetic)**, as shown in the image below. Once the settings have been updated, click on the “Launch Instance” button to create the instance.

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Amazon Linux 2 Kernel 5.10 AMI...[read more](#)  
ami-026b57f3c383c2e0c

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 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel **Launch Instance**

- You'll need to wait for a while until the instance is created. Once finished, click on the “View all Instances” button.

Success  
Successfully initiated launch of instance (i-053b5ce3f4eac049e)  
[Launch log](#)

**Next Steps**

**Get notified of estimated charges**  
[Create billing alerts](#) to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier)

**How to connect to your instance**  
Your instance is launching and it might be a few minutes until it is in the running state, when it will be ready for you to use  
Click [View Instances](#) to monitor your instance's status. Once your instance is in the 'running' state, you can connect to it from the Instances screen. Find out [how to connect to your instance](#)

[View more resources to get you started](#)

**View all instances**

- You'll be redirected to a new page where you can view all running EC2 instances. Click on the security tab and the security group ID.

The screenshot shows the AWS Management Console interface. At the top, there's a search bar and navigation tabs. The main content area displays the 'Instances (1/1)' page. A table lists the instance 'kafka\_instance' with ID 'i-0b904d6e0c40a2061', state 'Running', type 't2.medium', and status 'Initializing'. Below the table, the 'Instance: i-0b904d6e0c40a2061 (kafka\_instance)' details are shown. The 'Security' tab is selected, displaying the 'Security details' section. It shows the IAM Role as '-', Owner ID as '553517453308', and Launch time as 'Fri Sep 23 2022 16:00:05 GMT+0530 (India Standard Time)'. Under 'Security groups', the group 'sg-0628951589a93bd89 (launch-wizard-2)' is listed. The 'Inbound rules' section shows a single rule with ID 'sgr-092c1e1ca9787787b', port range '22', protocol 'TCP', source '0.0.0.0/0', and security group 'launch-wizard-2'.

- As you can see, there's only one inbound rule in the security group. Click on “**Edit Inbound Rules**”

The screenshot shows the AWS Management Console interface for the 'Security Groups' page. The breadcrumb trail is 'EC2 > Security Groups > sg-0628951589a93bd89 - launch-wizard-2'. The main heading is 'sg-0628951589a93bd89 - launch-wizard-2'. Below this, the 'Details' section shows the security group name 'launch-wizard-2', ID 'sg-0628951589a93bd89', description 'launch-wizard-2 created 2022-09-23T10:26:01.250Z', VPC ID 'vpc-0823844c6db5187b2', owner '553517453308', inbound rules count '1 Permission entry', and outbound rules count '1 Permission entry'. The 'Inbound rules' tab is selected, showing a single rule with ID 'sgr-092c1e1ca9787787b', IP version 'IPv4', type 'SSH', protocol 'TCP', port range '22', and source '0.0.0.0/0'. An 'Edit inbound rules' button is highlighted in yellow.

- Click on **Add Rule** button and configure the security group as shown below. Enter the following values as shown in the image below

**Type:** Custom TCP

**Port Range:** 8888

**Source:** Anywhere-IPv4

**Edit inbound rules** [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>		
sgr-092c1e1ca9787787b	SSH ▼	TCP	22	Custom ▼	Q	<input type="text"/>	Delete
-	Custom TCP ▼	TCP	8888	Anywhere-IPv4 ▼	Q	<input type="text"/>	Delete

Similarly, you need to add the following port numbers:

**2181, 9092, 9000, 8080**

Make sure that it looks like this:

**Inbound rules** [Info](#)

Security group rule ID	Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>		
sgr-0753441e9e15ebacb	SSH ▼	TCP	22	Custom ▼	Q	<input type="text"/>	Delete
-	Custom TCP ▼	TCP	8080	Anywhere-IPv4 ▼	Q	<input type="text"/>	Delete
-	Custom TCP ▼	TCP	8888	Anywhere-IPv4 ▼	Q	<input type="text"/>	Delete
-	Custom TCP ▼	TCP	9092	Anywhere-IPv4 ▼	Q	<input type="text"/>	Delete
-	Custom TCP ▼	TCP	9000	Anywhere-IPv4 ▼	Q	<input type="text"/>	Delete
-	Custom TCP ▼	TCP	2181	Anywhere-IPv4 ▼	Q	<input type="text"/>	Delete

**Note: This is a very important step. Double-check that all the port range and source are identical as shown in the image.**

Next click on the **Save rules**.



## ELASTIC IP CREATION INSTRUCTIONS

Whenever you start an EC2 instance, it generates a new IPv4 public IP. To keep the public IP constant, an elastic IP is required. A constant public IP is crucial in Kafka as you are required to create Kafka producers and consumers by writing code. You will understand the significance of having a constant IP when you will actually write the codes.

To create an elastic IP, do the following:

- Go to your EC2 instance page and on the left-hand side scroll till you find the **Network and Security** tab, on the left-hand side, under which you will see the **Elastic IPs** link, as shown in the image below.

The screenshot displays the AWS Management Console interface. On the left, the navigation menu is visible with 'Network & Security' expanded and 'Elastic IPs' selected. The main panel shows the 'Instances' page for a single instance named 'kafka\_instance'. The instance is in a 'Running' state. The 'Details' tab is selected, showing various attributes of the instance, including its public IPv4 address, instance state, and VPC ID.

- Click on the **Elastic IPs** link. You will be redirected to a new page. Click on the **Allocate Elastic IP address** button

The screenshot shows the 'Elastic IP addresses' page. At the top, there is a search bar and an 'Allocate Elastic IP address' button. Below this is a table with the following columns: Name, Allocated IPv4 address, Type, Allocation ID, and Reverse DNS record. The table is currently empty, and a message at the bottom indicates 'No Elastic IP addresses found in t'.

Once you do so, you will be redirected to a new page as shown below.  
Choose the settings and click on the “Allocate” button.

