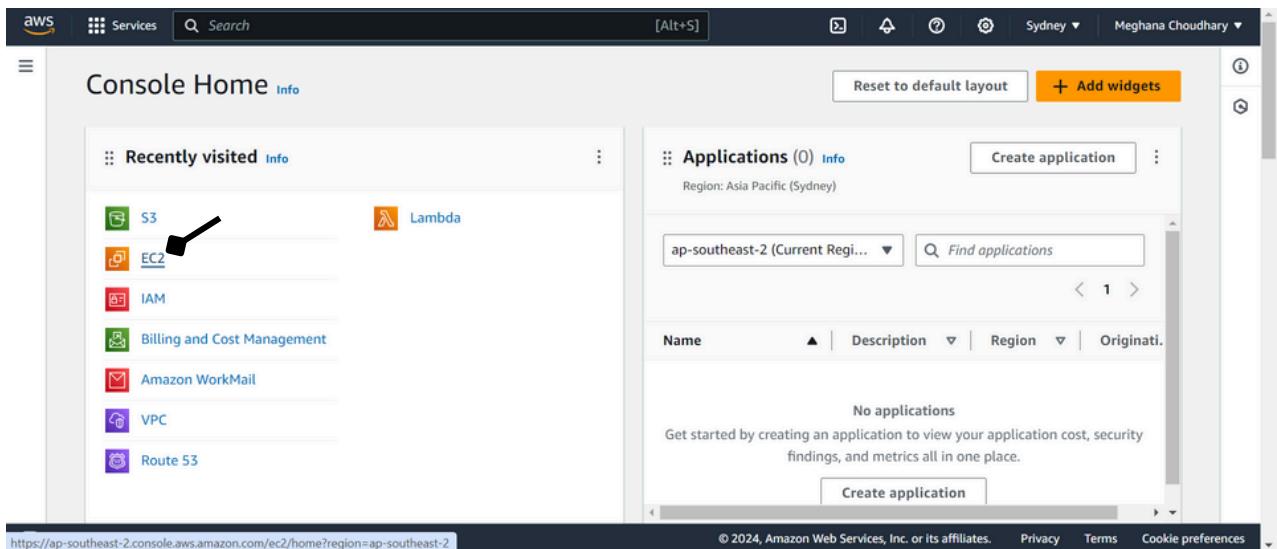


9. PROBLEM STATEMENT:

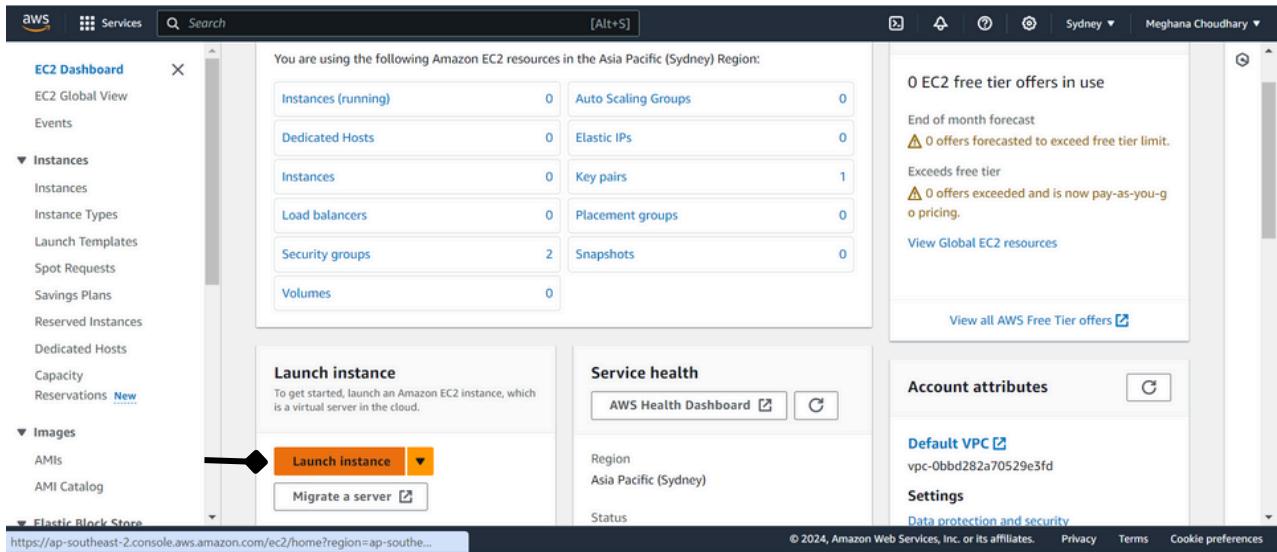
Deploy a project from GitHub to EC2

Creating a EC2 Instance with port 4000 open:-

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



2. Click on ‘Launch instance’.



3. Fill the required details.

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

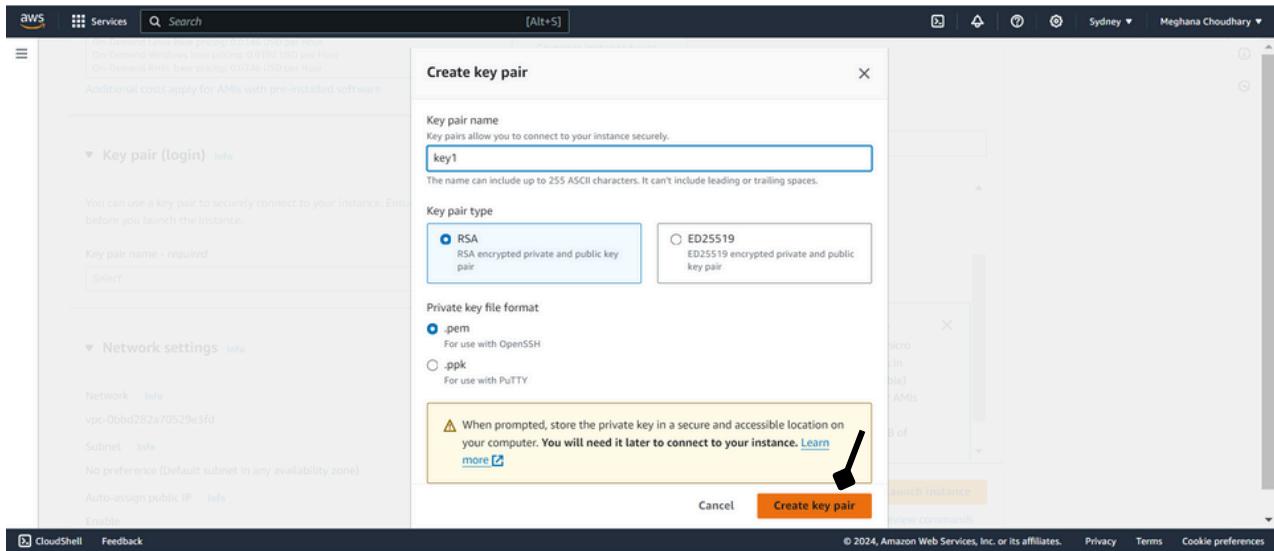
The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Name and tags' step, the 'Name' field contains 'nginx-server'. In the 'Application and OS Images (Amazon Machine Image)' step, the 'Software Image (AMI)' dropdown is set to 'Canonical, Ubuntu, 22.04 LTS'. Other settings include 'Virtual server type (instance type)' as 't2.micro', 'Firewall (security group)' as 'New security group', and 'Number of instances' as '1'. A 'Launch instance' button is visible at the bottom right.

The second part of the screenshot shows the 'Application and OS Images (Amazon Machine Image)' search results for 'Ubuntu Server 22.04 LTS (HVM, SSD Volume Type)'. It lists the AMI ID as ami-09c8d5d747253fb7a, architecture as '64-bit (x86)', and an option to 'Create new key pair'.

