

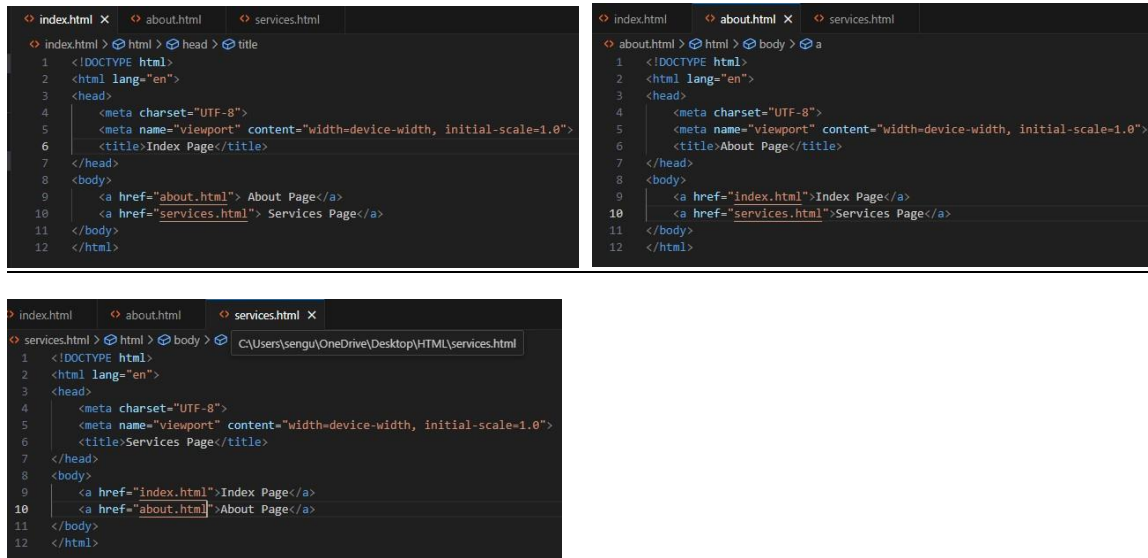
ASSIGNMENT 7 :-

PROBLEM STATEMENT :-

PROBLEM STATEMENT -> Hosting a Website on EC2.

#To host the website ->

STEP 1-> Create 3 Static Webpages using HTML



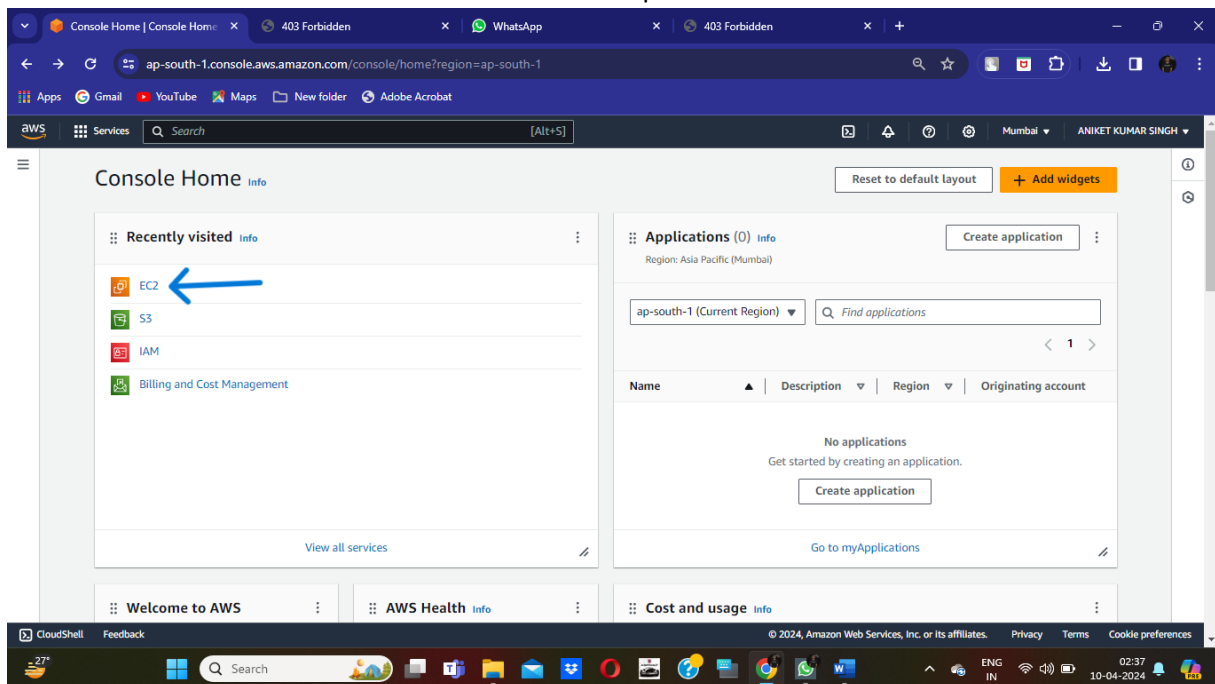
The first two screenshots show the code for index.html and about.html. The third screenshot shows the code for services.html.

```
index.html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Index Page</title>
7 </head>
8 <body>
9   <a href="about.html"> About Page</a>
10  <a href="services.html"> Services Page</a>
11 </body>
12 </html>
```