**Allocate Elastic IP address** [Info](#)

**Elastic IP address settings** [Info](#)

Network Border Group [Info](#)

us-east-1

Public IPv4 address pool

- ☒ Amazon's pool of IPv4 addresses
- ☐ Public IPv4 address that you bring to your AWS account (option disabled because no pools found) [Learn more](#)
- ☐ Customer owned pool of IPv4 addresses (option disabled because no customer owned pools found) [Learn more](#)

Global static IP addresses

AWS Global Accelerator can provide global static IP addresses that are announced worldwide using anycast from AWS edge locations. This can help improve the availability and latency for your user traffic by using the Amazon global network. [Learn more](#)

[Create accelerator](#)

**Tags - optional**

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

[Add new tag](#)

You can add up to 50 more tag

[Cancel](#) [Allocate](#)

- Click on **Allocate**. You will be redirected back to the elastic IP screen. Please note that only one Elastic IP address can't be allotted to multiple EC2 instances.

**Elastic IP address allocated successfully.**  
Elastic IP address 52.21.15.133

[Associate this Elastic IP address](#)

**Elastic IP addresses (1/1)**

[Filter Elastic IP addresses](#)

Public IPv4 address: 52.21.15.133 [Clear filters](#)

<input checked="" type="checkbox"/>	Name	Allocated IPv4 address	Type	Allocation ID	Reverse DNS record	Associated instance ID
<input checked="" type="checkbox"/>	-	52.21.15.133	Public IP	eipalloc-046779924386e333c	-	-

52.21.15.133

[Summary](#) [Tags](#)

- Click on the checkbox against your newly created elastic IP. Then, click on the Actions button, you will get a drop-down list. From this list, click on **Associate Address**.

Elastic IP address allocated successfully.  
Elastic IP address 52.21.15.133

Associate this Elastic IP address

Elastic IP addresses (1/1)

Filter Elastic IP addresses

Public IPv4 address: 52.21.15.133 X Clear filters

<input checked="" type="checkbox"/>	Name	Allocated IPv4 address	Type	Allocation ID	Reverse DNS	Instance ID
<input checked="" type="checkbox"/>	-	52.21.15.133	Public IP	eipalloc-046779924386e333c	-	

Actions

- View details
- Release Elastic IP addresses
- Associate Elastic IP address
- Disassociate Elastic IP address
- Update reverse DNS

- A new page will appear, as shown below. Click inside the Instance text box; you will get a list of available instances, as shown below. From the list, choose the EC2 instance with which you want to attach the elastic IP. If you have followed the steps in this document, then its name would be **kafka\_instance**. The Instance ID of that EC2 instance will appear in the text box, as shown in the example below.

## Associate Elastic IP address

Choose the instance or network interface to associate to this Elastic IP address (52.21.15.133)

**Elastic IP address: 52.21.15.133**

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

☒ Instance  
☐ Network interface

**⚠** If you associate an Elastic IP address to an instance that already has an Elastic IP address associated, this previously associated Elastic IP address will be disassociated but still allocated to your account. [Learn more](#)

Instance

Choose an instance

i-0b904d6e0c40a2061 (kafka\_instance) - 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 reassigned with a different resource if it already associated with a resource.

☐ Allow this Elastic IP address to be reassigned

Cancel Associate

- Next, click on the text-box next to **Private IP** and select the automatically generated **Private IP address** and then click on Associate.

## Associate Elastic IP address

Choose the instance or network interface to associate to this Elastic IP address (52.21.15.133)

**Elastic IP address: 52.21.15.133**

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

☒ Instance  
☐ Network interface

If you associate an Elastic IP address to an instance that already has an Elastic IP address associated, this previously associated Elastic IP address will be disassociated but still allocated to your account. [Learn more](#)

**Instance**

✕ ↻

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

**172.31.23.35**

☐ Allow this Elastic IP address to be reassociated

Cancel Associate

- You will be navigated to a new page where you will have confirmation that the elastic IP has been attached to your desired EC2 instance, as shown below and click on **Close**.

Elastic IP address associated successfully.  
Elastic IP address 52.21.15.133 has been associated with instance i-0b904d6e0c40a2061

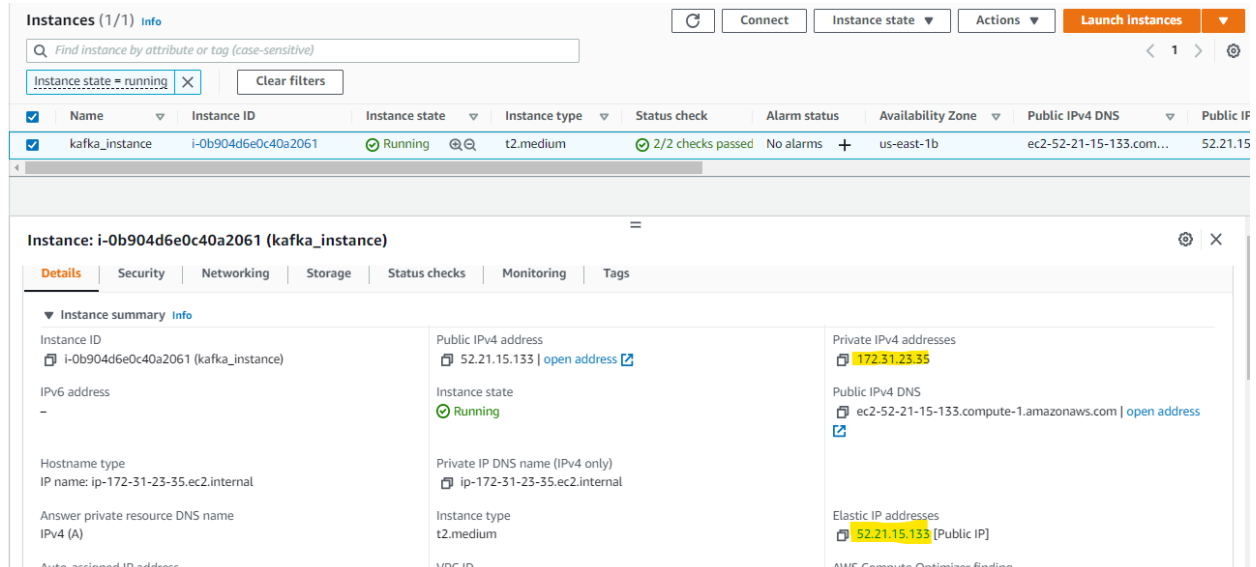
**Elastic IP addresses (1/1)** ↻ Actions ▾ Allocate Elastic IP address

