EC2 INSTANCE SETUP FOR KAFKA

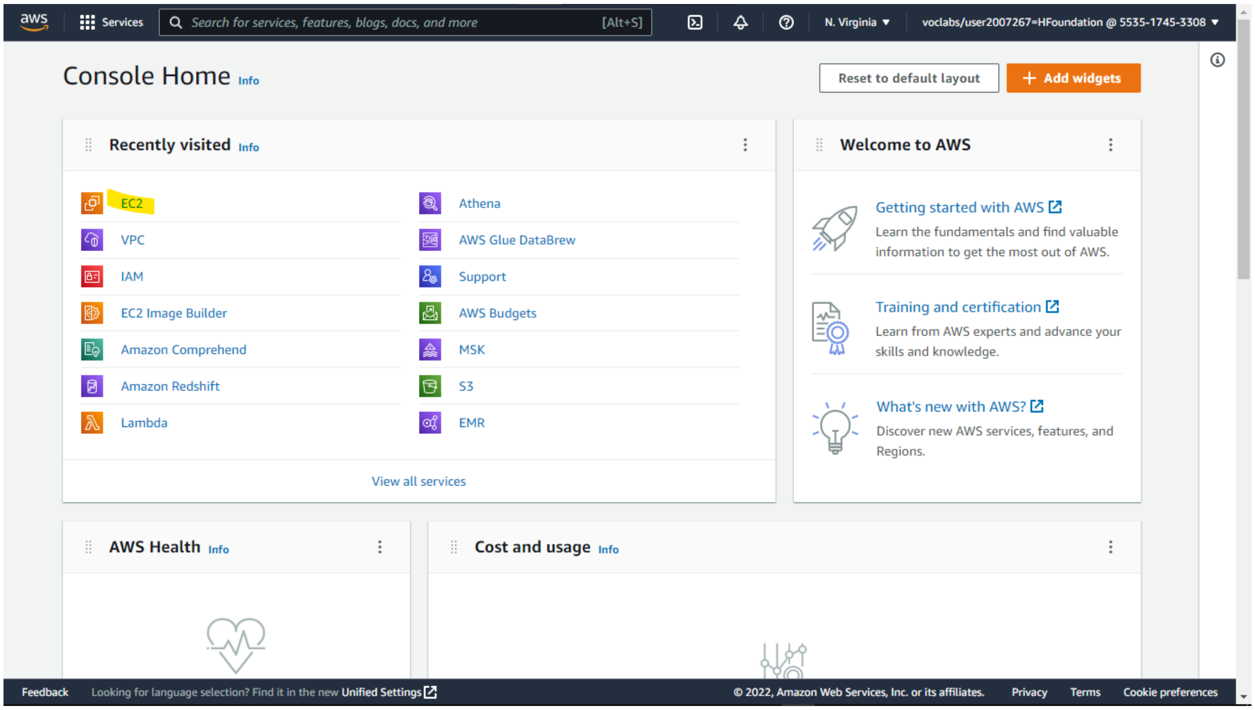
This document guides you through the setup of an EC2 instance which has been set up exclusively for Kafka. The following services are required for working with the Kafka service in the module:

1. Java 1.8
2. Zookeeper
3. Kafka
4. Jupyter Notebook

There’re OS images published as AMIs present in the AWS Marketplace that contain these services. In such cases you can directly use the AMI in your EC2 instance.

This documentation contains instructions to set up all the required services to an EC2 instance from scratch.

- Go to the AWS Management Console and click on “All Services” and then click on “EC2” under the heading “Compute.”

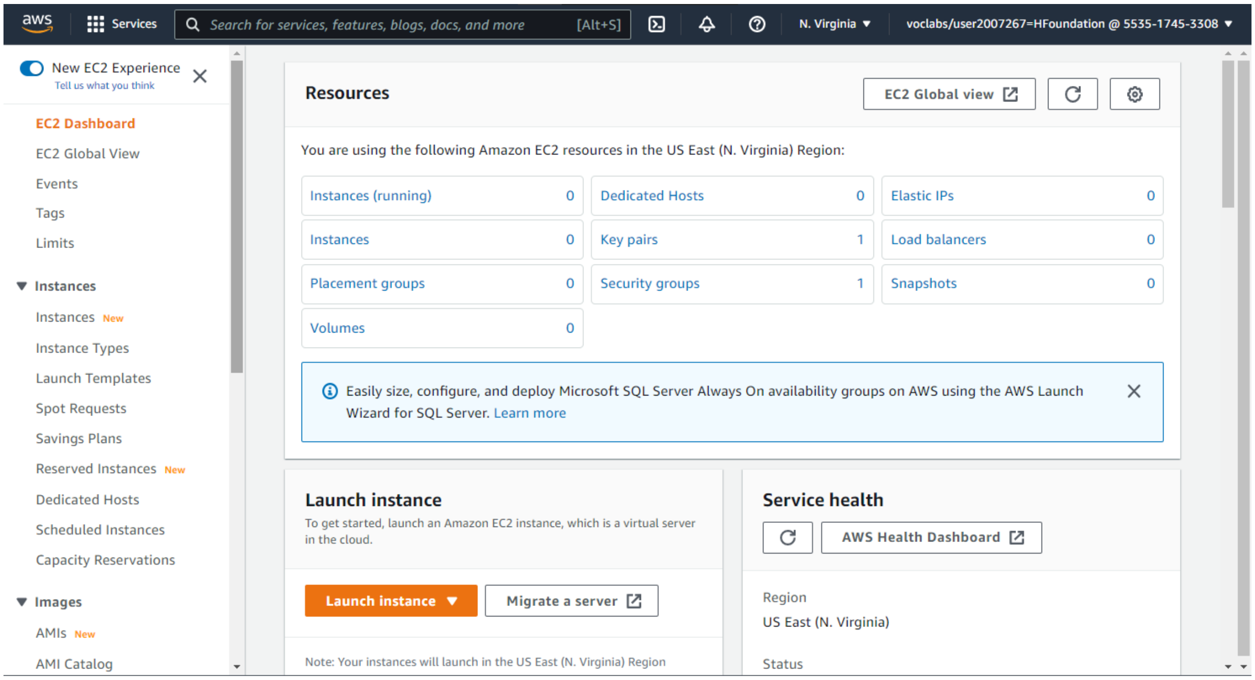


Learn the fundamentals and find valuable information to get the most out of AWS.

Learn from AWS experts and advance your skills and knowledge.

Discover new AWS services, features, and Regions.

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



aws | Services | Search for services, features, blogs, docs, and more [Alt+S] | N. Virginia ▼ | voclabs/user2007267=HFoundation @ 5535-1745-3308 ▼

New EC2 Experience
Tell us what you think

EC2 Dashboard
EC2 Global View
Events
Tags
Limits

▼ Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations

▼ Images
AMIs New
AMI Catalog

Resources

EC2 Global view

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

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				

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance ▼ | Migrate a server

Note: Your instances will launch in the US East (N. Virginia) Region

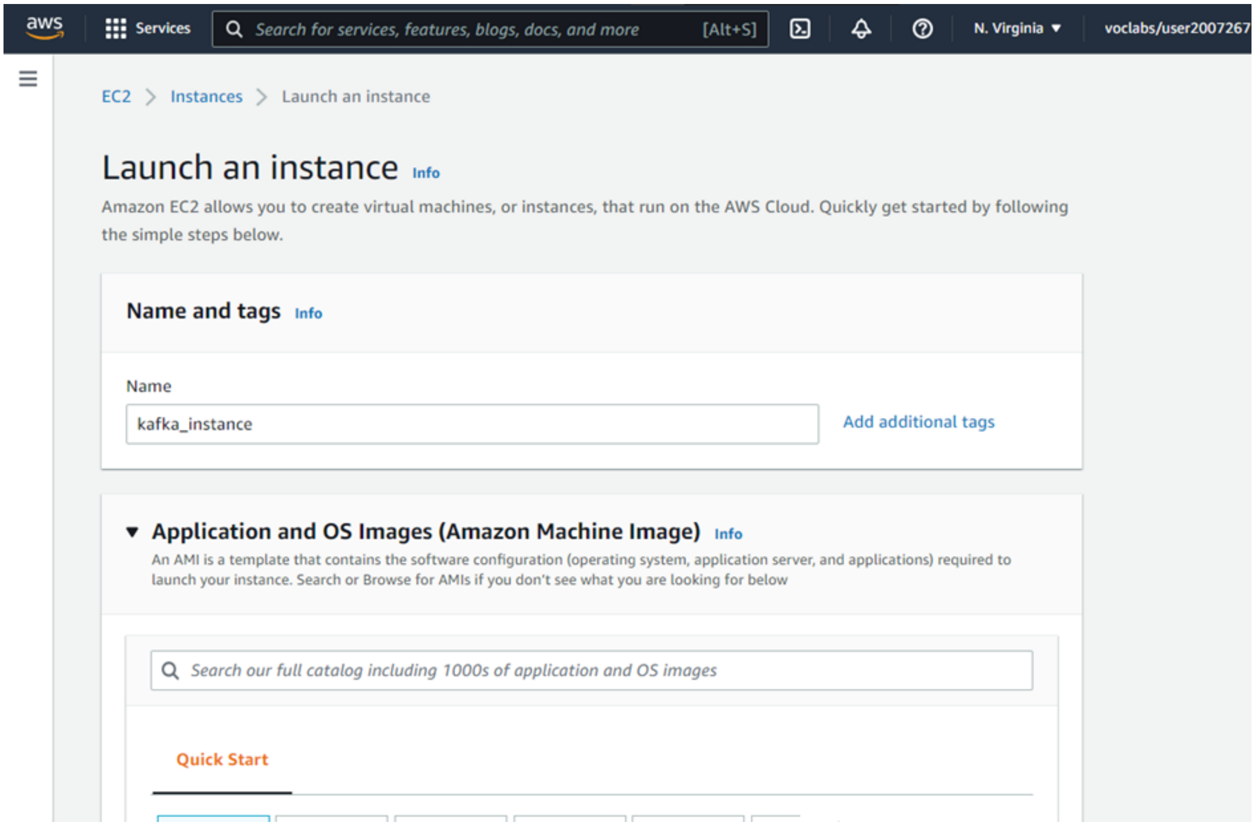
Service health

AWS Health Dashboard

Region
US East (N. Virginia)

Status

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



aws | Services | Search for services, features, blogs, docs, and more [Alt+S] | N. Virginia ▼ | voclabs/user2007267

EC2 > Instances > Launch an instance

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
kafka_instance

Add additional tags

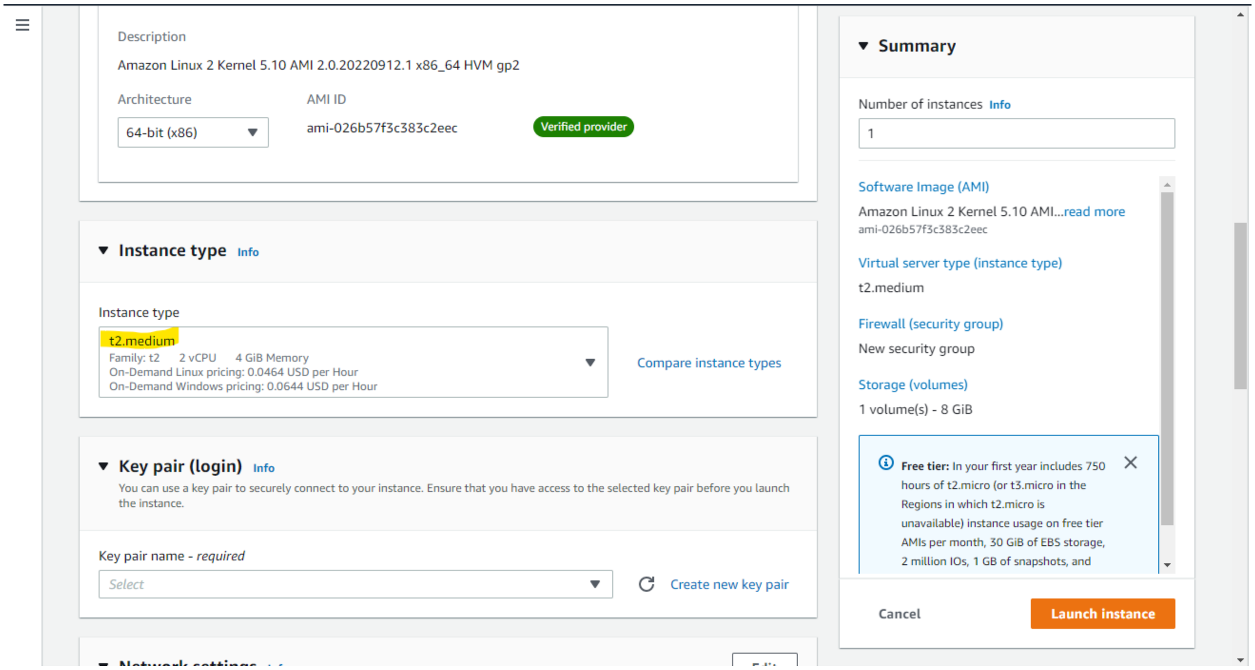
▼ **Application and OS Images (Amazon Machine Image)** Info

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

Quick Start

- 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*
Select

Create new key pair

▼ Network settings

▼ Summary

Number of instances Info
1

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

Virtual server type (instance type)
t2.medium

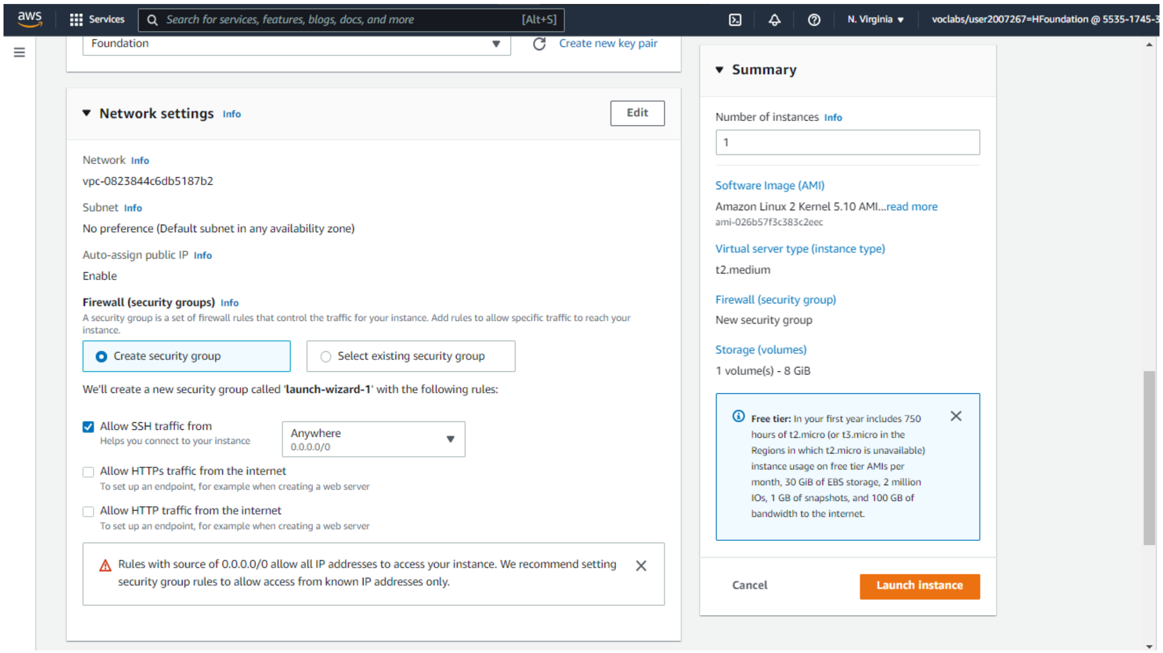
Firewall (security group)
New security group

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.



aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
voclabs/user2007267=HFoundation @ 5535-1745-3

Foundation

Create new key pair

▼ 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 — Anywhere 0.0.0.0/0
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
1

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

Virtual server type (instance type)
t2.medium

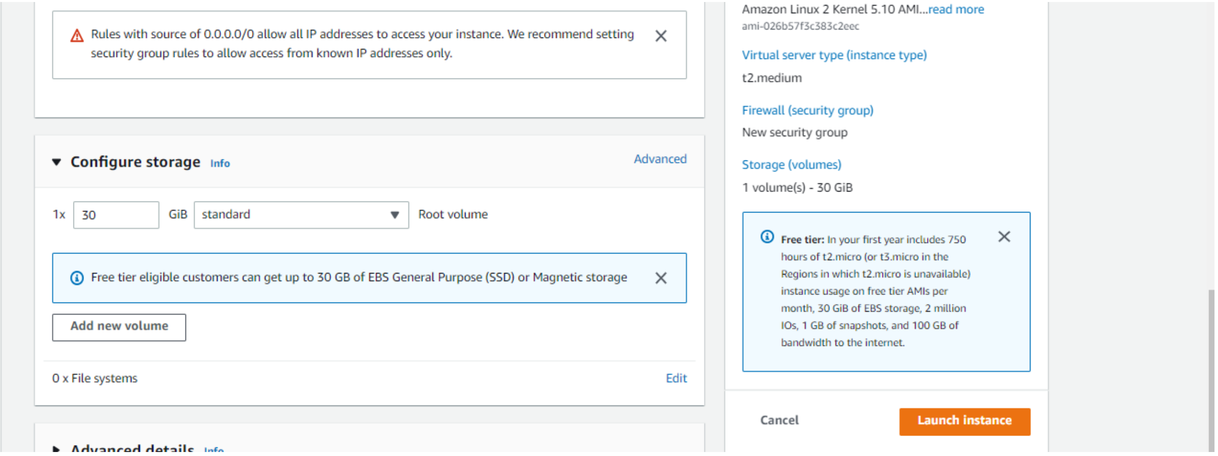
Firewall (security group)
New security group

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.

Configure storage Info

1x 30 GiB standard Root volume

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

0 x File systems

Amazon Linux 2 Kernel 5.10 AMI...read more
ami-026b57f3c383c2eec

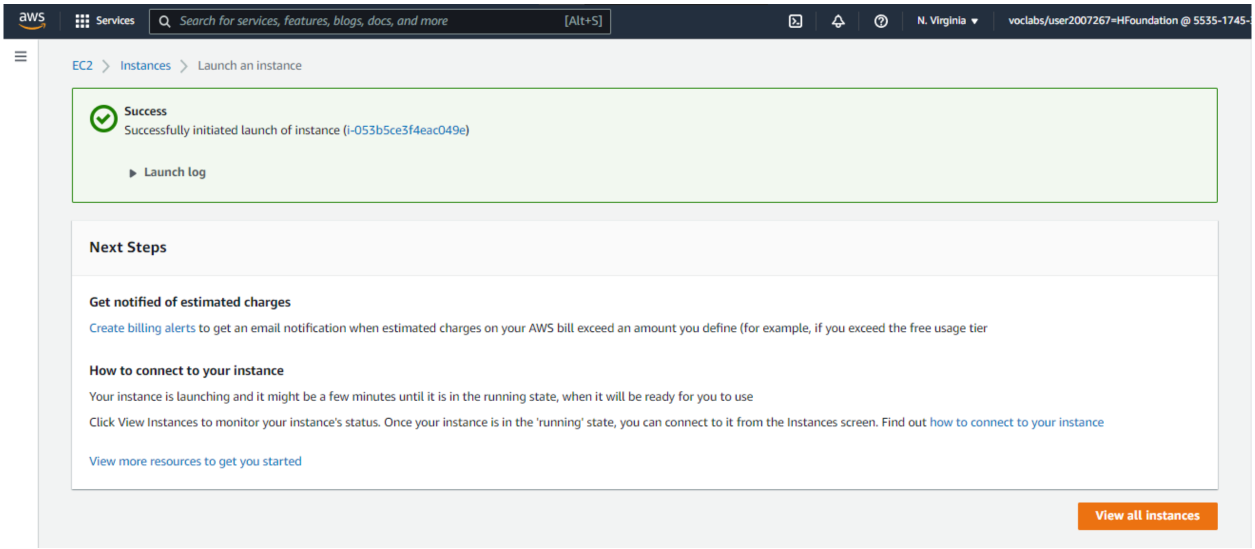
Virtual server type (instance type)
t2.medium

Firewall (security group)
New security group

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

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



aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
N. Virginia
voclabs/user2007267=HFoundation @ 5535-1745-
EC2 > Instances > Launch an instance
Launch log
View all instances

Success
Successfully initiated launch of instance (i-053b5ce3f4eac049e)

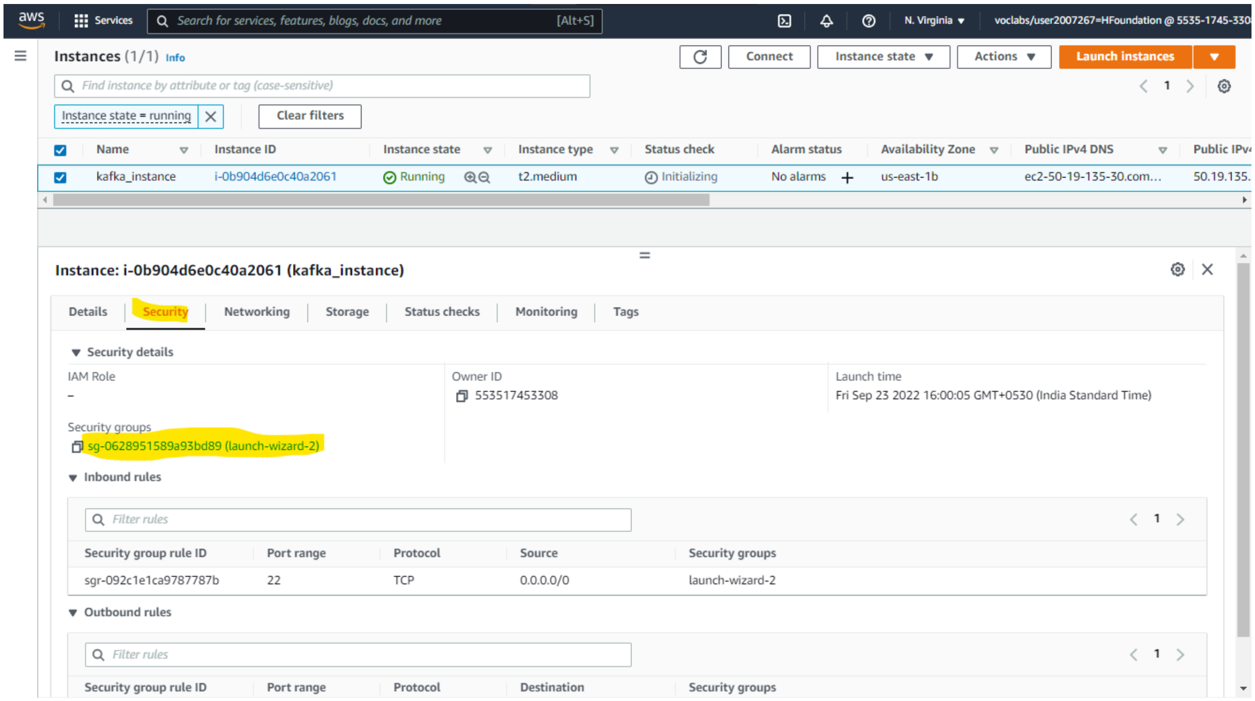
Next Steps

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

- 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.