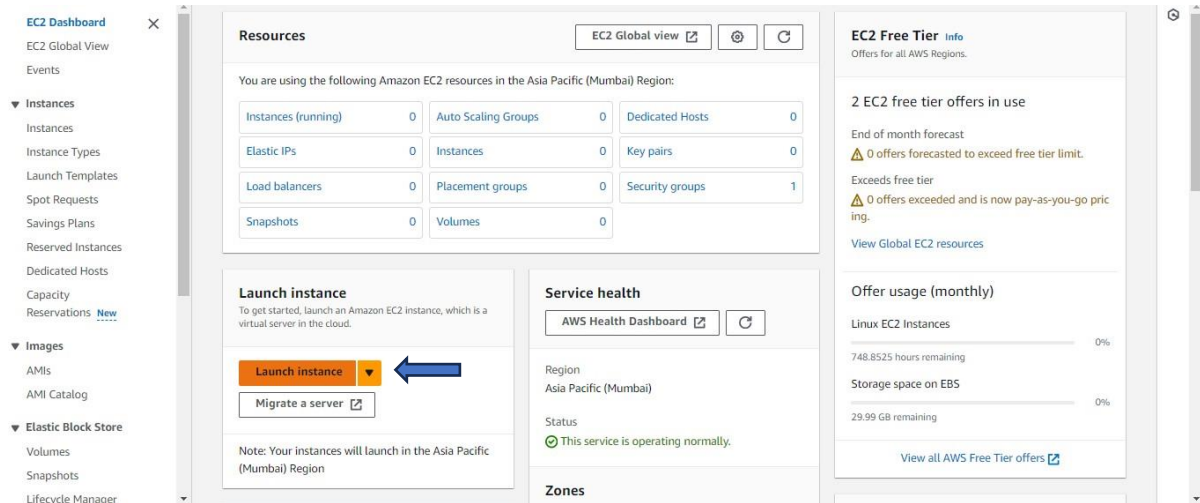
```
about.html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>About Page</title>
7 </head>
8 <body>
9   <a href="index.html">Index Page</a>
10  <a href="services.html">Services Page</a>
11 </body>
12 </html>
```

```
services.html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Services Page</title>
7 </head>
8 <body>
9   <a href="index.html">Index Page</a>
10  <a href="about.html">About Page</a>
11 </body>
12 </html>
```