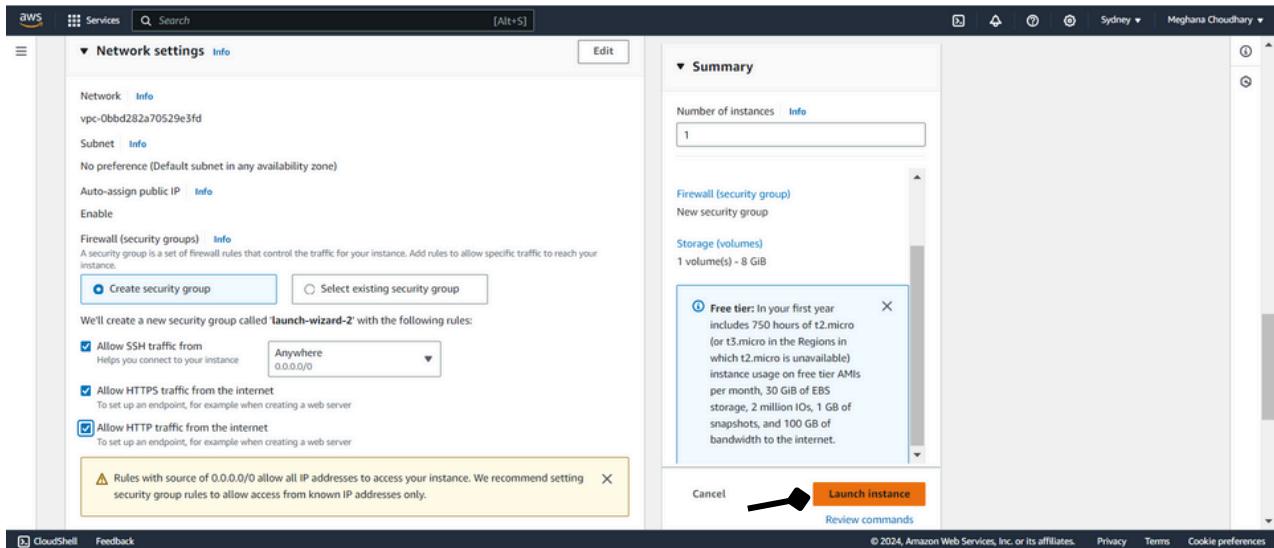
4. For Key pair, if you have a existing key pair select that else ‘Create e new key pair’.

The screenshot shows the 'Key pair (login)' step. The 'Key pair name - required' dropdown is set to 'Select'. A 'Create new key pair' button is located below the dropdown. To the right, there are sections for 'Firewall (security group)' (set to 'New security group'), 'Storage (volumes)' (set to '1 volume(s) - 8 GiB'), and a 'Free tier' information box.

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



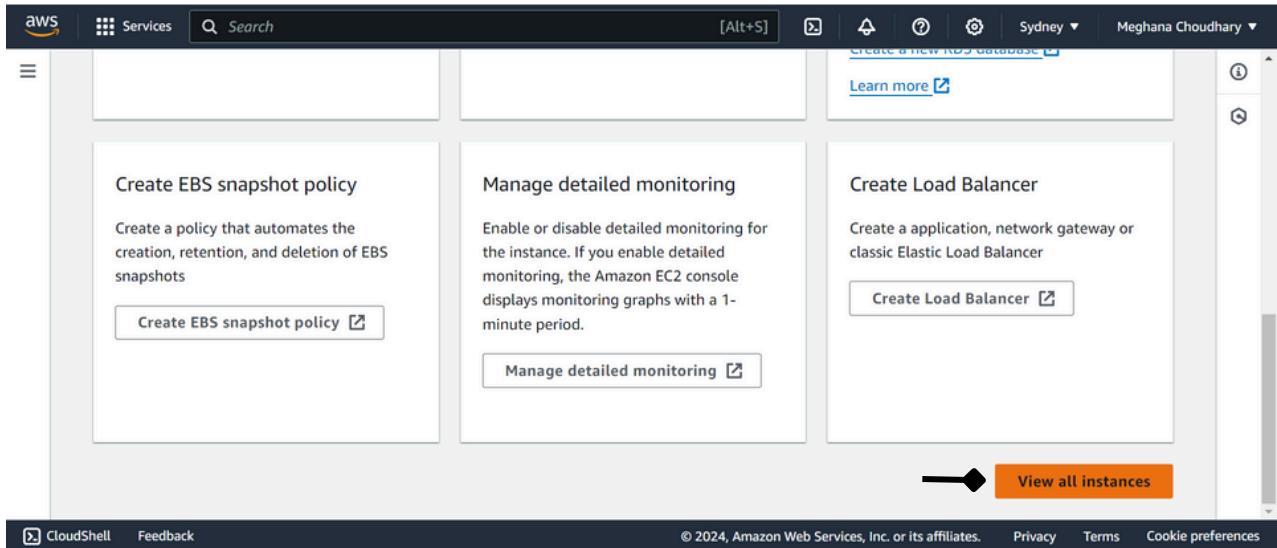
6. For 'Network Settings' Select all the check boxes(SSH, HTTPS,HTTP). Then Click on ‘Launch instance’.