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia ▼
voclabs/user2007267=HFoundation @ 5535-1745-330
Instances (1/1) Info
Connect
Instance state ▼
Actions ▼
Launch instances ▼
Find instance by attribute or tag (case-sensitive)
Instance state = running ✕
Clear filters
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Filter rules

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
☑	kafka_instance	i-0b904d6e0c40a2061	Running	t2.medium	Initializing	No alarms +	us-east-1b	ec2-50-19-135-30.com...	50.19.135.

Instance: i-0b904d6e0c40a2061 (kafka_instance)

▼ Security details

IAM Role
–

Owner ID
553517453308

Launch time
Fri Sep 23 2022 16:00:05 GMT+0530 (India Standard Time)

Security groups
sg-0628951589a93bd89 (launch-wizard-2)

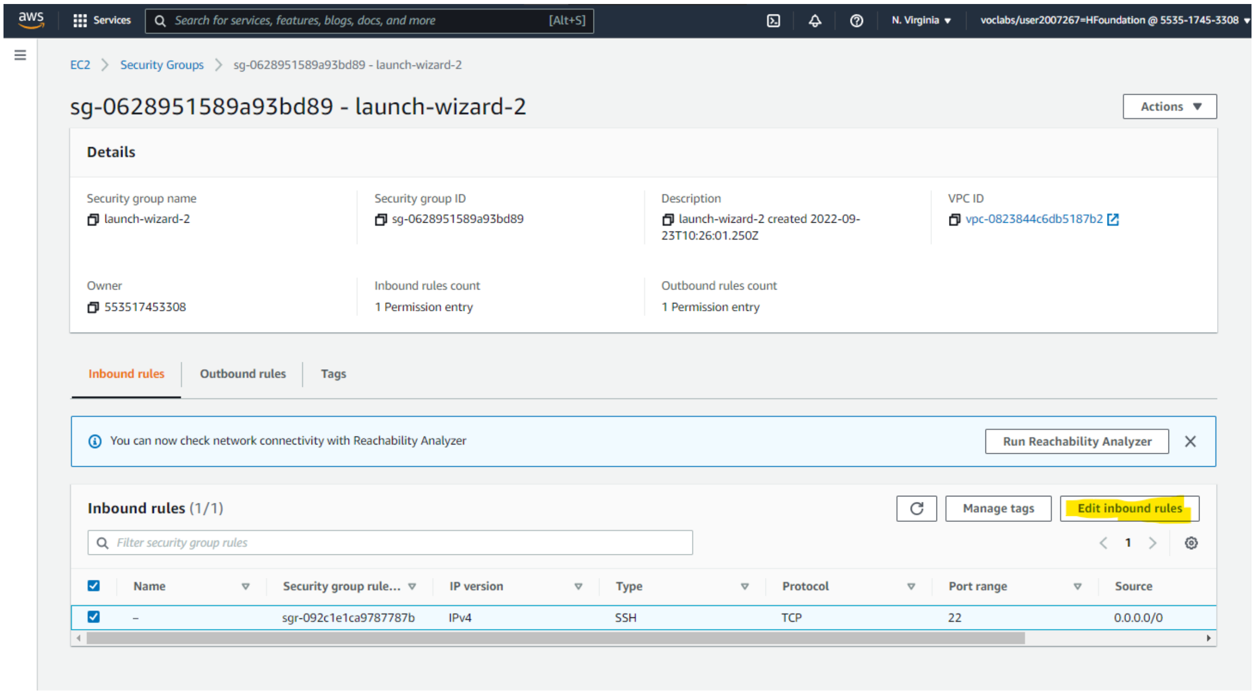
▼ Inbound rules

Security group rule ID	Port range	Protocol	Source	Security groups
sgr-092c1e1ca9787787b	22	TCP	0.0.0.0/0	launch-wizard-2

▼ Outbound rules

Security group rule ID	Port range	Protocol	Destination	Security groups

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


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia ▼
voclabs/user2007267=HFoundation @ 5535-1745-3308 ▼
EC2 > Security Groups > sg-0628951589a93bd89 - launch-wizard-2
Actions ▼
Inbound rules
Outbound rules
Tags
Run Reachability Analyzer
Manage tags
Edit inbound rules
Filter security group rules

sg-0628951589a93bd89 - launch-wizard-2

Details

Security group name
launch-wizard-2

Security group 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

Outbound rules count
1 Permission entry

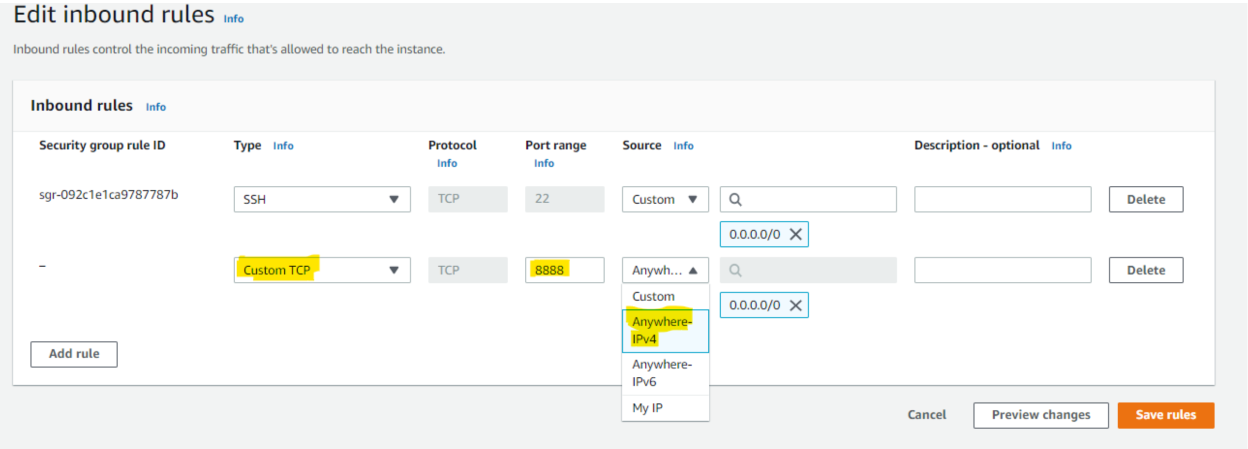
You can now check network connectivity with Reachability Analyzer

Inbound rules (1/1)

☑	Name	Security group rule...	IP version	Type	Protocol	Port range	Source
☑	–	sgr-092c1e1ca9787787b	IPv4	SSH	TCP	22	0.0.0.0/0

- 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



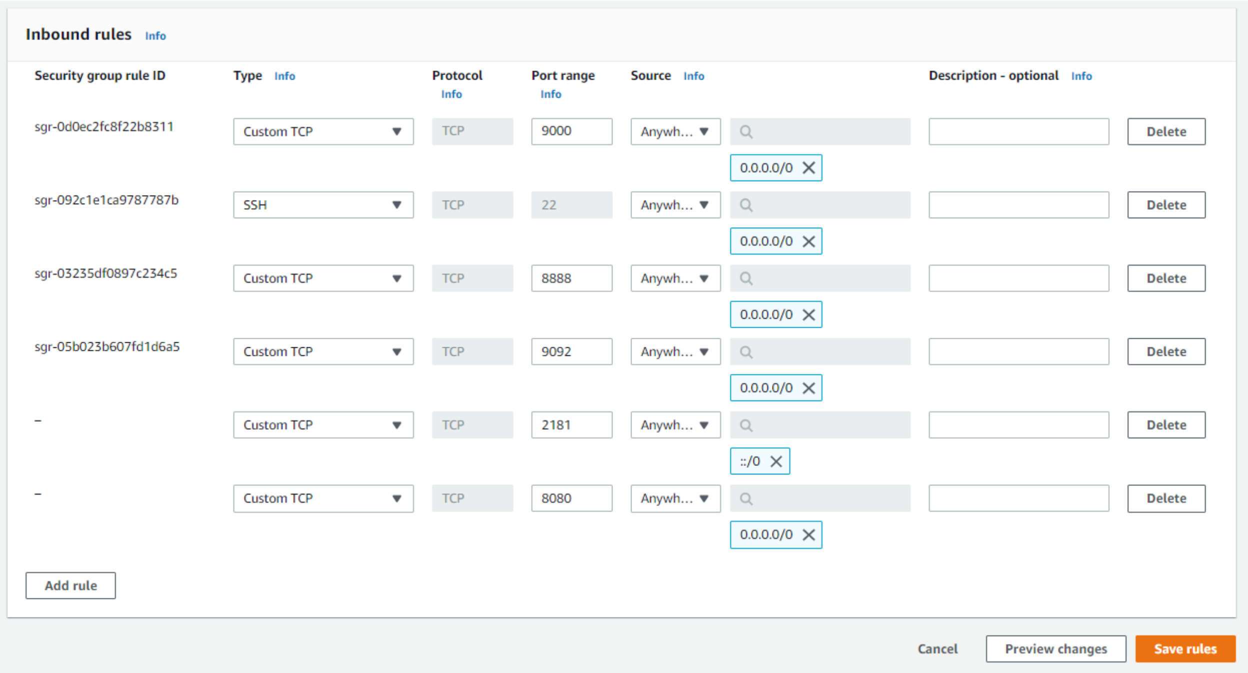
Edit inbound rules Info

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

Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-092c1e1ca9787787b	SSH	TCP	22	Custom	0.0.0.0/0		Delete
-	Custom TCP	TCP	8888	Anywh...	0.0.0.0/0		Delete

Similarly, you need to add the following port numbers:
2181, 8888, 9092, 9000, 8080
Make sure that it looks like this:



Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-0d0ec2fc8f22b8311	Custom TCP	TCP	9000	Anywh...	0.0.0.0/0		Delete
sgr-092c1e1ca9787787b	SSH	TCP	22	Anywh...	0.0.0.0/0		Delete
sgr-03235df0897c234c5	Custom TCP	TCP	8888	Anywh...	0.0.0.0/0		Delete
sgr-05b023b607fd1d6a5	Custom TCP	TCP	9092	Anywh...	0.0.0.0/0		Delete
-	Custom TCP	TCP	2181	Anywh...	::/0		Delete
-	Custom TCP	TCP	8080	Anywh...	0.0.0.0/0		Delete

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

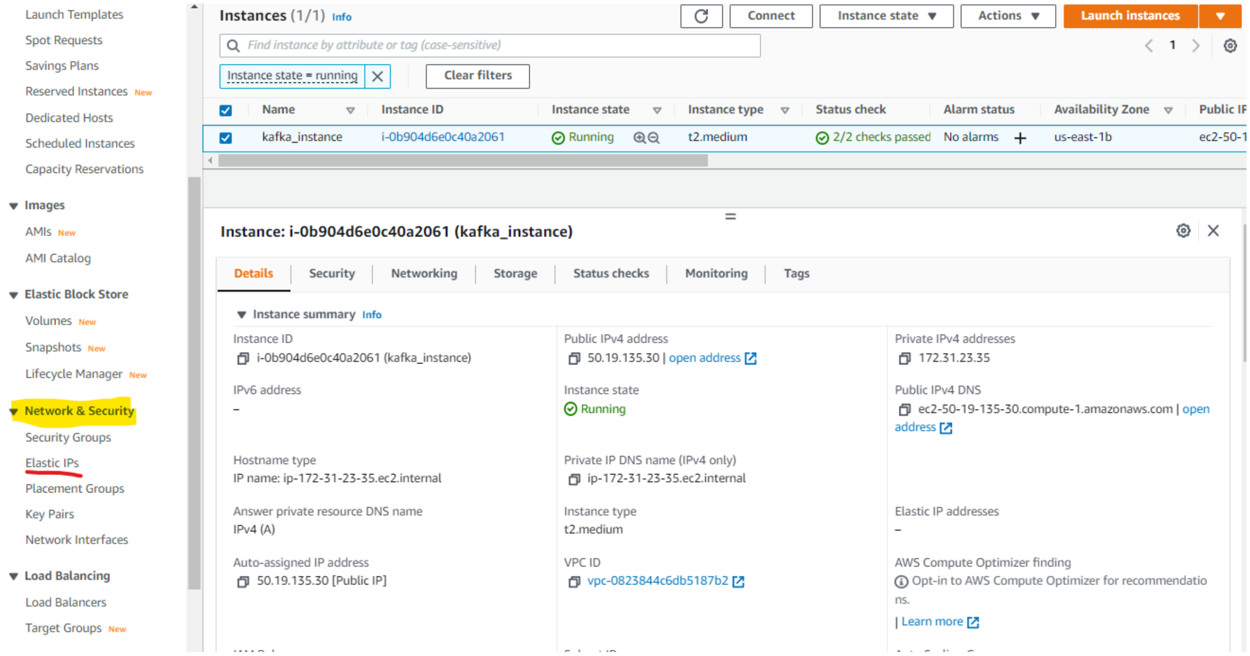
- Next click on the **Save rules.**

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.



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
☑	kafka_instance	i-0b904d6e0c40a2061	Running	t2.medium	2/2 checks passed	No alarms	us-east-1b	ec2-50-1

Instance: i-0b904d6e0c40a2061 (kafka_instance)

▼ Instance summary Info

Instance ID
i-0b904d6e0c40a2061 (kafka_instance)

IPv6 address
–

Hostname type
IP name: ip-172-31-23-35.ec2.internal

Answer private resource DNS name
IPv4 (A)

Auto-assigned IP address
50.19.135.30 [Public IP]

Public IPv4 address
50.19.135.30 | open address

Instance state
Running

Private IP DNS name (IPv4 only)
ip-172-31-23-35.ec2.internal

Instance type
t2.medium

VPC ID
vpc-0823844c6db5187b2

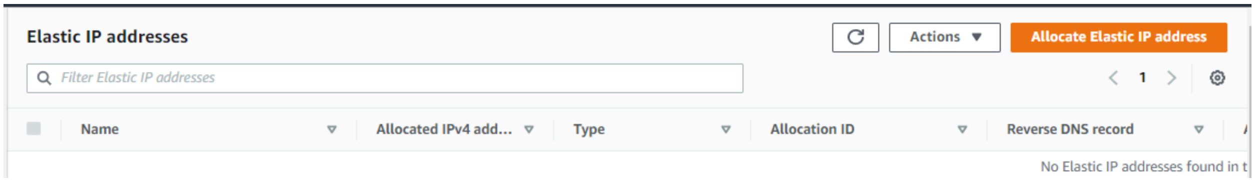
Private IPv4 addresses
172.31.23.35

Public IPv4 DNS
ec2-50-19-135-30.compute-1.amazonaws.com | open address

Elastic IP addresses
–

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations.
| Learn more

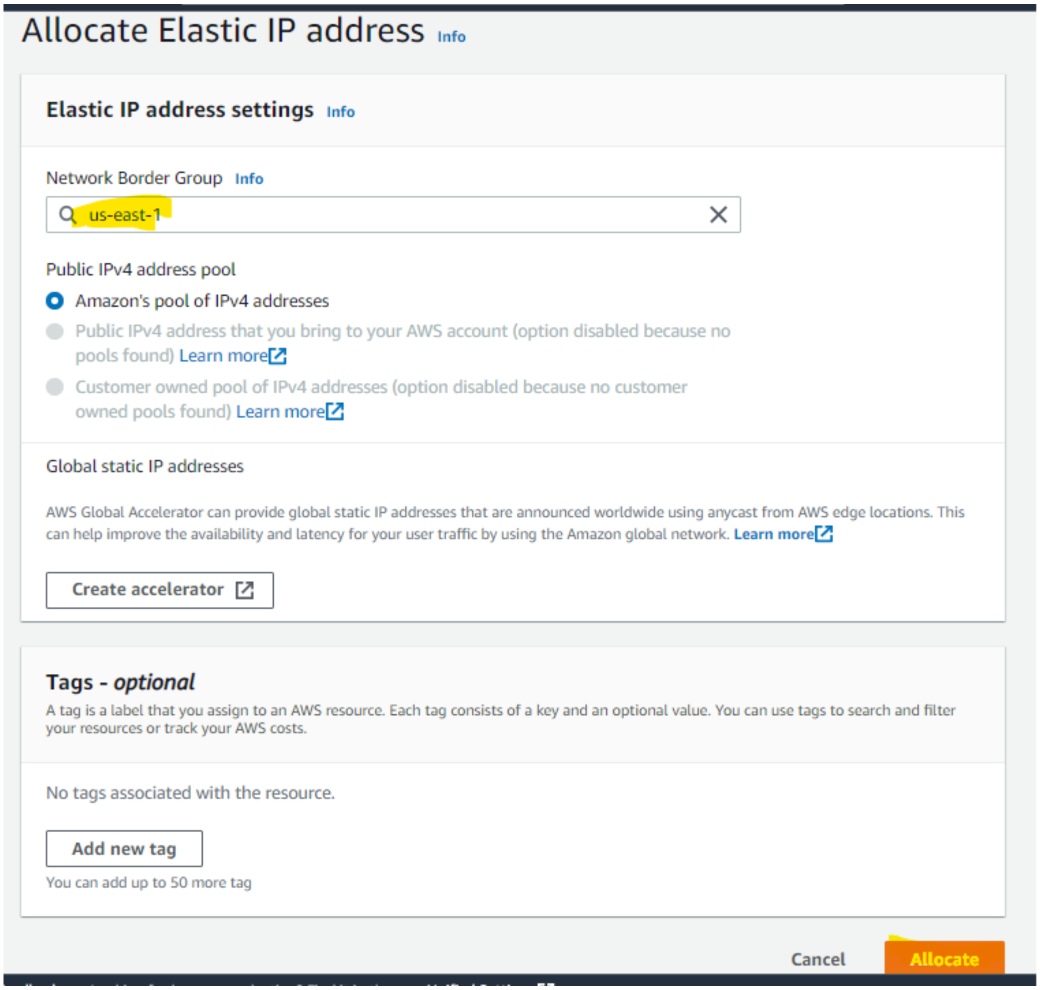
- Click on the **Elastic IPs** link. You will be redirected to a new page. Click on the **Allocate Elastic IP address** button



□	Name	Allocated IPv4 add...	Type	Allocation ID	Reverse DNS record

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

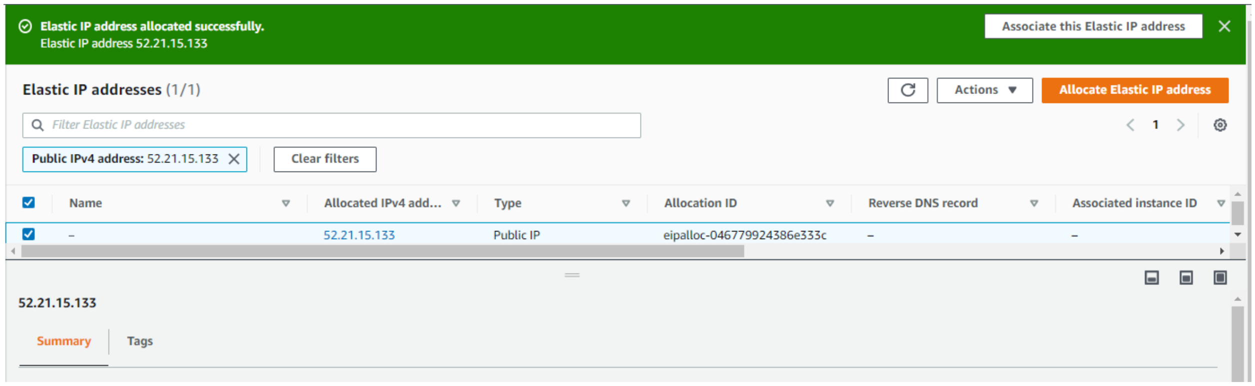
Tags - *optional*

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.