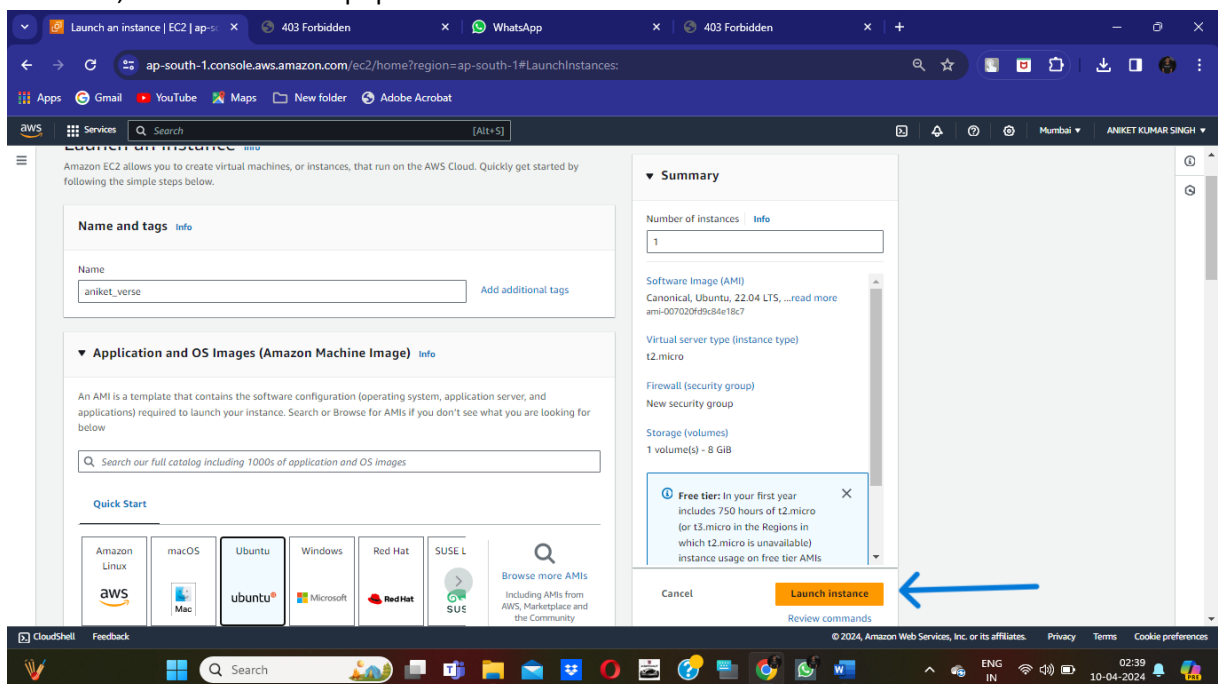
STEP 2-> Search for the “EC2”and Click on the “EC2” option.



STEP 3-> Click on "Launch Instance" button: Initiate the process of launching a new virtual server instance. This step begins the setup for hosting your website on an EC2 instance.



STEP 4-> Give a unique name to the instance & select "Ubuntu": Provide a distinctive name for your instance and choose the Ubuntu operating system. Naming helps identify and manage your instances, while Ubuntu is a popular choice for server environments..



STEP 5-> Create a new key pair by clicking "Create New Key Pair" button: Generate a new SSH key pair for secure access to the instance. SSH keys are used for secure authentication and access control.

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](#)

Please choose a key pair or choose the option to proceed with a key pair

STEP 6-> Give a name to the key pair, then click "Create Key Pair": Name your newly generated key pair and create it. This key pair will be used to securely connect to your EC2 instance.

Create key pair ×

Key pair name

Key pairs allow you to connect to your instance securely.

[Create new key pair](#)

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ RSA
RSA encrypted private and public key pair

☐ ED25519
ED25519 encrypted private and public key pair

Private key file format

☒ .pem
For use with OpenSSH

☐ .ppk
For use with PuTTY

When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

[Cancel](#) [Create key pair](#)

STEP 7-> Select all the Security options, then click on "Launch Instance": Configure security settings such as security groups and key pair. Security settings ensure your instance is protected and accessible only to authorized users.

Subnet [info](#)

No preference (Default: subnet in any availability zone)

Auto-assign public IP [info](#)

Enable

Additional charges apply when outside of free tier allowance

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-2** with the following rules:

☒ Allow SSH traffic from

☒ 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)

Canonical, Ubuntu, 22.04 LTS, ...[read more](#)
ami-007020f09c84e18c7

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

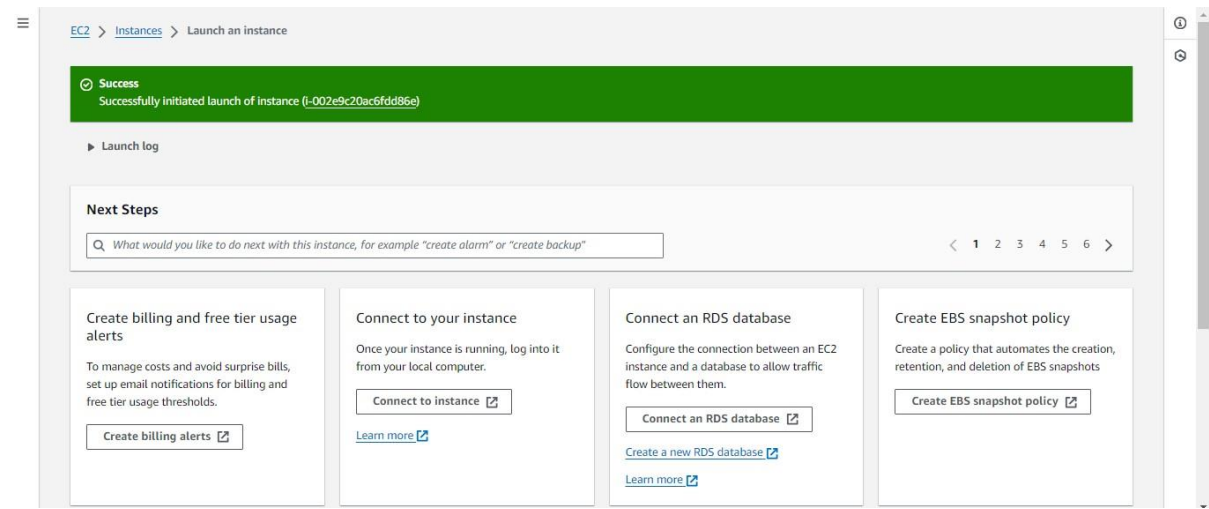
Storage (volumes)

