

ASSIGNMENT 7

Problem Statement: Upload a website on EC2.

1. Create two html file **Home.html** and **Next.html**.

*[Note-We should name the homepage of our website as **index.html** as nginx takes it as the starting page, but for better understanding in later steps we have taken names randomly]*

2. Open the **Amazon Web Services** home page (aws.amazon.com).

3. Choose **Sign into Console**.

4. Sign in as **Root user** using your email address and password.

5. Go to search and search EC2. Click on **Launch instance**.

The screenshot displays the AWS Management Console interface. At the top, the navigation bar indicates the user is logged in as 'MD SOHAIL' in the 'Mumbai' region. The main dashboard is organized into several panels:

- Resources:** A section titled 'You are using the following Amazon EC2 resources in the Asia Pacific (Mumbai) Region:' containing a grid of resource counts: Instances (running) 0, Elastic IPs 0, Load balancers 0, Snapshots 0, Auto Scaling Groups 0, Instances 0, Placement groups 0, Volumes 0, Dedicated Hosts 0, Key pairs 3, and Security groups 4.
- Account attributes:** A panel on the right with links for 'Supported platforms', 'Default VPC', 'Settings', 'EBS encryption', 'Zones', 'EC2 Serial Console', 'Default credit specification', and 'Console experiments'.
- Launch instance:** A panel on the bottom left with a large orange 'Launch instance' button and a 'Migrate a server' link.
- Service health:** A panel on the bottom middle showing the 'AWS Health Dashboard' and the current region 'Asia Pacific (Mumbai)'.
- Explore AWS:** A panel on the bottom right featuring promotional text: 'Get Up to 40% Better Price Performance' and 'Save up to 90% on EC2 with Spot Instances'.

6. In **Launch an instance page** page:

- a) Give the instance name in the Name and tags section. Give globally unique name as it is created globally.
- b) Select an OS from **Application and OS Images section**.

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

[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

Quick Start

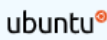
Amazon Linux




macOS




Ubuntu




Windows




Red Hat



S





[Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type

ami-0f8ca728008ff5af4 (64-bit (x86)) / ami-08795883c7b4b7140 (64-bit (Arm))

Free tier eligible ▼

- c) Keep the instance type as t2.micro .
- d) Then create a new key pair by clicking on **Create new key pair**
->In the create key pair page give it a name and select RSA
in **key pair type** and .pem in **Private key file format** and click on create a new key pair. It will download the the key pair in your local computer also.

▼ Instance type [Info](#)

Instance type

t2.micro

Free tier eligible

Family: t2 1 vCPU 1 GiB Memory

On-Demand Linux pricing: 0.0124 USD per Hour

On-Demand Windows pricing: 0.017 USD per Hour

On-Demand RHEL pricing: 0.0724 USD per Hour

On-Demand SUSE pricing: 0.0124 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*

50Key3



 [Create new key pair](#)

- Network settings . . .

744

▼ **Network settings** [Info](#)

Edit

Network **Info**

vpc-0faaa8cd9dd008a62

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-4**' with the following rules:

☒ Allow SSH traffic from

Helps you connect to your instance

Anywhere

0.0.0.0/0

▼

- ☒ Allow HTTPS traffic from the internet

To set up an endpoint, for example when creating a web server

- ☒ Allow HTTP traffic from the internet

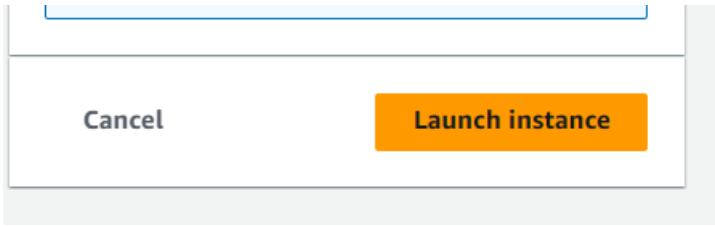
To set up an endpoint, for example when creating a web server



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.



e) Click on **Launch Instance** .