You can add up to 50 more tag

- 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)

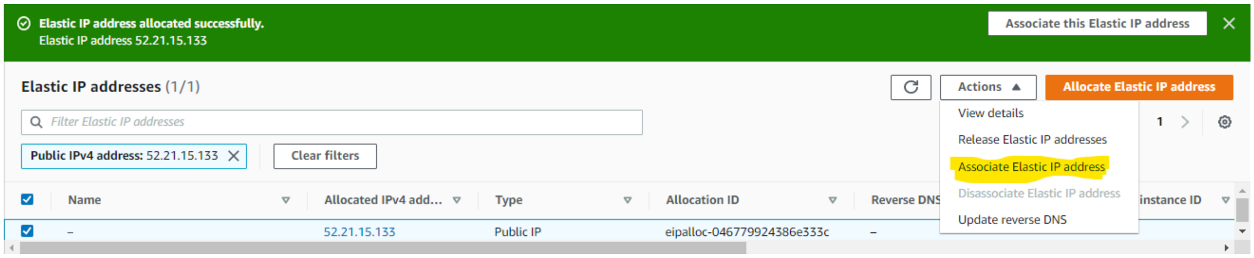
Actions
Allocate Elastic IP address
Filter Elastic IP addresses

Public IPv4 address: 52.21.15.133

☑	Name	Allocated IPv4 add...	Type	Allocation ID	Reverse DNS record	Associated instance ID
☑	–	52.21.15.133	Public IP	eipalloc-046779924386e333c	–	–

52.21.15.133

- 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

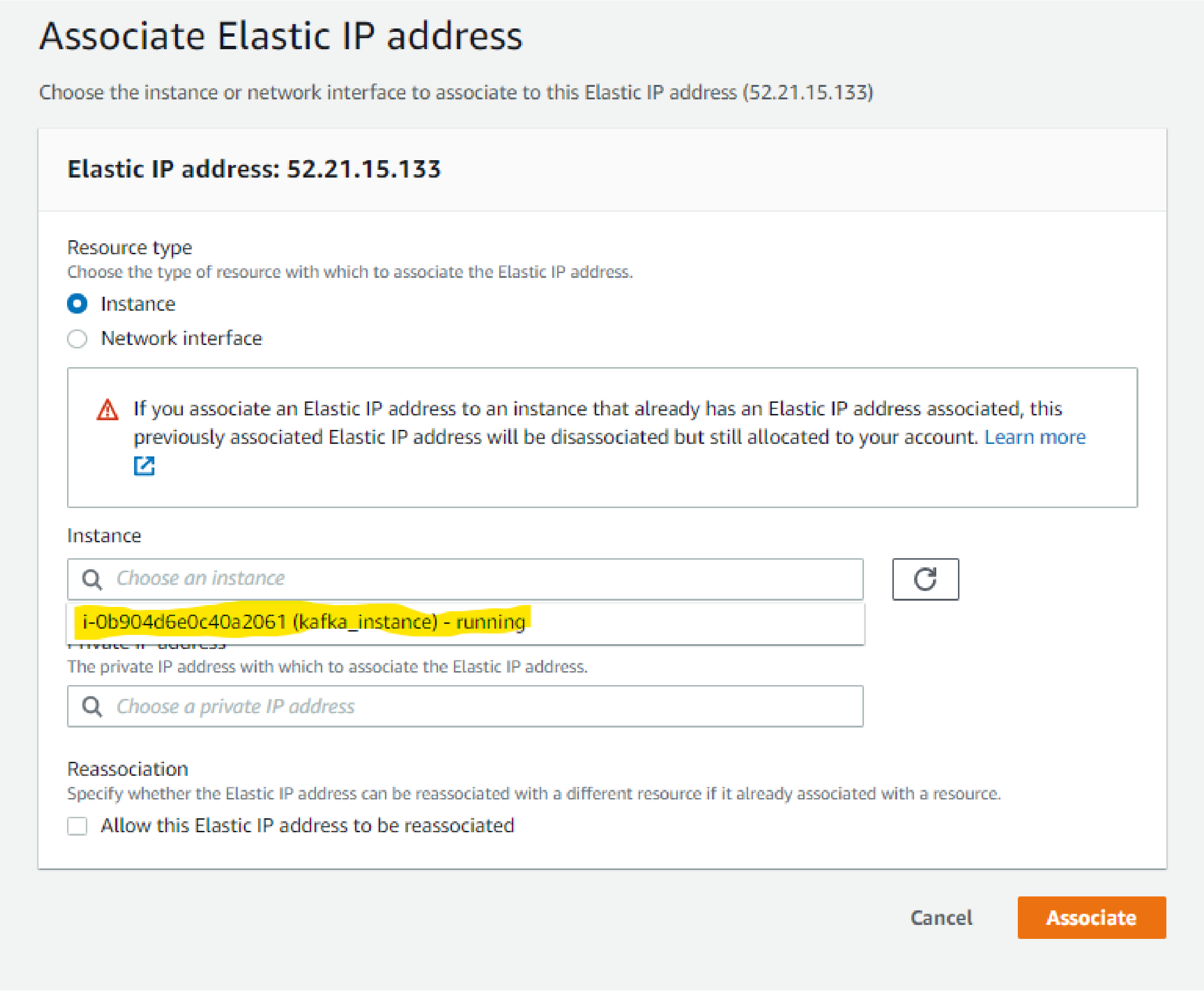
Elastic IP addresses (1/1)

Public IPv4 address: 52.21.15.133

☑	Name	Allocated IPv4 add...	Type	Allocation ID	Reverse DNS	instance ID
☑	–	52.21.15.133	Public IP	eipalloc-046779924386e333c	–	

Associate this Elastic IP address
Actions
Allocate Elastic IP address
Filter Elastic IP addresses
Clear filters
View details
Release Elastic IP addresses
Associate Elastic IP address
Disassociate Elastic IP address
Update reverse DNS
1

- 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

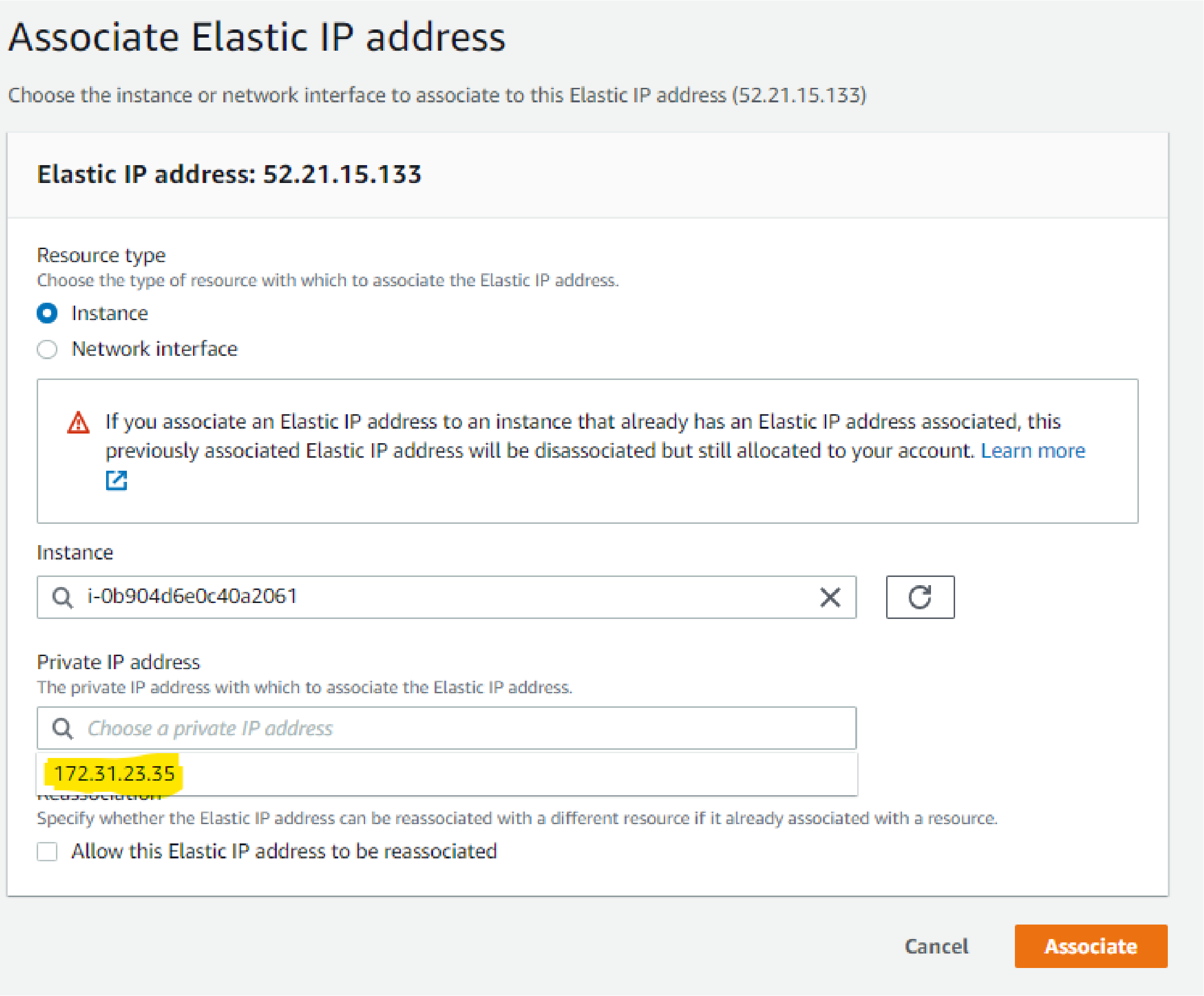
The private IP address with which to associate the Elastic IP address.

Choose a private IP address

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

□ Allow this Elastic IP address to be reassociated

- 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

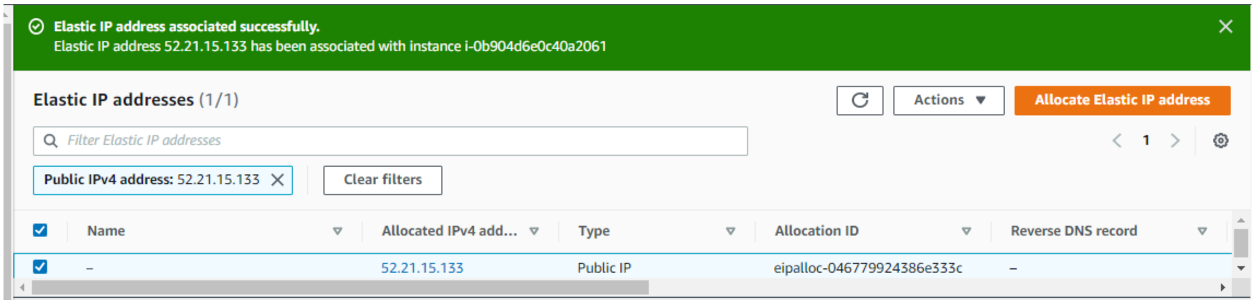
⚠ If you associate an Elastic IP address to an instance that already has an Elastic IP address associated, this previously associated Elastic IP address will be disassociated but still allocated to your account. Learn more

Instance
i-0b904d6e0c40a2061

Private IP address
The private IP address with which to associate the Elastic IP address.
Choose a private IP address
172.31.23.35

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

- 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**.



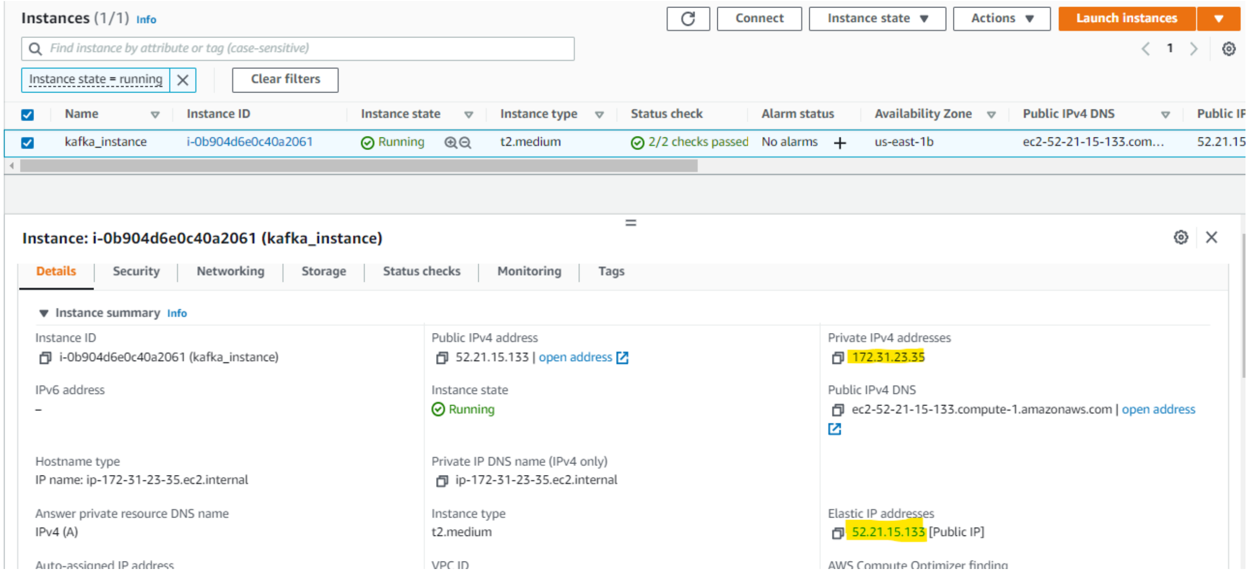
Elastic IP address associated successfully.
Elastic IP address 52.21.15.133 has been associated with instance i-0b904d6e0c40a2061

Elastic IP addresses (1/1)

Public IPv4 address: 52.21.15.133

☑	Name	Allocated IPv4 add...	Type	Allocation ID	Reverse DNS record
☑	–	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.



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IF
☑	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)

▼ 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

Answer private resource DNS name
IPv4 (A)

Instance type
t2.medium

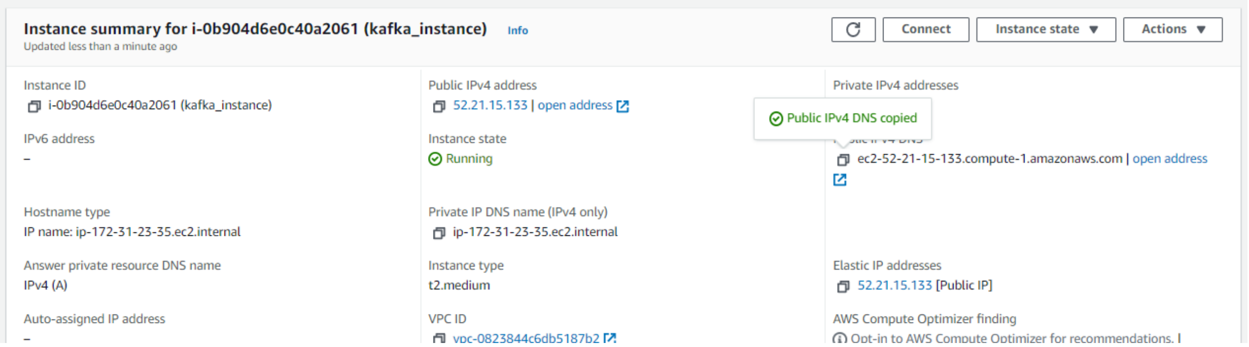
Elastic IP addresses
52.21.15.133 [Public IP]

Auto-assigned IP address

VPC ID

AWS Compute Optimizer finding

- 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

Instance ID
i-0b904d6e0c40a2061 (kafka_instance)

Public IPv4 address
52.21.15.133 | open address

Private IPv4 addresses

Public IPv4 DNS copied

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

Answer private resource DNS name
IPv4 (A)

Instance type
t2.medium

Elastic IP addresses
52.21.15.133 [Public IP]

Auto-assigned IP address
–

VPC ID
vpc-0823844c6db5187b2

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. |

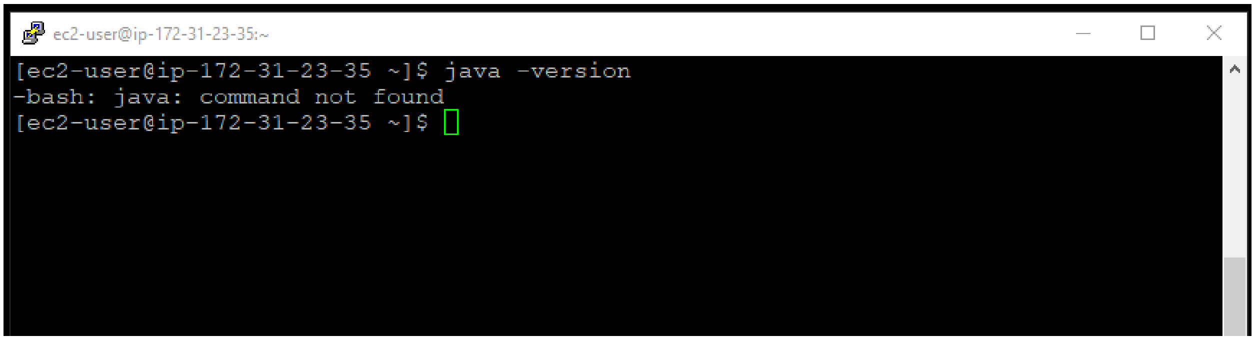
Now we'll instance all the required packages to work with Kafka. As mentioned in the beginning, the following services are required for working with the Kafka service in the module:

- Java 1.8
- Zookeeper
- Kafka
- Jupyter Notebook

Before you begin working, use the **sudo yum update** command to update the packages.

UPDATING JAVA ENVIRONMENT

- Check the Java version by using the **java -version** command

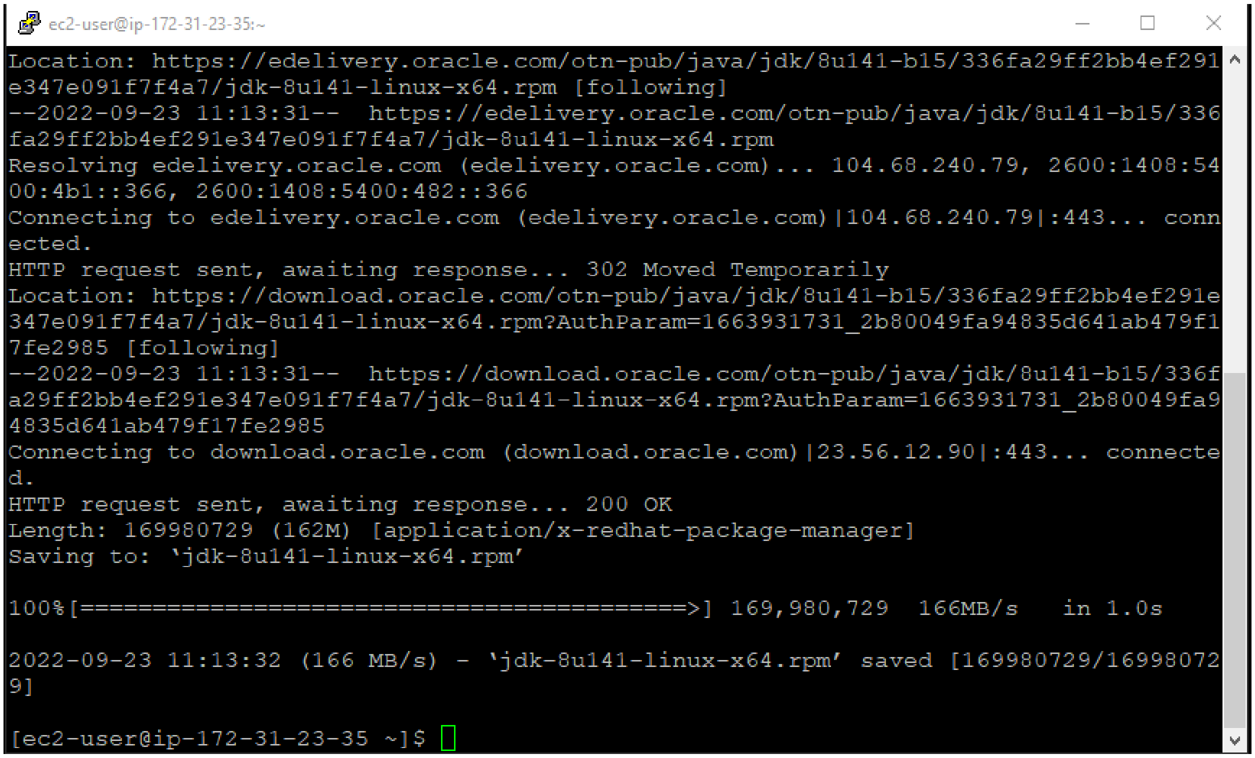


```
[ec2-user@ip-172-31-23-35 ~]$ java -version
-bash: java: command not found
[ec2-user@ip-172-31-23-35 ~]$ 
```

- If you find the above error, it means that the java package has not been installed in the instance yet.