< 1 >

Public IPv4 address: 52.21.15.133 ✕ Clear filters

<input checked="" type="checkbox"/>	Name	Allocated IPv4 add...	Type	Allocation ID	Reverse DNS record
<input checked="" type="checkbox"/>	-	52.21.15.133	Public IP	eipalloc-046779924386e333c	-

- Now, go to the Instances page using the Scroll bar on the left-hand side of the screen and click on the tick box against the EC2 instance with which you associated the elastic IP. You will see that the elastic IP has been attached to the EC2 instance, as shown below.



**Instances (1/1) info**

Find instance by attribute or tag (case-sensitive)

Instance state = running X Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
kafka_instance	i-0b904d6e0c40a2061	Running	t2.medium	2/2 checks passed	No alarms	us-east-1b	ec2-52-21-15-133.com...	52.21.15...

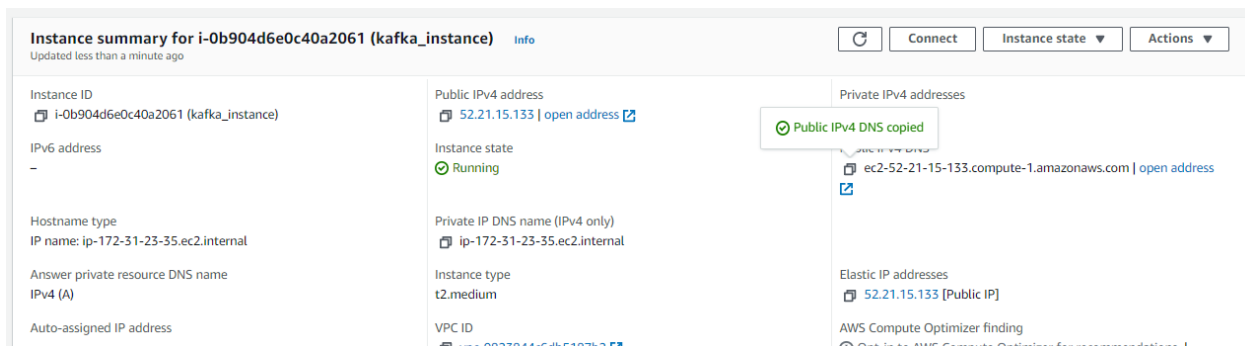
**Instance: i-0b904d6e0c40a2061 (kafka\_instance)**

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

**Instance summary info**

Instance ID i-0b904d6e0c40a2061 (kafka_instance)	Public IPv4 address 52.21.15.133   open address	Private IPv4 addresses 172.31.23.35
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-52-21-15-133.compute-1.amazonaws.com   open address
Hostname type IP name: ip-172-31-23-35.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-23-35.ec2.internal	Elastic IP addresses 52.21.15.133 [Public IP]
Answer private resource DNS name IPv4 (A)	Instance type t2.medium	AWS Compute Optimizer finding
Auto-assigned IP address -	VPC ID vpc-0823844c6db5187b2	

- You can see that the elastic IP has been attached to the EC2 instance and your Private IPv4 address has been updated as well. Now that you have setup your EC2 instance, you will now need to SSH into the instance using the Public IPv4 DNS of the EC2 instance.



**Instance summary for i-0b904d6e0c40a2061 (kafka\_instance) info**


Updated less than a minute ago

Connect Instance state Actions

Instance ID i-0b904d6e0c40a2061 (kafka_instance)	Public IPv4 address 52.21.15.133   open address	Private IPv4 addresses 172.31.23.35
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-52-21-15-133.compute-1.amazonaws.com   open address
Hostname type IP name: ip-172-31-23-35.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-23-35.ec2.internal	Elastic IP addresses 52.21.15.133 [Public IP]
Answer private resource DNS name IPv4 (A)	Instance type t2.medium	AWS Compute Optimizer finding Out-in to AWS Compute Optimizer for recommendations. I
Auto-assigned IP address -	VPC ID vpc-0823844c6db5187b2	

Public IPv4 DNS copied

- Next, you need to login to the EC2 instance with your key pair. After logging in, go to the **/home/ec2-user/downloads** directory where you will see the Kafka directory is already present, as shown in the image below.

 ec2-user@ip-172-31-71-97:~/downloads

```
[ec2-user@ip-172-31-71-97 ~]$ cd downloads/  
[ec2-user@ip-172-31-71-97 downloads]$ ls  
kafka_2.12-2.3.0  
[ec2-user@ip-172-31-71-97 downloads]$
```

# KAFKA SERVER CONFIGURATION

The following steps list the process to configure the Kafka Server so that you can connect your Kafka server with an IDE such as Eclipse.