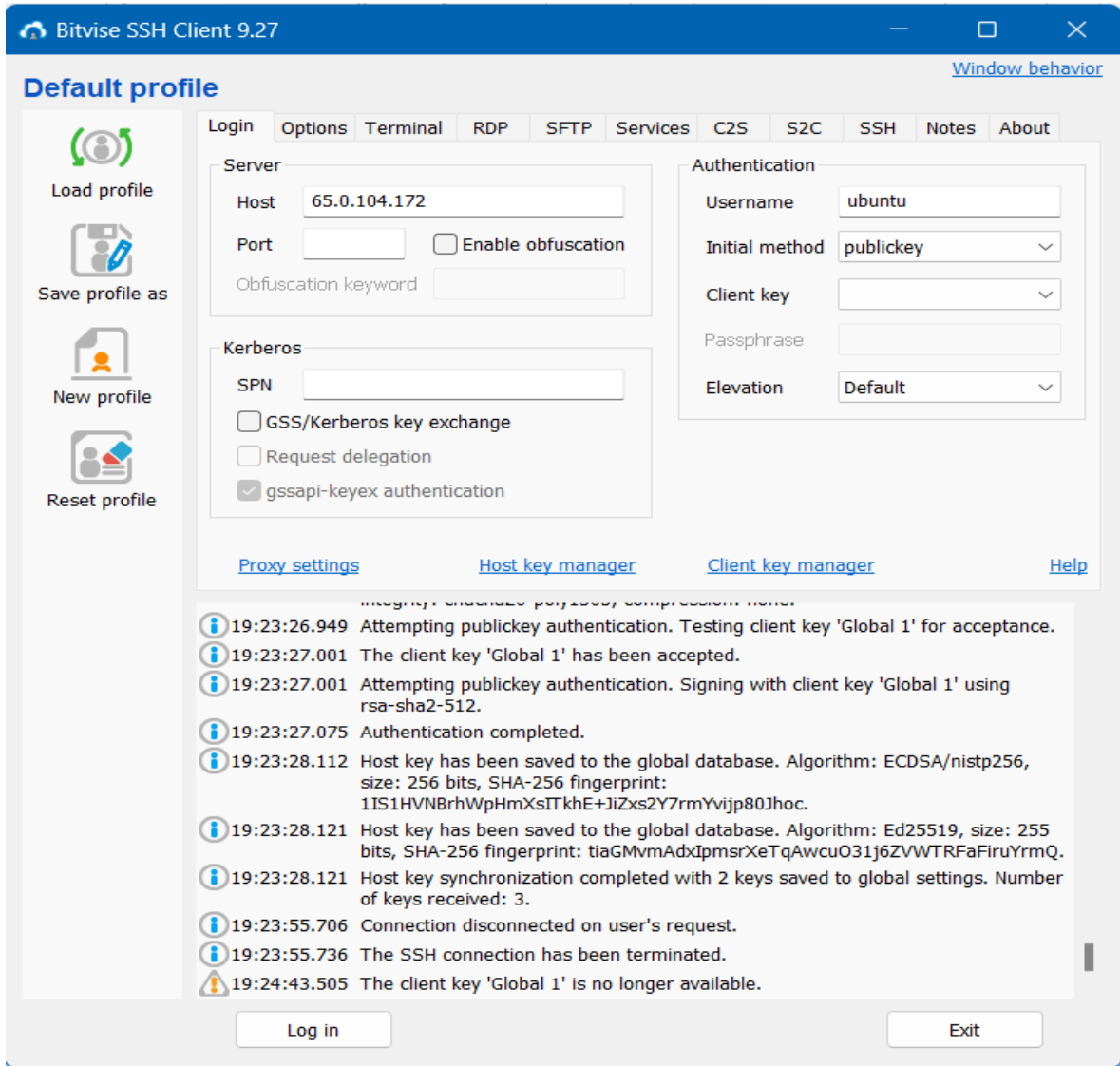


7. Go back to instances page. Select the instances and copy the public ipv4 from the **Details** page.

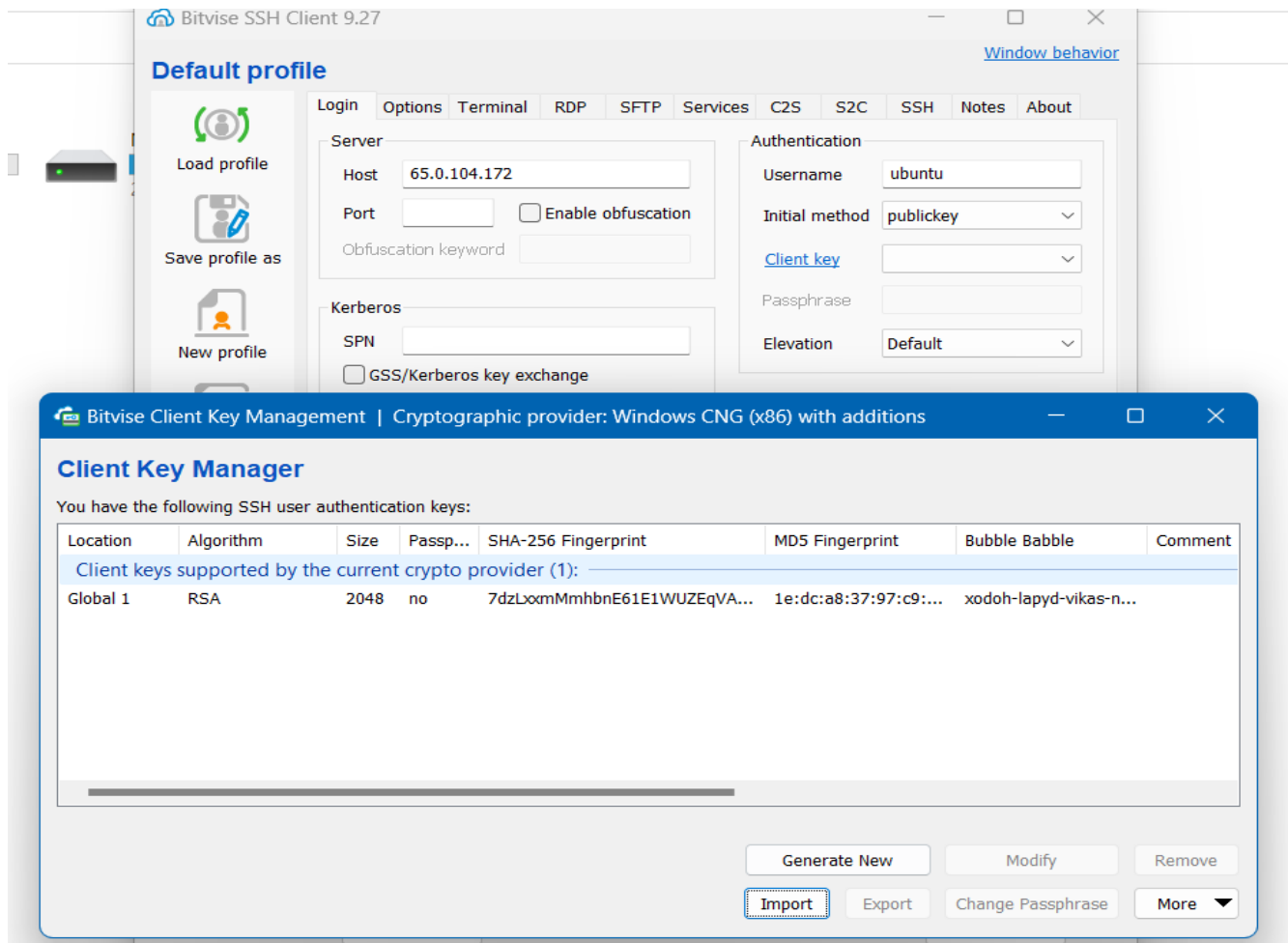
Instances (1) Info										
<div>Find instance by attribute or tag (case-sensitive)</div>										
<div><div>Refresh</div><div>Connect</div><div>Instance state</div><div>Actions</div><div>Launch instances</div></div>										
<div>< 1 > ⚙</div>										
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS		