Use the following command to download the RPM package of Oracle JDK (8u121)

```
wget --no-check-certificate --no-cookies --header "Cookie:
oraclelicense=accept-securebackup-cookie"
http://download.oracle.com/otn-pub/java/jdk/8u141-b15/336fa29ff2bb4ef291e34
7e091f7f4a7/jdk-8u141-linux-x64.rpm
```



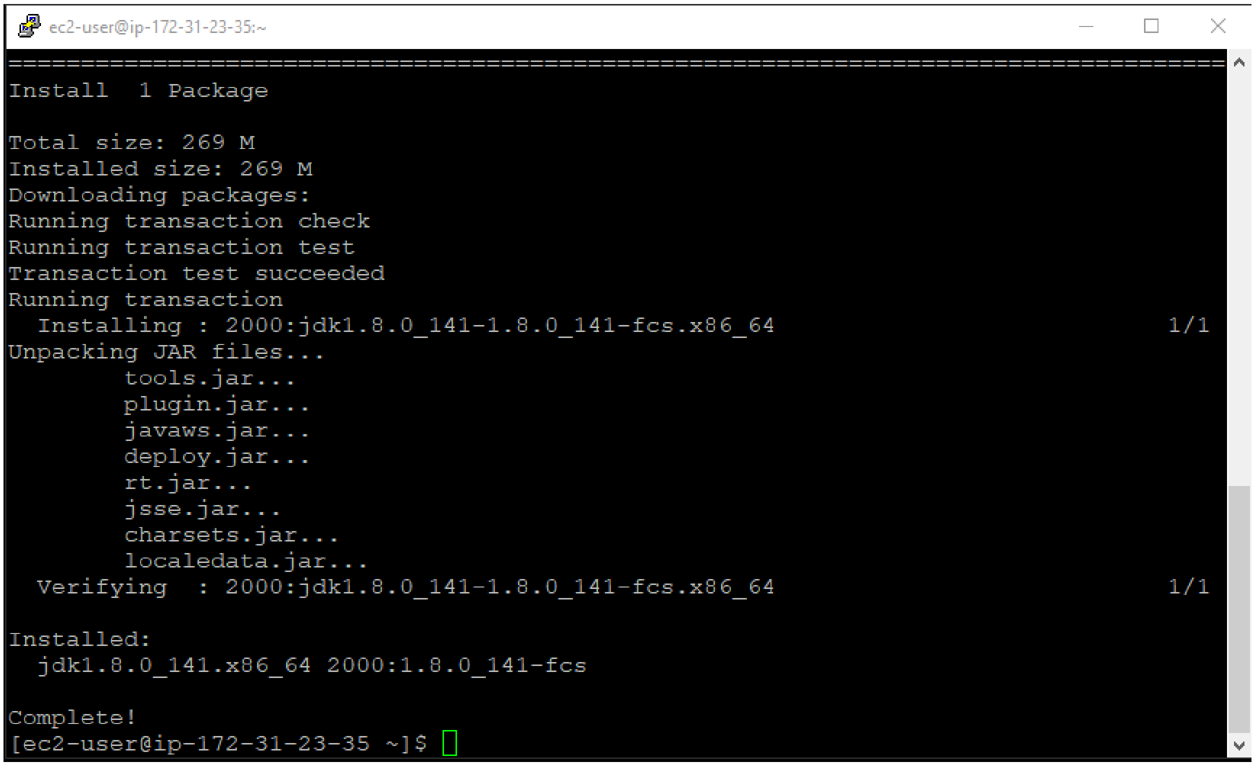
```
Location: https://edelivery.oracle.com/otn-pub/java/jdk/8u141-b15/336fa29ff2bb4ef291
e347e091f7f4a7/jdk-8u141-linux-x64.rpm [following]
--2022-09-23 11:13:31--  https://edelivery.oracle.com/otn-pub/java/jdk/8u141-b15/336
fa29ff2bb4ef291e347e091f7f4a7/jdk-8u141-linux-x64.rpm
Resolving edelivery.oracle.com (edelivery.oracle.com)... 104.68.240.79, 2600:1408:54
00:4b1::366, 2600:1408:5400:482::366
Connecting to edelivery.oracle.com (edelivery.oracle.com)|104.68.240.79|:443... conn
ected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://download.oracle.com/otn-pub/java/jdk/8u141-b15/336fa29ff2bb4ef291e
347e091f7f4a7/jdk-8u141-linux-x64.rpm?AuthParam=1663931731_2b80049fa94835d641ab479f1
7fe2985 [following]
--2022-09-23 11:13:31--  https://download.oracle.com/otn-pub/java/jdk/8u141-b15/336f
a29ff2bb4ef291e347e091f7f4a7/jdk-8u141-linux-x64.rpm?AuthParam=1663931731_2b80049fa9
4835d641ab479f17fe2985
Connecting to download.oracle.com (download.oracle.com)|23.56.12.90|:443... connecte
d.
HTTP request sent, awaiting response... 200 OK
Length: 169980729 (162M) [application/x-redhat-package-manager]
Saving to: 'jdk-8u141-linux-x64.rpm'

100%[==========================================>] 169,980,729  166MB/s   in 1.0s

2022-09-23 11:13:32 (166 MB/s) - 'jdk-8u141-linux-x64.rpm' saved [169980729/16998072
9]

[ec2-user@ip-172-31-23-35 ~]$ 
```

NOTE: It is important to manually instruct wget to send the cookie and here this is done using **–header “Cookie: oraclelicense=accept-securebackup-cookie”** where oraclelicense is the name of the cookie and accept-securebackup-cookie is its value.

- Now install JDK 8 using the command below:

```
sudo yum install -y jdk-8u141-linux-x64.rpm
```

When you’ve successfully installed the Java Development Kit, you’ll see the command prompt as the image below

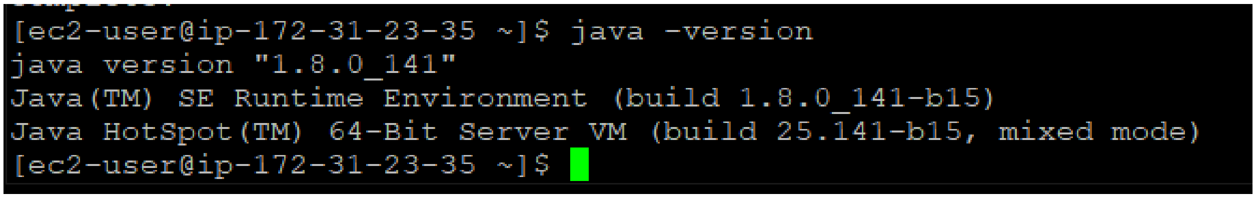


```
================================================================================
Install  1 Package

Total size: 269 M
Installed size: 269 M
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 2000:jdk1.8.0_141-1.8.0_141-fcs.x86_64                        1/1
Unpacking JAR files...
        tools.jar...
        plugin.jar...
        javaws.jar...
        deploy.jar...
        rt.jar...
        jsse.jar...
        charsets.jar...
        localedata.jar...
  Verifying  : 2000:jdk1.8.0_141-1.8.0_141-fcs.x86_64                        1/1

Installed:
  jdk1.8.0_141.x86_64 2000:1.8.0_141-fcs

Complete!
[ec2-user@ip-172-31-23-35 ~]$ 
```

- Verify oracle JDK version using the following command

```
java -version
```

If the JDK installation has been successful, you will see the image below



```
[ec2-user@ip-172-31-23-35 ~]$ java -version
java version "1.8.0_141"
Java(TM) SE Runtime Environment (build 1.8.0_141-b15)
Java HotSpot(TM) 64-Bit Server VM (build 25.141-b15, mixed mode)
[ec2-user@ip-172-31-23-35 ~]$ 
```

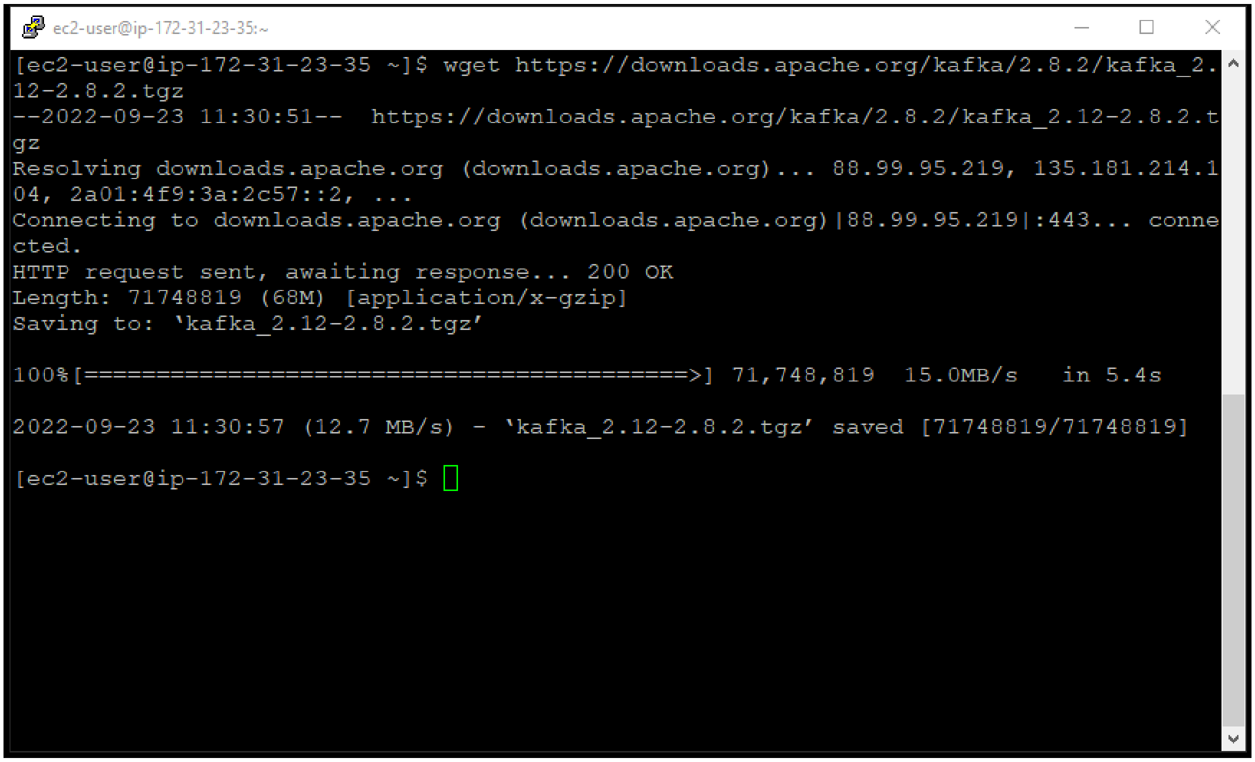
- The java version is now updated. After installing Java, follow the steps mentioned below in sequence to start Kafka on your instance.

INSTALLING KAFKA

We'll now install kafka_2.12-2.8.2 to the EC2 instance. Please note that in the module though we've installed a different version of kafka_2.12-2.3.0, the commands for working with Kafka will still remain the same.

- Using the following wget command, download the Kafka package

```
wget https://downloads.apache.org/kafka/2.8.2/kafka_2.12-2.8.2.tgz
```



```
[ec2-user@ip-172-31-23-35 ~]$ wget https://downloads.apache.org/kafka/2.8.2/kafka_2.
12-2.8.2.tgz
--2022-09-23 11:30:51--  https://downloads.apache.org/kafka/2.8.2/kafka_2.12-2.8.2.t
gz
Resolving downloads.apache.org (downloads.apache.org)... 88.99.95.219, 135.181.214.1
04, 2a01:4f9:3a:2c57::2, ...
Connecting to downloads.apache.org (downloads.apache.org)|88.99.95.219|:443... conne
cted.
HTTP request sent, awaiting response... 200 OK
Length: 71748819 (68M) [application/x-gzip]
Saving to: 'kafka_2.12-2.8.2.tgz'

100%[==========================================>] 71,748,819  15.0MB/s   in 5.4s

2022-09-23 11:30:57 (12.7 MB/s) - 'kafka_2.12-2.8.2.tgz' saved [71748819/71748819]

[ec2-user@ip-172-31-23-35 ~]$ 
```

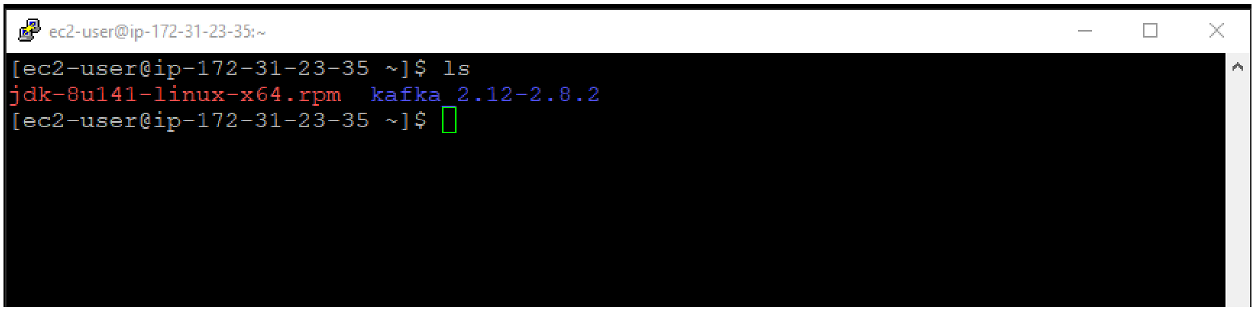
- The kafka file will be saved to your instance as .tgz file. Now use the following command to extract the .tgz package

```
tar -xvzf kafka_2.12-2.8.2.tgz
```

- Since the original file is of no use, we can remove it using the following command:

```
rm kafka_2.12-2.8.2.tgz
```

Use the **ls** command to see the files in your EC2 instance.



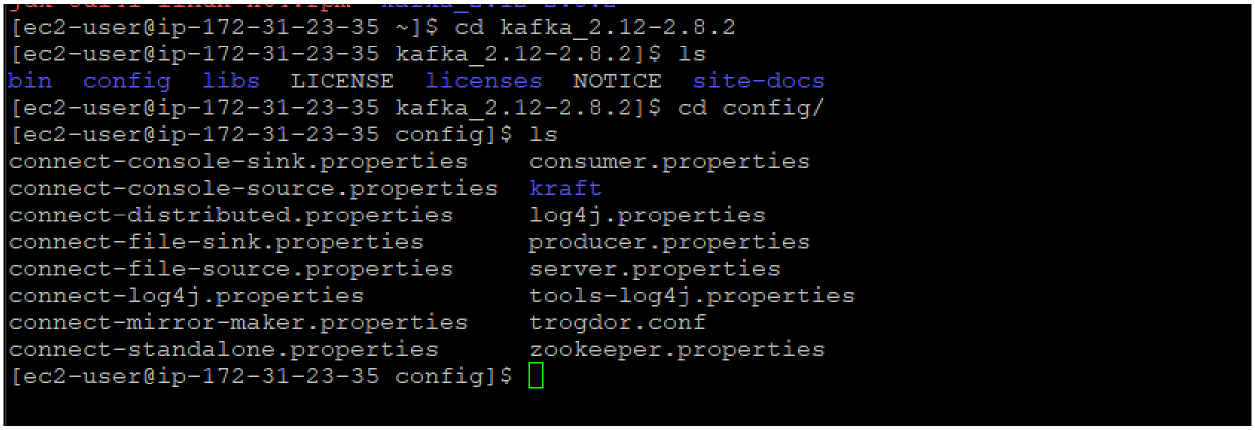
```
[ec2-user@ip-172-31-23-35 ~]$ ls
jdk-8u141-linux-x64.rpm  kafka_2.12-2.8.2
[ec2-user@ip-172-31-23-35 ~]$
```

KAFKA SERVER CONFIGURATION

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

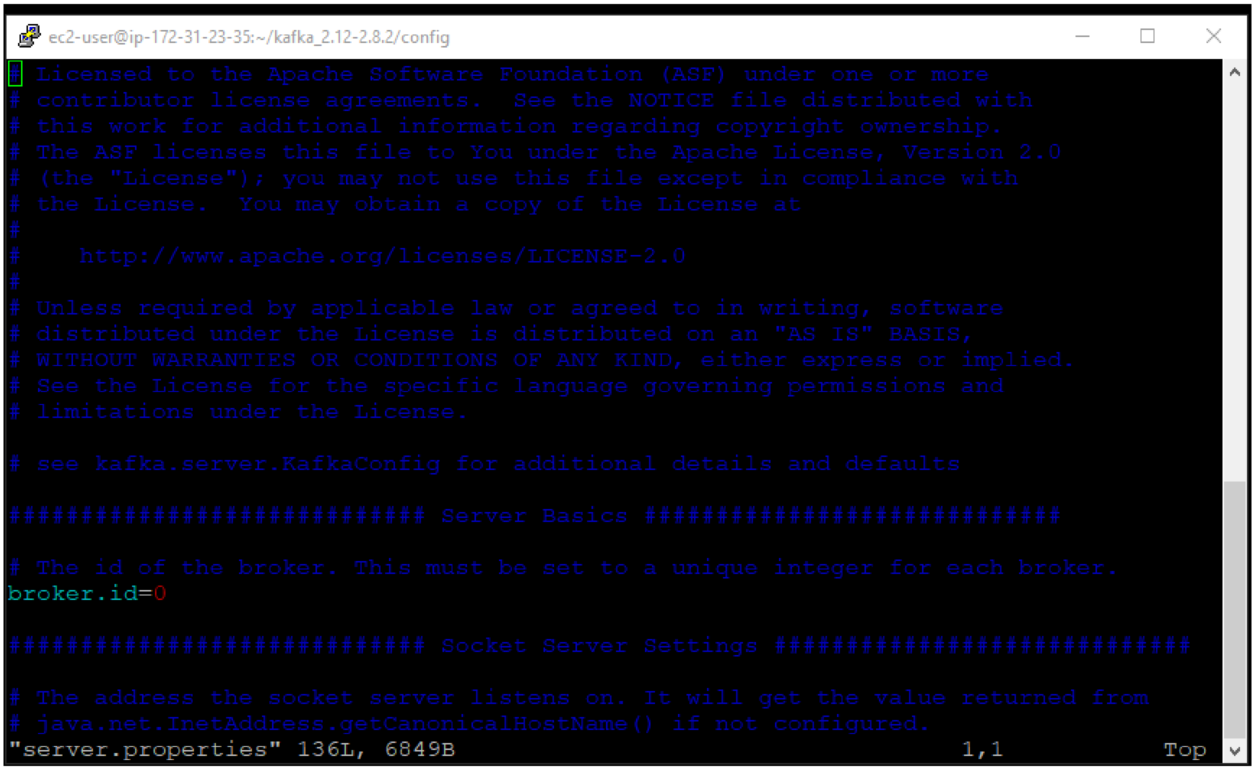
To configure the Kafka server, you need to make changes to the server.properties file for which you need to go through the following steps:

- Login to the EC2 instance and go inside the Kafka directory present there. The command for the same is **cd /kafka_2.12-2.8.2.** 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.



```
jdk-8u141-linux-x64.rpm  kafka_2.12-2.8.2
[ec2-user@ip-172-31-23-35 ~]$ cd kafka_2.12-2.8.2
[ec2-user@ip-172-31-23-35 kafka_2.12-2.8.2]$ ls
bin  config  libs  LICENSE  licenses  NOTICE  site-docs
[ec2-user@ip-172-31-23-35 kafka_2.12-2.8.2]$ cd config/
[ec2-user@ip-172-31-23-35 config]$ ls
connect-console-sink.properties    consumer.properties
connect-console-source.properties  kraft
connect-distributed.properties     log4j.properties
connect-file-sink.properties       producer.properties
connect-file-source.properties     server.properties
connect-log4j.properties           tools-log4j.properties
connect-mirror-maker.properties    trogdor.conf
connect-standalone.properties      zookeeper.properties
[ec2-user@ip-172-31-23-35 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.



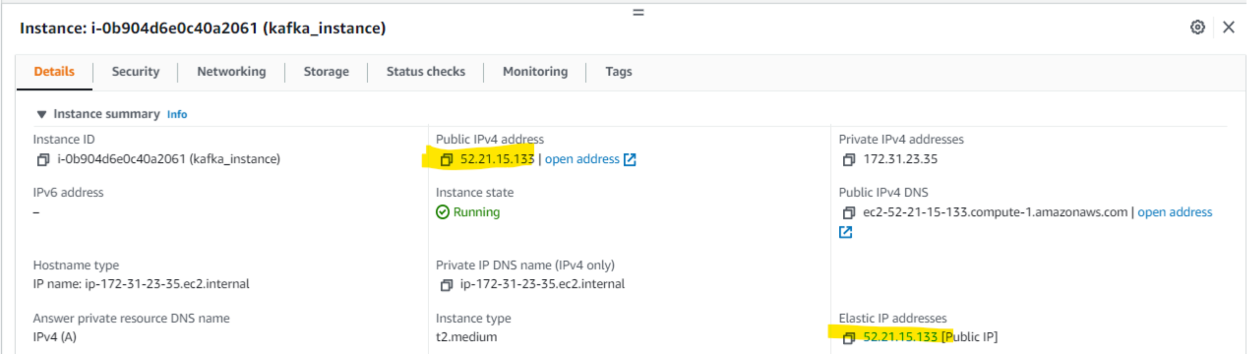
ec2-user@ip-172-31-23-35:~/kafka_2.12-2.8.2/config

```
# 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.
"server.properties" 136L, 6849B                                1,1           Top
```

- 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)

▼ 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

Answer private resource DNS name
IPv4 (A)

Instance type
t2.medium

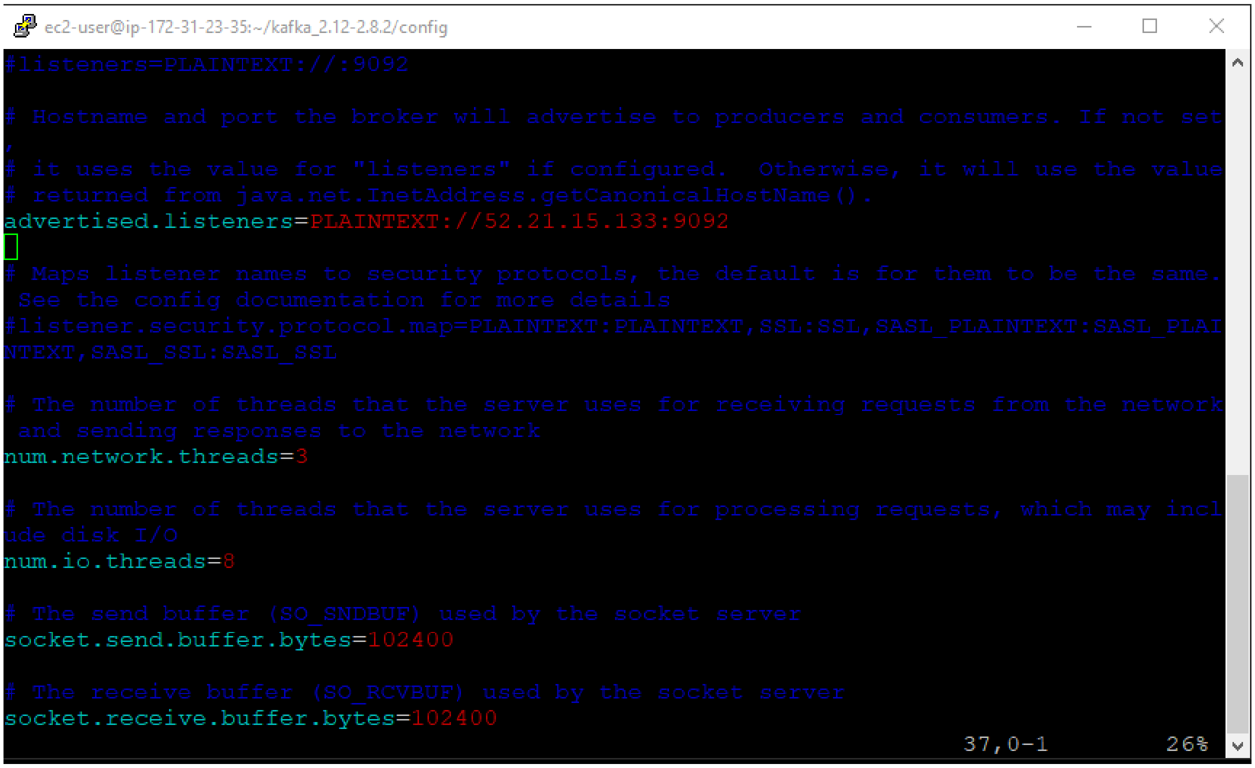
Elastic IP addresses
52.21.15.133 [Public IP]

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. For instance, the line would read as follows:

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

Your screen should look something as shown below:



```
#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
                                                              37,0-1          26%
```

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.

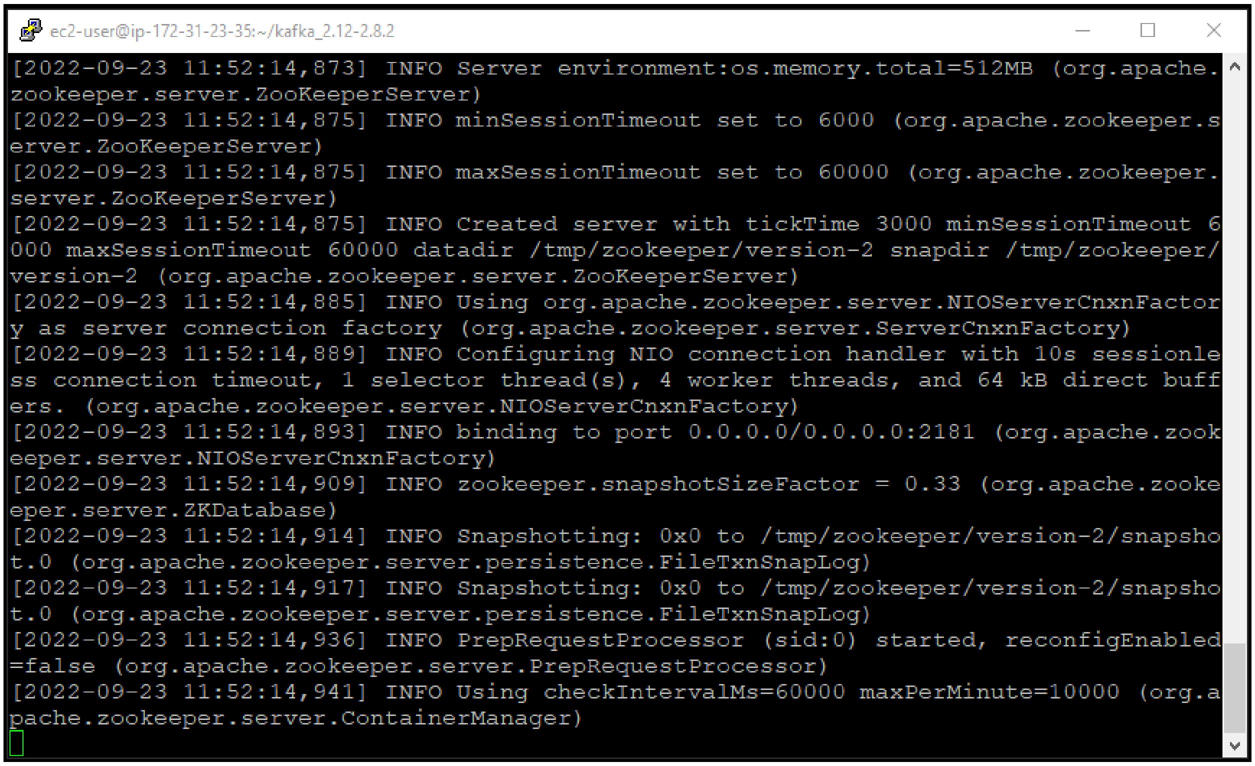
With these steps, you have configured the Kafka server to connect it with an IDE such as Eclipse.

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_2.12-2.8.2/** command and then start the Zookeeper server using the **bin/zookeeper-server-start.sh config/zookeeper.properties** command. You should get the following output.



```
[2022-09-23 11:52:14,873] INFO Server environment:os.memory.total=512MB (org.apache.
zookeeper.server.ZooKeeperServer)
[2022-09-23 11:52:14,875] INFO minSessionTimeout set to 6000 (org.apache.zookeeper.s
erver.ZooKeeperServer)
[2022-09-23 11:52:14,875] INFO maxSessionTimeout set to 60000 (org.apache.zookeeper.
server.ZooKeeperServer)
[2022-09-23 11:52:14,875] INFO Created server with tickTime 3000 minSessionTimeout 6
000 maxSessionTimeout 60000 datadir /tmp/zookeeper/version-2 snapdir /tmp/zookeeper/
version-2 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-09-23 11:52:14,885] INFO Using org.apache.zookeeper.server.NIOServerCnxnFactor
y as server connection factory (org.apache.zookeeper.server.ServerCnxnFactory)
[2022-09-23 11:52:14,889] INFO Configuring NIO connection handler with 10s sessionle
ss connection timeout, 1 selector thread(s), 4 worker threads, and 64 kB direct buff
ers. (org.apache.zookeeper.server.NIOServerCnxnFactory)
[2022-09-23 11:52:14,893] INFO binding to port 0.0.0.0/0.0.0.0:2181 (org.apache.zook
eeper.server.NIOServerCnxnFactory)
[2022-09-23 11:52:14,909] INFO zookeeper.snapshotSizeFactor = 0.33 (org.apache.zooke
eper.server.ZKDatabase)
[2022-09-23 11:52:14,914] INFO Snapshotting: 0x0 to /tmp/zookeeper/version-2/snapsho
t.0 (org.apache.zookeeper.server.persistence.FileTxnSnapLog)
[2022-09-23 11:52:14,917] INFO Snapshotting: 0x0 to /tmp/zookeeper/version-2/snapsho
t.0 (org.apache.zookeeper.server.persistence.FileTxnSnapLog)
[2022-09-23 11:52:14,936] INFO PrepRequestProcessor (sid:0) started, reconfigEnabled
=false (org.apache.zookeeper.server.PrepRequestProcessor)
[2022-09-23 11:52:14,941] INFO Using checkIntervalMs=60000 maxPerMinute=10000 (org.a
pache.zookeeper.server.ContainerManager)
```

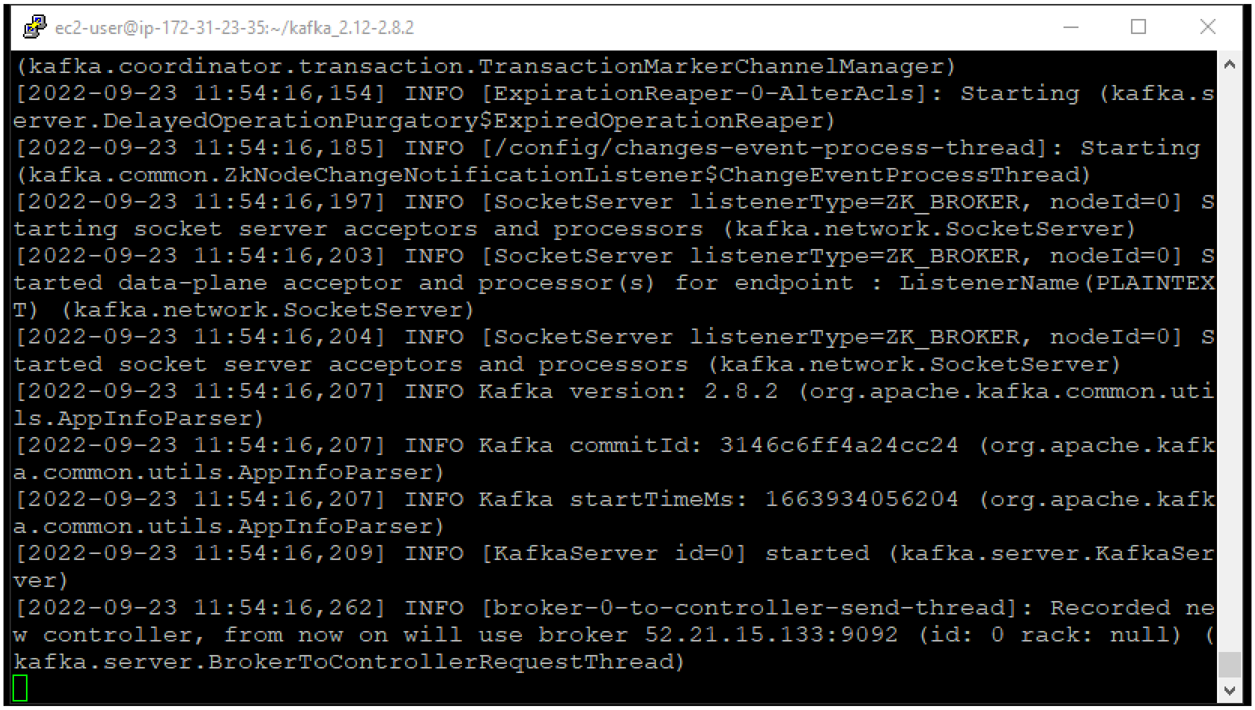
At the bottom of the screen, you would get something like:
INFO binding to port 0.0.0.0/0.0.0.0:2181 (org.apache.zookeeper.server.NIOServerCnxnFactory)

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_2.12-2.8.2** 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)**”

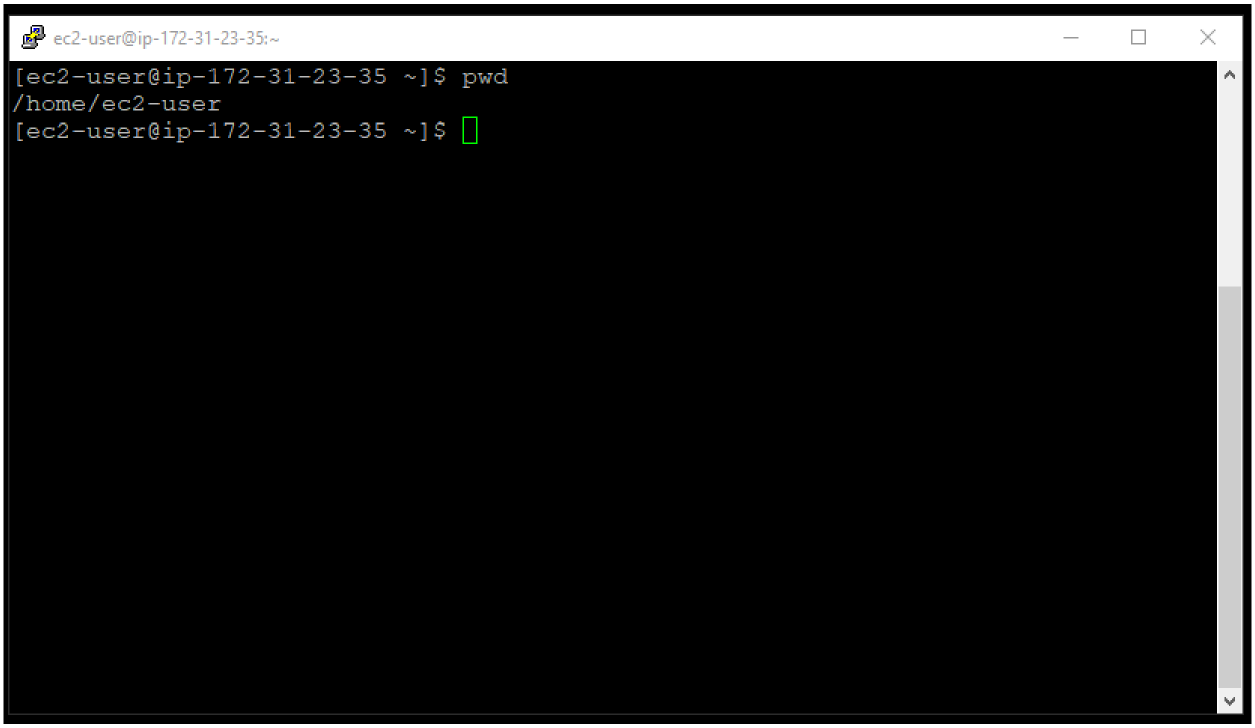


```
(kafka.coordinator.transaction.TransactionMarkerChannelManager)
[2022-09-23 11:54:16,154] INFO [ExpirationReaper-0-AlterAcls]: Starting (kafka.s
erver.DelayedOperationPurgatory$ExpiredOperationReaper)
[2022-09-23 11:54:16,185] INFO [/config/changes-event-process-thread]: Starting
(kafka.common.ZkNodeChangeNotificationListener$ChangeEventProcessThread)
[2022-09-23 11:54:16,197] INFO [SocketServer listenerType=ZK_BROKER, nodeId=0] S
tarting socket server acceptors and processors (kafka.network.SocketServer)
[2022-09-23 11:54:16,203] INFO [SocketServer listenerType=ZK_BROKER, nodeId=0] S
tarted data-plane acceptor and processor(s) for endpoint : ListenerName(PLAINTEX
T) (kafka.network.SocketServer)
[2022-09-23 11:54:16,204] INFO [SocketServer listenerType=ZK_BROKER, nodeId=0] S
tarted socket server acceptors and processors (kafka.network.SocketServer)
[2022-09-23 11:54:16,207] INFO Kafka version: 2.8.2 (org.apache.kafka.common.uti
ls.AppInfoParser)
[2022-09-23 11:54:16,207] INFO Kafka commitId: 3146c6ff4a24cc24 (org.apache.kafk
a.common.utils.AppInfoParser)
[2022-09-23 11:54:16,207] INFO Kafka startTimeMs: 1663934056204 (org.apache.kafk
a.common.utils.AppInfoParser)
[2022-09-23 11:54:16,209] INFO [KafkaServer id=0] started (kafka.server.KafkaSer
ver)
[2022-09-23 11:54:16,262] INFO [broker-0-to-controller-send-thread]: Recorded ne
w controller, from now on will use broker 52.21.15.133:9092 (id: 0 rack: null) (
kafka.server.BrokerToControllerRequestThread)
```

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

INSTALLING JUPYTER

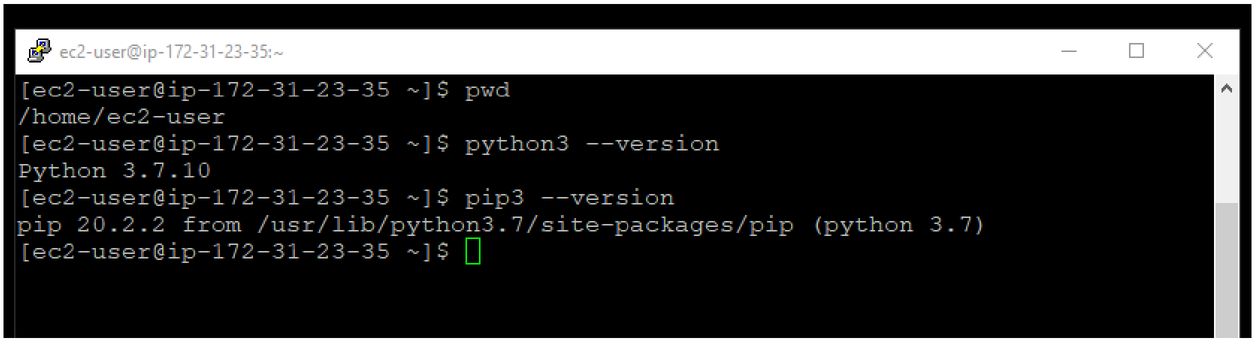
1. Open a new Terminal window and log in to the EC2 instance.
   Make sure that the current directory is **/home/ec2-user** by running the **pwd** command.
   Once you are inside the **/home/ec2-user** directory, follow the steps listed below.



