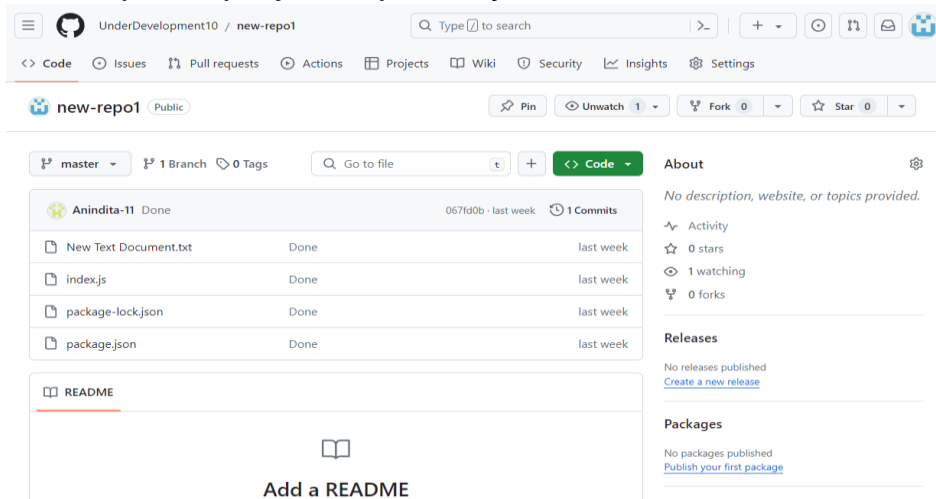


Assignment – 9

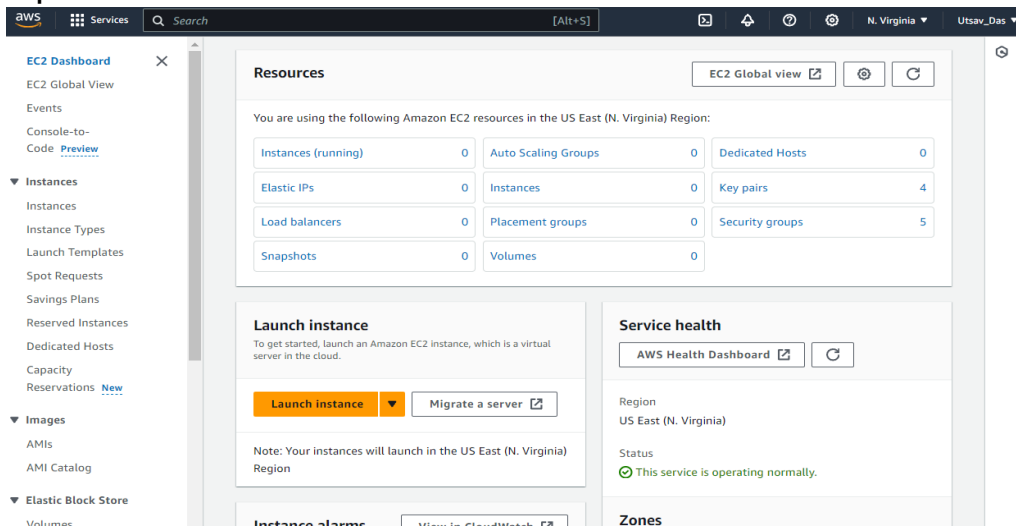
Problem Statement:

Deploy a project from GitHub on EC2.