To configure the Kafka server, you need to make changes to the `server.properties` file for which you need to go through the following steps:

- Login to the EC2 instance and go inside the Kafka directory present there. The command for the same is `cd /home/ec2-user/downloads/kafka_2.12-2.3.0`. Once you are inside this directory, you need to go inside the config directory. The command for the same is `cd config/`. Once you are inside the directory, enter the command `ls`. This will list you the different files present inside the directory.

```
ec2-user@ip-172-31-71-97:~/downloads/kafka_2.12-2.3.0/config
[ec2-user@ip-172-31-71-97 kafka_2.12-2.3.0]$ cd /home/ec2-user/downloads/kafka_2.12-2.3.0
[ec2-user@ip-172-31-71-97 kafka_2.12-2.3.0]$ cd config/
[ec2-user@ip-172-31-71-97 config]$ ls
connect-console-sink.properties  connect-file-sink.properties  connect-standalone.properties  producer.properties  trogdor.conf
connect-console-source.properties  connect-file-source.properties  consumer.properties  server.properties  zookeeper.properties
connect-distributed.properties  connect-log4j.properties  log4j.properties  tools-log4j.properties
[ec2-user@ip-172-31-71-97 config]$
```

- Here you need to make changes to the `server.properties` file. To edit this file enter the command `vi server.properties`. You would get a screen, as shown below.

```

# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
# see kafka.server.KafkaConfig for additional details and defaults
##### Server Basics #####
# The id of the broker. This must be set to a unique integer for each broker.
broker.id=0

##### Socket Server Settings #####
# The address the socket server listens on. It will get the value returned from
# java.net.InetAddress.getCanonicalHostName() if not configured.
#   FORMAT:
#   listeners = listener_name://host_name:port
#   EXAMPLE:
#   listeners = PLAINTEXT://your.host.name:9092
#listeners=PLAINTEXT://:9092
#
# Hostname and port the broker will advertise to producers and consumers. If not set,
# it uses the value for "listeners" if configured. Otherwise, it will use the value
# returned from java.net.InetAddress.getCanonicalHostName().
#advertised.listeners=PLAINTEXT://your.host.name:9092
#
# Maps listener names to security protocols, the default is for them to be the same. See the config documentation for more details
#listener.security.protocol.map=PLAINTEXT:PLAINTEXT,SSL:SSL,SASL_PLAINTEXT:SASL_PLAINTEXT,SASL_SSL:SASL_SSL
#
# The number of threads that the server uses for receiving requests from the network and sending responses to the network
num.network.threads=3
#
# The number of threads that the server uses for processing requests, which may include disk I/O
num.io.threads=8
#
# The send buffer (SO_SNDBUF) used by the socket server
socket.send.buffer.bytes=102400
"server.properties" [noeol] 136L, 6851C

```

- Here you need to make changes in the line which reads as follows:

```
#advertised.listeners=PLAINTEXT://your.host.name:9092
```

- You need to uncomment this line and in place of your.host.name you need to enter the IPv4 Public IP of your EC2 Instance. This is the same as the Elastic IP associated with your EC2 instance. So press i, and you will enter insert mode.

Instance: i-0b904d6e0c40a2061 (kafka\_instance)

Details	Security	Networking	Storage	Status checks	Monitoring	Tags
<div>▼ Instance summary info</div> <div> <div> <div>Instance ID</div> <div>i-0b904d6e0c40a2061 (kafka_instance)</div> </div> <div> <div>Public IPv4 address</div> <div>52.21.15.133   open address</div> </div> <div> <div>Private IPv4 addresses</div> <div>172.31.23.35</div> </div> <div> <div>IPv6 address</div> <div>-</div> </div> <div> <div>Instance state</div> <div>Running</div> </div> <div> <div>Public IPv4 DNS</div> <div>ec2-52-21-15-133.compute-1.amazonaws.com   open address</div> </div> <div> <div>Hostname type</div> <div>IP name: ip-172-31-23-35.ec2.internal</div> </div> <div> <div>Private IP DNS name (IPv4 only)</div> <div>ip-172-31-23-35.ec2.internal</div> </div> <div> <div>Answer private resource DNS name</div> <div>IPv4 (A)</div> </div> <div> <div>Elastic IP addresses</div> <div>52.21.15.133 [Public IP]</div> </div> <div> <div>Instance type</div> <div>t2.medium</div> </div> </div>						

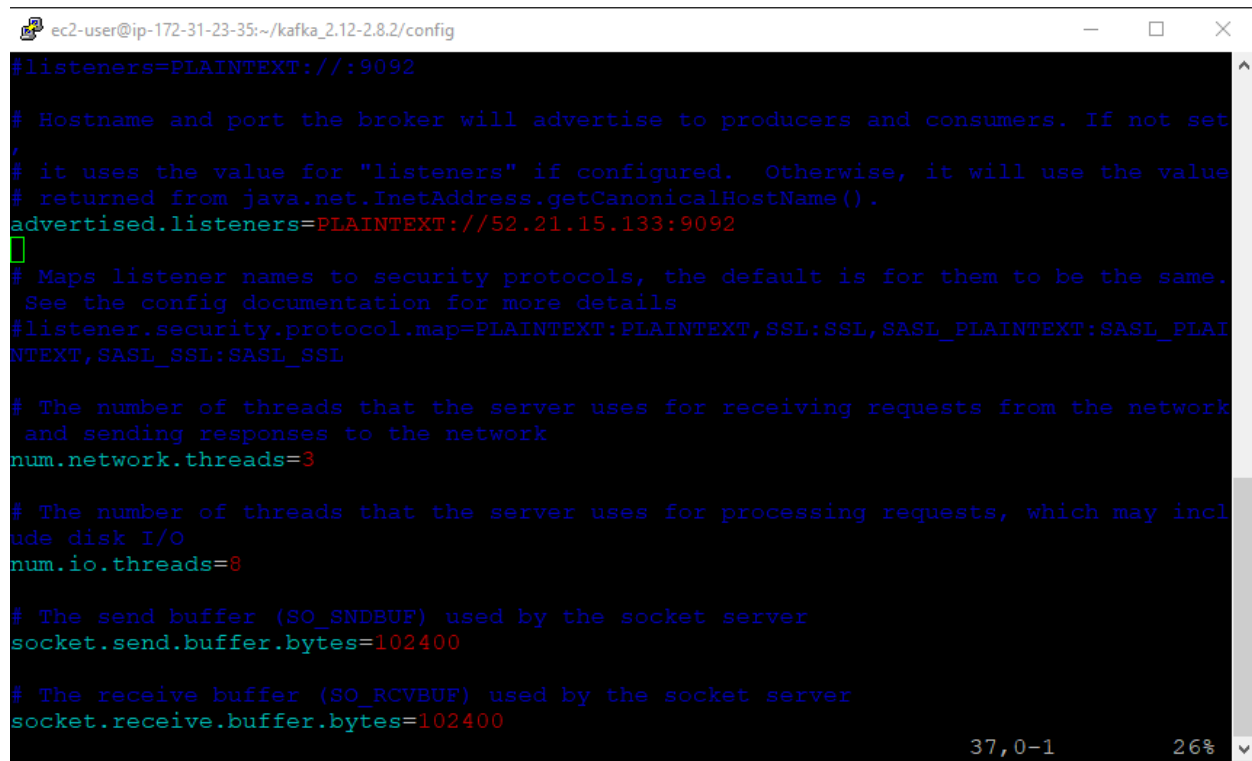
You can see the IP in my case is **52.21.15.133**.



