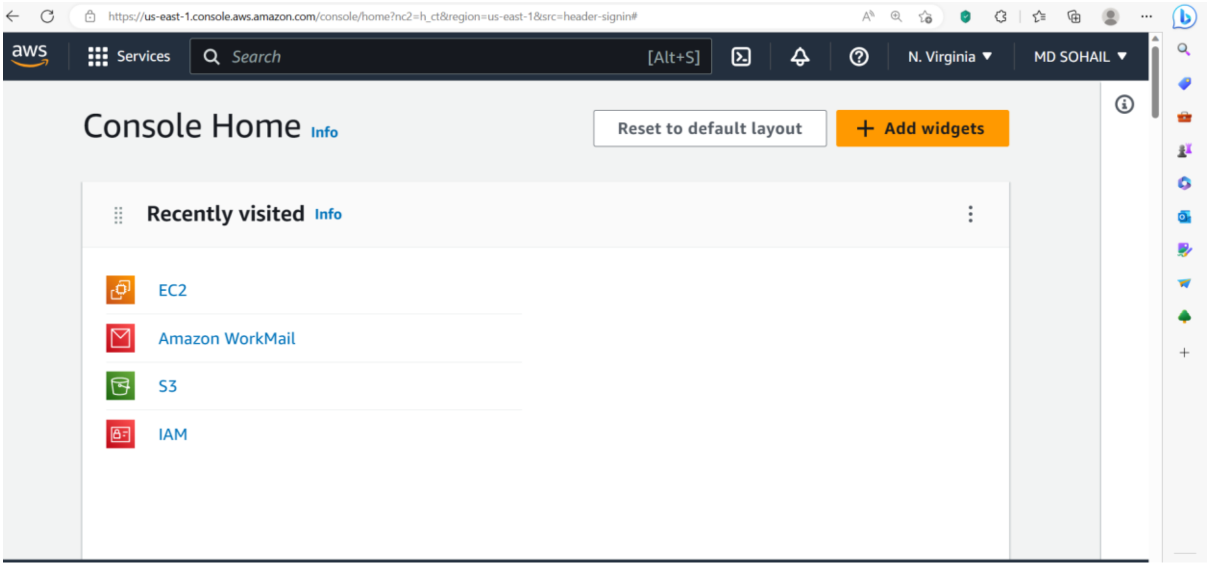
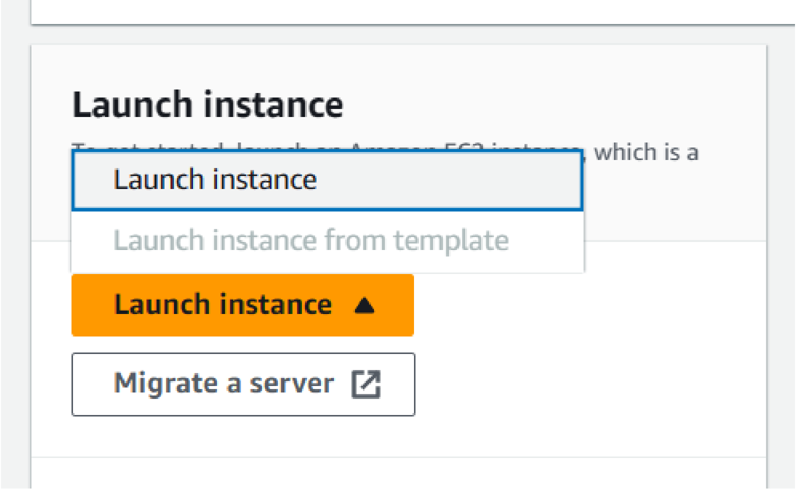
ASSIGNMENT 12

***Problem Statement*: Deploy and run project in AWS without using port.**

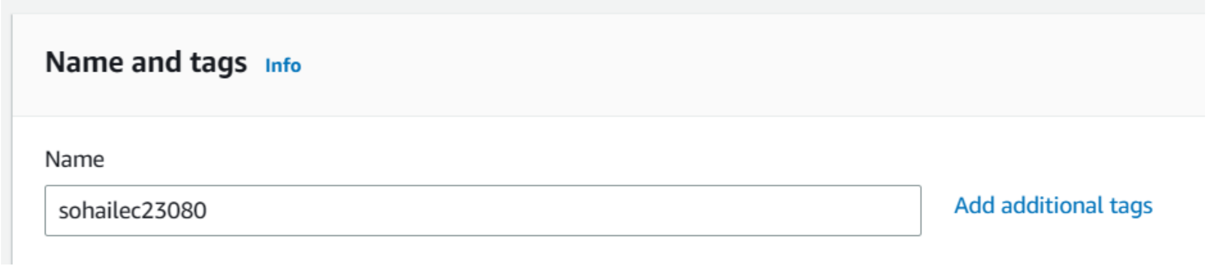
1. Sign in to your AWS account.