<input type="checkbox"/>	Sohailec23080	i-0bb4290134f0a7b94	Running	t2.micro	-	No alarms	ap-south-1a	ec2-65-0-104-172.ap-s...	65	

8. Download and install **Bitvise SSH Client**. Open **Bitvise SSH Client**. In **Bitvise SSH Client**

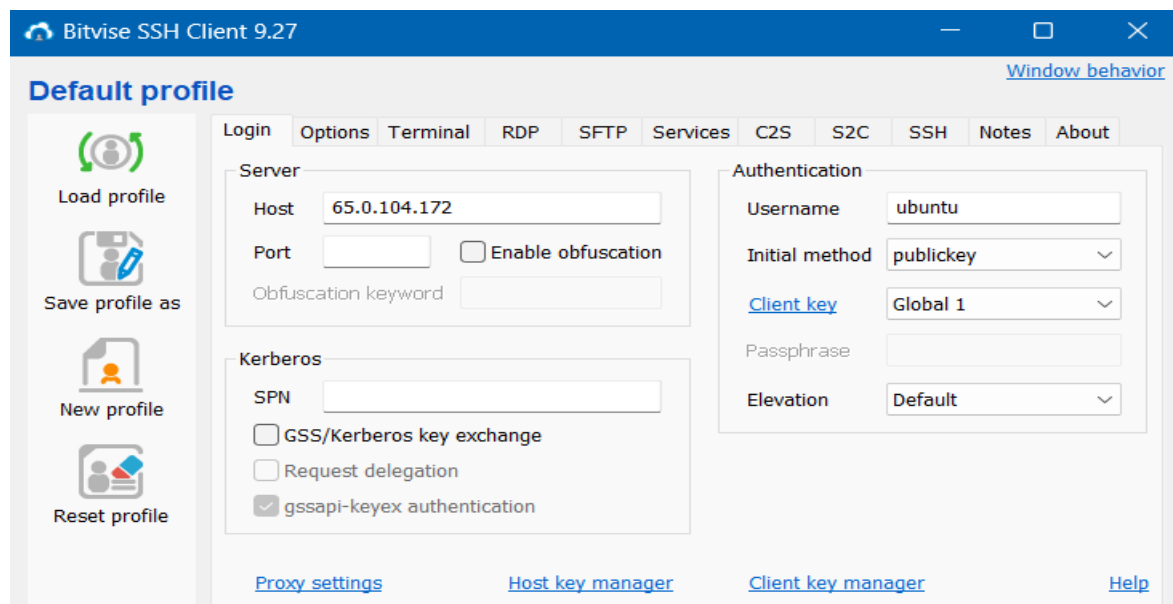
- a) Paste the public ipv4 in the host in **Server** section.
- b) Keep the Username ubuntu in **Authentication** section as we using ubuntu in EC2 and publickey as **Initial method**.



