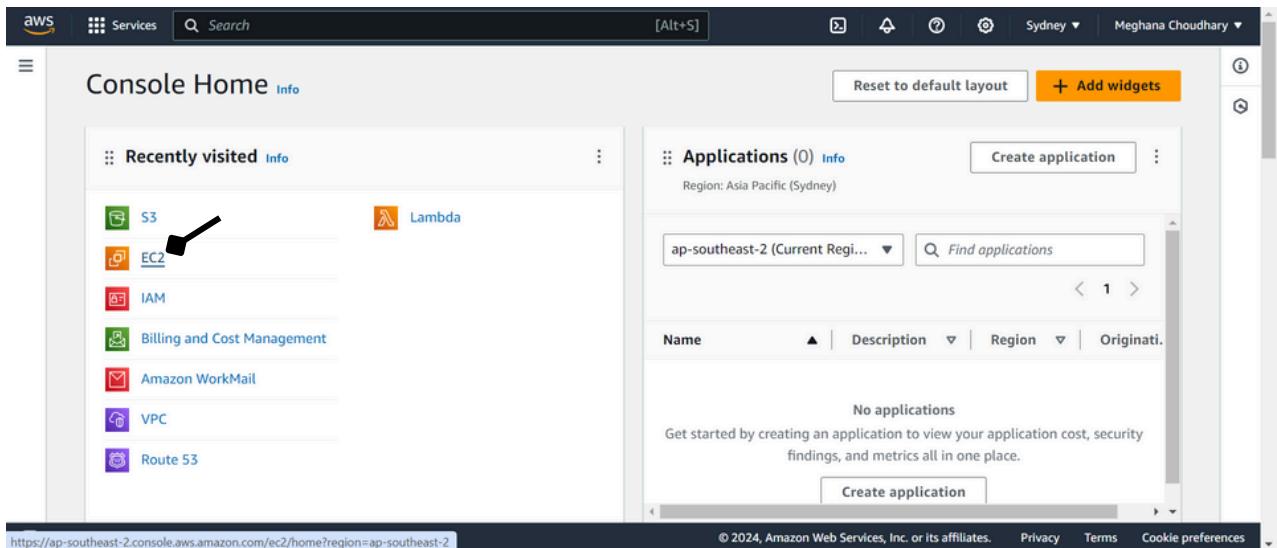


12. PROBLEM STATEMENT:

Deploy and run the project in AWS without using the port.

