Assignment : 07

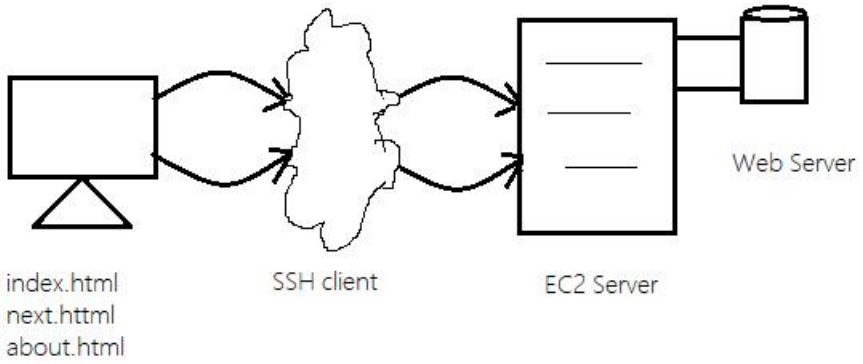
Title : Upload a static website in EC2 server.

Elastic Compute Cloud(EC2) :

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

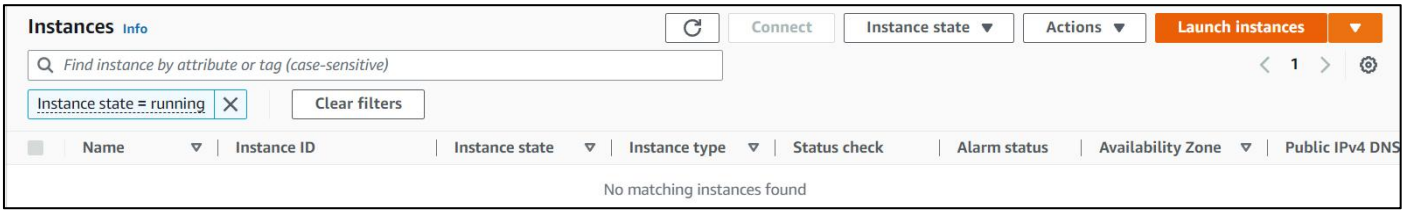
Components Required to upload a static website on EC2 :

1. Server → Hardware (t2Micro) / Platform (Ubuntu)
2. Web Server (NGINX)
3. SSH Client (bitvise)
4. Public key



Steps to create an instance on EC2 :

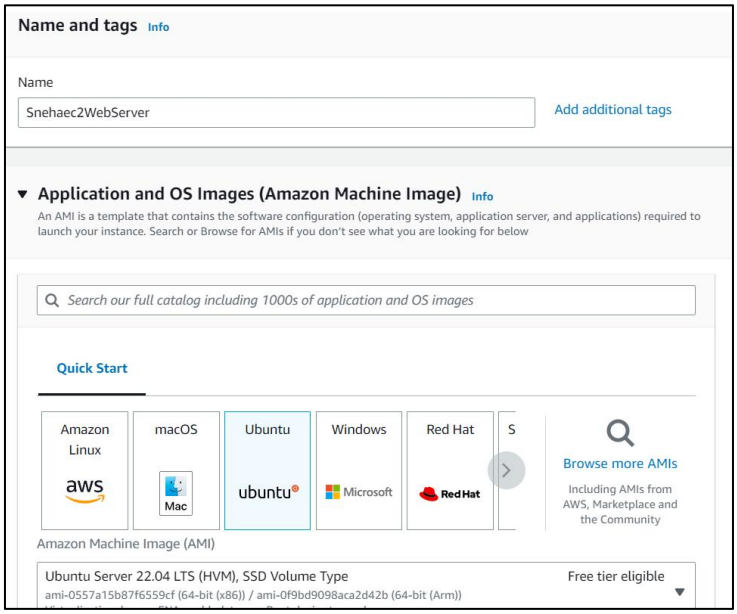
1. Open the Amazon EC2 console.
2. From the EC2 console dashboard, Click on **Instances(Running)**, choose **Launch instance.**



and The **Launch an instance** page opens..