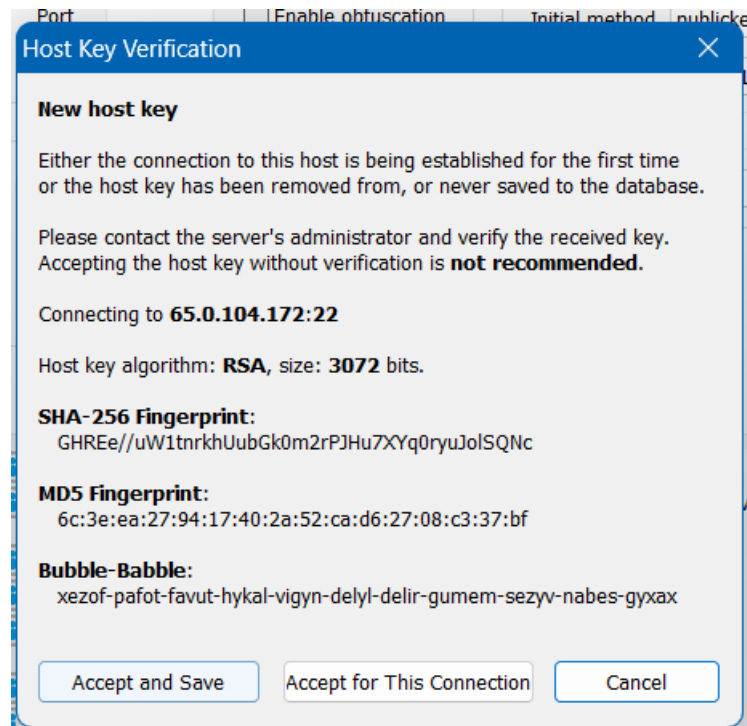
c) Click on client key manager, remove if any key exists, next import your key pair, next close client key manager.



Next, Choose the client key that you imported



9. Click on login and click on accept and a save.



10. Click on **New terminal console** on the left sidebar

In the terminal run command,

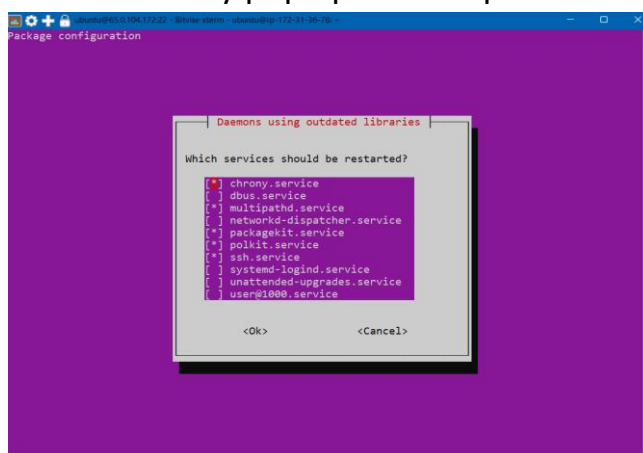
a) `pwd` //To show present working directory