1 volume(s) - 8 GiB

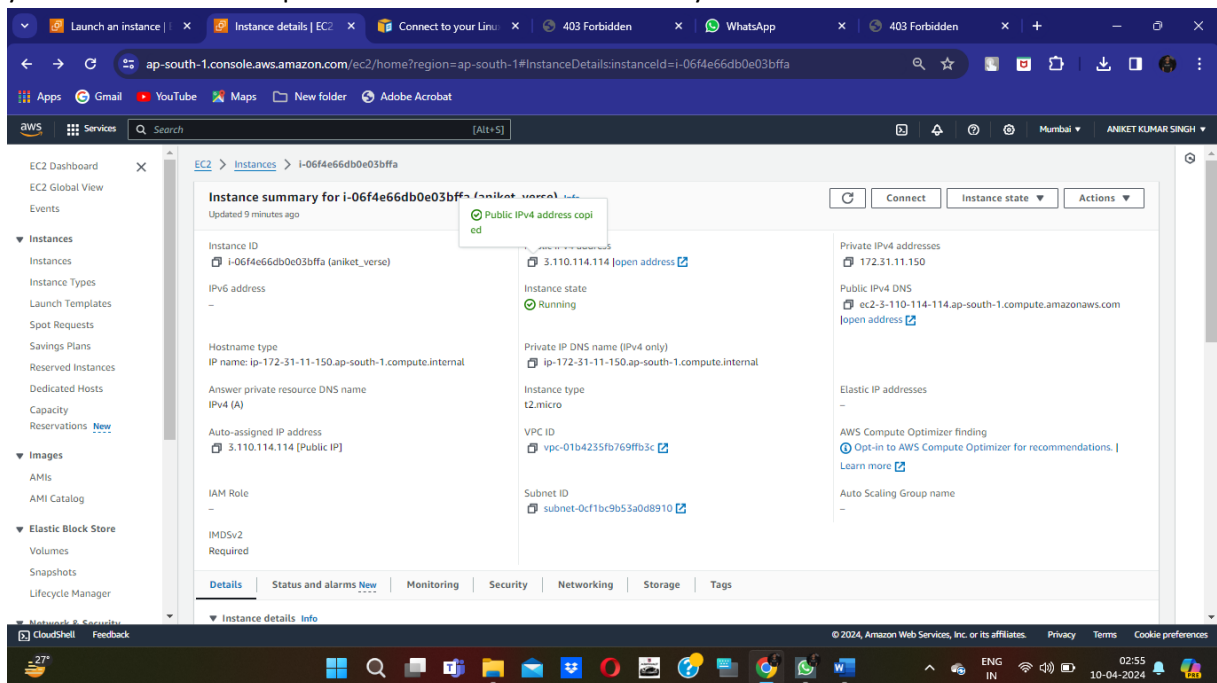
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which ...)