2. Go to EC2 and Click on Launch instance.

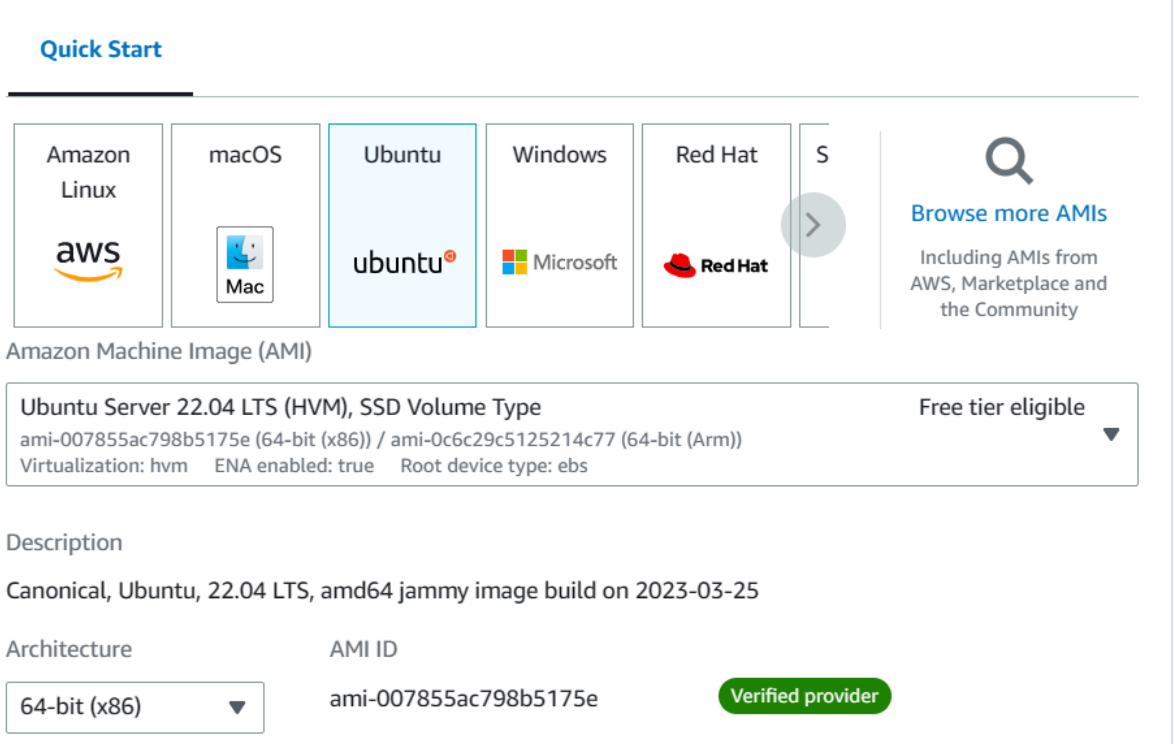

Launch instance
To get started, launch an Amazon EC2 instance, which is a
Launch instance
Launch instance from template
Launch instance
Migrate a server

a) Enter a name for the instance.



Name: sohailec23080

b) Select an OS for your server. [Here we have selected Ubuntu]



Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-007855ac798b5175e (64-bit (x86)) / ami-0c6c29c5125214c77 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

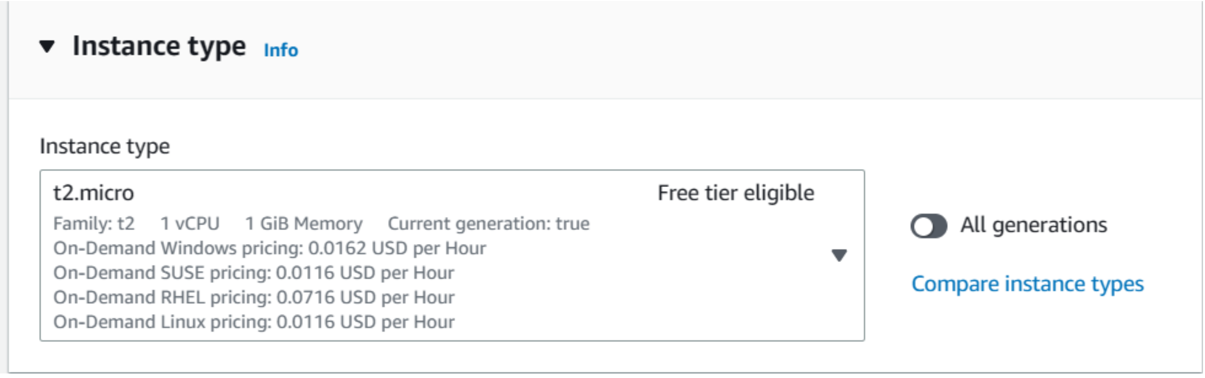
Description

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-03-25

Architecture: 64-bit (x86)