b) `sudo apt-get update`

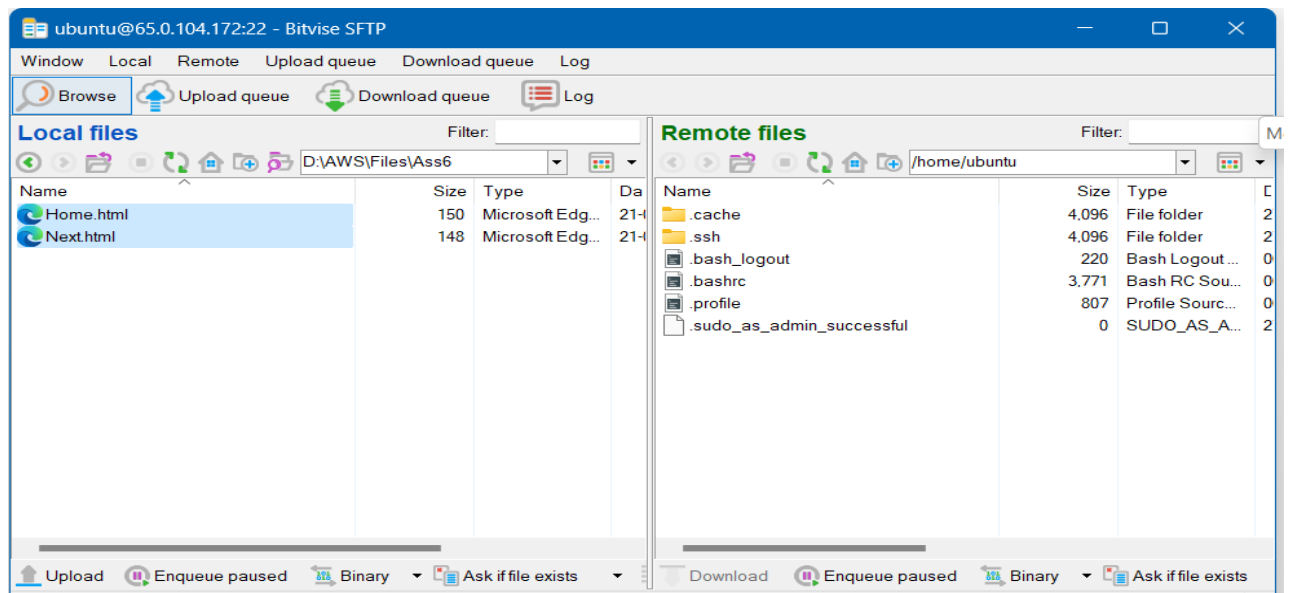
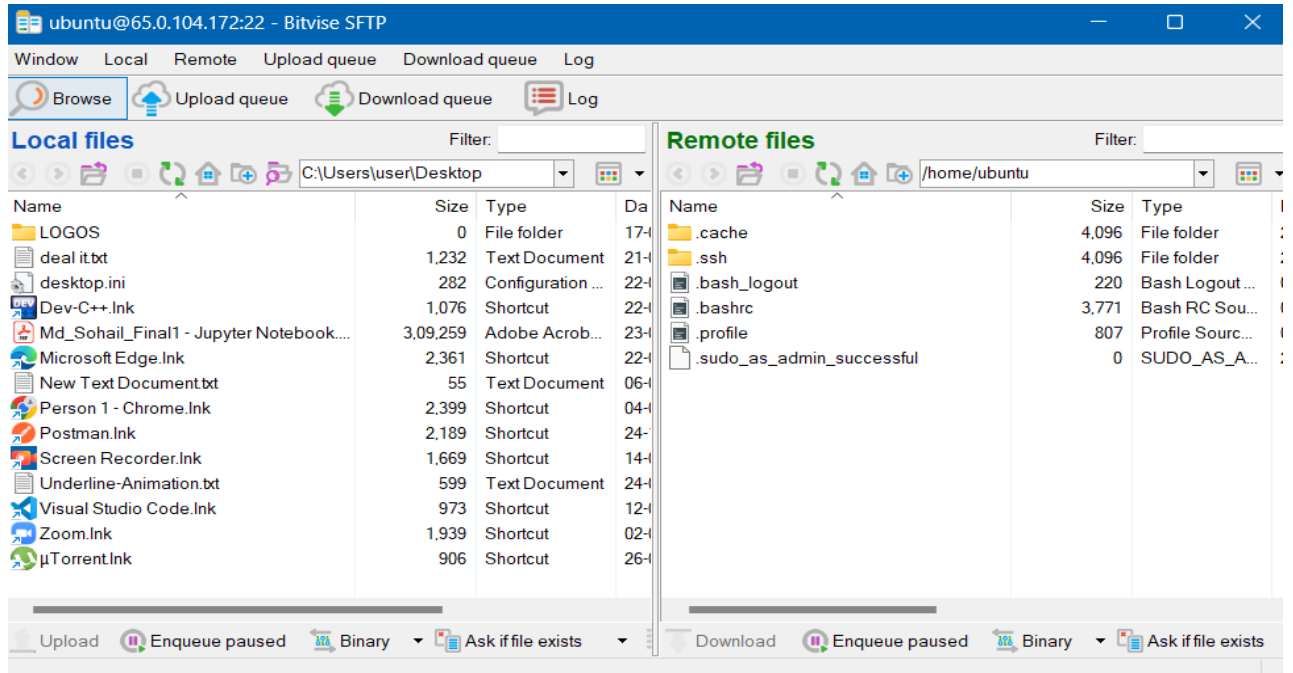
c) `sudo apt-get upgrade`