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

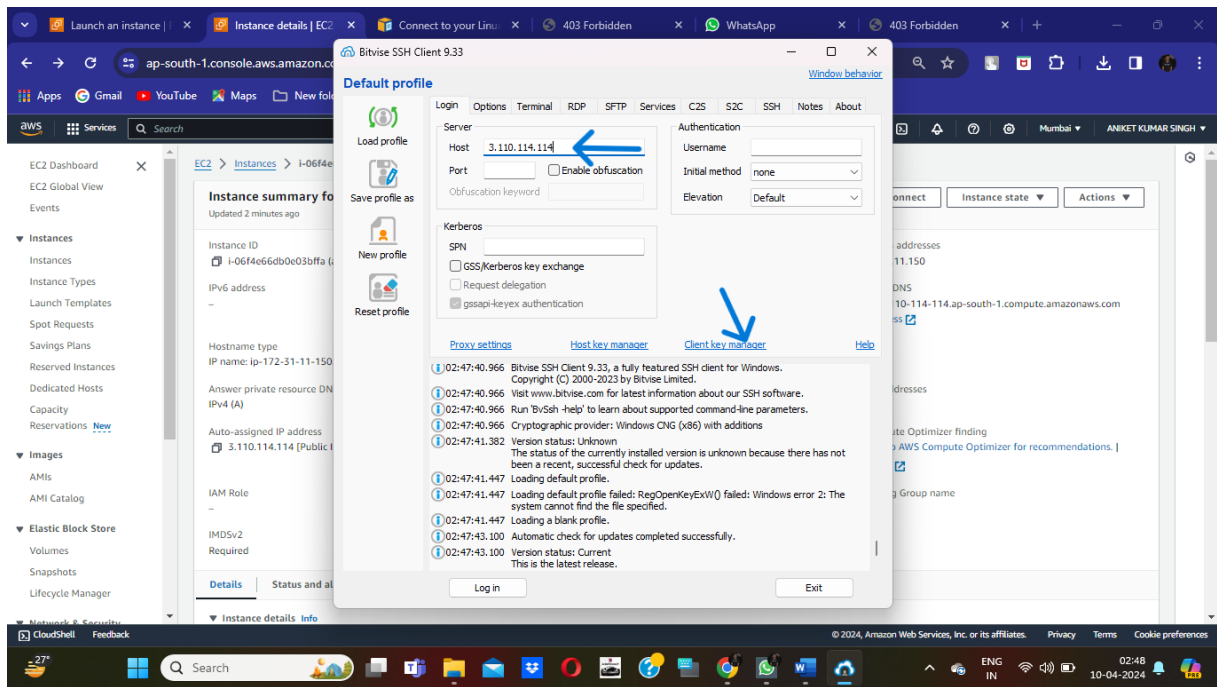
STEP 8-> Click on the instance ID to access it: Navigate to the details page of the newly created instance. This is where you manage and configure your EC2 instance settings.



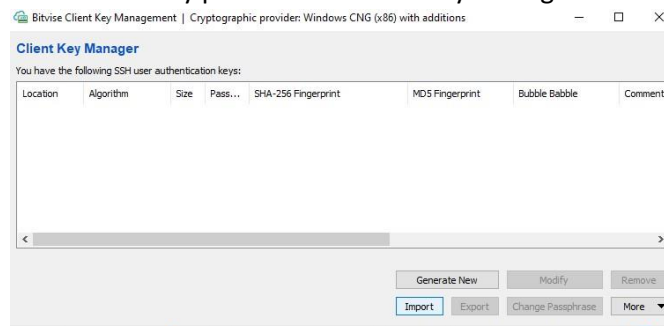
STEP 9-> Copy the "Public IPv4 Address" of the instance: Obtain the public IP address assigned to your EC2 instance. The public IP address is used to access your website hosted on the EC2 instance.



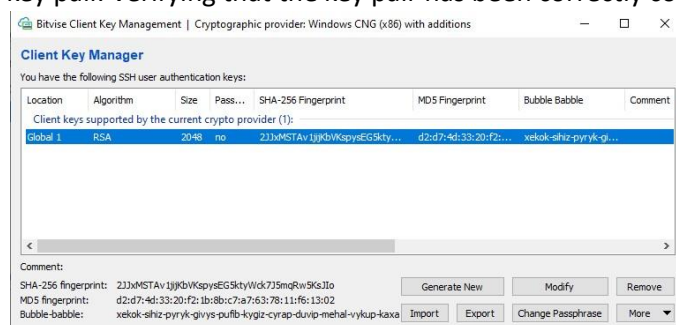
STEP 10-> Paste IP address under Host, then go to "Client Key Manager" option: Enter the copied IP address in the appropriate field and proceed to manage client keys.