AMI ID: ami-007855ac798b5175e

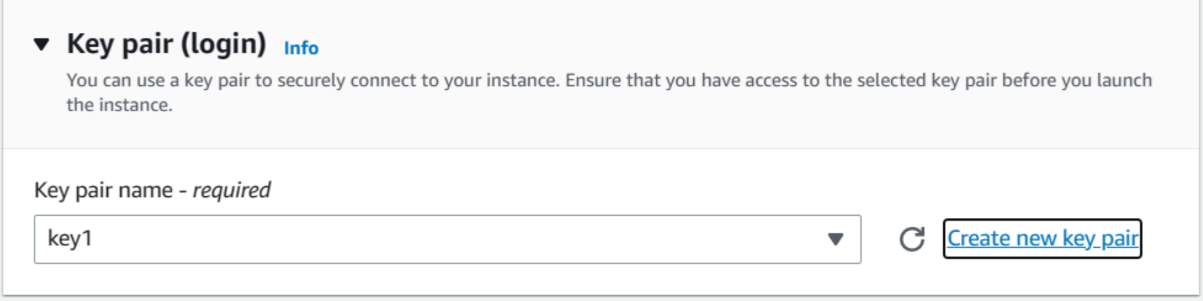
c) Select the instance type as t2.micro



Instance type

t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows pricing: 0.0162 USD per Hour
On-Demand SUSE pricing: 0.0116 USD per Hour
On-Demand RHEL pricing: 0.0716 USD per Hour
On-Demand Linux pricing: 0.0116 USD per Hour

d) Select a key pair if you already have created one otherwise create a new key pair.

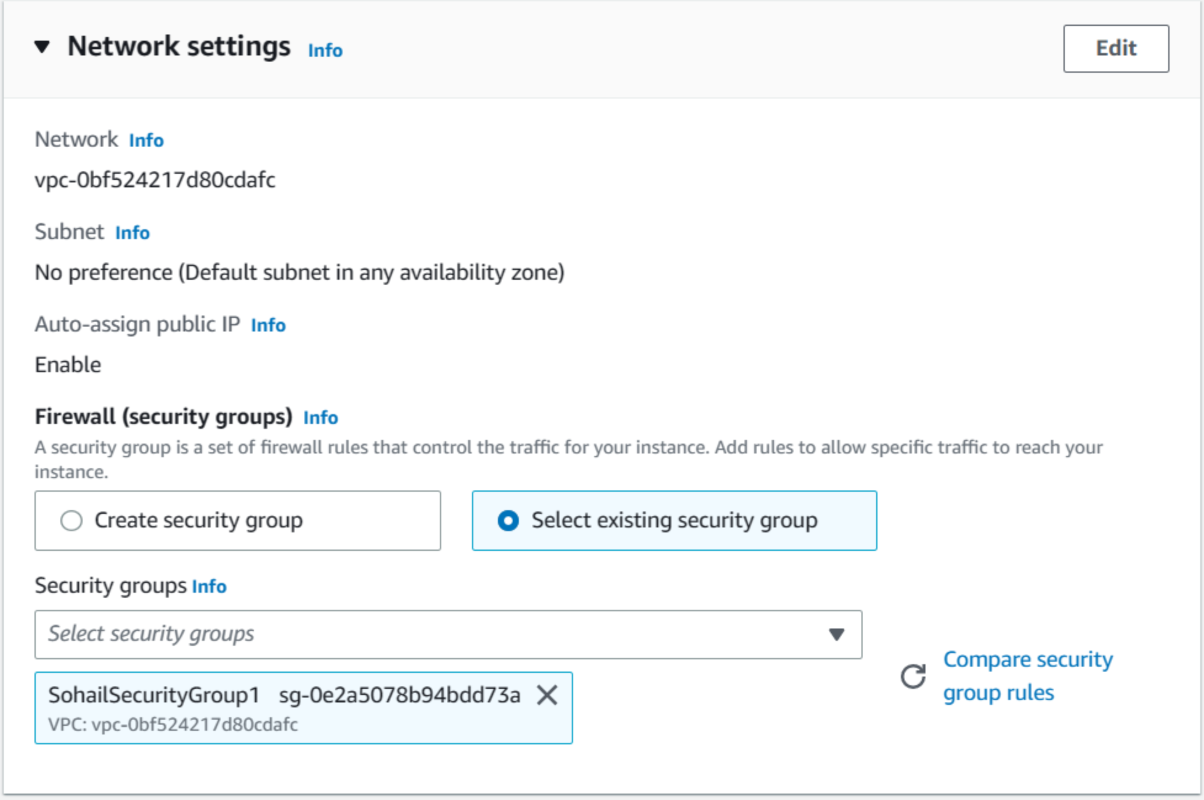


You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

key1

e) In the Network settings, select the existing security group.