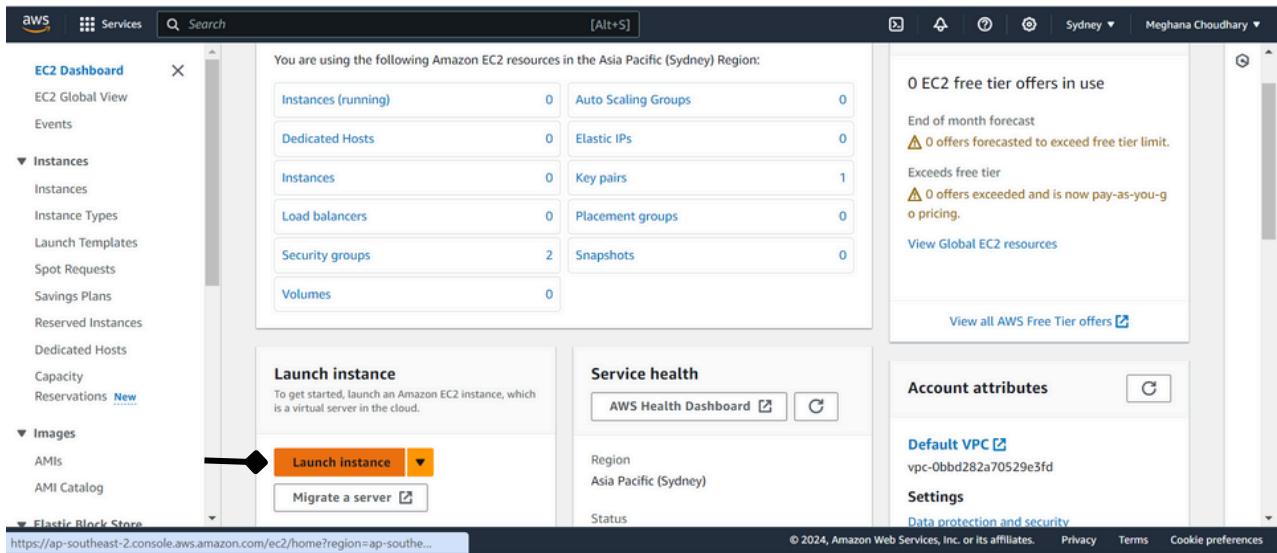
Creating a EC2 Instance-

1. Log into AWS console and click on “EC2”.



The screenshot shows the AWS Console Home page. On the left, there is a sidebar with 'Recently visited' services: S3, EC2 (which has a black arrow pointing to it), IAM, Billing and Cost Management, Amazon WorkMail, VPC, and Route 53. To the right, there is a section titled 'Applications (0)' with a 'Create application' button. Below this is a message: 'No applications. Get started by creating an application to view your application cost, security findings, and metrics all in one place.' At the bottom of the page, there is a footer with links for 'Privacy', 'Terms', and 'Cookie preferences'.

2. Click on ‘Launch instance’.



The screenshot shows the EC2 Dashboard. On the left, there is a sidebar with 'EC2 Dashboard' selected, followed by 'EC2 Global View', 'Events', 'Instances' (with sub-options: Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, Reservations), 'Images' (with sub-options: AMIs, AMI Catalog), and 'Elastic Block Store'. The main area displays statistics for running instances, Auto Scaling Groups, Dedicated Hosts, Elastic IPs, Key pairs, Load balancers, Placement groups, Security groups, and Snapshots. Below this is a 'Launch instance' section with a 'Launch instance' button (which has a black arrow pointing to it) and a 'Migrate a server' button. To the right, there is a 'Service health' section with a 'AWS Health Dashboard' link, and 'Account attributes' sections for 'Default VPC' (vpc-0bbd282a70529e3fd) and 'Settings' (Data protection and security). The footer includes links for 'Privacy', 'Terms', and 'Cookie preferences'.

3. Fill the required details.

Give a name to your server and choose ubuntu Server as the OS Image.

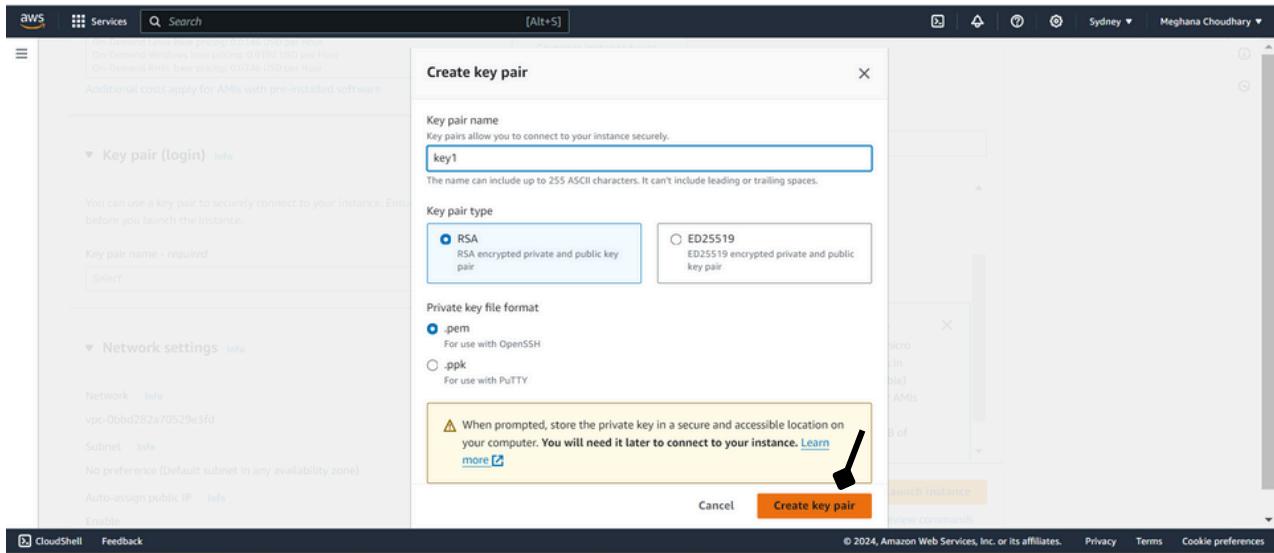
The screenshot shows the 'Launch an instance' wizard. In the 'Name and tags' section, the 'Name' field contains 'nginx-server'. In the 'Application and OS Images (Amazon Machine Image)' section, the 'Software Image (AMI)' dropdown is set to 'Canonical, Ubuntu, 22.04 LTS'. The 'Virtual server type (instance type)' is 't2.micro'. The 'Firewall (security group)' is 'New security group'. The 'Number of instances' is set to 1. At the bottom right, there are 'Cancel', 'Launch instance', and 'Review commands' buttons.