```
[ec2-user@ip-172-31-23-35 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-23-35 ~]$ 
```

2. Once you have logged into the EC2 instance, the first step is to check for the prerequisites. You will require the pip3 package to install the Boto3 package. You can run the following commands:

python3 --version
pip3 --version



```
[ec2-user@ip-172-31-23-35 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-23-35 ~]$ python3 --version
Python 3.7.10
[ec2-user@ip-172-31-23-35 ~]$ pip3 --version
pip 20.2.2 from /usr/lib/python3.7/site-packages/pip (python 3.7)
[ec2-user@ip-172-31-23-35 ~]$ 
```

If the packages are installed, you can skip the **3rd and 4th** steps and jump to the 5th step.

3. As you can see in the image above, you must first install the Python3 and pip3 package. To install Python, you must run the following commands:

   sudo yum -y update

   This command will help you to update the Linux repository

   Then to install python use

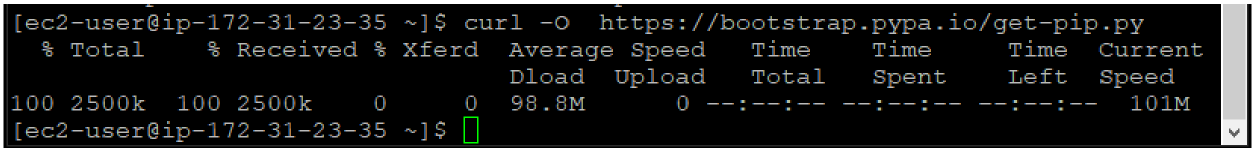
   sudo yum install python36

   This command will install the Python 3.6 on your instance
   Note: In case you receive an error, you can run the following command:

   sudo yum -y install python37

4. Once you have installed Python, the next step is to install the pip package corresponding to Python3

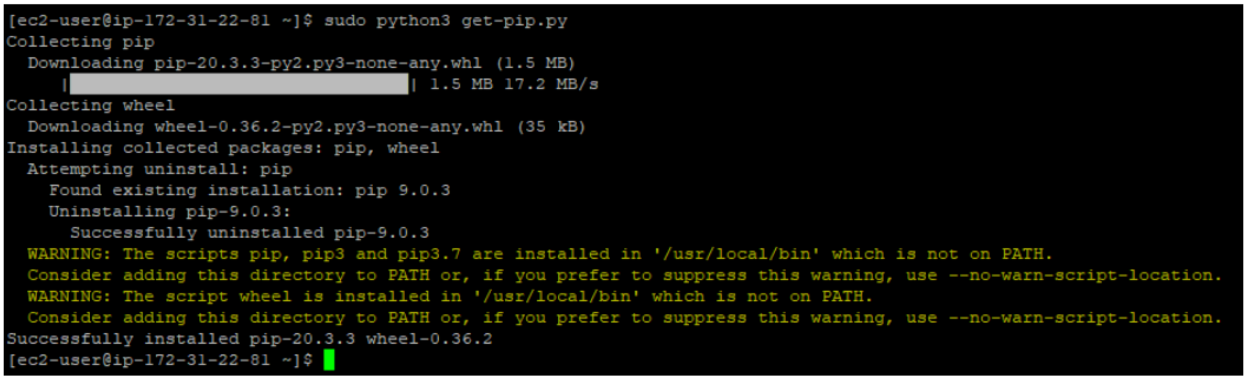
   curl -O https://bootstrap.pypa.io/get-pip.py



```
[ec2-user@ip-172-31-23-35 ~]$ curl -O  https://bootstrap.pypa.io/get-pip.py
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 2500k  100 2500k    0     0  98.8M      0 --:--:-- --:--:-- --:--:--  101M
[ec2-user@ip-172-31-23-35 ~]$
```

   This command will install the pip corresponding to Python3 on your instance.

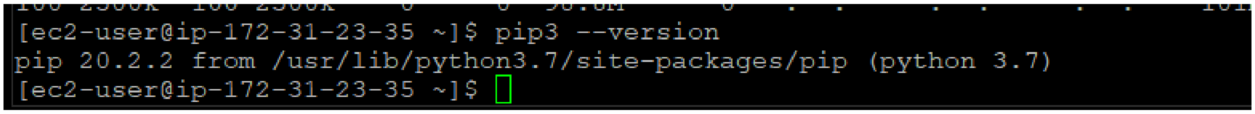
   sudo python3 get-pip.py



```
[ec2-user@ip-172-31-22-81 ~]$ sudo python3 get-pip.py
Collecting pip
  Downloading pip-20.3.3-py2.py3-none-any.whl (1.5 MB)
     |████████████████████████████████| 1.5 MB 17.2 MB/s
Collecting wheel
  Downloading wheel-0.36.2-py2.py3-none-any.whl (35 kB)
Installing collected packages: pip, wheel
  Attempting uninstall: pip
    Found existing installation: pip 9.0.3
    Uninstalling pip-9.0.3:
      Successfully uninstalled pip-9.0.3
  WARNING: The scripts pip, pip3 and pip3.7 are installed in '/usr/local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
  WARNING: The script wheel is installed in '/usr/local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed pip-20.3.3 wheel-0.36.2
[ec2-user@ip-172-31-22-81 ~]$
```

5. Now, you have the required packages installed. However, there is one check required at this step. You must be able to call the functions associated with pip directly. You must not specify the complete path of the package to run the commands.
   To check this, run the following command:

   pip3 --version



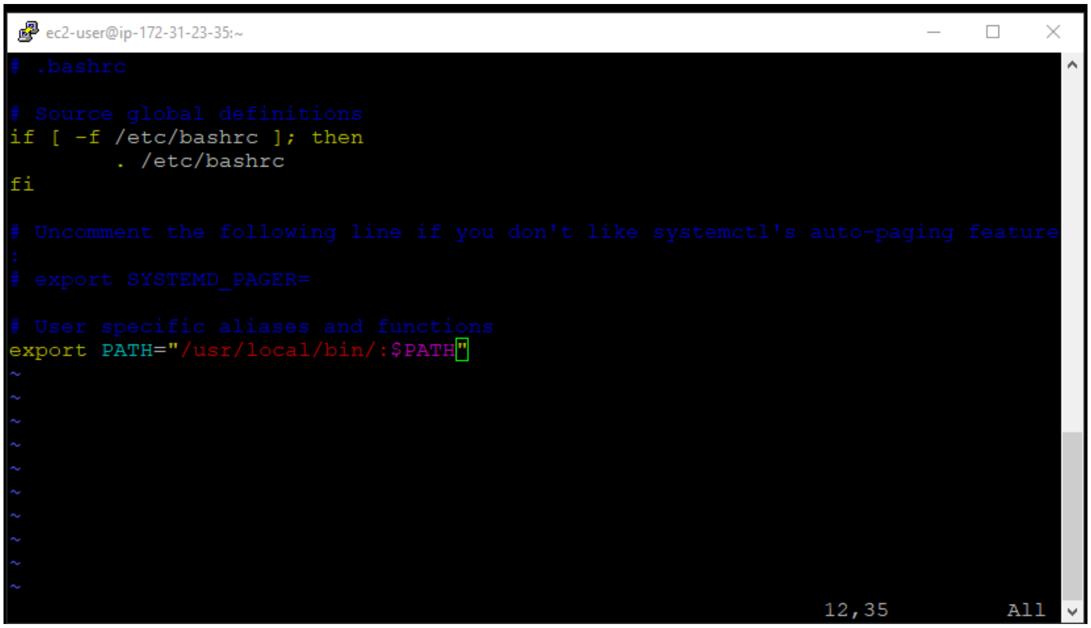
```
[ec2-user@ip-172-31-23-35 ~]$ pip3 --version
pip 20.2.2 from /usr/lib/python3.7/site-packages/pip (python 3.7)
[ec2-user@ip-172-31-23-35 ~]$ 
```

You must follow the steps discussed ahead to make sure that you don't receive an error

- Run this command to open the bash file:

  vi .bashrc

  To edit the file, press the 'i' key.
- You are expected to add the following command at the end of the file.
  export PATH="/usr/local/bin/:$PATH"

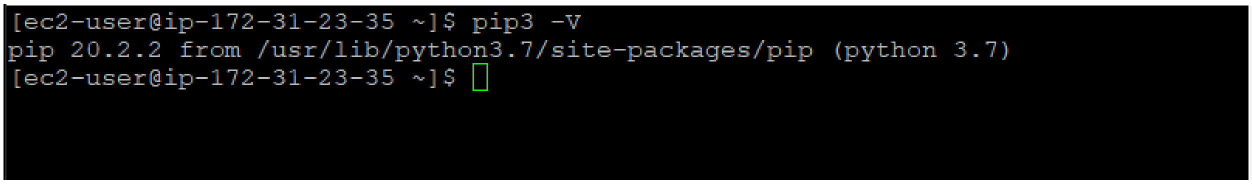


```
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-paging feature
:
# export SYSTEMD_PAGER=

# User specific aliases and functions
export PATH="/usr/local/bin/:$PATH"
~
~
~
~
~
~
~
~
~
~
                                                              12,35         All
```

- Do not make any other changes in the file. Once you have added this, press Esc and type the command :**wq!** and press enter to exit the bash file.

- You must finally execute the file to load the changes using the following command:
  source .bashrc

- Now verify using the command
  pip3 -V



```
[ec2-user@ip-172-31-23-35 ~]$ pip3 -V
pip 20.2.2 from /usr/lib/python3.7/site-packages/pip (python 3.7)
[ec2-user@ip-172-31-23-35 ~]$ 
```

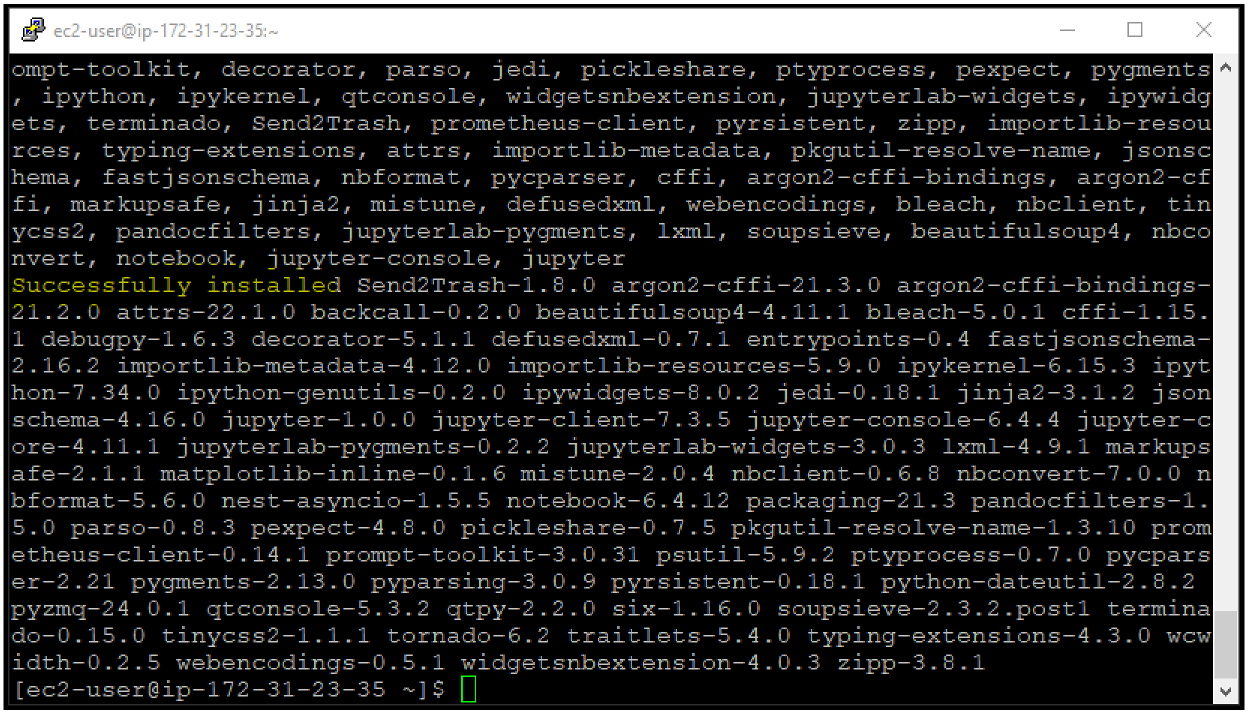
You are now ready with all the prerequisites to install the Jupyter library in the EC2 instance.

INSTALLING JUPYTER ON EC2 INSTANCE

1. You can run the following command on the EC2 terminal to install the jupyter server:

pip3 install jupyter --user

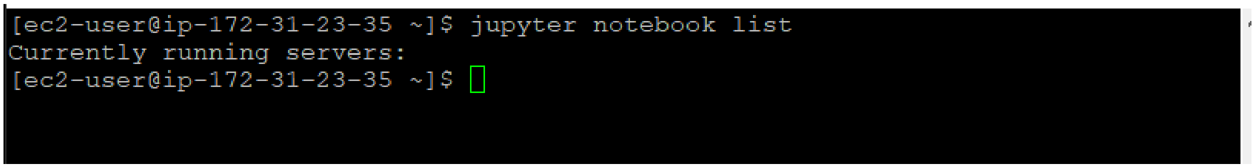
You'll get a "Successfully installed" message in your terminal.



```
ompt-toolkit, decorator, parso, jedi, pickleshare, ptyprocess, pexpect, pygments
, ipython, ipykernel, qtconsole, widgetsnbextension, jupyterlab-widgets, ipywidg
ets, terminado, Send2Trash, prometheus-client, pyrsistent, zipp, importlib-resou
rces, typing-extensions, attrs, importlib-metadata, pkgutil-resolve-name, jsonsc
hema, fastjsonschema, nbformat, pycparser, cffi, argon2-cffi-bindings, argon2-cf
fi, markupsafe, jinja2, mistune, defusedxml, webencodings, bleach, nbclient, tin
ycss2, pandocfilters, jupyterlab-pygments, lxml, soupsieve, beautifulsoup4, nbco
nvert, notebook, jupyter-console, jupyter
Successfully installed Send2Trash-1.8.0 argon2-cffi-21.3.0 argon2-cffi-bindings-
21.2.0 attrs-22.1.0 backcall-0.2.0 beautifulsoup4-4.11.1 bleach-5.0.1 cffi-1.15.
1 debugpy-1.6.3 decorator-5.1.1 defusedxml-0.7.1 entrypoints-0.4 fastjsonschema-
2.16.2 importlib-metadata-4.12.0 importlib-resources-5.9.0 ipykernel-6.15.3 ipyt
hon-7.34.0 ipython-genutils-0.2.0 ipywidgets-8.0.2 jedi-0.18.1 jinja2-3.1.2 json
schema-4.16.0 jupyter-1.0.0 jupyter-client-7.3.5 jupyter-console-6.4.4 jupyter-c
ore-4.11.1 jupyterlab-pygments-0.2.2 jupyterlab-widgets-3.0.3 lxml-4.9.1 markups
afe-2.1.1 matplotlib-inline-0.1.6 mistune-2.0.4 nbclient-0.6.8 nbconvert-7.0.0 n
bformat-5.6.0 nest-asyncio-1.5.5 notebook-6.4.12 packaging-21.3 pandocfilters-1.
5.0 parso-0.8.3 pexpect-4.8.0 pickleshare-0.7.5 pkgutil-resolve-name-1.3.10 prom
etheus-client-0.14.1 prompt-toolkit-3.0.31 psutil-5.9.2 ptyprocess-0.7.0 pycpars
er-2.21 pygments-2.13.0 pyparsing-3.0.9 pyrsistent-0.18.1 python-dateutil-2.8.2
pyzmq-24.0.1 qtconsole-5.3.2 qtpy-2.2.0 six-1.16.0 soupsieve-2.3.2.post1 termina
do-0.15.0 tinycss2-1.1.1 tornado-6.2 traitlets-5.4.0 typing-extensions-4.3.0 wcw
idth-0.2.5 webencodings-0.5.1 widgetsnbextension-4.0.3 zipp-3.8.1
[ec2-user@ip-172-31-23-35 ~]$
```

2. Verify jupyter server status using below command:

jupyter notebook list

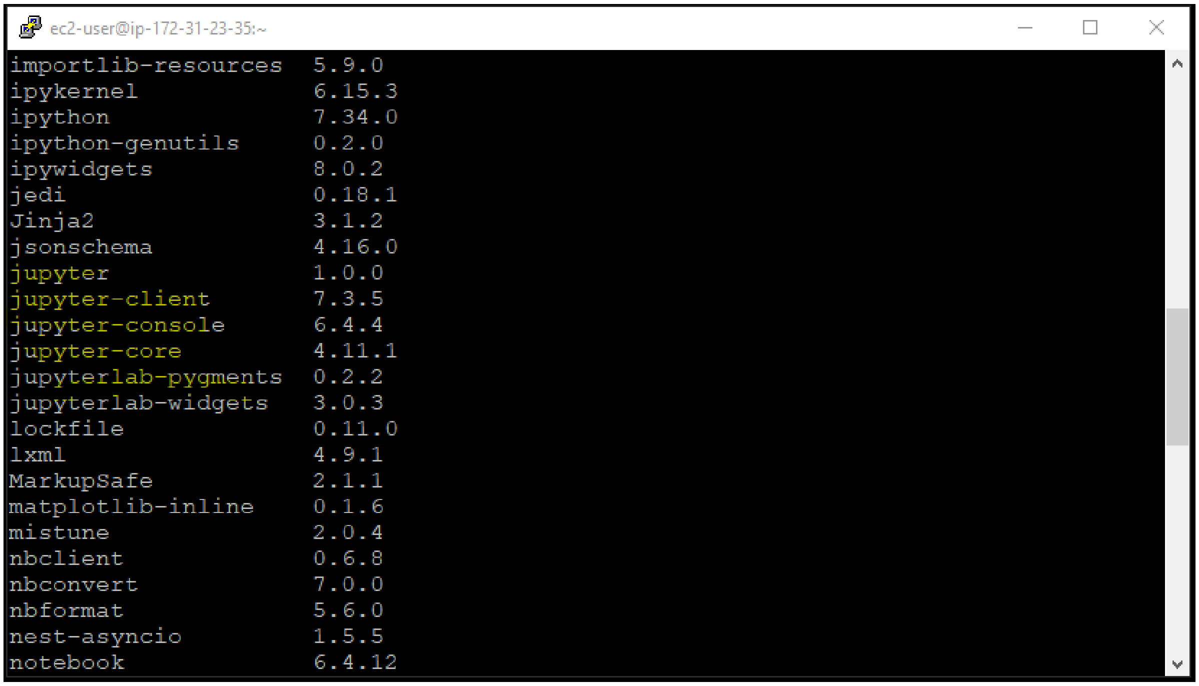


```
[ec2-user@ip-172-31-23-35 ~]$ jupyter notebook list
Currently running servers:
[ec2-user@ip-172-31-23-35 ~]$
```

Currently, there are no running jupyter servers.

Alternatively, you can view the list of installed libraries by using the following command:

pip3 list

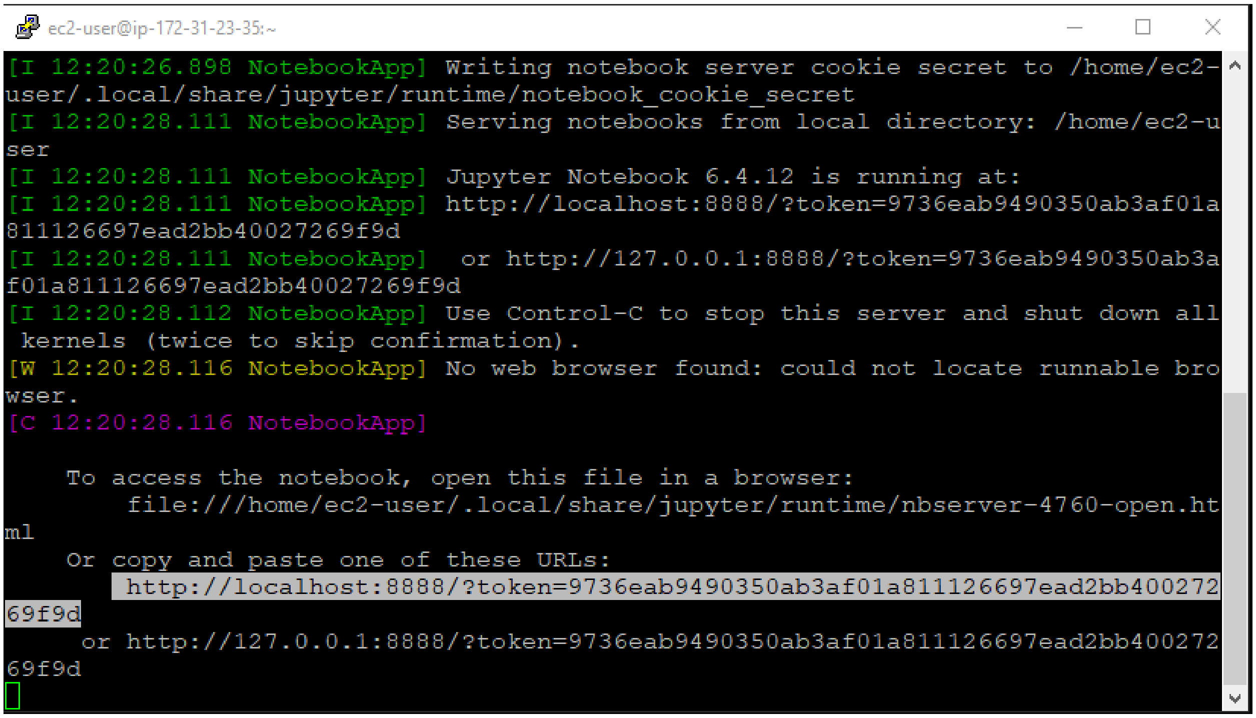