- Then go to the above line and uncomment it by removing the `#`. Next in place of your.host.name enter the IPv4 Public IP of your EC2 instance. In this case, that line would read as follows:

```
advertised.listeners=PLAINTEXT://52.21.15.133:9092
```

Your screen should look something as shown below:



```
ec2-user@ip-172-31-23-35:~/kafka_2.12-2.8.2/config
#listeners=PLAINTEXT://:9092
# Hostname and port the broker will advertise to producers and consumers. If not set
# it uses the value for "listeners" if configured.  Otherwise, it will use the value
# returned from java.net.InetAddress.getCanonicalHostName().
advertised.listeners=PLAINTEXT://52.21.15.133:9092
# Maps listener names to security protocols, the default is for them to be the same.
# See the config documentation for more details
#listener.security.protocol.map=PLAINTEXT:PLAINTEXT,SSL:SSL,SASL_PLAINTEXT:SASL_PLAI
NTEXT,SASL_SSL:SASL_SSL
# The number of threads that the server uses for receiving requests from the network
# and sending responses to the network
num.network.threads=3
# The number of threads that the server uses for processing requests, which may incl
ude disk I/O
num.io.threads=8
# The send buffer (SO_SNDBUF) used by the socket server
socket.send.buffer.bytes=102400
# The receive buffer (SO_RCVBUF) used by the socket server
socket.receive.buffer.bytes=102400
```

**Make sure that you enter the IP address of your EC2 instance.**

Once you have done these steps, press **esc** and then type **:wq** to come out of the file.

With these steps, you have configured the Kafka server to connect it with an IDE such as Eclipse.

## STEPS TO VERIFY THE INSTALLATION

### Zookeeper

To verify the zookeeper installation, follow the steps listed below.