This screenshot shows the 'Application and OS Images (Amazon Machine Image)' step. It lists various operating systems: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux. A search bar at the top says 'Search our full catalog including 1000s of application and OS images'. Below the search bar is a 'Quick Start' section with icons for each system. A specific AMI is selected: 'Ubuntu Server 22.04 LTS (HVM), SSD Volume Type'. The description below it states: 'Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2024-03-01'. The 'Architecture' is '64-bit (x86)'. The 'AMI ID' is 'ami-09c8d5d747253fb7a'. A 'Verified provider' button is present. On the right side, the 'Summary' section shows the same configuration as the previous screenshot: 1 instance of Canonical, Ubuntu, 22.04 LTS, t2.micro, New security group, and a storage volume of 8 GiB. A tooltip for the 'Free tier' is visible, stating: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable)'. The 'Launch instance' button is at the bottom right.

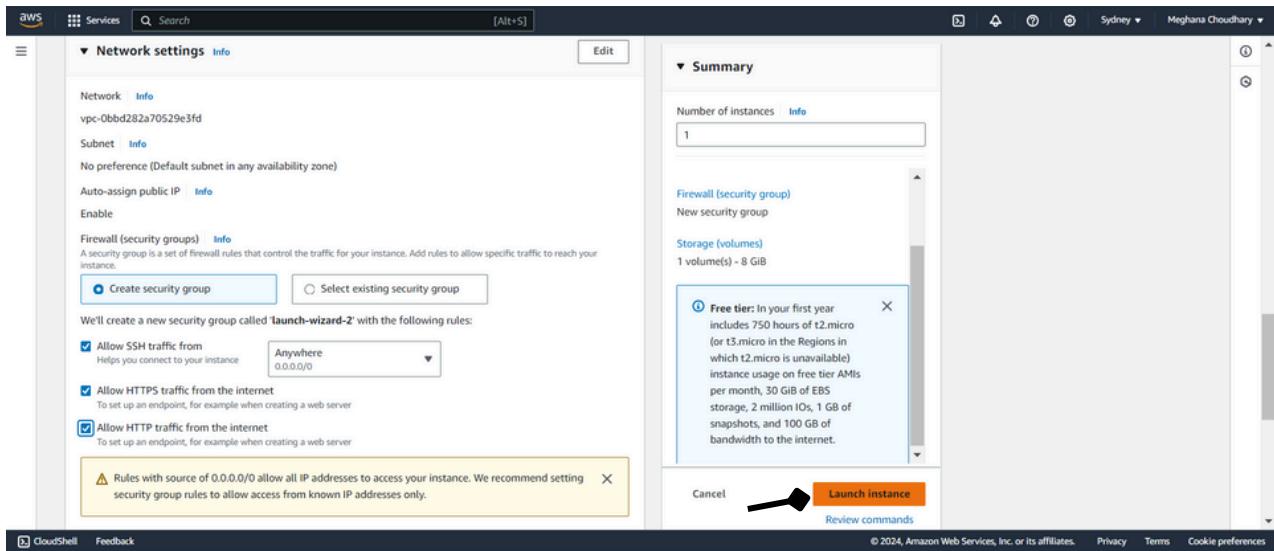
4. For Key pair, if you have a existing key pair select that else 'Create e new key pair' .