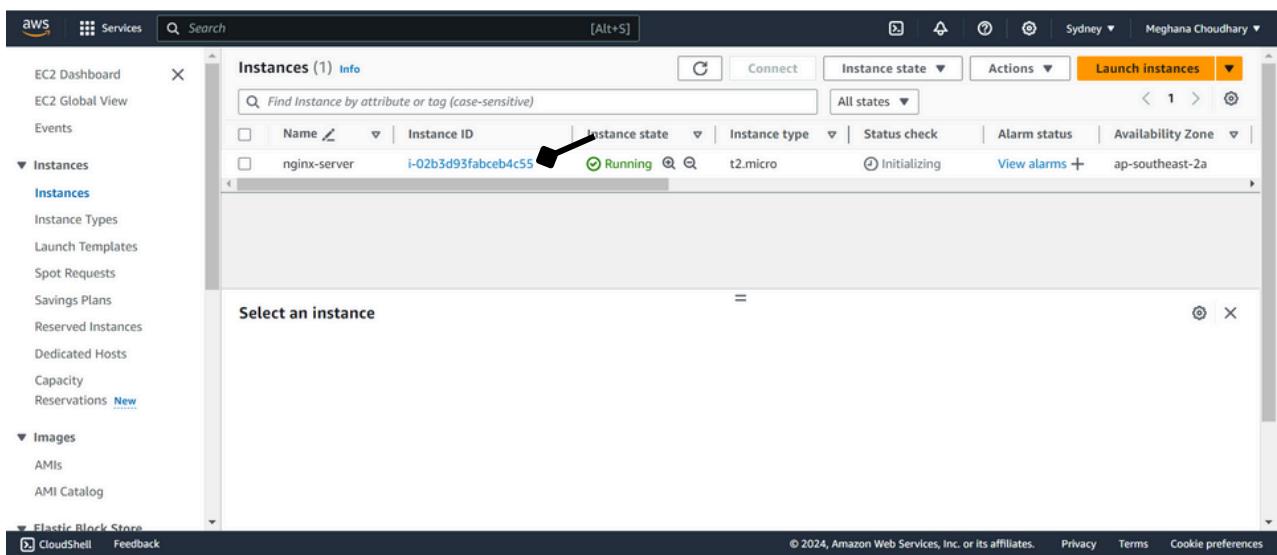
7. Your instance is created successfully.



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



9. You can see your instance is created and running. Click on the server Instance ID



10. Scroll Down, Go to ‘Security’ then ‘Security groups’

The screenshot shows the AWS EC2 Dashboard. The left sidebar includes sections for EC2 Dashboard, EC2 Global View, Events, Instances (with sub-options like Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots). The main content area has tabs for Details, Status and alarms, Monitoring, Security (which is highlighted with a black arrow), Networking, Storage, and Tags. Under the Security tab, there's a 'Security details' section showing IAM Role (None), Owner ID (381491890643), and Launch time (Wed Apr 24 2024 04:22:10 GMT+0530 (India Standard Time)). Below that is a 'Security groups' section with a list containing 'sg-07285c04c2c0c7033 (launch-wizard-1)'. Further down are sections for Inbound rules and Outbound rules, each with a filter input field and a table of rules.

11. Now ‘Edit inbound rules’

The screenshot shows the EC2 Security Groups page for the security group 'sg-07285c04c2c0c7033 - launch-wizard-1'. The left sidebar is identical to the previous screenshot. The main content area shows the security group details: name (sg-07285c04c2c0c7033), owner (381491890643), and VPC ID (vpc-07d67f2ecedab99bd). Below the details are tabs for Inbound rules (which is highlighted with a black arrow), Outbound rules, and Tags. The Inbound rules section displays three entries with columns for Name, Security group rule ID, Port range, Protocol, and Source. There are also buttons for C (Create), Manage tags, and Edit inbound rules.