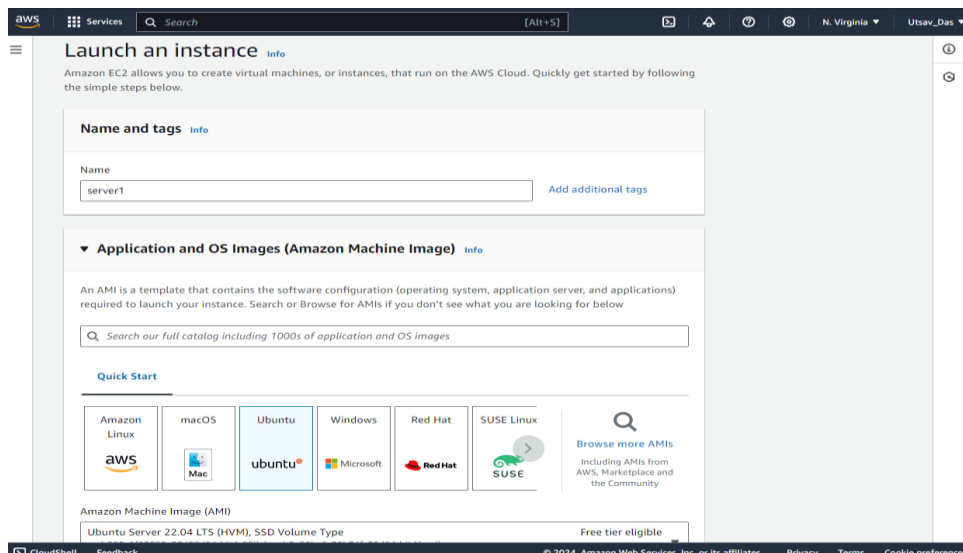
1) First, upload project repository to Github.



2) Open EC2 and click on Launch Instance.



3) Give the name of server and select Ubuntu application and OS image inside Quick Start.



- 4) As we have pre-existing key pair, select the key pair from drop down menu.

The screenshot shows the AWS Management Console interface. At the top, there's a navigation bar with the AWS logo, 'Services', a search bar, and a user profile 'Utsav_Das'. Below the navigation bar, there's a section titled 'Additional costs apply for AMIs with pre-installed software'. The main content area is divided into two sections: 'Key pair (login)' and 'Network settings'. The 'Key pair (login)' section has a description: '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.' It features a dropdown menu for 'Key pair name - required' with 'key3' selected, and a 'Create new key pair' button. The 'Network settings' section has an 'Edit' button and shows the following configuration: Network: vpc-05e13defc1bb34724, Subnet: No preference (Default subnet in any availability zone), Auto-assign public IP: Enable, and Firewall (security groups): No preference (Default security group in any availability zone). A note states: 'Additional charges apply when outside of free tier allowance'.

- 5) Select Create security group and click all checkboxes below it. Then, click on Launch Instance.

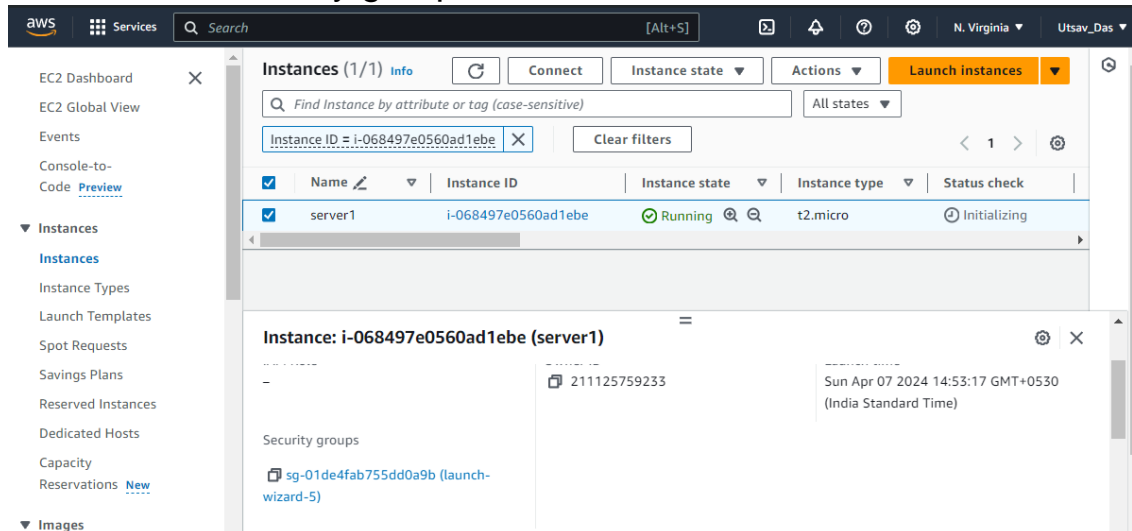
The screenshot shows the AWS Management Console interface. The 'Network settings' section is on the left, and the 'Summary' section is on the right. The 'Network settings' section has an 'Edit' button and shows the following configuration: Network: vpc-05e13defc1bb34724, Subnet: No preference (Default subnet in any availability zone), Auto-assign public IP: Enable, and Firewall (security groups): No preference (Default security group in any availability zone). A note states: 'Additional charges apply when outside of free tier allowance'. Below this, there's a section for 'Firewall (security groups)' with a description: '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.' It features two radio buttons: 'Create security group' (selected) and 'Select existing security group'. Below the radio buttons, there's a section for 'We'll create a new security group called 'launch-wizard-5' with the following rules:'. It has three checkboxes: 'Allow SSH traffic from' (checked), 'Allow HTTPS traffic from the internet' (checked), and 'Allow HTTP traffic from the internet' (checked). The 'Summary' section shows the following configuration: Number of instances: 1, Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ...read more, Virtual server type (instance type): t2.micro, Firewall (security group): New security group, and Storage (volumes): 1 volume(s) - 8 GiB. A 'Launch instance' button is at the bottom right. A 'Free tier' notification box is also 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) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million ...'.