Network Info

vpc-0bf524217d80cdafc

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

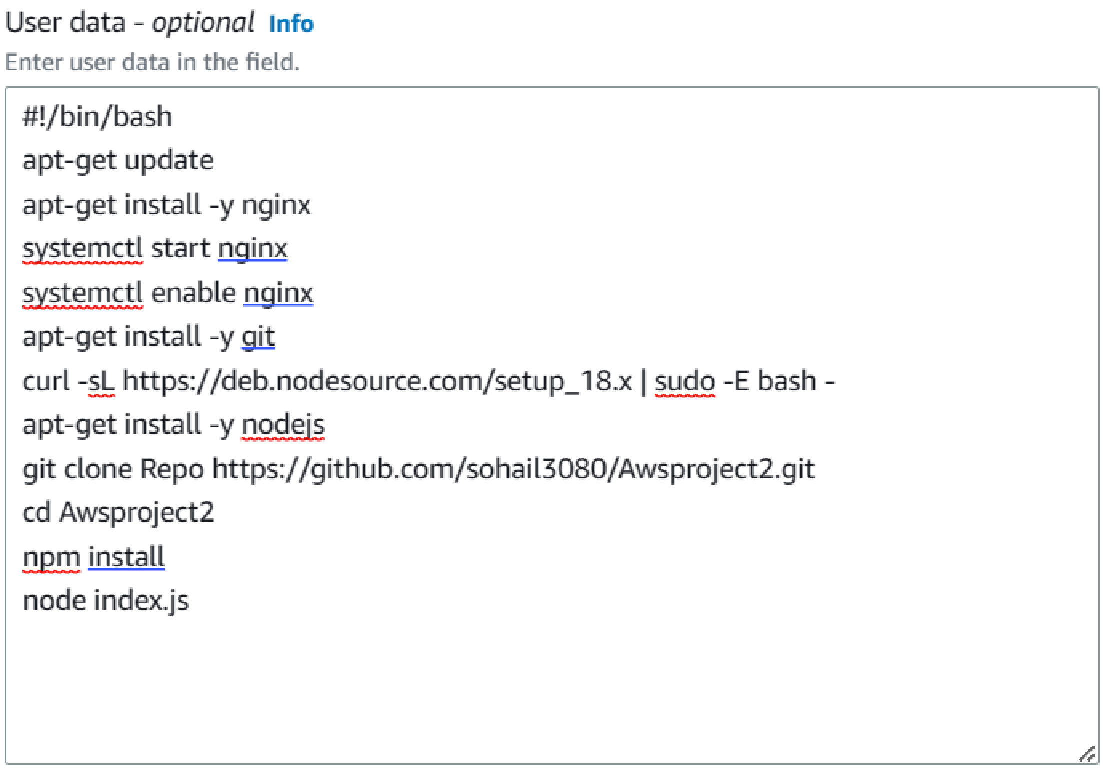
Security groups Info

Select security groups ▼

SohailSecurityGroup1 sg-0e2a5078b94bdd73a ✕
VPC: vpc-0bf524217d80cdafc

f) In Advanced Details, Enter the following commands in *User data* section-

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
git clone Repo link
cd Repo name
npm install
node index.js
```

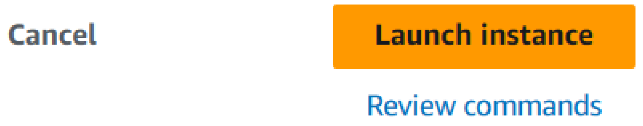


User data - *optional* **Info**

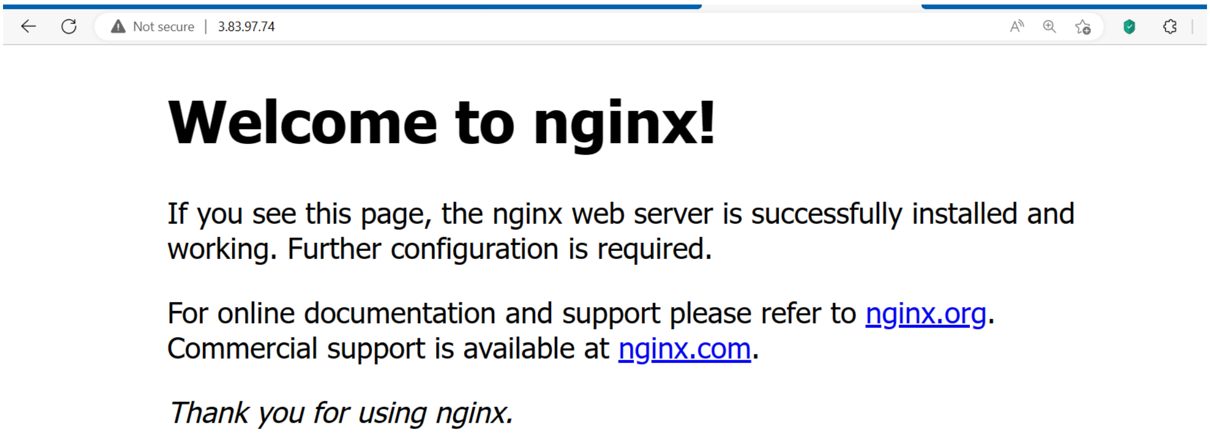
Enter user data in the field.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
git clone Repo https://github.com/sohail3080/Awsproject2.git
cd Awsproject2
npm install
node index.js
```

g) Now Click on Launch instance



h) As we can see, we have started nginx server and deployed the project successfully.