12. Add a Rule allowing traffic on port 4000 from anywhere.

The screenshot shows the AWS Management Console with the Services menu selected. The main area displays the 'Inbound rules' for a security group. A new rule is being added at the bottom of the list:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0b40e85fe84d3df87	HTTP	TCP	80	Cu... ▾	0.0.0.0/0 X
sgr-0740233220d1df5b4	HTTPS	TCP	443	Cu... ▾	0.0.0.0/0 X
sgr-00ae4727950be6fcfd	SSH	TCP	22	Cu... ▾	0.0.0.0/0 X
-	Custom TCP	TCP	4000	An... ▾	0.0.0.0/0 X

A modal dialog is open for the new rule, showing the configuration: Type: Custom TCP, Protocol: TCP, Port range: 4000, Source: Anywhere (0.0.0.0/0). The 'Add rule' button is visible at the bottom left of the list.

13. Then 'Save rules'.

The screenshot shows the same AWS Security Groups page after saving the changes. A yellow warning message is displayed: "⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only." A large black arrow points to the 'Save rules' button at the bottom right of the page.

You'll Get a confirmation message.

The screenshot shows the EC2 Dashboard with a confirmation message: "Inbound security group rules successfully modified on security group (sg-07285c04c2c0c7033 | launch-wizard-1)". The message includes a link to the 'Details' page for the security group.

The 'Details' page for the security group 'sg-07285c04c2c0c7033 - launch-wizard-1' is shown. The page lists the following details:

Security group name	Security group ID	Description	VPC ID
launch-wizard-1	sg-07285c04c2c0c7033	launch-wizard-1 created 2024-01-27T20:54:56Z	vpc-023456789012345678

Connecting to the Instance with Bitvise SSH and deploying a project from Github:-

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' section with information about offers and a 'View Global EC2 resources' link. At the bottom, there's a URL bar with the address https://ap-southeast-2.console.aws.amazon.com/ec2/home?region=ap-southeast-2#Instances:instanceState=running.

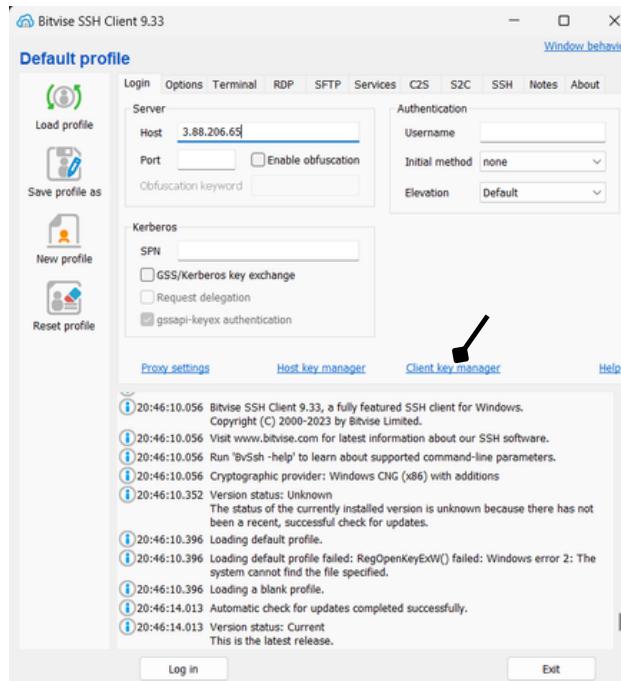
2. Click on the 'Instance ID' of your instance.

The screenshot shows the 'Instances (1)' page. The sidebar is identical to the previous dashboard. The main table shows one instance named 'nginx-server' with the instance ID 'i-02b3d93fabceb4c55'. A magnifying glass icon points to the instance ID. The instance status is 'Running'. There are buttons for 'Connect', 'Actions', and 'Launch instances' at the top of the table.