The screenshot shows the 'Key pair (login)' step. It asks, 'Can you 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.' A dropdown menu for 'Key pair name - required' has 'Select' as the current option. To the right, there are sections for 'Firewall (security group)' (New security group), 'Storage (volumes)' (1 volume(s) - 8 GiB), and another 'Free tier' tooltip for t2.micro. The 'Launch instance' button is at the bottom right.

5. Give a name to the key pair. Make sure the key pair type is 'RSA' and the file extension is '.pem'.



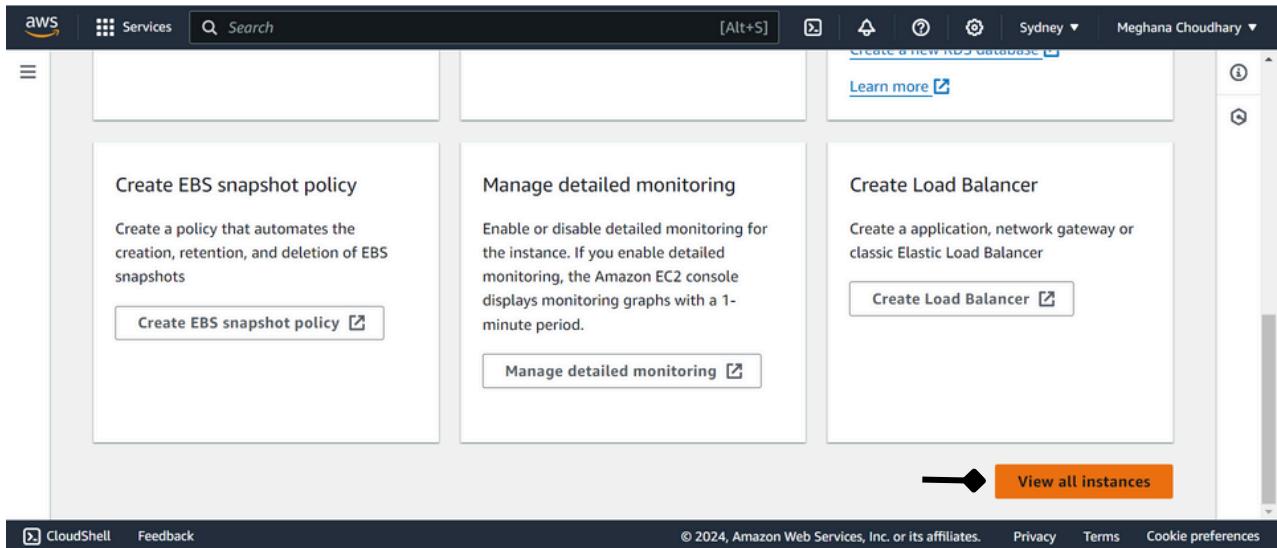
6. For 'Network Settings' Select all the check boxes(SSH, HTTPS,HTTP). Then Click on 'Launch instance'.