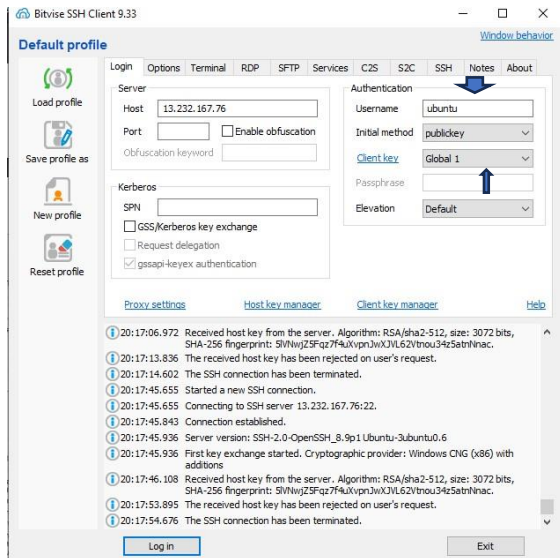
STEP 11-> Click on "Import" button & select the key, then click "Import": Import the previously created SSH key pair into the client key manager.



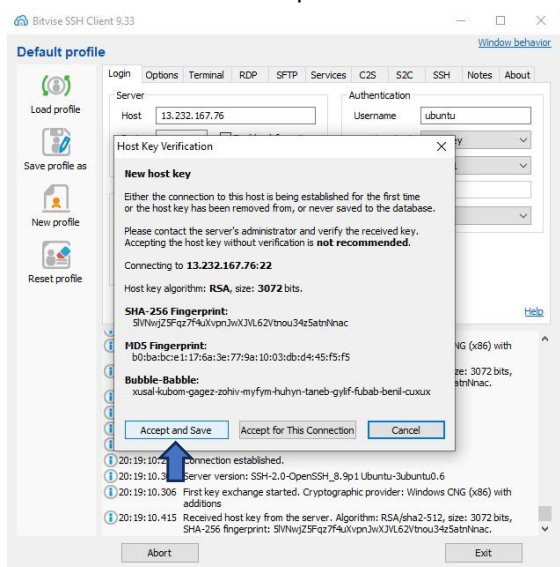
STEP 12-> The new key is successfully added, click "Import": Confirm the successful import of the SSH key pair. Verifying that the key pair has been correctly configured for authentication.



STEP 13-> Give the username as "ubuntu," select "Public Key" & "Global 1", then click the "Log in" button: Provide the username and select authentication method for logging into the EC2 instance. Configuring login credentials and authentication method for SSH access.

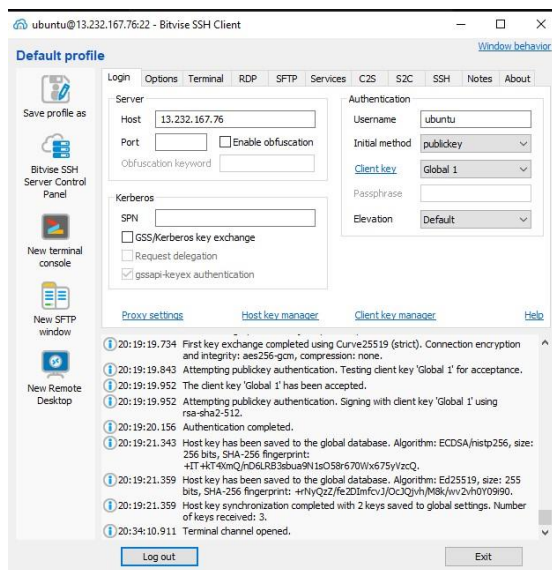


STEP 14-> Click on "Accept & Save" button: Accept and save the connection.

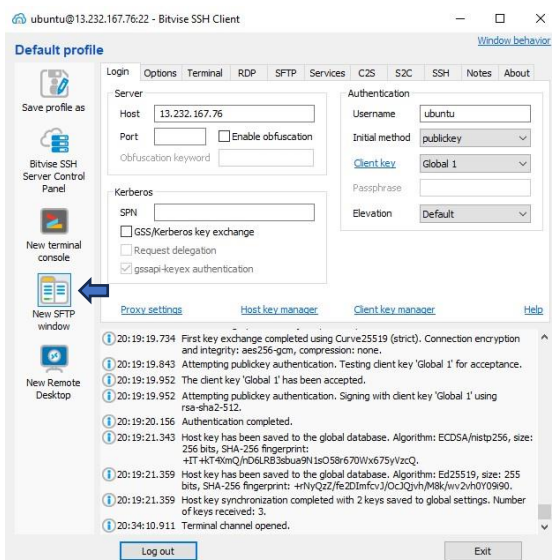


STEP 15-> Now the client is connected: Verify that the client has successfully established a connection to the EC2 instance.