d) `sudo apt-get install nginx`

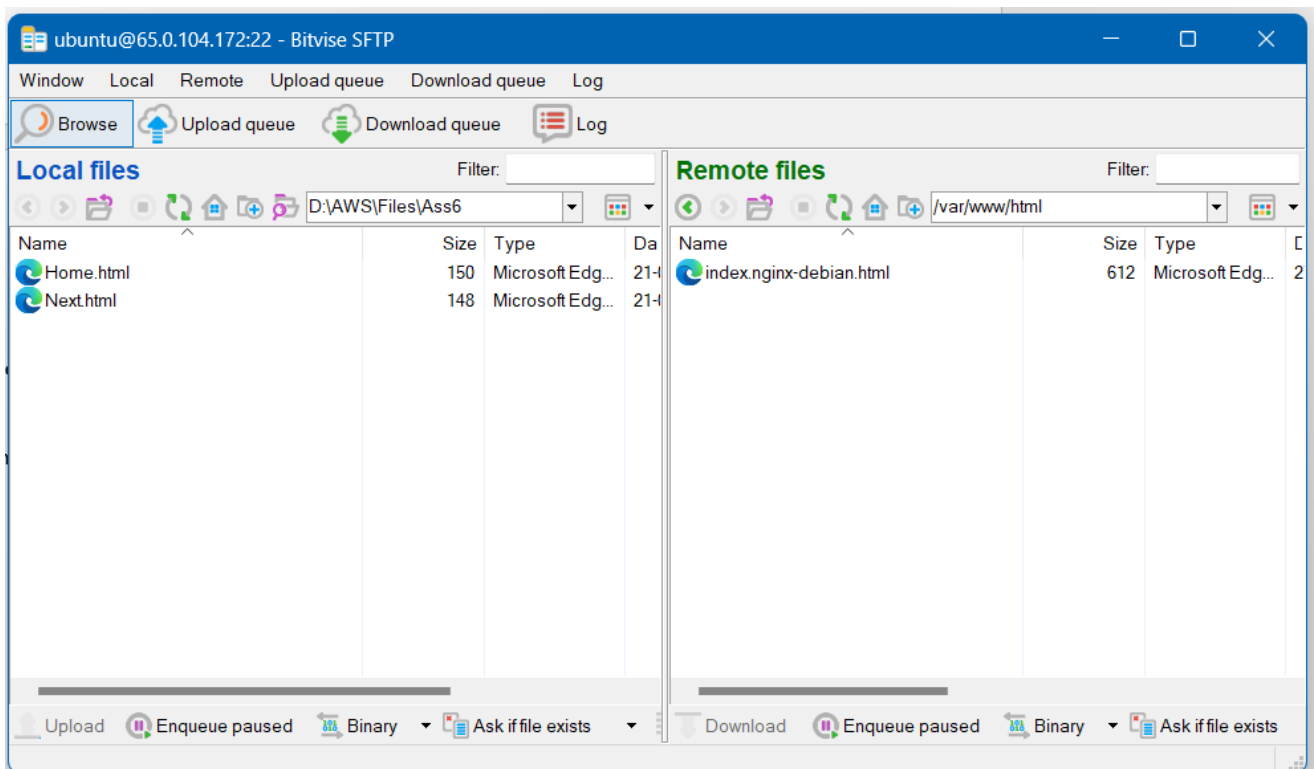
Select ok if any pop up shows up



11.. Go back to the **Bitvise SSH Client**. Click on **new SFTP window** in the left side panel. In the SFTP panel in the **Local Files** go to your folder where you kept the **HTML** files.



Next, In remote files go to root by clicking on **Up** or using the **path "/"**, then open the **var**, open **www**, open **html** folder.



12. Copy the EC2 instance IPv4 and open in a new browser window. You can see the Welcome to “nginx!” page.



13. Go back to the **terminal console**.

In the terminal console run,

a) `nginx -v` // To show the version info of the nginx

```
ubuntu@ip-172-31-36-76:~$ nginx -v
nginx version: nginx/1.18.0 (Ubuntu)
```

b) `cd /` //To go back to the root in the terminal

```
ubuntu@ip-172-31-36-76:~$ cd /
ubuntu@ip-172-31-36-76:/$ pwd
/
```