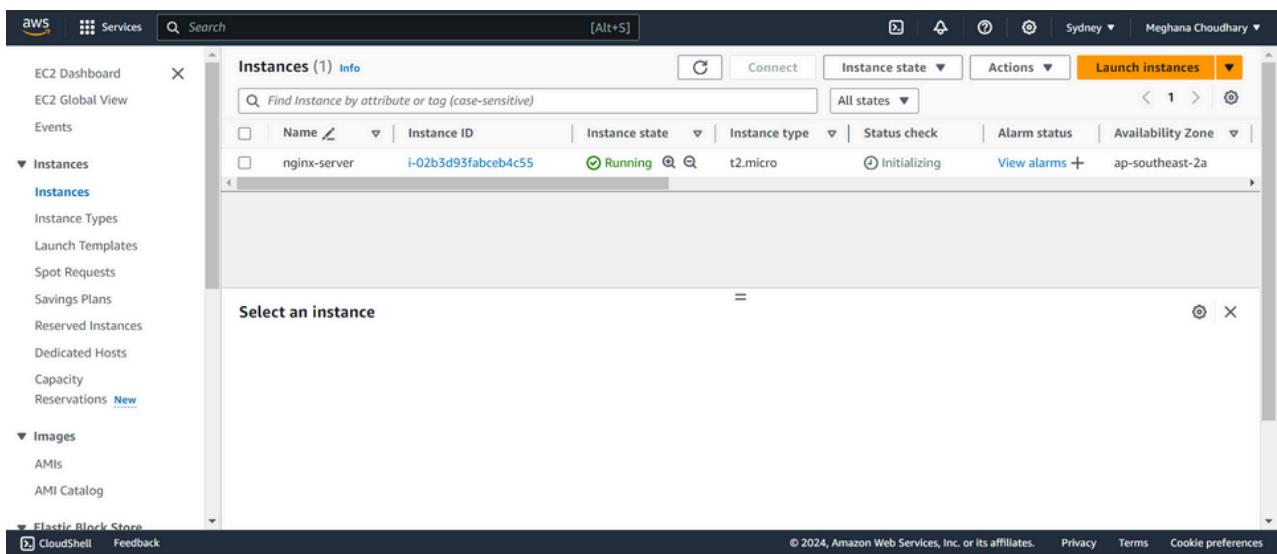
7. Your instance is created successfully.



8. Then Scroll Down and Click on 'View all instances'.



9. You can see your instance is created and running.



Connecting to the Instance with Bitvise SSH and Routing the website running in port 4000 through port 80 with Nginx:-

1. Log into AWS console, search for EC2 and open your EC2 dashboard and click on 'Instances'.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with options like EC2 Dashboard, Services, and a search bar. Under 'Instances', there are several sub-options: Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, and Reservations. The 'Instances' option is selected. In the main content area, there's a 'Resources' summary table and a 'Launch instance' button. To the right, there's a 'EC2 Free Tier' info panel and an 'Account attributes' section. The URL in the address bar is <https://ap-southeast-2.console.aws.amazon.com/ec2/home?region=ap-southeast-2#Instances:instanceState=running>.

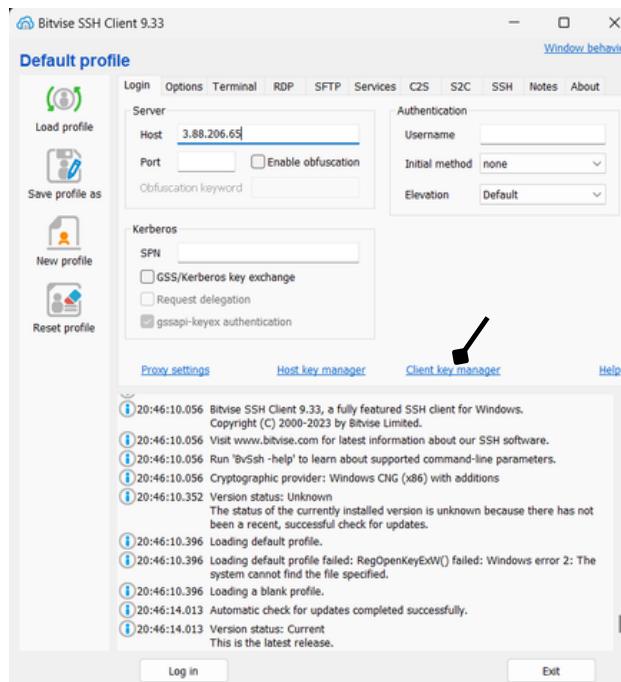
2. Click on the 'Instance ID' of your instance.