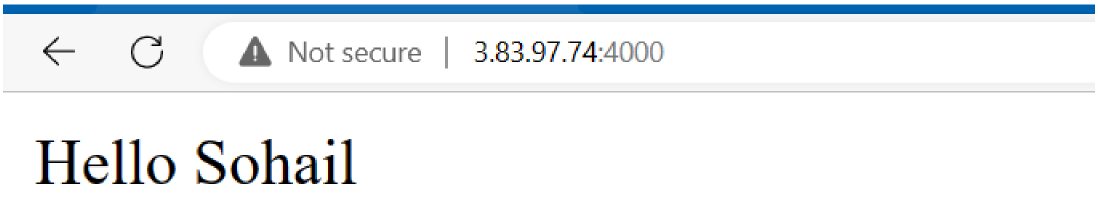


Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

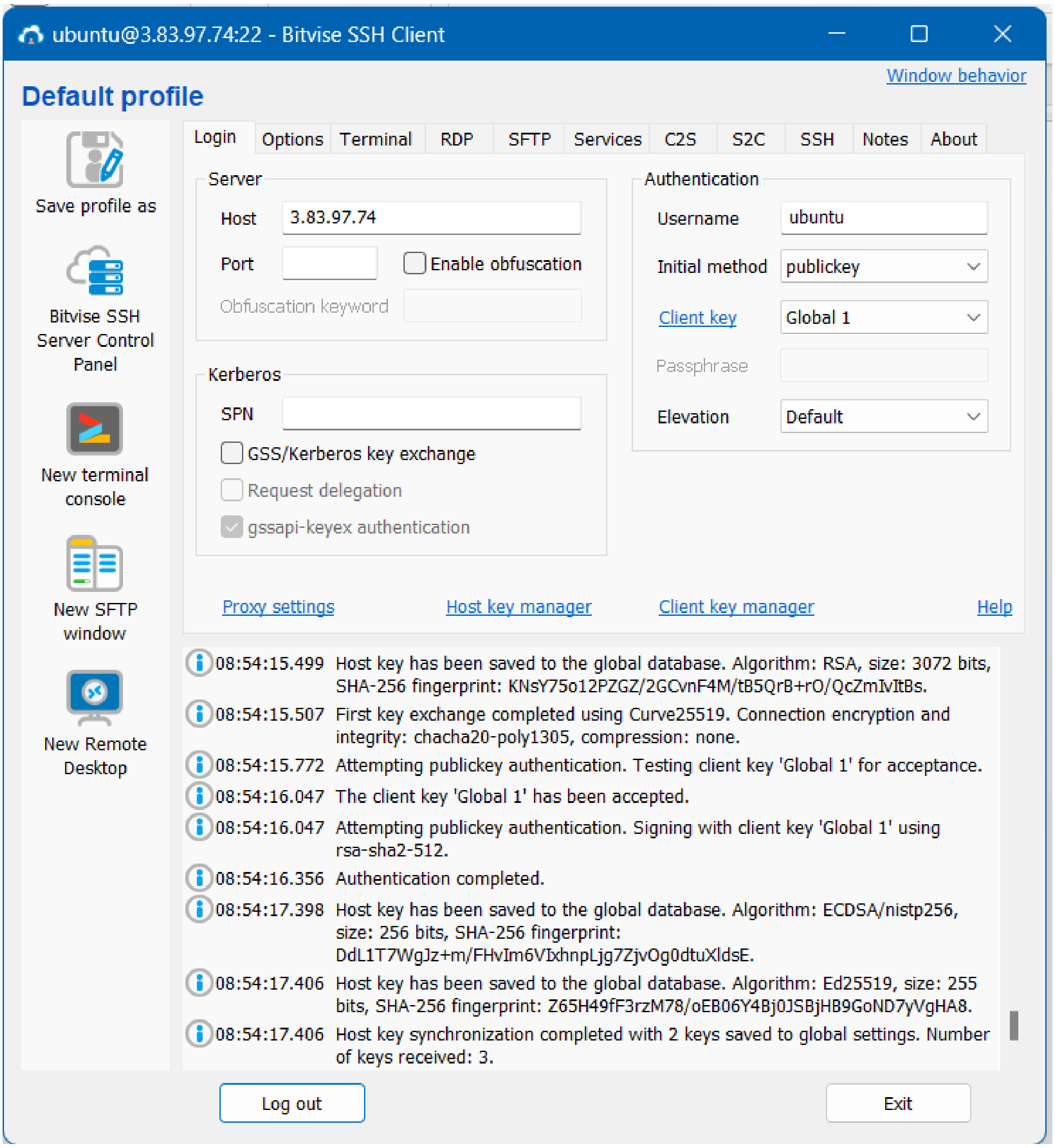
Thank you for using nginx.



Not secure | 3.83.97.74:4000

Hello Sohail

3. Now, Connect the instance using Bitvise SSH Client.



Server

Host: 3.83.97.74

Port:

□ Enable obfuscation

Obfuscation keyword:

Kerberos

SPN:

□ GSS/Kerberos key exchange

□ Request delegation

☑ gssapi-keyex authentication

Authentication

Username: ubuntu

Initial method: publickey

Client key: Global 1

Passphrase:

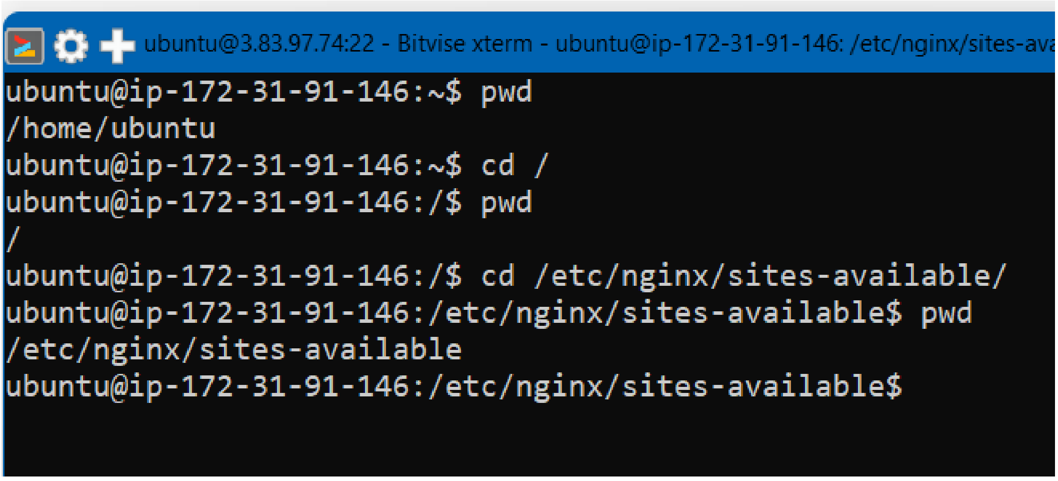
Elevation: Default

```
08:54:15.499 Host key has been saved to the global database. Algorithm: RSA, size: 3072 bits,
             SHA-256 fingerprint: KNsY75o12PZGZ/2GCvnF4M/tB5QrB+rO/QcZmIvItBs.
08:54:15.507 First key exchange completed using Curve25519. Connection encryption and
             integrity: chacha20-poly1305, compression: none.
08:54:15.772 Attempting publickey authentication. Testing client key 'Global 1' for acceptance.
08:54:16.047 The client key 'Global 1' has been accepted.
08:54:16.047 Attempting publickey authentication. Signing with client key 'Global 1' using
             rsa-sha2-512.
08:54:16.356 Authentication completed.
08:54:17.398 Host key has been saved to the global database. Algorithm: ECDSA/nistp256,
             size: 256 bits, SHA-256 fingerprint:
             DdL1T7WgJz+m/FHvIm6VIxhnpLjg7ZjvOg0dtuXldsE.
08:54:17.406 Host key has been saved to the global database. Algorithm: Ed25519, size: 255
             bits, SHA-256 fingerprint: Z65H49fF3rzM78/oEB06Y4Bj0JSBjHB9GoND7yVgHA8.
08:54:17.406 Host key synchronization completed with 2 keys saved to global settings. Number
             of keys received: 3.
```

4. In New terminal console,

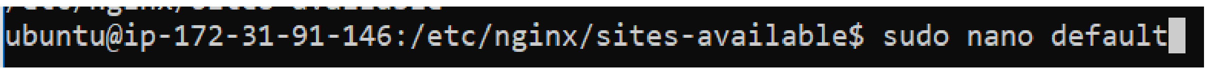
a) Enter the following commands-

- pwd [*to check the present working directory]*
- cd / *[to go to the root directory]*
- cd /etc/nginx/sites-available/


ubuntu@3.83.97.74:22 - Bitvise xterm - ubuntu@ip-172-31-91-146: /etc/nginx/sites-ava

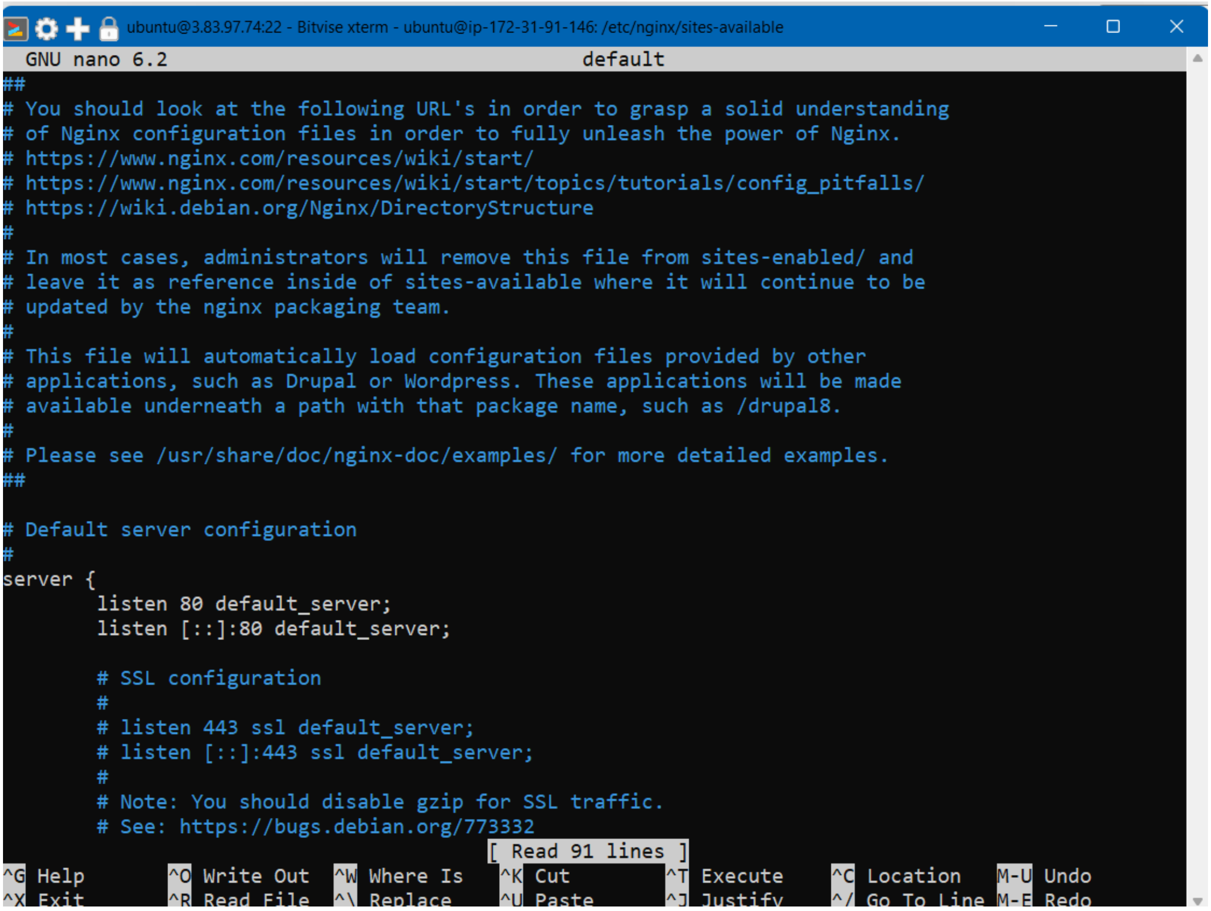
```
ubuntu@ip-172-31-91-146:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-91-146:~$ cd /
ubuntu@ip-172-31-91-146:/$ pwd
/
ubuntu@ip-172-31-91-146:/$ cd /etc/nginx/sites-available/
ubuntu@ip-172-31-91-146:/etc/nginx/sites-available$ pwd
/etc/nginx/sites-available
ubuntu@ip-172-31-91-146:/etc/nginx/sites-available$
```