```
importlib-resources  5.9.0
ipykernel            6.15.3
ipython              7.34.0
ipython-genutils     0.2.0
ipywidgets           8.0.2
jedi                 0.18.1
Jinja2               3.1.2
jsonschema           4.16.0
jupyter              1.0.0
jupyter-client       7.3.5
jupyter-console      6.4.4
jupyter-core         4.11.1
jupyterlab-pygments  0.2.2
jupyterlab-widgets   3.0.3
lockfile             0.11.0
lxml                 4.9.1
MarkupSafe           2.1.1
matplotlib-inline    0.1.6
mistune              2.0.4
nbclient             0.6.8
nbconvert            7.0.0
nbformat             5.6.0
nest-asyncio         1.5.5
notebook             6.4.12
```

You can see that the jupyter library has been installed with its necessary components

3. You can start the Jupyter Notebook server using the command provided below:

jupyter notebook



```
[I 12:20:26.898 NotebookApp] Writing notebook server cookie secret to /home/ec2-
user/.local/share/jupyter/runtime/notebook_cookie_secret
[I 12:20:28.111 NotebookApp] Serving notebooks from local directory: /home/ec2-u
ser
[I 12:20:28.111 NotebookApp] Jupyter Notebook 6.4.12 is running at:
[I 12:20:28.111 NotebookApp] http://localhost:8888/?token=9736eab9490350ab3af01a
811126697ead2bb40027269f9d
[I 12:20:28.111 NotebookApp]  or http://127.0.0.1:8888/?token=9736eab9490350ab3a
f01a811126697ead2bb40027269f9d
[I 12:20:28.112 NotebookApp] Use Control-C to stop this server and shut down all
 kernels (twice to skip confirmation).
[W 12:20:28.116 NotebookApp] No web browser found: could not locate runnable bro
wser.
[C 12:20:28.116 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///home/ec2-user/.local/share/jupyter/runtime/nbserver-4760-open.ht
ml
    Or copy and paste one of these URLs:
        http://localhost:8888/?token=9736eab9490350ab3af01a811126697ead2bb400272
69f9d
     or http://127.0.0.1:8888/?token=9736eab9490350ab3af01a811126697ead2bb400272
69f9d
```

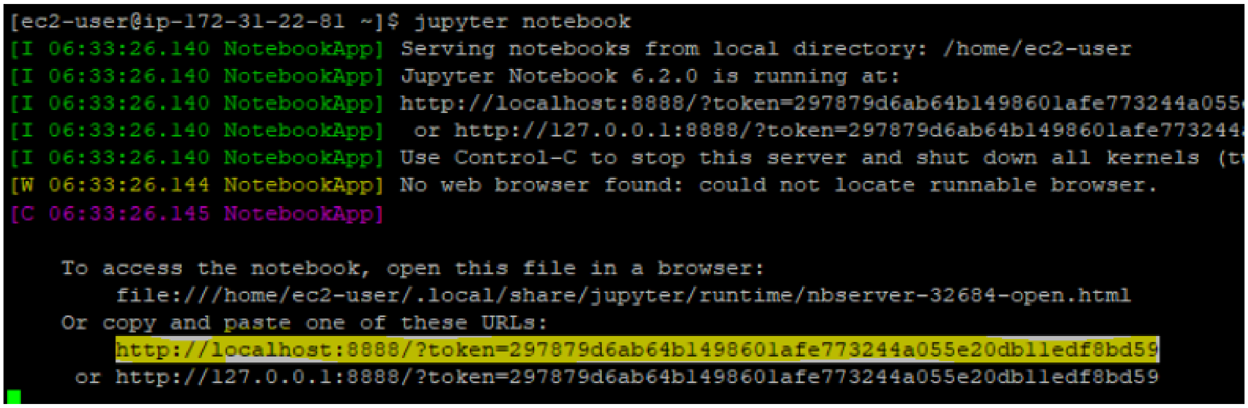
NOTE: Now, the Jupyter Notebook is running on the EC2 instance. However, you need to access it through a browser using SSH tunneling. This step differs for Windows and Linux/Mac users.

CONNECTING TO A JUPYTER NOTEBOOK - LINUX/MAC

1. After you have launched the Jupyter notebook server, you must open a new terminal window and run the following commands:

   ssh -i “keypair1.pem” -N -f -L 8888:localhost:8888 ec2-user@IPv4_address_of_EC2 instance

2. Next, you can select the url from the previous terminal window and paste to the local browser. You should not use Cont + C to copy as it raises the prompt to end the session. The text is automatically copied by selecting it in the window.



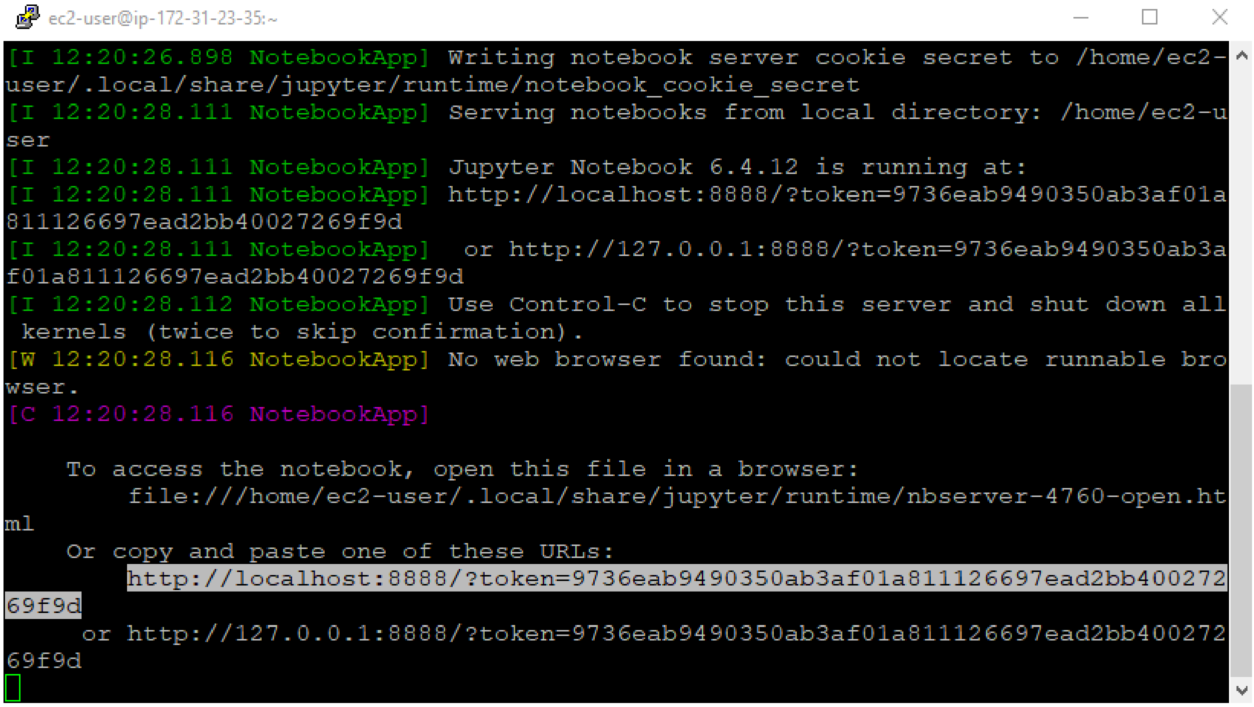
```
[ec2-user@ip-172-31-22-81 ~]$ jupyter notebook
[I 06:33:26.140 NotebookApp] Serving notebooks from local directory: /home/ec2-user
[I 06:33:26.140 NotebookApp] Jupyter Notebook 6.2.0 is running at:
[I 06:33:26.140 NotebookApp] http://localhost:8888/?token=297879d6ab64b149860lafe773244a055
[I 06:33:26.140 NotebookApp]  or http://127.0.0.1:8888/?token=297879d6ab64b149860lafe773244
[I 06:33:26.140 NotebookApp] Use Control-C to stop this server and shut down all kernels (t
[W 06:33:26.144 NotebookApp] No web browser found: could not locate runnable browser.
[C 06:33:26.145 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///home/ec2-user/.local/share/jupyter/runtime/nbserver-32684-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/?token=297879d6ab64b149860lafe773244a055e20db1ledf8bd59
     or http://127.0.0.1:8888/?token=297879d6ab64b149860lafe773244a055e20db1ledf8bd59
```

The Jupyter notebook will be launched in the browser

CONNECTING TO A JUPYTER NOTEBOOK - WINDOWS

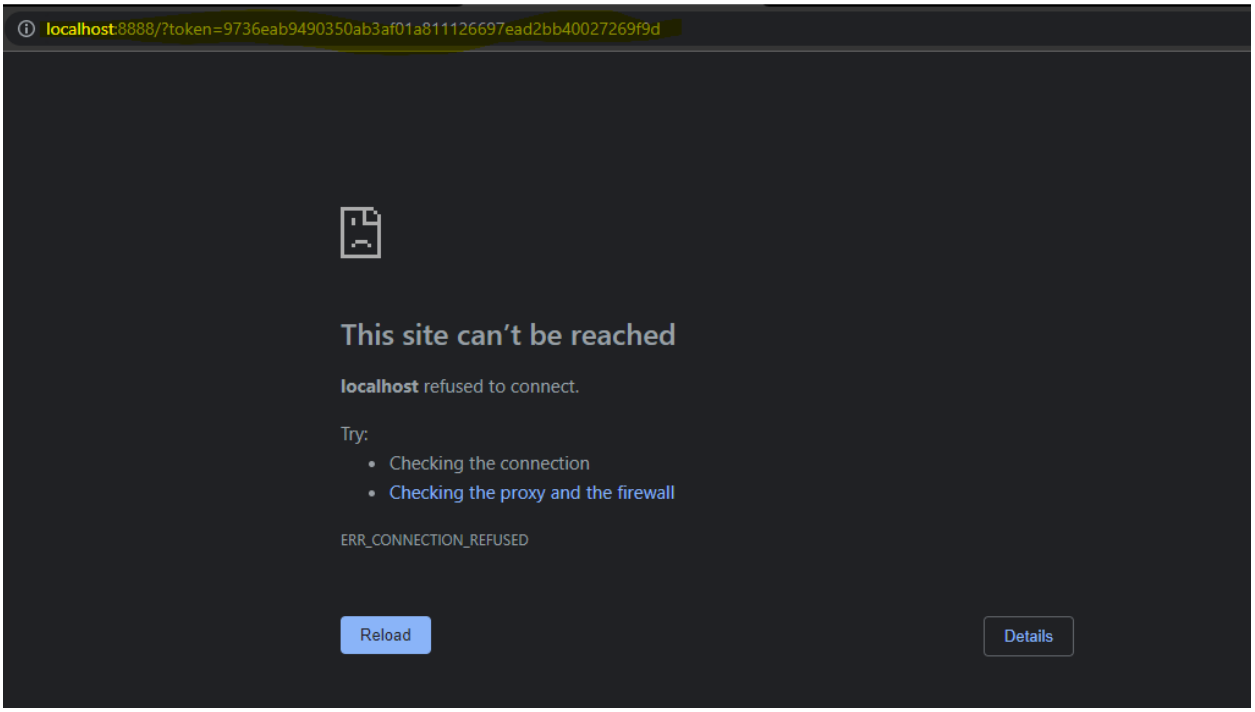
This is a step by step approach to connect to a Jupyter Notebook once it has been installed in the EC2 instance. For this task, you will require the PuTTY software when working with the Windows machine.

1. You can select the url from the PuTTY window and paste in the local browser. You should not use Cont + C to copy as it raises the prompt to end the session. The text is automatically copied by selecting it in the window.



```
[I 12:20:26.898 NotebookApp] Writing notebook server cookie secret to /home/ec2-
user/.local/share/jupyter/runtime/notebook_cookie_secret
[I 12:20:28.111 NotebookApp] Serving notebooks from local directory: /home/ec2-u
ser
[I 12:20:28.111 NotebookApp] Jupyter Notebook 6.4.12 is running at:
[I 12:20:28.111 NotebookApp] http://localhost:8888/?token=9736eab9490350ab3af01a
811126697ead2bb40027269f9d
[I 12:20:28.111 NotebookApp]  or http://127.0.0.1:8888/?token=9736eab9490350ab3a
f01a811126697ead2bb40027269f9d
[I 12:20:28.112 NotebookApp] Use Control-C to stop this server and shut down all
 kernels (twice to skip confirmation).
[W 12:20:28.116 NotebookApp] No web browser found: could not locate runnable bro
wser.
[C 12:20:28.116 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///home/ec2-user/.local/share/jupyter/runtime/nbserver-4760-open.ht
ml
    Or copy and paste one of these URLs:
        http://localhost:8888/?token=9736eab9490350ab3af01a811126697ead2bb400272
69f9d
     or http://127.0.0.1:8888/?token=9736eab9490350ab3af01a811126697ead2bb400272
69f9d
```

You will find that the Jupyter Notebook will not be launched.



localhost:8888/?token=9736eab9490350ab3af01a811126697ead2bb40027269f9d

This site can’t be reached

localhost refused to connect.

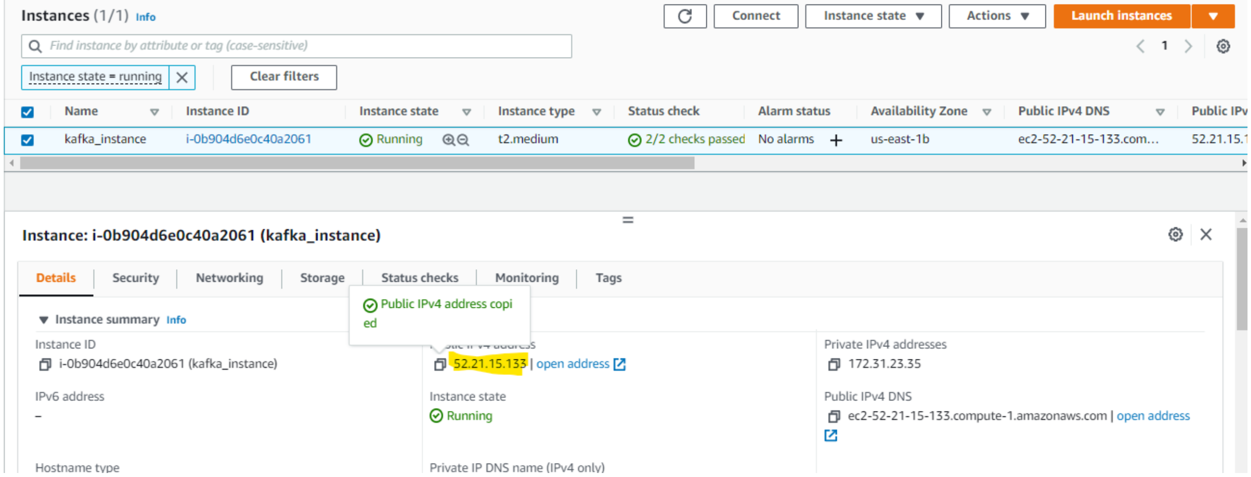
Try:

- Checking the connection
- Checking the proxy and the firewall

ERR_CONNECTION_REFUSED

The conclusion is that we can not access this url without the SSH tunneling. For this, we will again use the services offered by PuTTY.

2. You must first copy the public IP address of the EC2 instance from the AWS Console



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv
☑	kafka_instance	i-0b904d6e0c40a2061	✓ Running	t2.medium	✓ 2/2 checks passed	No alarms +	us-east-1b	ec2-52-21-15-133.com...	52.21.15.1

Instance: i-0b904d6e0c40a2061 (kafka_instance)

✓ Public IPv4 address copied

▼ 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
–

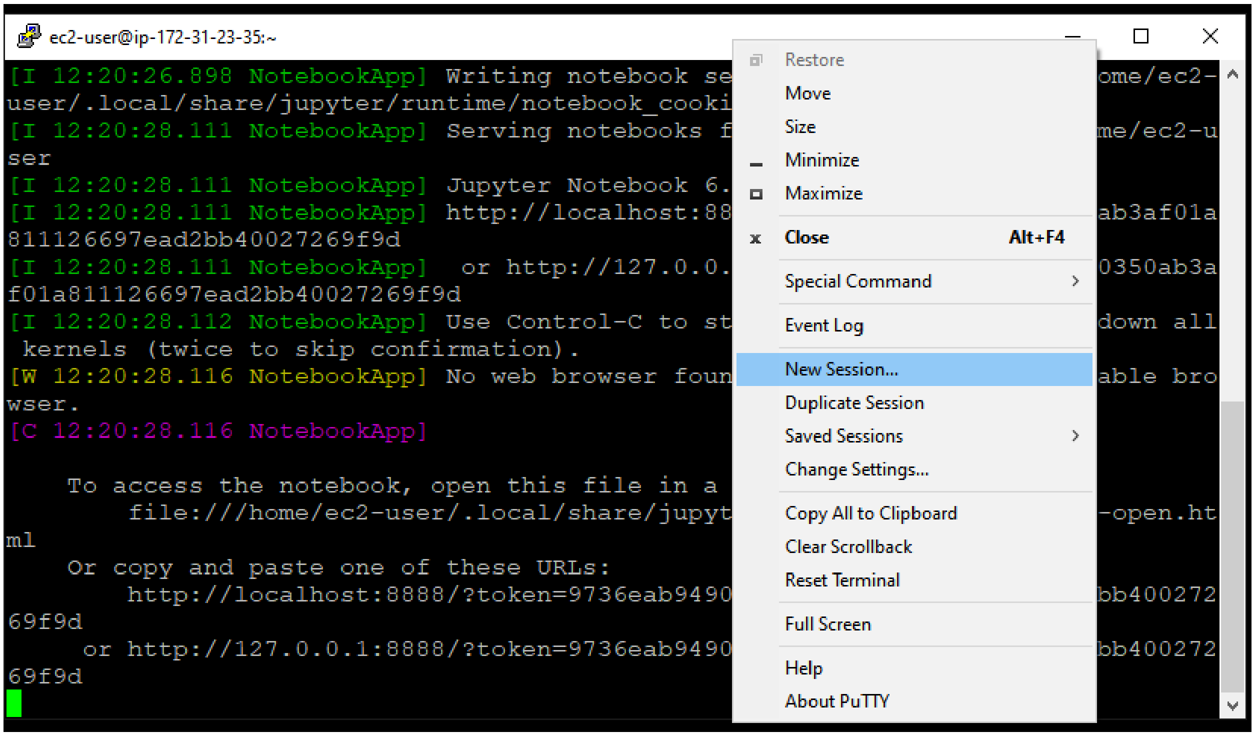
Instance state
✓ Running

Public IPv4 DNS
ec2-52-21-15-133.compute-1.amazonaws.com | open address

Hostname type

Private IP DNS name (IPv4 only)

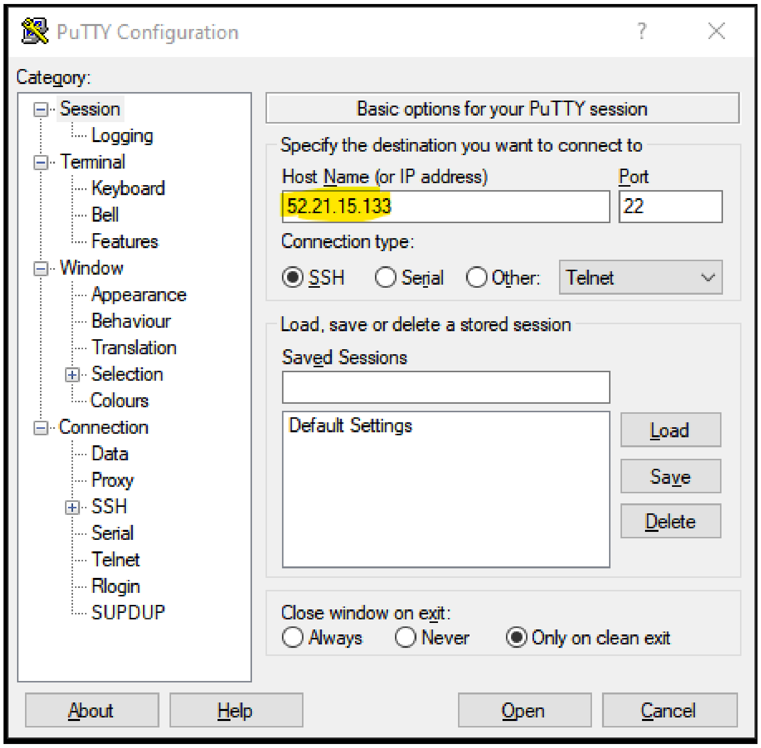
3. Next, open a new PuTTY session by right clicking on the existing PuTTY window



ec2-user@ip-172-31-23-35:~