The screenshot shows the 'Instances' page with one instance listed. The instance is named 'nginx-server' and has an ID of 'i-02b3d93fabceb4c55'. It is currently 'Running'. A black arrow points to the instance ID 'i-02b3d93fabceb4c55'. The URL in the address bar is <https://ap-southeast-2.console.aws.amazon.com/ec2/instances?region=ap-southeast-2>.

3. Copy the Public IPV4 address of your instance.

The screenshot shows the 'Instance summary' page for the instance 'i-02b3d93fabceb4c55'. The instance is named 'nginx-server'. The 'Public IPv4 address' is listed as '3.27.226.106'. A black arrow points to this address. Other details shown include the instance state ('Running'), IP name ('ip-172-31-5-56.ap-southeast-2.compute.internal'), and VPC ID ('vpc-0bbd282a70529e3fd'). The URL in the address bar is <https://3.27.226.106/>.

4. In Bitvise SSH Client, paste the ‘Public IPV4 address’ and click on ‘Client key Manager’.



5. Under ‘Client key manager’, if there is any existing key then remove it and click on ‘Import’, select the same key that you used while creating the instance, from your file directory and then again click on ‘Import’ and then the key is successfully imported.

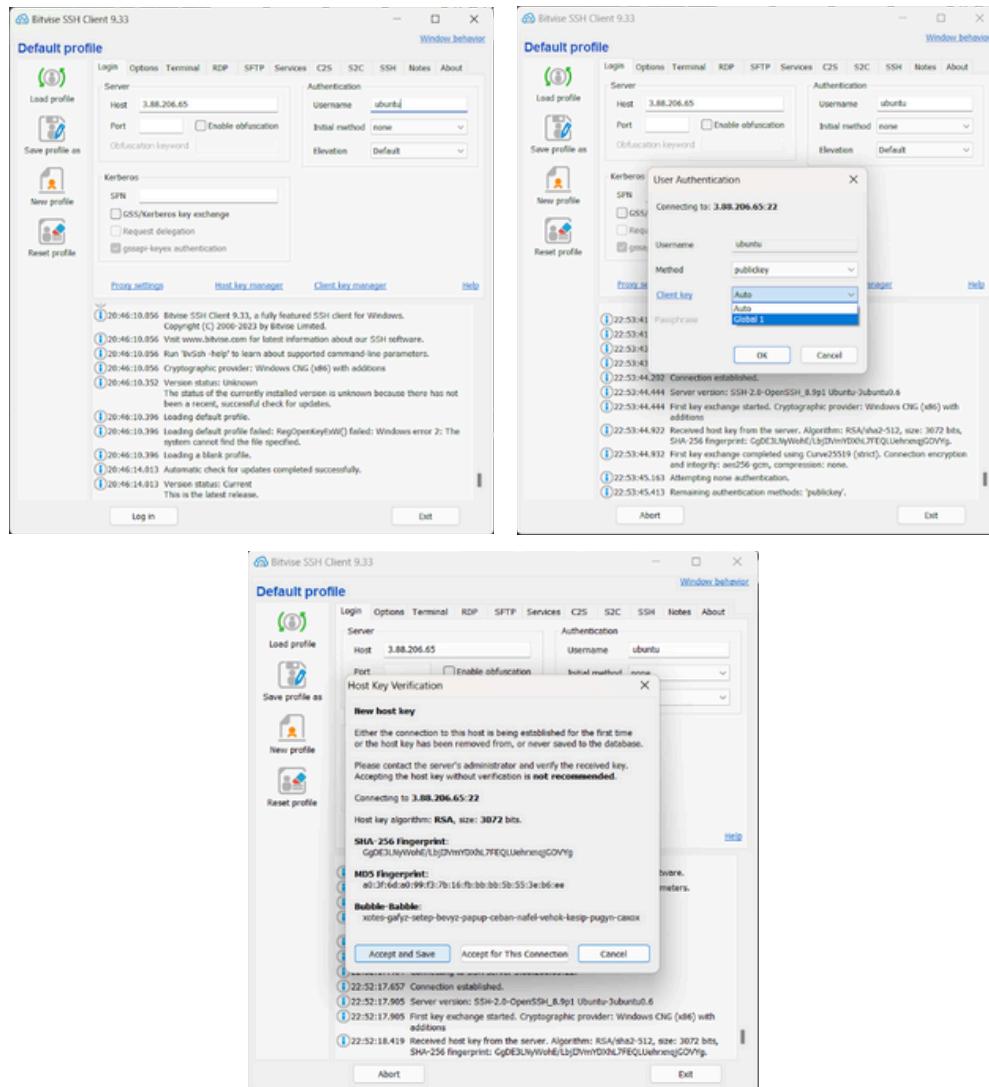
Import Client Key Dialog (Top Screenshot):