- 6) Go to the Github project and then to index.html, where you can see 4000 in app.listen() which is port no that will be added to custom TCP.

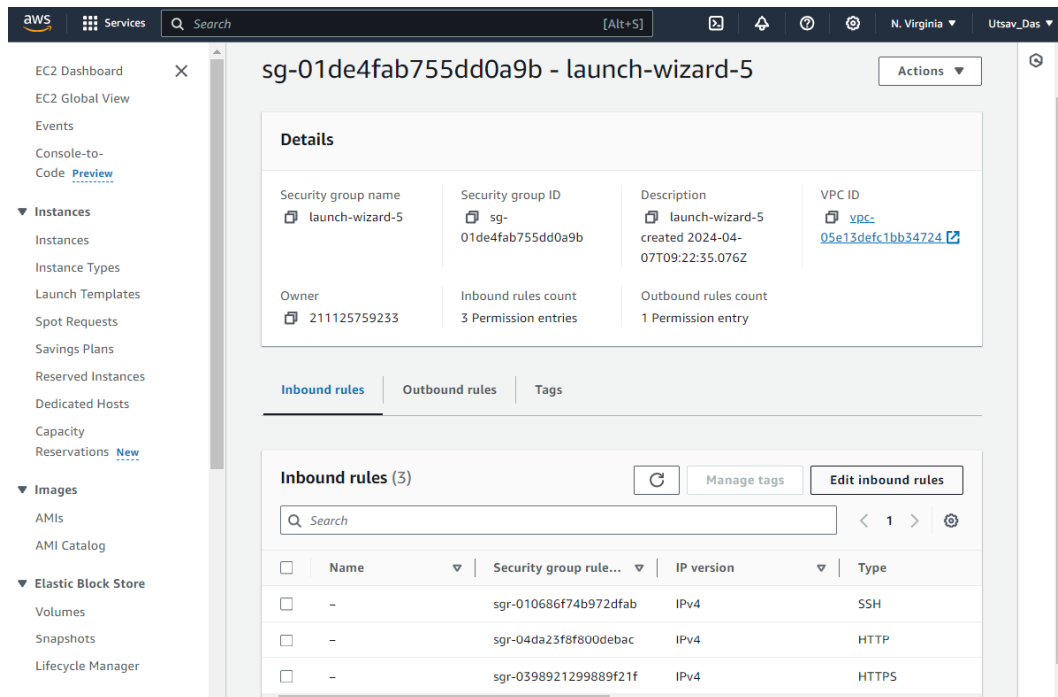
The screenshot shows the Github repository 'new-repo1' by 'UnderDevelopment10'. The 'Code' tab is selected, and the 'index.js' file is open. The file content is as follows:

```
1 const express = require('express')
2 const app = express()
3
4 app.get('/', function (req, res) {
5   res.send('Hello World')
6 })
7
8 app.listen(4000, ()=>{
9   console.log("Started server");
10 })
11 )
```