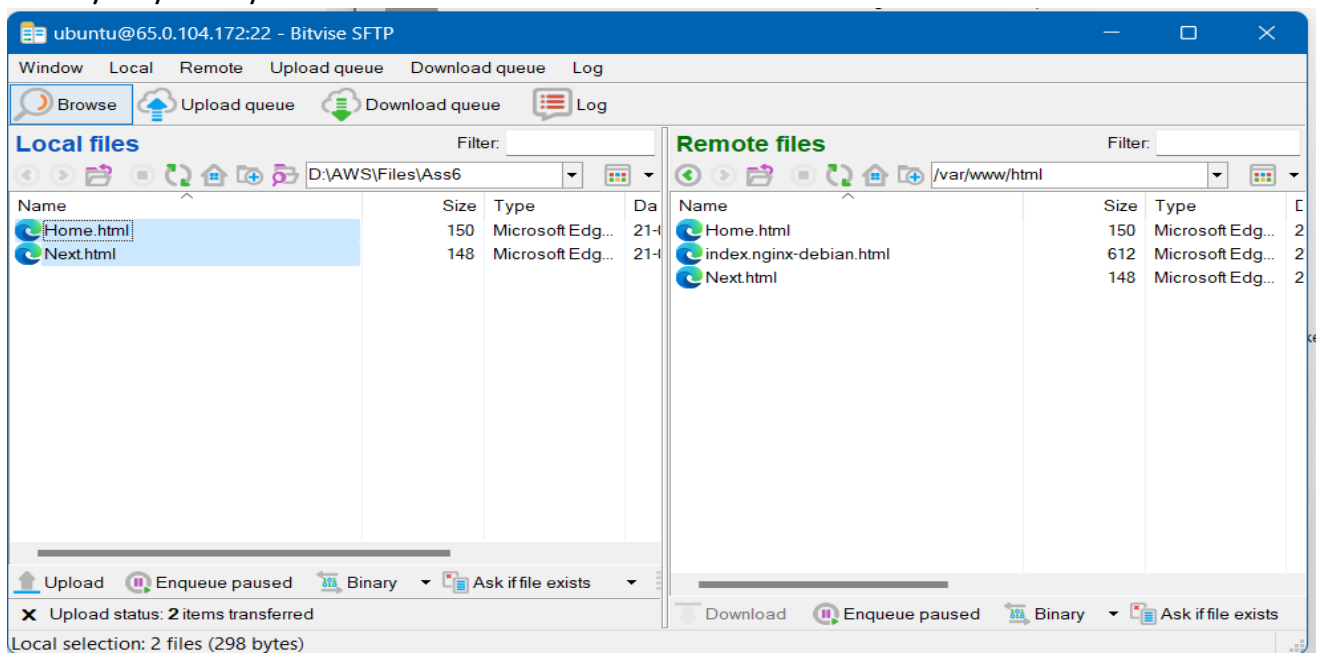
- c) `cd /var/www` //To go to the www folder
d) `ls` //To show the files

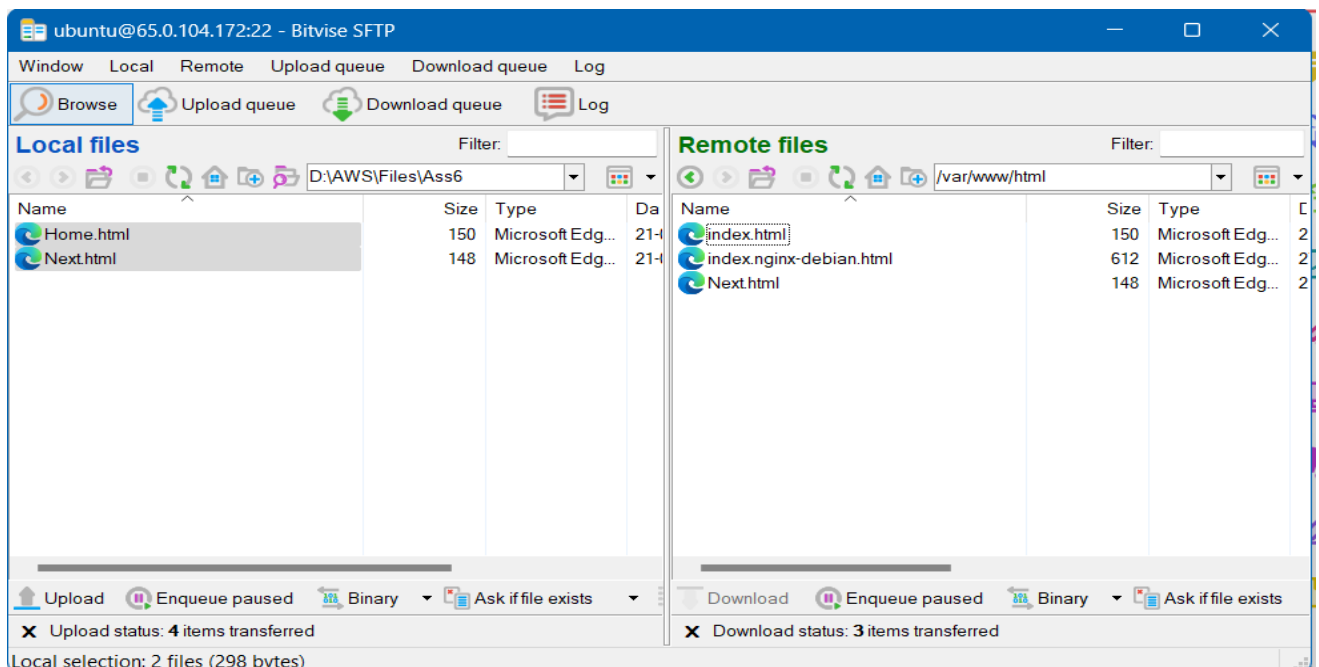
```
ubuntu@ip-172-31-36-76:/$ cd /var/www
ubuntu@ip-172-31-36-76:/var/www$ ls
html
```