Location	Algorithm	Size	Passph...	SHA-256 Fingerprint	MDS Fingerprint	Bubble Babble	Comment
Global	RSA	2048	no	1b21BZ0264RywAtezBgrB6+hudSN/IchJH9Zu1N/zsU	33:11:97:ba:3b:0d:04:4e:e4:9f:dc:de:07:a2:7f:77	xogem-gonis-tylyl-canat-tynam-kyzuk-potuh-fyzod-femok-nilek-lolex	

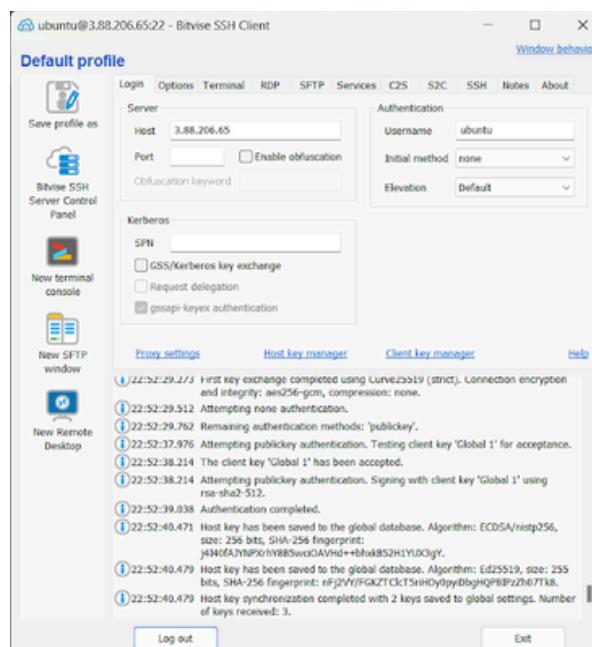
Main Client Key Manager Window (Bottom Screenshot):

Location	Algorithm	Size	Passph...	SHA-256 Fingerprint	MDS Fingerprint	Bubble Babble	Comment
Global 1	RSA	2048	no	1b21BZ0264RywAtezBgrB6+hudSN/IchJH9Zu1N/zsU	33:11:97:ba:3b:0d:04:4e:e4:9f:dc:de:07:a2:7f:77	xogem-gonis-tylyl-canat-tynam-kyzuk-potuh-fyzod-femok-nilek-lolex	