- You need to get inside the Kafka directory. 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. You should get the following output.

```
0.5.0.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/commons-lang3-3.8.1.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/connect-api-2.3.0.jar:/home/
kafka_2.12-2.3.0/bin/./libs/connect-basic-auth-extension-2.3.0.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/connect-file-2.3.0.jar:/home/ec2-user/downloads/
n/./libs/connect-json-2.3.0.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/connect-runtime-2.3.0.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/cor
0.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/guava-20.0.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/hk2-api-2.5.0.jar:/home/ec2-user/downlo
/bin/./libs/hk2-locator-2.5.0.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/hk2-utils-2.5.0.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/jackso
jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/jackson-core-2.9.9.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/jackson-databind-2.9.9.jar:/home/ec
kafka_2.12-2.3.0/bin/./libs/jackson-dataformat-csv-2.9.9.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/jackson-datatype-jdk8-2.9.9.jar:/home/ec2-user/downloa
in/./libs/jackson-jaxrs-base-2.9.9.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/jackson-jaxrs-json-provider-2.9.9.jar:/home/ec2-user/downloads/kafka_2.12-2.
kson-module-jaxb-annotations-2.9.9.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/jackson-module-paranamer-2.9.9.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/
-module-scala_2.12-2.9.9.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/jakarta.annotation-api-1.3.4.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/
0.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/jakarta.ws.rs-api-2.1.5.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/javassist-3.22.0-CR2.jar:/h
ds/kafka_2.12-2.3.0/bin/./libs/javax.servlet-api-3.1.0.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/javax.ws.rs-api-2.1.1.jar:/home/ec2-user/downloads/kafka
libs/jaxb-api-2.3.0.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/jersey-client-2.28.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/jersey-common-2
ser/downloads/kafka_2.12-2.3.0/bin/./libs/jersey-container-servlet-2.28.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/jersey-container-servlet-core-2.28.jar:/
loads/kafka_2.12-2.3.0/bin/./libs/jersey-hk2-2.28.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/jersey-media-jaxb-2.28.jar:/home/ec2-user/downloads/kafka_2.1
/jersey-server-2.28.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/jetty-client-9.4.18.v20190429.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/jett
18.v20190429.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/jetty-http-9.4.18.v20190429.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/jetty-io-9.4.
ome/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/jetty-security-9.4.18.v20190429.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/jetty-server-9.4.18.v2019042
r/downloads/kafka_2.12-2.3.0/bin/./libs/jetty-servlet-9.4.18.v20190429.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/jetty-servlets-9.4.18.v20190429.jar:/hom
s/kafka_2.12-2.3.0/bin/./libs/jetty-util-9.4.18.v20190429.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/jopt-simple-5.0.4.jar:/home/ec2-user/downloads/kafka/
ibs/jsr305-3.0.2.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/kafka_2.12-2.3.0.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/kafka_2.12-2.3.0-sou
user/downloads/kafka_2.12-2.3.0/bin/./libs/kafka-clients-2.3.0.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/kafka-log4j-appender-2.3.0.jar:/home/ec2-user/d
2.3.0/bin/./libs/kafka-streams-2.3.0.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/kafka-streams-examples-2.3.0.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0
streams-scala_2.12-2.3.0.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/kafka-streams-test-utils-2.3.0.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./lib
.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/log4j-1.2.17.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/lz4-java-1.6.0.jar:/home/ec2-user/downlo
0/bin/./libs/maven-artifact-3.6.1.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/metrics-core-2.2.0.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/
oc-1.0.1.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/paranamer-2.8.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/plexus-utils-3.2.0.jar:/home/ec
kafka_2.12-2.3.0/bin/./libs/reflections-0.9.11.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/rocksdbjni-5.19.3.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bi
ary-2.12.8.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/scala-logging_2.12-3.9.0.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/scala-reflect-2.1
ser/downloads/kafka_2.12-2.3.0/bin/./libs/slf4j-api-1.7.26.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/slf4j-log4j12-1.7.26.jar:/home/ec2-user/downloads/ka
./libs/snappy-java-1.1.7.3.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/spotbugs-annotations-3.1.9.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs
1.Final.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/zkclient-0.11.jar:/home/ec2-user/downloads/kafka_2.12-2.3.0/bin/./libs/zookeeper-3.4.14.jar:/home/ec2-
a_2.12-2.3.0/bin/./libs/zstd-jni-1.4.0-1.jar (org.apache.zookeeper.server.ZooKeeperServer)
[2020-09-20 07:04:21,917] INFO Server environment:java.library.path=/usr/java/packages/lib/amd64:/usr/lib64:/lib64:/lib:/usr/lib (org.apache.zookeeper.server.ZooKeeperSer
[2020-09-20 07:04:21,917] INFO Server environment:java.io.tmpdir=/tmp (org.apache.zookeeper.server.ZooKeeperServer)
[2020-09-20 07:04:21,917] INFO Server environment:java.compiler=<NA> (org.apache.zookeeper.server.ZooKeeperServer)
[2020-09-20 07:04:21,917] INFO Server environment:os.name=Linux (org.apache.zookeeper.server.ZooKeeperServer)
[2020-09-20 07:04:21,917] INFO Server environment:os.arch=amd64 (org.apache.zookeeper.server.ZooKeeperServer)
[2020-09-20 07:04:21,917] INFO Server environment:os.version=4.14.193-149.317.amzn2.x86_64 (org.apache.zookeeper.server.ZooKeeperServer)
[2020-09-20 07:04:21,917] INFO Server environment:user.name=ec2-user (org.apache.zookeeper.server.ZooKeeperServer)
[2020-09-20 07:04:21,917] INFO Server environment:user.home=/home/ec2-user (org.apache.zookeeper.server.ZooKeeperServer)
[2020-09-20 07:04:21,917] INFO Server environment:user.dir=/home/ec2-user/downloads/kafka_2.12-2.3.0 (org.apache.zookeeper.server.ZooKeeperServer)
[2020-09-20 07:04:21,917] INFO tickTime set to 3000 (org.apache.zookeeper.server.ZooKeeperServer)
[2020-09-20 07:04:21,953] INFO minSessionTimeout set to -1 (org.apache.zookeeper.server.ZooKeeperServer)
[2020-09-20 07:04:21,953] INFO maxSessionTimeout set to -1 (org.apache.zookeeper.server.ZooKeeperServer)
[2020-09-20 07:04:22,086] INFO Using org.apache.zookeeper.server.NIOServerCnxnFactory as server connection factory (org.apache.zookeeper.server.ServerCnxnFactory)
[2020-09-20 07:04:22,118] INFO binding to port 0.0.0.0/0.0.0.0:2181 (org.apache.zookeeper.server.NIOServerCnxnFactory)
```

At the bottom of the screen, you would get something like:  
**INFO binding to port 0.0.0.0/0.0.0.0:2181**  
**(org.apache.zookeeper.server.NIOServerCnxnFactory)**

This means that your zookeeper server has successfully started.

## Kafka

Before going through this step, please ensure that the Zookeeper server is running.

To verify the Kafka installation, follow the steps listed below:

- Leave the previous terminal window as it is and login to your EC2 instance using another terminal.
- Go to the Kafka directory using the `cd downloads/kafka_2.12-2.3.0` command.
- Start the Kafka server using the `bin/kafka-server-start.sh config/server.properties` command.
- You should get an output which displays a message something like “**INFO [KafkaServer id=0] started (kafka.server.KafkaServer)**”

```
transaction.state.log.segment.bytes = 104857600
transactional.id.expiration.ms = 604800000
unclean.leader.election.enable = false
zookeeper.connect = localhost:2181
zookeeper.connection.timeout.ms = 6000
zookeeper.max.in.flight.requests = 10
zookeeper.session.timeout.ms = 6000
zookeeper.set.acl = false
zookeeper.sync.time.ms = 2000
(kafka.server.KafkaConfig)
(2020-09-05 16:38:40,040) INFO [ThrottledChannelReaper-Fetch]: Starting (kafka.server.ClientQuotaManager$ThrottledChannelReaper)
(2020-09-05 16:38:40,041) INFO [ThrottledChannelReaper-Producer]: Starting (kafka.server.ClientQuotaManager$ThrottledChannelReaper)
(2020-09-05 16:38:40,044) INFO [ThrottledChannelReaper-Request]: Starting (kafka.server.ClientQuotaManager$ThrottledChannelReaper)
(2020-09-05 16:38:40,108) INFO Loading logs. (kafka.log.LogManager)
(2020-09-05 16:38:40,120) INFO Logs loading complete in 11 ms. (kafka.log.LogManager)
(2020-09-05 16:38:40,135) INFO Starting log cleanup with a period of 300000 ms. (kafka.log.LogManager)
(2020-09-05 16:38:40,144) INFO Starting log flusher with a default period of 9223372036854775807 ms. (kafka.log.LogManager)
(2020-09-05 16:38:40,683) INFO Awaiting socket connections on 0.0.0.0:9092. (kafka.network.Acceptor)
(2020-09-05 16:38:40,735) INFO [SocketServer brokerId=0] Started 1 acceptor threads (kafka.network.SocketServer)
(2020-09-05 16:38:40,771) INFO [ExpirationReaper-0-Producer]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
(2020-09-05 16:38:40,775) INFO [ExpirationReaper-0-Fetch]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
(2020-09-05 16:38:40,782) INFO [ExpirationReaper-0-DeleteRecords]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
(2020-09-05 16:38:40,800) INFO [LogDirFailureHandler]: Starting (kafka.server.ReplicaManager$LogDirFailureHandler)
(2020-09-05 16:38:40,857) INFO Creating /brokers/ids/0 (is it secure? false) (kafka.zk.KafkaZkClient)
(2020-09-05 16:38:40,862) INFO Result of znode creation at /brokers/ids/0 is: OK (kafka.zk.KafkaZkClient)
(2020-09-05 16:38:40,863) INFO Registered broker 0 at path /brokers/ids/0 with addresses: ArrayBuffer(EndPoint(54.81.125.202,9092,ListenerName(PLAINTEXT),PLAINTEXT)) (kafka.zk.KafkaZkClient)
(2020-09-05 16:38:40,976) INFO [ExpirationReaper-0-topic]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
(2020-09-05 16:38:40,971) INFO Creating /controller (is it secure? false) (kafka.zk.KafkaZkClient)
(2020-09-05 16:38:40,977) INFO [ExpirationReaper-0-Heartbeat]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
(2020-09-05 16:38:40,992) INFO [ExpirationReaper-0-Rebalance]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
(2020-09-05 16:38:40,995) INFO Result of znode creation at /controller is: OK (kafka.zk.KafkaZkClient)
(2020-09-05 16:38:41,018) INFO [GroupCoordinator 0]: Starting up. (kafka.coordinator.group.GroupCoordinator)
(2020-09-05 16:38:41,020) INFO [GroupCoordinator 0]: Startup complete. (kafka.coordinator.group.GroupCoordinator)
(2020-09-05 16:38:41,044) INFO [GroupMetadataManager brokerId=0] Removed 0 expired offsets in 23 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
(2020-09-05 16:38:41,057) INFO [ProducerId Manager 0]: Acquired new producerId block (brokerId:0,blockStartProducerId:6000,blockEndProducerId:6999) by writing to zk with path version 7 (kafka.coordinator.transaction.ProducerIdManager)
(2020-09-05 16:38:41,139) INFO [TransactionCoordinator id=0] Starting up. (kafka.coordinator.transaction.TransactionCoordinator)
(2020-09-05 16:38:41,144) INFO [TransactionCoordinator id=0] Startup complete. (kafka.coordinator.transaction.TransactionCoordinator)
(2020-09-05 16:38:41,155) INFO [Transaction Marker Channel Manager 0]: Starting (kafka.coordinator.transaction.TransactionMarkerChannelManager)
(2020-09-05 16:38:41,290) INFO [/config/changes-event-process-thread]: Starting (kafka.common.ZkNodeChangeNotificationListener$ChangeEventProcessThread)
(2020-09-05 16:38:41,309) INFO [SocketServer brokerId=0] Started processors for 1 acceptors (kafka.network.SocketServer)
(2020-09-05 16:38:41,317) INFO Kafka version : 2.9.0 (org.apache.kafka.common.utils.AppInfoParser)
(2020-09-05 16:38:41,317) INFO Kafka commitId : 3402a8361b734732 (org.apache.kafka.common.utils.AppInfoParser)
(2020-09-05 16:38:41,318) INFO [KafkaServer id=0] started (kafka.server.KafkaServer)
```

This means the instance creation is successful, and you can continue with the next steps.

## STARTING JUPYTER NOTEBOOKS

- Open a new Terminal window and log in to the EC2 instance.
  - Make sure that the current directory is **/home/ec2-user** by running the **pwd** command.
- Once you are inside the **/home/ec2-user** directory, follow the steps listed below.

```
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Sun Sep 20 07:08:50 2020 from 157.35.236.42

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

https://aws.amazon.com/amazon-linux-2/
No packages needed for security: 1 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-71-97 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-71-97 ~]$
```

- Enter the command **cd .jupyter/** and then enter the **ls** command.

```
https://aws.amazon.com/amazon-linux-2/
No packages needed for security; 1 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-71-97 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-71-97 ~]$ cd .jupyter/
[ec2-user@ip-172-31-71-97 .jupyter]$ ls
jupyter_notebook_config.py
[ec2-user@ip-172-31-71-97 .jupyter]$
```

- Enter the command **vi jupyter\_notebook\_config.py**. At the bottom you would see the following two lines:

**c.NotebookApp.allow\_origin = ''**

**c.NotebookApp.ip = '0.0.0.0'**

```

# NotebookNotary configuration
#-----

# A class for computing and verifying notebook signatures.

# The hashing algorithm used to sign notebooks.
# c.NotebookNotary.algorithm = 'sha256'

# The number of notebook signatures to cache. When the number of signatures
# exceeds this value, the oldest 25% of signatures will be culled.
# c.NotebookNotary.cache_size = 65535

# The sqlite file in which to store notebook signatures. By default, this will
# be in your Jupyter runtime directory. You can set it to ':memory:' to disable
# sqlite writing to the filesystem.
# c.NotebookNotary.db_file = u''

# The secret key with which notebooks are signed.
# c.NotebookNotary.secret = ''

# The file where the secret key is stored.
# c.NotebookNotary.secret_file = u''

#-----
# KernelSpecManager configuration
#-----

# If there is no Python kernelspec registered and the IPython kernel is
# available, ensure it is added to the spec list.
# c.KernelSpecManager.ensure_native_kernel = True

# The kernel spec class. This is configurable to allow subclassing of the
# KernelSpecManager for customized behavior.
# c.KernelSpecManager.kernel_spec_class = 'jupyter_client.kernelspec.KernelSpec'

# Whitelist of allowed kernel names.
#
# By default, all installed kernels are allowed.
# c.KernelSpecManager.whitelist = set([])
c.NotebookApp.allow_origin = '*'
c.NotebookApp.ip = '0.0.0.0'
"jupyter_notebook_config.py" 582L, 21326C

```

If you see these two lines, which you most likely would, you do not need to do anything. Just press **:wq** and come out the config file.