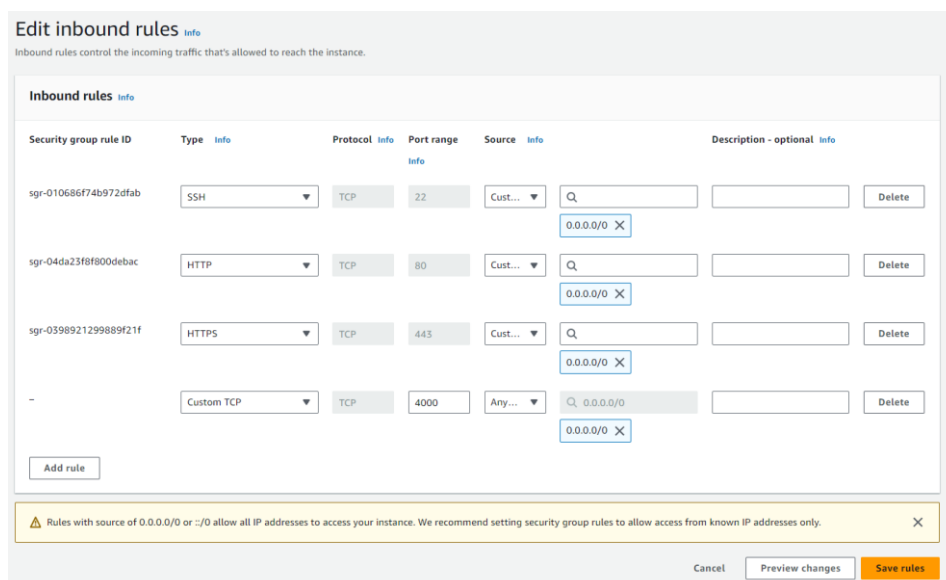
- 7) Go back to EC2 instances and select that instance newly made and click on Security groups.



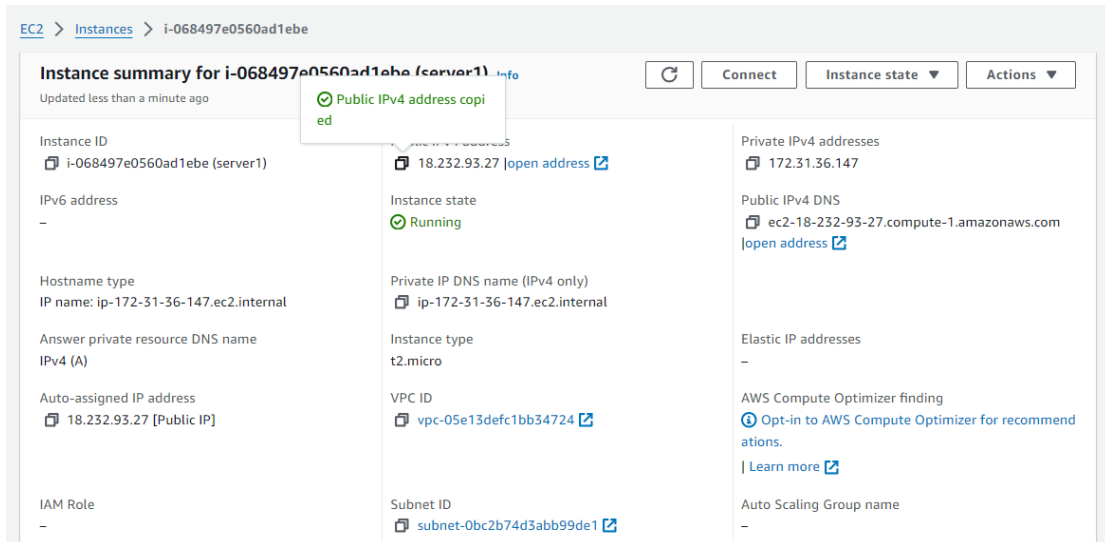
- 8) After in Inbound rules section click on Edit inbound rules.