3. Copy the Public IPV4 address of your instance.

The screenshot shows the 'Instance summary for i-02b3d93fabceb4c55 (nginx-server)' page. The sidebar is the same. The main content includes sections for Instance ID (i-02b3d93fabceb4c55), IPv6 address (empty), Hostname type (IP name: ip-172-31-5-56.ap-southeast-2.compute.internal), Answer private resource DNS name (IPv4 (A)), Auto-assigned IP address (3.27.226.106 [Public IP]), IAM Role (empty), Public IPv4 address (3.27.226.106), Instance state (Running), Private IP DNS name (ip-172-31-5-56.ap-southeast-2.compute.internal), Instance type (t2.micro), VPC ID (vpc-0bbd282a70529e3fd), Subnet ID (empty), and Auto Scaling Group name (empty). A magnifying glass icon points to the Public IPv4 address field. The URL in the address bar is https://3.27.226.106/feedback.

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.

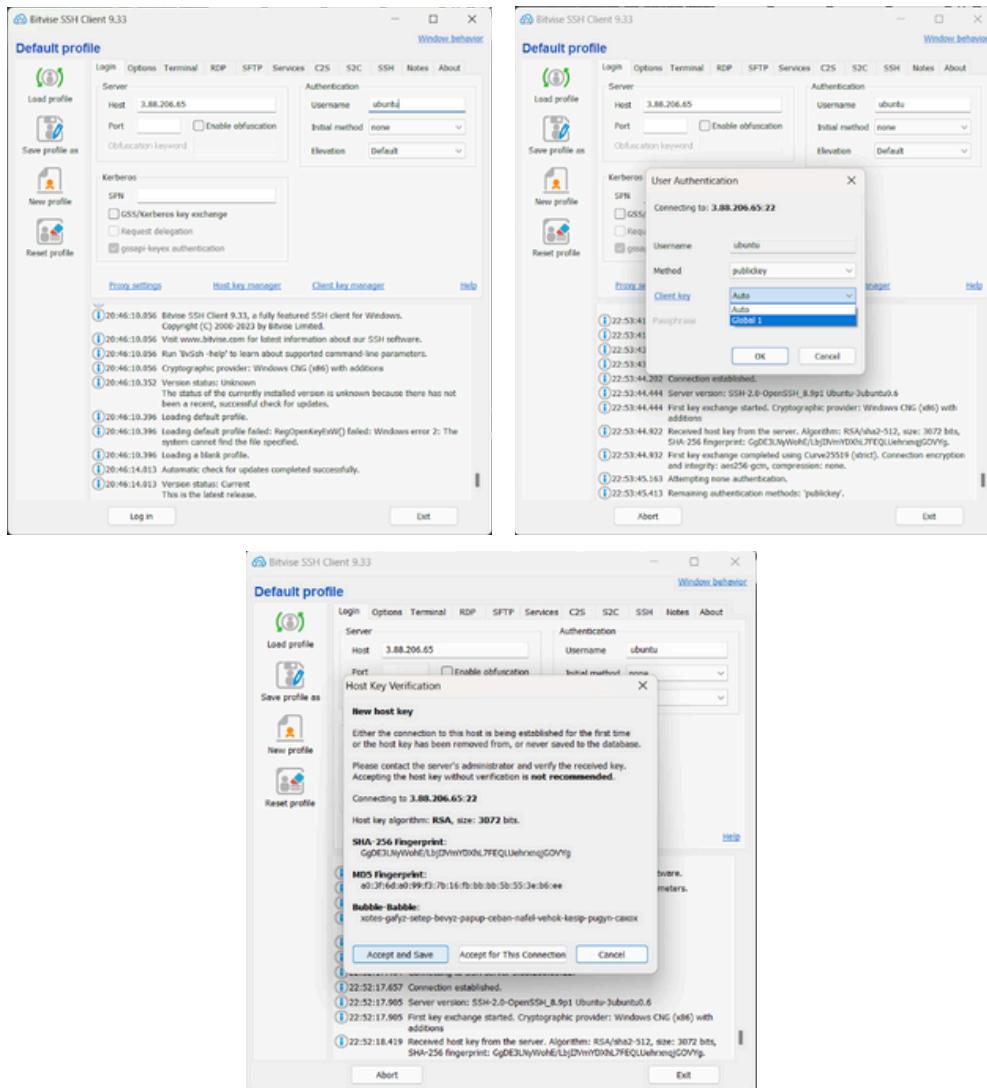
Import Client Key Dialog (Top Screenshot):