**Optional:** If these two lines are not present, then you need to copy the above two lines and then write it into the **jupyter\_notebook\_config.py** file. For that press **i** and then enter these two lines and then press **esc** and then type **:wq** to come save it and then come out of the file.

Next you need to enter the `cd ..` to come to the `/home/ec2-user` directory.

```
https://aws.amazon.com/amazon-linux-2/
No packages needed for security; 1 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-71-97 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-71-97 ~]$ cd .jupyter/
[ec2-user@ip-172-31-71-97 .jupyter]$ ls
jupyter_notebook_config.py
[ec2-user@ip-172-31-71-97 .jupyter]$ vi jupyter_notebook_config.py
[ec2-user@ip-172-31-71-97 .jupyter]$ vi jupyter_notebook_config.py
[ec2-user@ip-172-31-71-97 .jupyter]$ cd ..
[ec2-user@ip-172-31-71-97 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-71-97 ~]$ █
```

Once you are in the `/home/ec2-user` directory, you need to enter the command `nohup jupyter notebook &`.

```
[ec2-user@ip-172-31-71-97 .jupyter]$ vi jupyter_notebook_config.py
[ec2-user@ip-172-31-71-97 .jupyter]$ vi jupyter_notebook_config.py
[ec2-user@ip-172-31-71-97 .jupyter]$ cd ..
[ec2-user@ip-172-31-71-97 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-71-97 ~]$ nohup jupyter notebook &
[1] 4482
[ec2-user@ip-172-31-71-97 ~]$ nohup: ignoring input and appending output to 'nohup.out'
█
```

Then you need to press `Ctrl+C` and then need to run the command `cat nohup.out`.

```
[ec2-user@ip-172-31-71-97 ~]$ cat nohup.out
[W 07:21:47.832 NotebookApp] Unrecognized JSON config file version, assuming version 1
[I 07:21:49.946 NotebookApp] [nb_conda_kernels] enabled, 1 kernels found
[I 07:21:49.968 NotebookApp] Writing notebook server cookie secret to /run/user/1000/jupyter/notebook_cookie_secret
[I 07:21:50.535 NotebookApp] ✓ nbpresent HTML export ENABLED
[W 07:21:50.535 NotebookApp] X nbpresent PDF export DISABLED: No module named nbbrowserpdf.exporters.pdf
[I 07:21:50.549 NotebookApp] [nb_conda] enabled
[I 07:21:51.018 NotebookApp] [nb_anacondacloud] enabled
[I 07:21:51.019 NotebookApp] Serving notebooks from local directory: /home/ec2-user
[I 07:21:51.019 NotebookApp] 0 active kernels
[I 07:21:51.019 NotebookApp] The Jupyter Notebook is running at: http://0.0.0.0:8888/
[I 07:21:51.019 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[W 07:21:51.020 NotebookApp] No web browser found: could not locate runnable browser.
[ec2-user@ip-172-31-71-97 ~]$
```

Here you will get a line which says “The Jupyter Notebook is running at: <http://0.0.0.0:8888/>”.

Copy this url and replace 0.0.0.0 with your elastic ip associated with this EC2 instance. You also need to make sure that the port number is always **8888** and not any other port number. Once these are done, open the URL in your browser. So the url should be something like this:

**52.21.15.133:8888** and the screen which you will get will look something as shown below.



This means that the Jupyter Notebook has started successfully.

**Note:** Always make sure that you always open the port number is **8888**. Sometimes you might need to run the **nohup jupyter notebook &** command and the **cat nohup.out** command again. This might happen sometimes.

You can upload the Jupyter notebooks using the **Upload** button from your local machine to EC2 instance. Once you select the notebook, you need to upload you get a screen as shown below.



Select items to perform actions on them.

Upload New ↻

<input type="checkbox"/>		
<input type="checkbox"/>	 Kafka Python Producer.ipynb	Upload Cancel
<input type="checkbox"/>	anaconda2	
<input type="checkbox"/>	downloads	
<input type="checkbox"/>	Notebook	
<input type="checkbox"/>	Anaconda2-4.1.1-Linux-x86_64.sh	
<input type="checkbox"/>	jdk-8u161-linux-x64.tar.gz	
<input type="checkbox"/>	nohup.out	

Click on the **Upload** button next to it and the notebook will get uploaded.

Select items to perform actions on them.

Upload New ↻

<input type="checkbox"/>		
<input type="checkbox"/>	anaconda2	
<input type="checkbox"/>	downloads	
<input type="checkbox"/>	Notebook	
<input type="checkbox"/>	 Kafka Python Producer.ipynb	
<input type="checkbox"/>	Anaconda2-4.1.1-Linux-x86_64.sh	
<input type="checkbox"/>	jdk-8u161-linux-x64.tar.gz	
<input type="checkbox"/>	nohup.out	

If you login to EC2 instance and enter the ls command you will see the notebook is present in the /home/ec2-user directory.

```

login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Sun Sep 20 07:11:02 2020 from 157.35.236.42

 _ _ _ _ _
| | | | |
|_|_|_|_|_|

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
No packages needed for security: 1 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-71-97 ~]$ ls
anaconda2  Anaconda2-4.1.1-Linux-x86_64.sh  downloads  jdk-8u161-linux-x64.tar.gz  Kafka Python Producer.ipynb  nohup.out  Notebook
[ec2-user@ip-172-31-71-97 ~]$

```