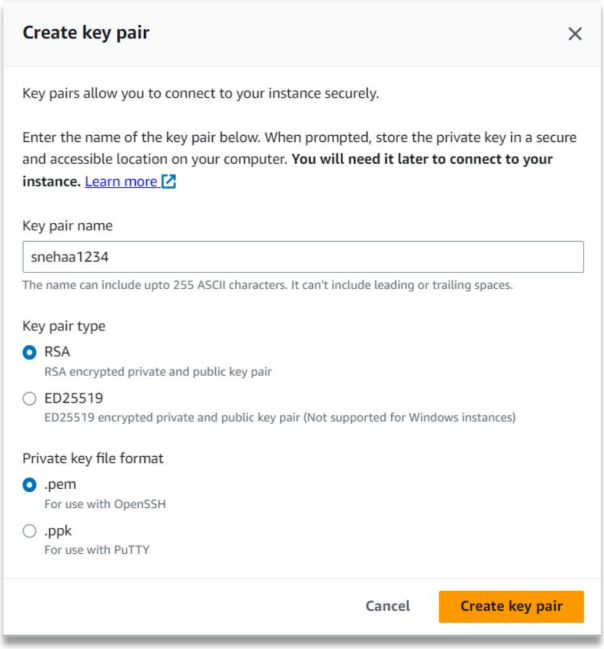
3. Under **Name and tags**, for **Name**, enter a descriptive name for your instance like ' **Snehaec2WebServer** '.
4. Under **Application and OS Images (Amazon Machine Image)**, do the following:

Choose **Quick Start**, and then choose **Ubuntu**. This is the operating system (OS) for your instance, which is Free Tier Eligible.



Under **Instance type**, from the **Instance type** list, you can select the hardware configuration for your instance. Choose the `t2.micro` instance type, which is selected by default. The **`t2.micro`** instance type is eligible for the free tier.

5. Under **Key pair (login)**, for **Key pair name**, choose the key pair that you created already or Choose **Create new key pair**. A dialogue box opens – Give a name to the key pair under the **Key pair name** like **snehaa1234**

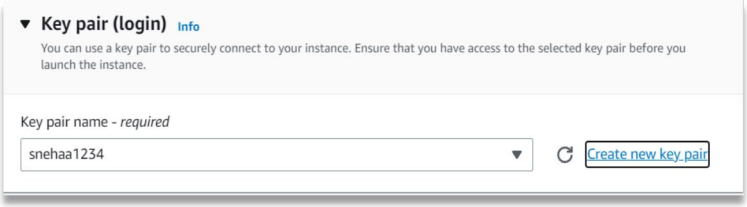


The key pair generated is of:

i. Type – **RSA**

ii. File format – **.pem**

Click on **Create key pair** and the .pem file of your key pair is automatically downloaded. And is saved for further use.



6. In **Network settings** , under the **Firewall (Security groups)** there is a by default selection of **Create security Groups** under which check or select all the three boxes namely :

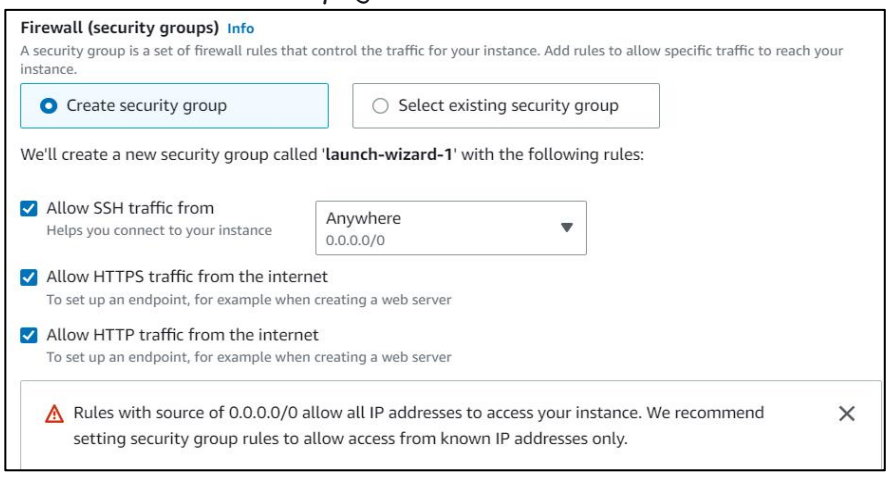
☑ **Allow SSH traffic from** – Helps you connect to your instance