- e) `sudo chmod 777 html` //To give read and write permissions to the HTML folder.

```
ubuntu@ip-172-31-36-76:/var/www$ sudo chmod 777 html
ubuntu@ip-172-31-36-76:/var/www$ ls -l
total 4
drwxrwxrwx 2 root root 4096 Mar 27 14:14 html
```

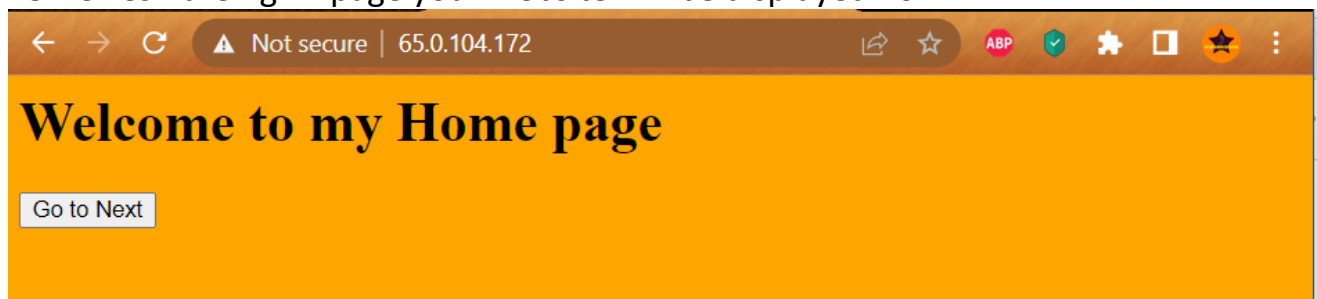
14. Now go to the Bitvise's SFTP panel and move your file from local files to remote files “/var/www/html”.





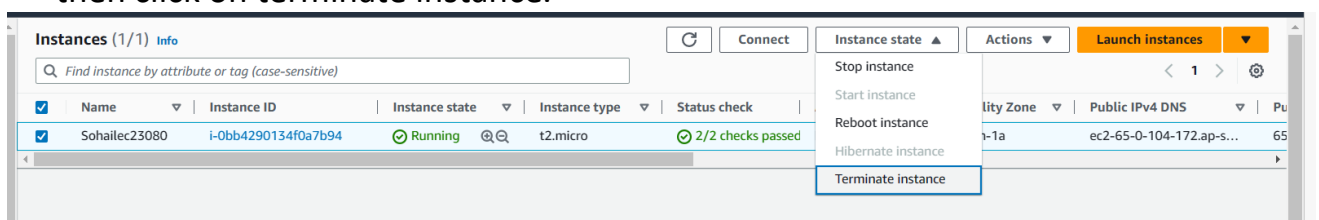
Make sure the home of your webpage is “**index.html**”. If it is not, rename it to **index.html** as nginx takes it as the starting page of the website.

15.Refresh the nginx page your website will be displayed now.



16.Now for **Termination**:

a) Click on instance state On the EC2 Instances page and click on **instance state** then click on terminate instance.



b) Click on **Abort** in Bitvise SSH Client.

- 19:27:09.210 Authentication completed.
- 19:30:34.768 SFTP channel opened.
- 19:30:36.822 SFTP channel closed by user.
- 19:30:37.945 Terminal channel opened.
- 19:33:30.624 SFTP channel opened.
- 19:45:04.860 SFTP channel closed by user.
- 19:45:12.251 SFTP channel opened.
- 20:11:49.511 The SSH connection has terminated with error. Reason: Error class: LocalSshDisconn, code: ConnectionLost, message: FlowSshTransport: received EOF.
- 20:11:49.545 Automatic reconnection started.
Reconnection attempt is scheduled at 20:11:52.
- 20:11:51.526 Started a new SSH connection.
- 20:11:51.548 Connecting to SSH server 65.0.104.172:22.

Abort

Exit