- sudo nano default



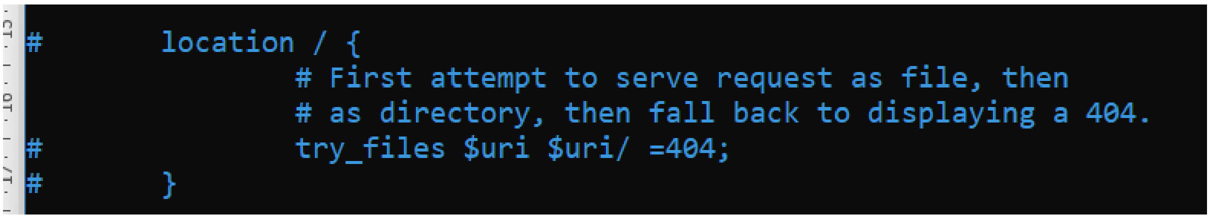
```
ubuntu@ip-172-31-91-146:/etc/nginx/sites-available$ sudo nano default
```

b) Now a PHP default code will open.


ubuntu@3.83.97.74:22 - Bitvise xterm - ubuntu@ip-172-31-91-146: /etc/nginx/sites-available

```
  GNU nano 6.2                                        default
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/Nginx/DirectoryStructure
#
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
#
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
        listen 80 default_server;
        listen [::]:80 default_server;

        # SSL configuration
        #
        # listen 443 ssl default_server;
        # listen [::]:443 ssl default_server;
        #
        # Note: You should disable gzip for SSL traffic.
        # See: https://bugs.debian.org/773332
                                         [ Read 91 lines ]
^G Help         ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Location     M-U Undo
^X Exit         ^R Read File    ^\ Replace      ^U Paste        ^J Justify      ^/ Go To Line   M-E Redo
```

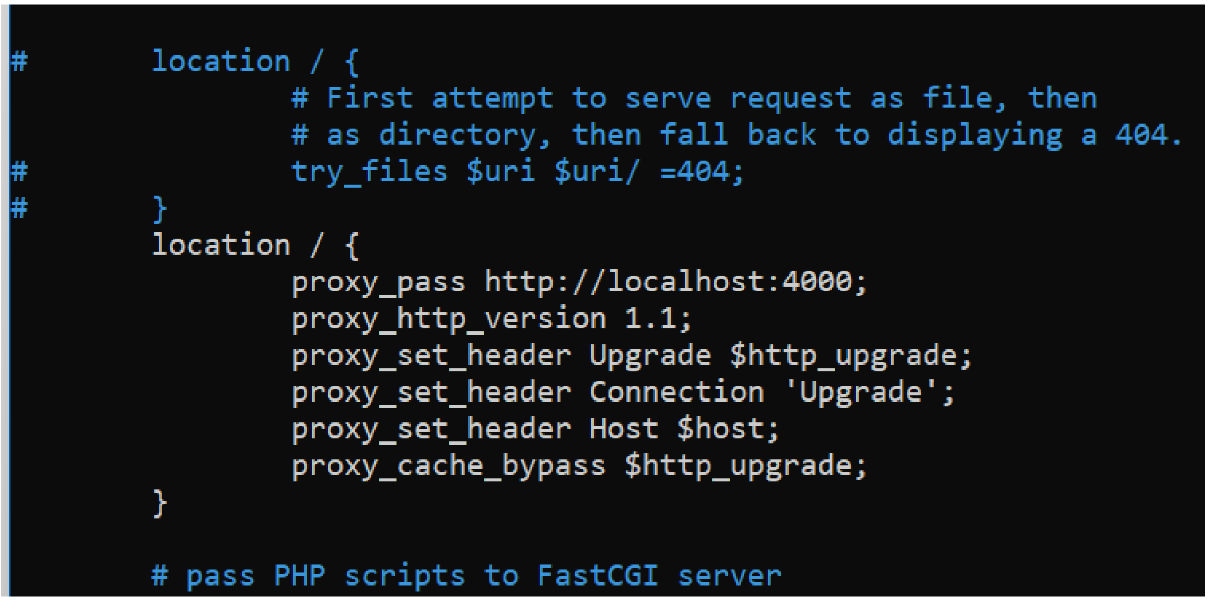
c) Go down until you see the “location” part of the code. Comment the three lines of that as shown in the image below.



```
#       location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
#               try_files $uri $uri/ =404;
#       }
```