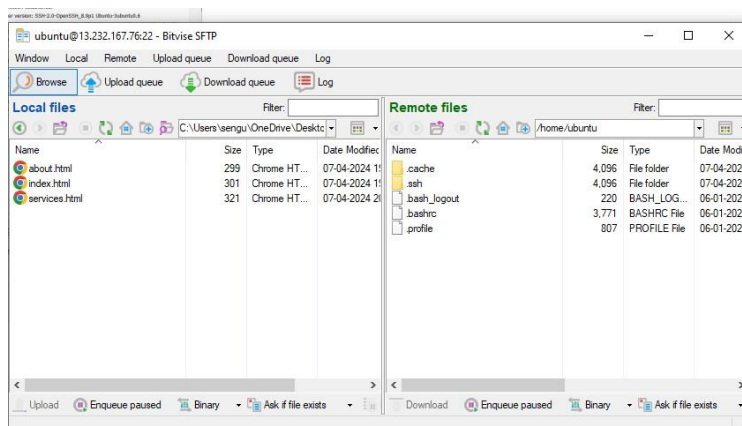
```
ubuntu@ip-172-31-44-99:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-44-99:~$
```

STEP 16-> Click on "New SFTP Window" button: Open a new SFTP (Secure File Transfer Protocol) window



STEP 17-> Under Local Files, open the folder where the HTML files are present: Navigate to the directory containing the HTML files on the local machine. Locating the HTML files that will be uploaded to the EC2 instance.



STEP 18-> Go to the terminal and type the following commands. “ sudo apt-get update “ , “ sudo apt-get upgrade “ , “ sudo apt-get install nginx ” sudo apt-get update: Updates the local package index to reflect the latest changes in repositories. sudo apt-get upgrade: Upgrades installed packages to their latest available versions. sudo apt-get install nginx: Installs the Nginx web server on the system.

```
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-44-99:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-44-99:~$ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
```

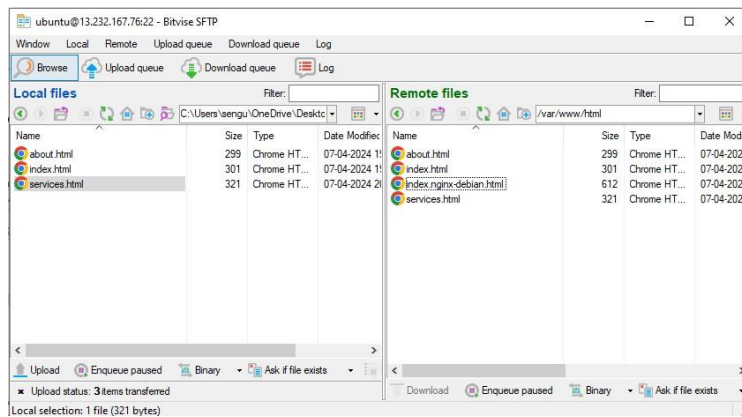
```
Last login: Sun Apr  7 15:04:13 2024 from 152.58.180.135
ubuntu@ip-172-31-44-99:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  linux-aws linux-headers-aws linux-image-aws ubuntu-advantage-tools ubuntu-pro-client-l10n
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
ubuntu@ip-172-31-44-99:~$ sudo apt-get install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libdeflate0 libfontconfig1 libgd3 libjpeg8 libjpeg-turbo8
  libjpeg8 libnghttp3 libnghttp3-dev libnghttp3-doc libnghttp3-mod-http-image-filter libnghttp3-mod-http-xslt-filter
  libnghttp3-mod-mail libnghttp3-mod-stream libnghttp3-mod-stream-geoip2 libtiff5 libwebp7 libxpm4
  nginx-common nginx-core
Suggested packages:
```

STEP 19-> Type the command “sudo chmod 777 html” and press “Enter”.