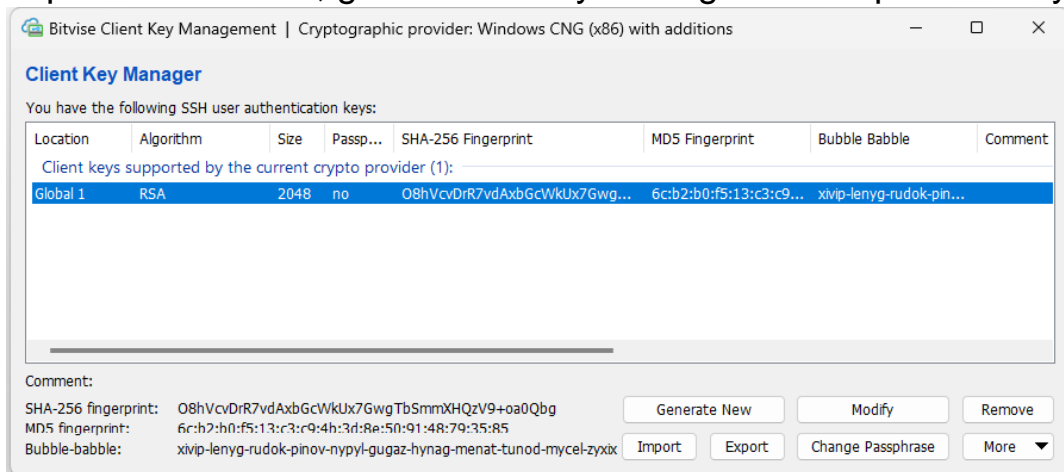
- 9) In Inbound rules, click on Add rule select Custom TCP, give port range 4000 and select 0.0.0.0/0. Click on Save Rules.



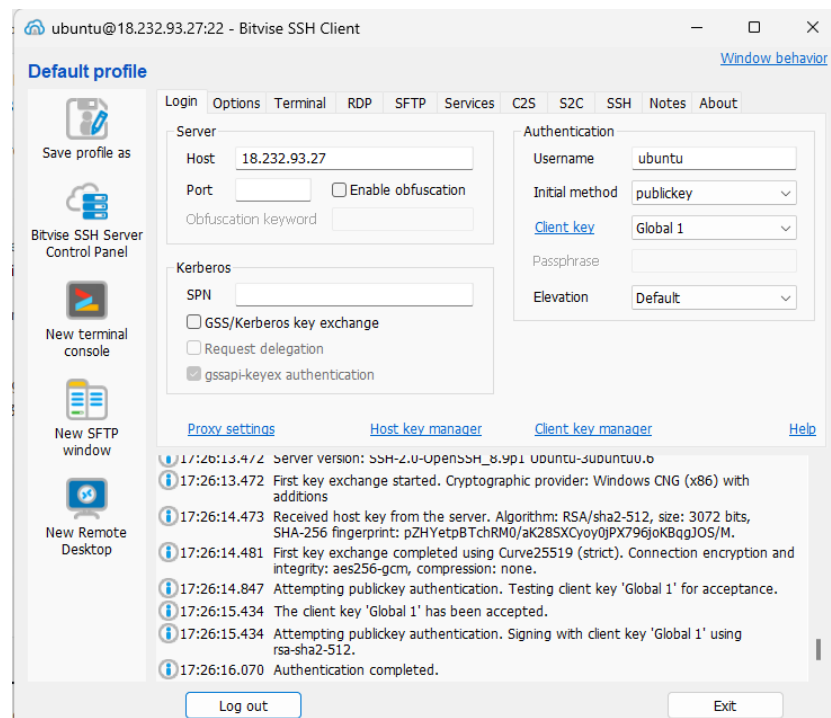
10) Go back to EC2 recently created instance and copy the public IPv4 address.



11) Open Bitvise SSH, go to Client key manager and import the key pair.



12) Paste that copied IPv4 public address in host and then click Login and click Accept and Save.



13) Open New Terminal Console and type command `pwd` to know current location.

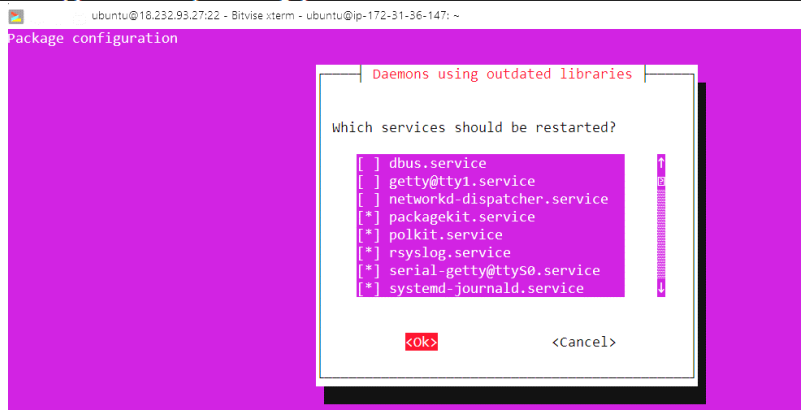
```
ubuntu@ip-172-31-36-147:~$ pwd
/home/ubuntu
```