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-tyl-canat-tynam-kyzuk-potuh-fyzod-femok-nilek-loxex	

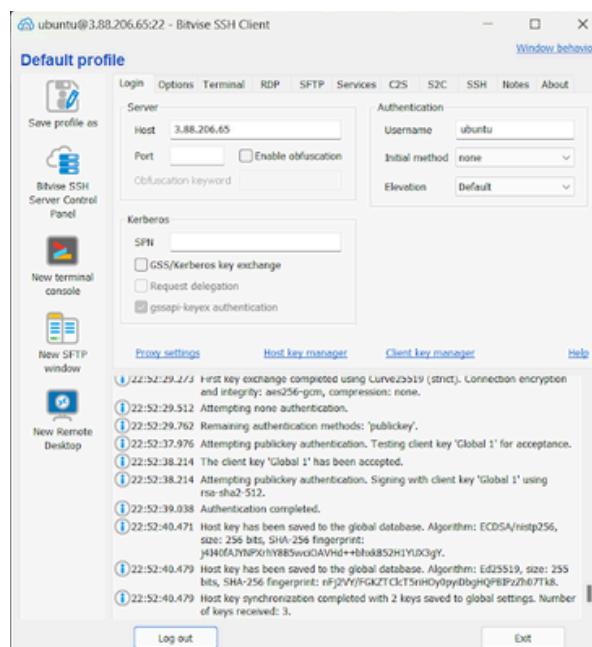
Main Client Key Manager Window (Bottom Screenshot):

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-tyl-canat-tynam-kyzuk-potuh-fyzod-femok-nilek-loxex	

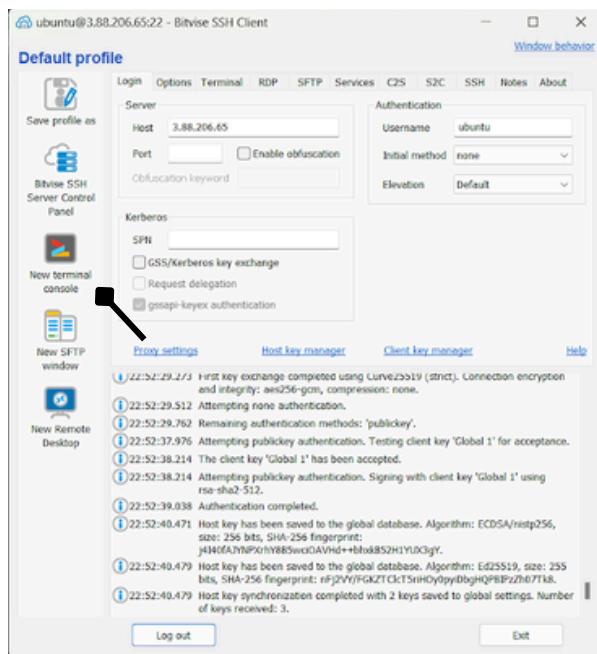
6. In ‘Bitvise SSH Client’, under ‘Default profile’ give the ‘Username’ as ‘ubuntu’ then select ‘Global1’ from ‘Client Key Manager’ then click on ‘Log in’ & ‘Accept and save’.



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
```

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
```

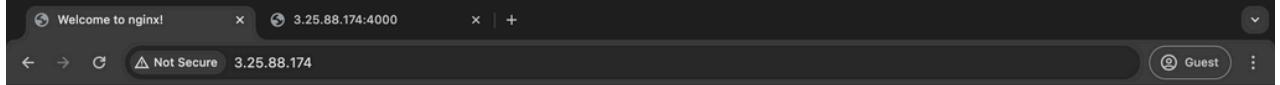
```
ubuntu@ip-172-31-4-9:~/aws_proj$ npm install
```

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 the Nginx default welcome page, through the default HTTP port, port 80.



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.

Adding :4000 after the IPV4 address, we get to our Node.js server through our specified port 4000.