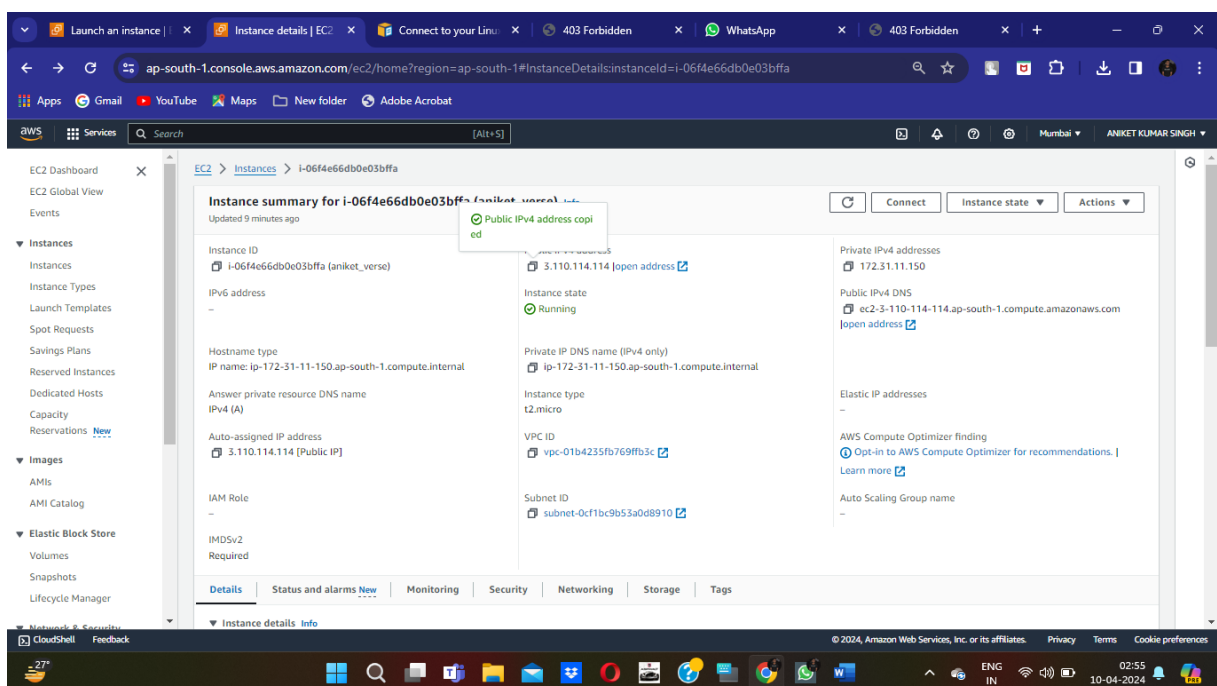
The command "sudo chmod 777 html" is used to change the permissions of the directory named "html" to allow full read, write, and execute permissions for all users.

```
ubuntu@ip-172-31-44-99:~$ sudo chmod 777 html
chmod: cannot access 'html': No such file or directory
ubuntu@ip-172-31-44-99:~$ sudo chmod 777 HTML
chmod: cannot access 'HTML': No such file or directory
ubuntu@ip-172-31-44-99:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-44-99:~$ cd /
ubuntu@ip-172-31-44-99:/ $ pwd /
ubuntu@ip-172-31-44-99:/ $ pwd /
ubuntu@ip-172-31-44-99:/ $ ls
bin  dev  home  lib32  libx32  media  opt  root  sbin  srv  tmp  var
boot  etc  lib  lib64  lost+found  mnt  proc  run  snap  sys  usr
ubuntu@ip-172-31-44-99:/ $ cd Desktop
-bash: cd: Desktop: No such file or directory
ubuntu@ip-172-31-44-99:/ $ cd home
ubuntu@ip-172-31-44-99:/home $ ls
ubuntu
ubuntu@ip-172-31-44-99:/home $ cd /
ubuntu@ip-172-31-44-99:/ $ cd /
ubuntu@ip-172-31-44-99:/ $ cd var
ubuntu@ip-172-31-44-99:/var $ ls
backups  cache  crash  lib  local  lock  log  mail  opt  run  snap  spool  tmp  www
ubuntu@ip-172-31-44-99:/var $ cd www
ubuntu@ip-172-31-44-99:/var/www $ ls
html
ubuntu@ip-172-31-44-99:/var/www $ cd html
ubuntu@ip-172-31-44-99:/var/www/html $ ls
index.nginx-debian.html
ubuntu@ip-172-31-44-99:/var/www/html $ cd ..
ubuntu@ip-172-31-44-99:/var/www $ cd ..
ubuntu@ip-172-31-44-99:/var $ cd www
ubuntu@ip-172-31-44-99:/var/www $ sudo chmod 777 html
ubuntu@ip-172-31-44-99:/var/www $
```

STEP 20-> Now going back to the "SFTP Window," under the "Remote Files," open the HTML directory and drag & drop the HTML files: Transfer the HTML files from the local machine to the EC2 instance using SFTP. Uploading the website content to the EC2 instance for hosting



STEP 21-> Now go back to the “AWS Window” and copy the public IPV4 address and paste in the new tab. Anyone can access it.



STEP 22-> A new window will open with the webpage.