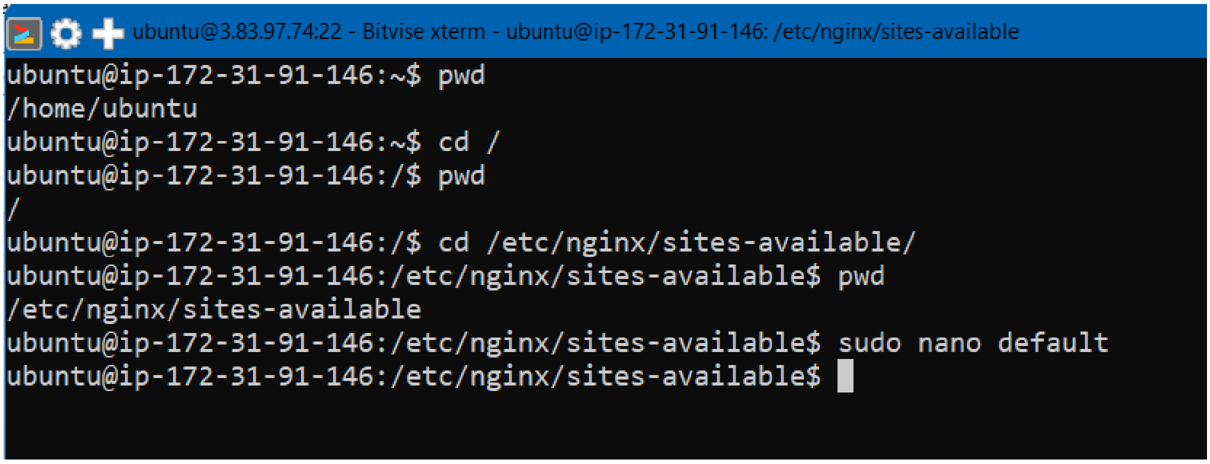
d) Then paste the location code (given below) under the hashed location part.

```
location / {
        proxy_pass http://localhost:4000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'Upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
 }
```



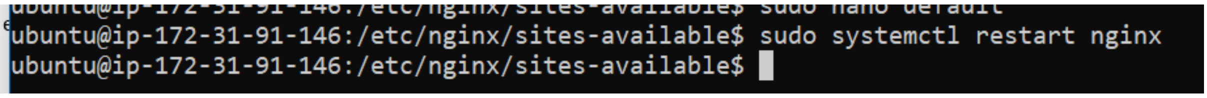
```
#       location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
#               try_files $uri $uri/ =404;
#       }
        location / {
                proxy_pass http://localhost:4000;
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection 'Upgrade';
                proxy_set_header Host $host;
                proxy_cache_bypass $http_upgrade;
        }

        # pass PHP scripts to FastCGI server
```

e) Next, Press “***Ctrl+X → Y →Enter***” respectively to exit and save your changes.


ubuntu@3.83.97.74:22 - Bitvise xterm - ubuntu@ip-172-31-91-146: /etc/nginx/sites-available

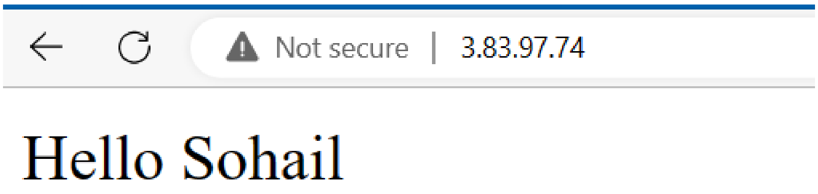
```
ubuntu@ip-172-31-91-146:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-91-146:~$ cd /
ubuntu@ip-172-31-91-146:/$ pwd
/
ubuntu@ip-172-31-91-146:/$ cd /etc/nginx/sites-available/
ubuntu@ip-172-31-91-146:/etc/nginx/sites-available$ pwd
/etc/nginx/sites-available
ubuntu@ip-172-31-91-146:/etc/nginx/sites-available$ sudo nano default
ubuntu@ip-172-31-91-146:/etc/nginx/sites-available$ 
```

f) Next, Enter the following command:
sudo systemctl restart nginx



```
ubuntu@ip-172-31-91-146:/etc/nginx/sites-available$ sudo nano default
ubuntu@ip-172-31-91-146:/etc/nginx/sites-available$ sudo systemctl restart nginx
ubuntu@ip-172-31-91-146:/etc/nginx/sites-available$ 
```

5. Now run the Public IPv4 Address in a web browser without using the port number.



Not secure | 3.83.97.74

Hello Sohail

Hence, we have successfully deployed the project without using Port number.