14) Type command `sudo apt-get update` to fetch all packages.

```
ubuntu@ip-172-31-36-147:~$ sudo apt-get update
```

15) Type `sudo apt upgrade` to upgrade all outdated packages.

```
ubuntu@ip-172-31-36-147:~$ sudo apt-get upgrade
```



16) Type command `sudo apt-get install nginx`.

```
ubuntu@ip-172-31-36-147:~$ sudo apt-get install nginx
```

17) To execute JavaScript, we need to install NodeJS. Type command `curl -SL https://deb.nodesource.com/setup_16.x | sudo -E bash -`

```
ubuntu@ip-172-31-36-147:~$ curl -SL https://deb.nodesource.com/setup_16.x | sudo -E bash -
```

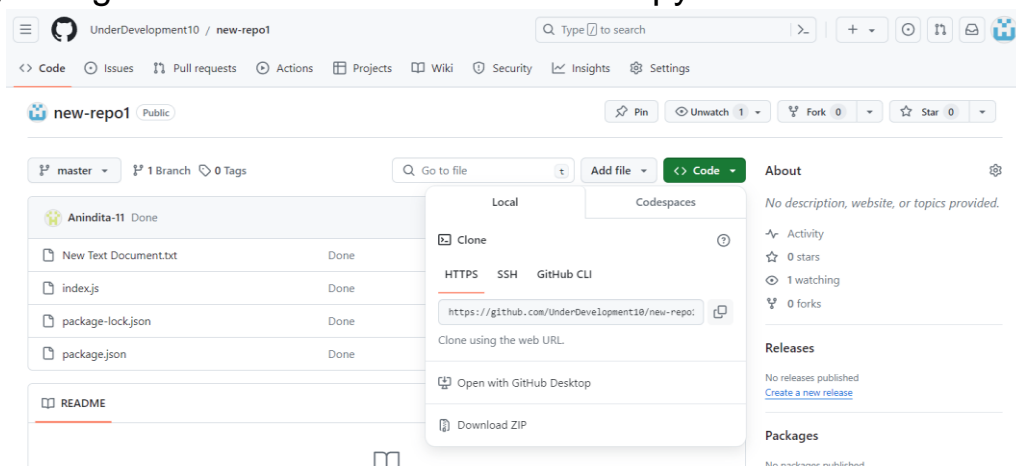
18) To install type `sudo apt install nodejs`.

```
ubuntu@ip-172-31-36-147:~$ sudo apt install nodejs
```

19) Type `node --version` to see what version of NodeJS installed.

```
ubuntu@ip-172-31-36-147:~$ node --version
v16.20.2
```

20) Now go back to GitHub and in Code copy HTTPS URL.



21) Type `git clone` and paste the URL to clone the repository.

```
ubuntu@ip-172-31-36-147:~$ git clone https://github.com/UnderDevelopment10/new-repo1.git
```

22) Type `ls` to see if project has been cloned or not. Then type command `cd (project name)` to enter into the project and then type `ls` to see what files have been cloned.

```
ubuntu@ip-172-31-36-147:~$ ls
new-repo1
ubuntu@ip-172-31-36-147:~$ cd new-repo1/
ubuntu@ip-172-31-36-147:~/new-repo1$ ls
'New Text Document.txt'  index.js  package-lock.json  package.json
```

23) Now, to execute node command we install node package manager (npm). Type `npm install`.

```
ubuntu@ip-172-31-36-147:~/new-repo1$ npm install
```

24) Type `node index.js`

```
ubuntu@ip-172-31-36-147:~/new-repo1$ node index.js
Started server
```

25) Copy public IPv4 address in EC2 instance and then paste it in another tab URL section.

```
< > ↻ 📖 ⚠ Not secure | 18.232.93.27
```

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.

26) To add port number in URL at end write `':4000'`.

```
< > ↻ 📖 ⚠ Not secure | 18.232.93.27:4000
```

Hello World

27) At the end, to close server in new terminal press `Ctrl+C` to stop server and press Logout in Bitwise SSH window.