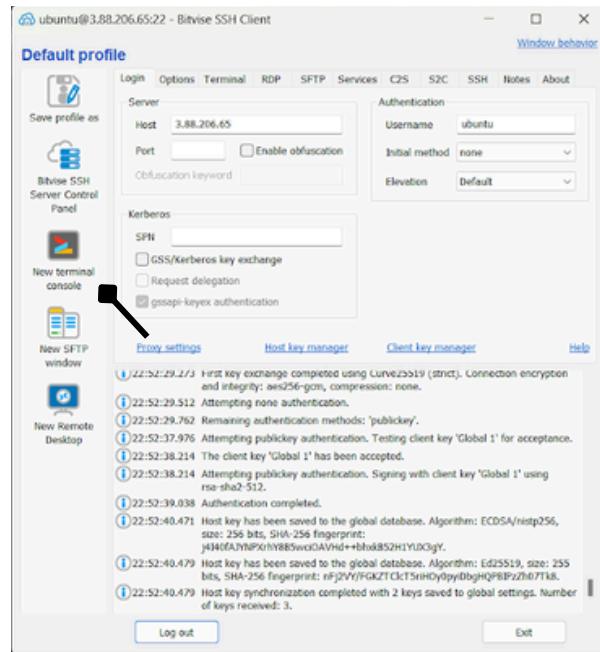
6. In ‘Bitvise SSH Client’, under ‘Default profile’ give the ‘Username’ as ‘ubuntu’ then select ‘Global1’ from ‘Client Key Manager’ then click on ‘Log in’ & ‘Accept and save’.



If the ‘Log In’ button changes to ‘Log Out’, then you are logged in successfully.



7. Click on ‘New terminal consol’, to access the terminal of your instance.



8. Once you are inside the teminal of your instance run
To update the system packages and upgrade them to the latest version.

```
sudo apt update -y && sudo apt upgrade -y
```

```
ubuntu@ip-172-31-10-86:~$ sudo apt update -y && sudo apt upgrade -y
```

Download Node.js from the source

```
curl -fsSL https://deb.nodesource.com/setup_current.x | sudo bash -  
ubuntu@ip-172-31-2-183:~$ curl -fsSL https://deb.nodesource.com/setup_current.x | sudo bash -
```

Install Nginx, Git, Node.js with

```
sudo apt install git nginx nodejs -y
```

```
ubuntu@ip-172-31-2-183:~$ sudo apt install git nginx nodejs -y
```

Start and Enable the Nginx Service

```
sudo systemctl start nginx && sudo systemctl enable nginx
```

```
ubuntu@ip-172-31-10-86:~$ sudo systemctl start nginx && sudo systemctl enable nginx
```

Clone the Github Repository

```
git clone https://github.com/meghana-choudhary/aws_proj.git
```

```
ubuntu@ip-172-31-4-9:~$ git clone https://github.com/meghana-choudhary/aws_proj.git
```

Edit the Configuration file for Nginx with a text editor(Nano or Vim)

```
sudo vim /etc/nginx/sites-available/default
```

```
ubuntu@ip-172-31-41-206:~$ sudo vim /etc/nginx/sites-available/default
```

Scroll down to the location section in the file, comment out the existing code and add the following code-

```
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;  
}
```

```
7  
6 #     location / {  
5 #         # First attempt to serve request as file, then  
4 #         # as directory, then fall back to displaying a 404.  
3 #             try_files $uri $uri/ =404;  
2 #         }  
1  
54     location / {  
1         proxy_pass http://localhost:4000;  
2         proxy_http_version 1.1;  
3         proxy_set_header Upgrade $http_upgrade;  
4         proxy_set_header Connection 'upgrade';  
5         proxy_set_header Host $host;  
6         proxy_cache_bypass $http_upgrade;  
7     }  
8
```

Go to the repository folder.

```
cd aws_proj
```

```
ubuntu@ip-172-31-4-9:~/aws_proj$ cd aws_proj
```

Install Dependencies described in the package.json

```
npm install
```

```
ubuntu@ip-172-31-4-9:~/aws_proj$ npm install
```

Start the Node server

```
node index.js
```

```
ubuntu@ip-172-31-4-9:~/aws_proj$ node index.js
```

9. Now pasting the Public IPV4 address of your EC2 instance in a web browser gets us to our website.