```
[I 12:20:26.898 NotebookApp] Writing notebook se                        ome/ec2-
user/.local/share/jupyter/runtime/notebook_cooki
[I 12:20:28.111 NotebookApp] Serving notebooks f                        me/ec2-u
ser
[I 12:20:28.111 NotebookApp] Jupyter Notebook 6.
[I 12:20:28.111 NotebookApp] http://localhost:88                        ab3af01a
811126697ead2bb40027269f9d
[I 12:20:28.111 NotebookApp]  or http://127.0.0.                        0350ab3a
f01a811126697ead2bb40027269f9d
[I 12:20:28.112 NotebookApp] Use Control-C to st                        down all
 kernels (twice to skip confirmation).
[W 12:20:28.116 NotebookApp] No web browser foun                        able bro
wser.
[C 12:20:28.116 NotebookApp]

    To access the notebook, open this file in a
        file:///home/ec2-user/.local/share/jupyt                        -open.ht
ml
    Or copy and paste one of these URLs:
        http://localhost:8888/?token=9736eab9490                        bb400272
69f9d
     or http://127.0.0.1:8888/?token=9736eab9490                        bb400272
69f9d
```

4. You must paste the copied Public IP address in the PuTTY window under the Host Name



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 52.21.15.133

Port: 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

Load, save or delete a stored session

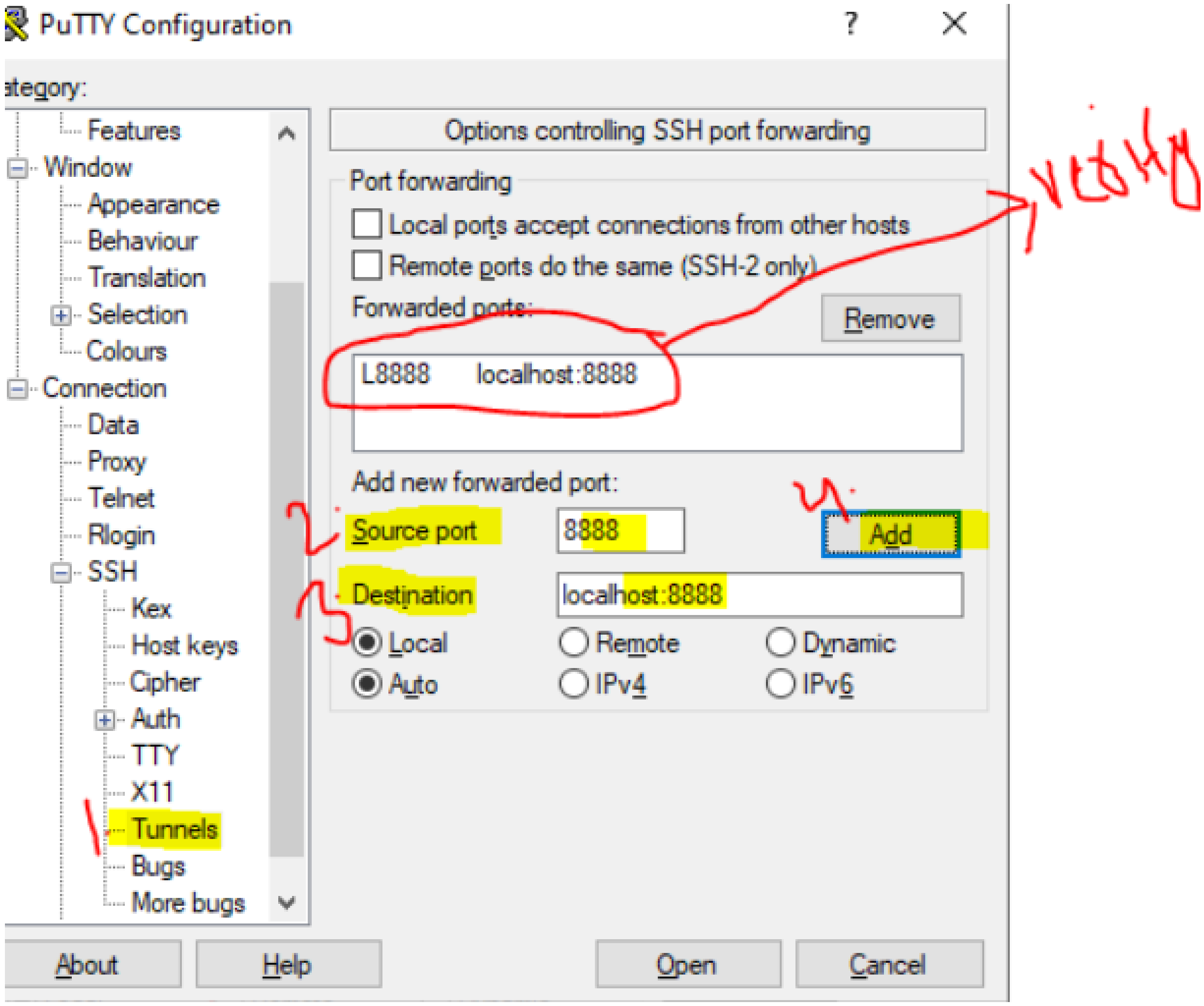
Saved Sessions: Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

5. Next, you must establish a connection between the 8888 ports of both the machines This port will be used in launching the Jupyter Notebook hosted on the EC2 instance on your local machine. For this, go to the 'Tunnels' section under 'SSH' to add the connection. Add the following elements as shown in the image:

SSH > Tunnels
Source port: 8888
Destination: localhost:8888

Once you have added the details, click on the 'Add' button.



PuTTY Configuration

Options controlling SSH port forwarding

Port forwarding

☐ Local ports accept connections from other hosts
☐ Remote ports do the same (SSH-2 only)

Forwarded ports:

L8888 localhost:8888

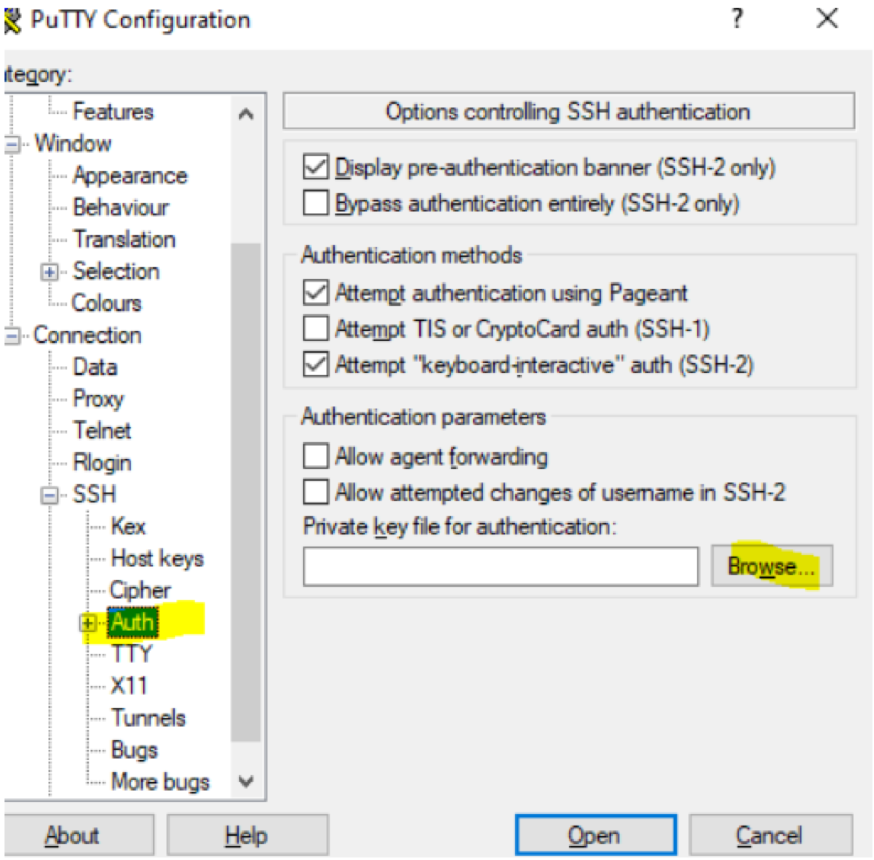
Add new forwarded port:

Source port: 8888
Destination: localhost:8888

● Local ○ Remote ○ Dynamic
● Auto ○ IPv4 ○ IPv6

6. The next step is to provide the Key pair file that you must have saved when the instance was created. You must provide the key under the Auth tab of the SSH Section.

SSH > Auth > Browse > ppk file path



Options controlling SSH authentication

☑ Display pre-authentication banner (SSH-2 only)
☐ Bypass authentication entirely (SSH-2 only)

Authentication methods

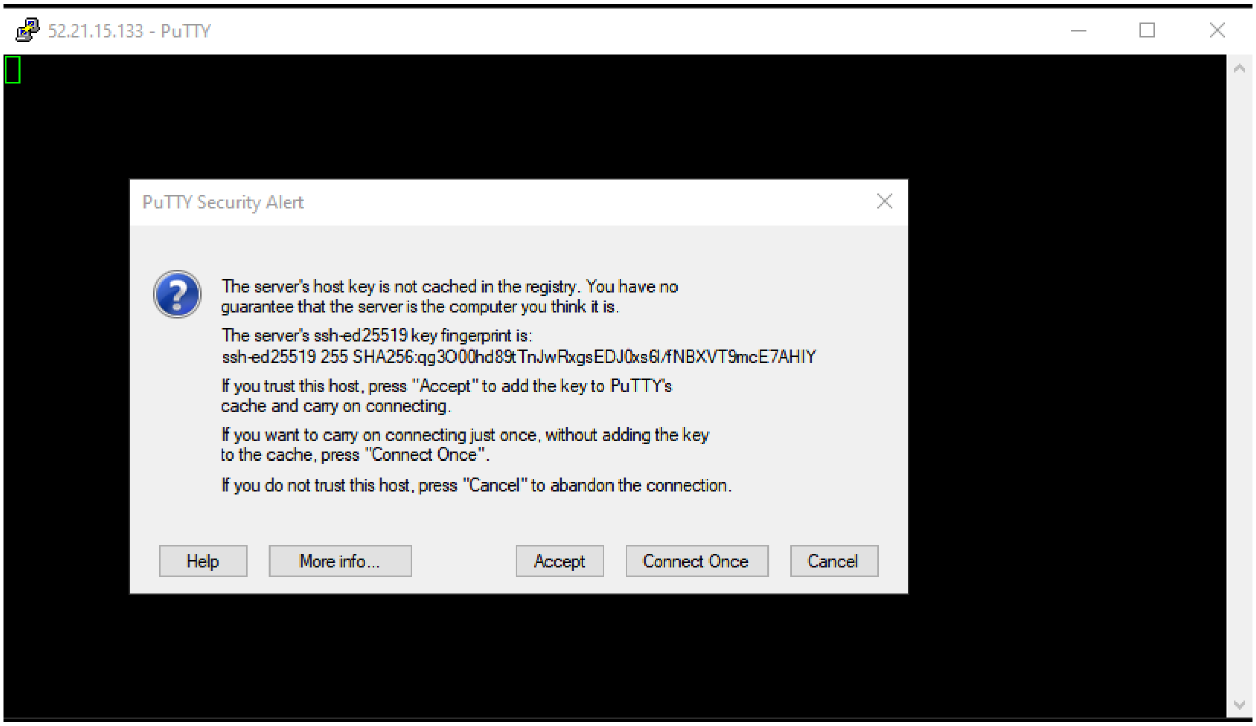
☑ Attempt authentication using Pageant
☐ Attempt TIS or CryptoCard auth (SSH-1)
☑ Attempt "keyboard-interactive" auth (SSH-2)

Authentication parameters

☐ Allow agent forwarding
☐ Allow attempted changes of username in SSH-2

Private key file for authentication:

7. After you have successfully selected the file, click on ‘Open’ to launch the EC2 instance. Click on ‘Accept’ in the dialogue box when prompted.



The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.

The server's ssh-ed25519 key fingerprint is:
ssh-ed25519 255 SHA256:qg3O00hd89tTnJwRxgsEDJ0xs6l/fNBXVT9mcE7AHIY

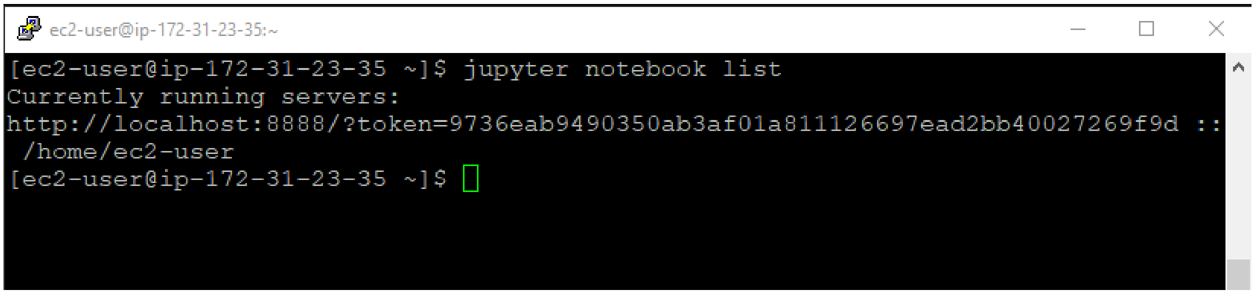
If you trust this host, press "Accept" to add the key to PuTTY's cache and carry on connecting.

If you want to carry on connecting just once, without adding the key to the cache, press "Connect Once".

If you do not trust this host, press "Cancel" to abandon the connection.

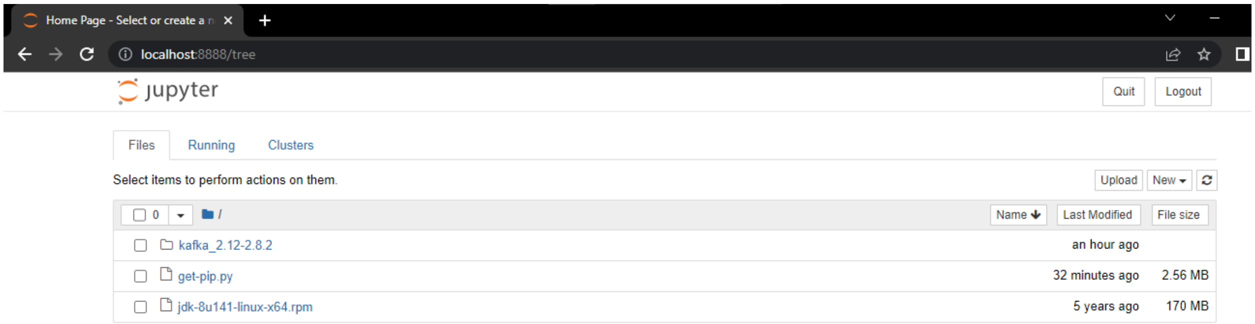
7. In the EC2 window, login with the username as ‘ec2-user’. After logging in, run the following command to check the running servers:

jupyter notebook list



```
[ec2-user@ip-172-31-23-35 ~]$ jupyter notebook list
Currently running servers:
http://localhost:8888/?token=9736eab9490350ab3af01a811126697ead2bb40027269f9d ::
 /home/ec2-user
[ec2-user@ip-172-31-23-35 ~]$ 
```

You can now select the link mentioned in this window to copy, and then paste it in the local browser to access the Jupyter notebook. You can see that the notebook contains the files present in your EC2 instance.

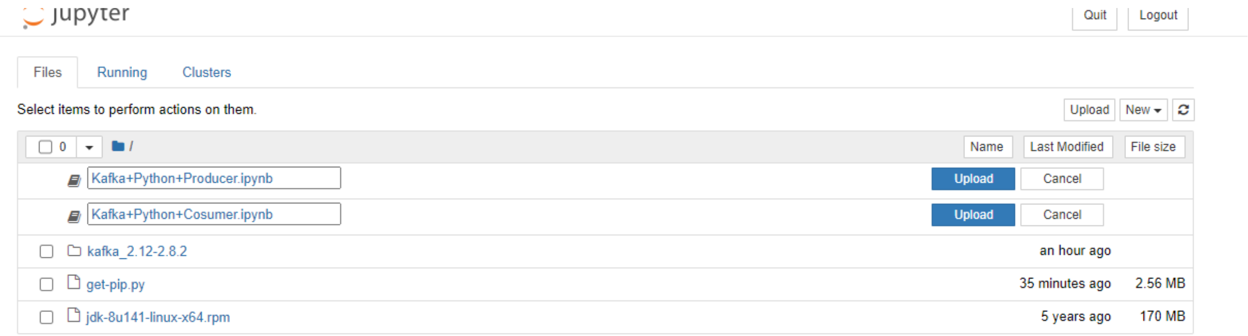


Select items to perform actions on them.

Name	Last Modified	File size
kafka_2.12-2.8.2	an hour ago	
get-pip.py	32 minutes ago	2.56 MB
jdk-8u141-linux-x64.rpm	5 years ago	170 MB

You can now upload or create any python notebooks and run codes on the EC2 instance.

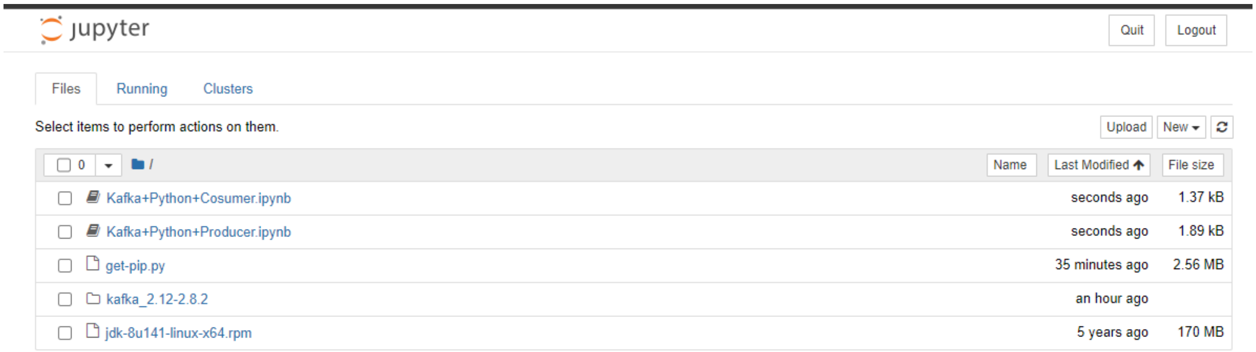
- You can upload the Jupyter notebooks using the **Upload** button from your local machine to EC2 instance. Once you select the notebook, you need to upload and you get the screen as shown below.



Select items to perform actions on them.

Name	Last Modified	File size
Kafka+Python+Producer.ipynb	Upload Cancel	
Kafka+Python+Cosumer.ipynb	Upload Cancel	
kafka_2.12-2.8.2	an hour ago	
get-pip.py	35 minutes ago	2.56 MB
jdk-8u141-linux-x64.rpm	5 years ago	170 MB

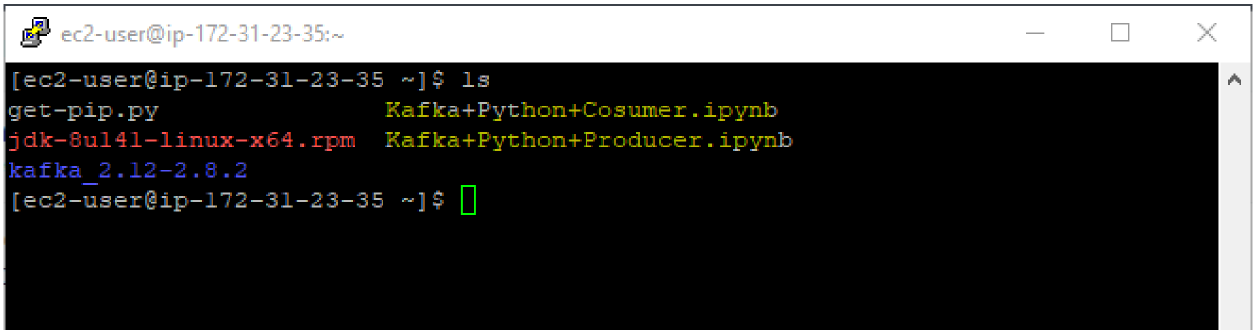
- Click on the **Upload** button next to it and the notebook will get uploaded.



Select items to perform actions on them.

Name	Last Modified	File size
Kafka+Python+Cosumer.ipynb	seconds ago	1.37 kB
Kafka+Python+Producer.ipynb	seconds ago	1.89 kB
get-pip.py	35 minutes ago	2.56 MB
kafka_2.12-2.8.2	an hour ago	
jdk-8u141-linux-x64.rpm	5 years ago	170 MB

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



```
[ec2-user@ip-172-31-23-35 ~]$ ls
get-pip.py                 Kafka+Python+Cosumer.ipynb
jdk-8u141-linux-x64.rpm    Kafka+Python+Producer.ipynb
kafka_2.12-2.8.2
[ec2-user@ip-172-31-23-35 ~]$ 
```