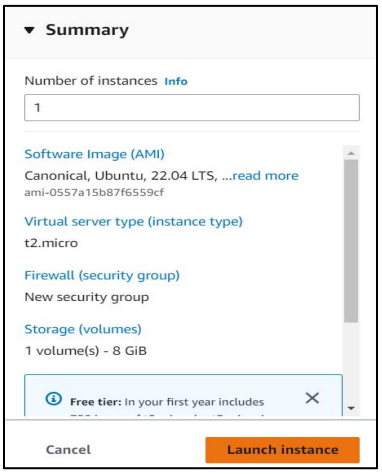
☑ **Allow HTTPS traffic from the internet** – To set up an end point.

☑ **Allow HTTP traffic from the internet** – To set up an endpoint .

7. Keep the default selections for the other configuration settings for your instance.Review a summary of your instance configuration in the **Summary** panel, and when you're ready, choose **Launch instance.**

A confirmation page lets you know that your instance is launching. Choose **View all instances** to close the confirmation page and return to the console.





Steps to link Server and client using Bitvise SSH :

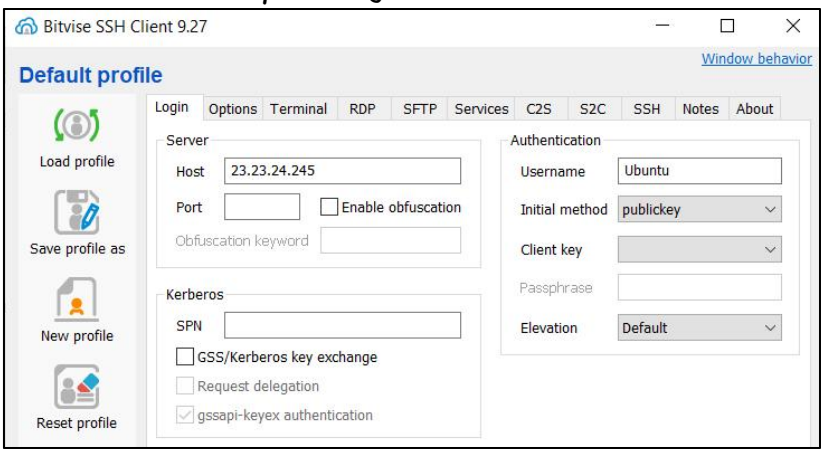
1. Click on the **Instance ID** of the instance you created. The instance summary opens .
2. Copy the **Public IPv4 Address**.

3. Download the **Bitvise SSH client** from browser - Install it and open the application to move further.
4. Under **Login** section ,In **Server** - **Host** paste the **public IPv4 address** of the instance

In the **Authentication** part do as follows:

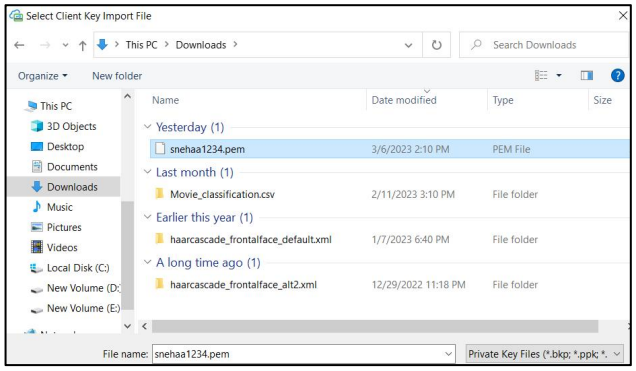
Username - **Ubuntu**

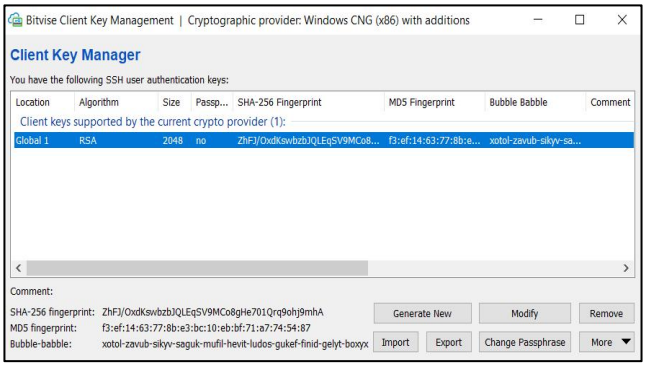
Initial method - **publickey**



Click on **Client key manager** , in the dialogue box Click on **Import** . Import the **key pair** generated while making the instance ->choose **open** -> **import**.

It is visible in the client key manager as **Global 1**. Return back (close the window)

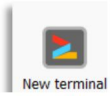




5. In the Authentication section , **Client key - Global 1**

Click On **log in -> Accept & Save.**

6. Open **new terminal console** - and type



(i) Sudo apt-get update && sudo apt-get upgrade

Followed by typing y when asked for yes/no and then pressing enter when finished.

(ii) Sudo apt-get install nginx

Followed by typing y when asked for yes/no and then pressing enter when finished.

```
ubuntu@ip-172-31-31-167:~$ sudo apt-get update && sudo apt-get upgrade
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [107 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [941 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [879 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [173 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [17
9 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [139 kB]
Fetched 2486 kB in 1s (1795 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
```

```
ubuntu@ip-172-31-31-167:~$ 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 libjbig0 libjpeg-turbo8
  libjpeg8 libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip2 libtiff5 libwebp7 libxpm4
  nginx-common nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core libdeflate0 libfontconfig1 libgd3 libjbig0 libjpeg-turbo8
  libjpeg8 libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip2 libtiff5 libwebp7 libxpm4 ngin
  nginx-common nginx-core
```

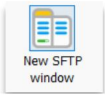
(iii) To check the version :

ubuntu@ip-172-31-31-167:~$ **nginx -v**

nginx version: nginx/1.18.0 (Ubuntu)

7. Go to **New SFTP window** , a window opens having two parts namely - **Local files** and **Remote files**.

In local files select the location where your html files are located. Then-



Follow the Steps:

a. Click on the icon until you reach the Root file with path "/".

b. Choose var->www->html , the final path will be "/var/www/html" as shown in the image below.

Now if you try to drag and drop the html files from local to remote it will still give you and error. This means we need to still edit some permissions.

i) Open **new Terminal console**

ii) Enter the correct path

iii) Change permissions

For which You need to apply following commands ->>>>

Commands used:

ubuntu@ip-172-31-31-167:~$ **pwd** *// will show present working directory*

/home/ubuntu

ubuntu@ip-172-31-31-167:~$ **cd ..** *// move up into previous directory*

ubuntu@ip-172-31-31-167:/home$ **cd ..**

ubuntu@ip-172-31-31-167:/$ **cd var/www/** *// move into the path provided*

ubuntu@ip-172-31-31-167:/var/www$ **pwd**

/var/www

ubuntu@ip-172-31-31-167:/var/www$ **sudo chmod 777 html** *// change permissions*

```
ubuntu@ip-172-31-31-167:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-31-167:~$ cd ..
ubuntu@ip-172-31-31-167:/home$ cd ..
ubuntu@ip-172-31-31-167:/$ cd var/www/
ubuntu@ip-172-31-31-167:/var/www$ pwd
/var/www
ubuntu@ip-172-31-31-167:/var/www$ sudo chmod 777 html
ubuntu@ip-172-31-31-167:/var/www$
```

8. Now drag and drop the files from **local to remote** .

9. Paste the **public IPv4 address** in a new web window , the html file opens .

That means we have successfully ***hosted a static website on EC2***.