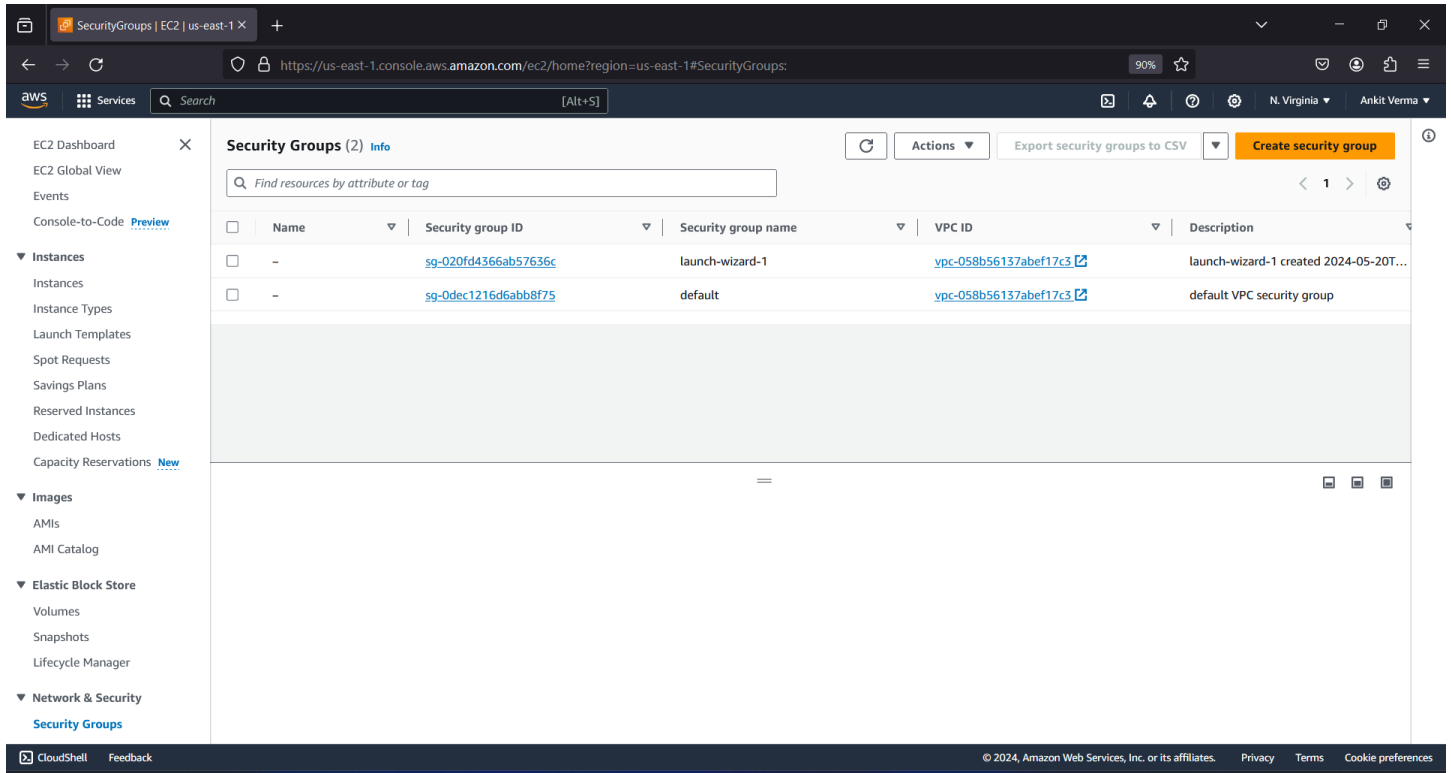


Assignment: 12

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

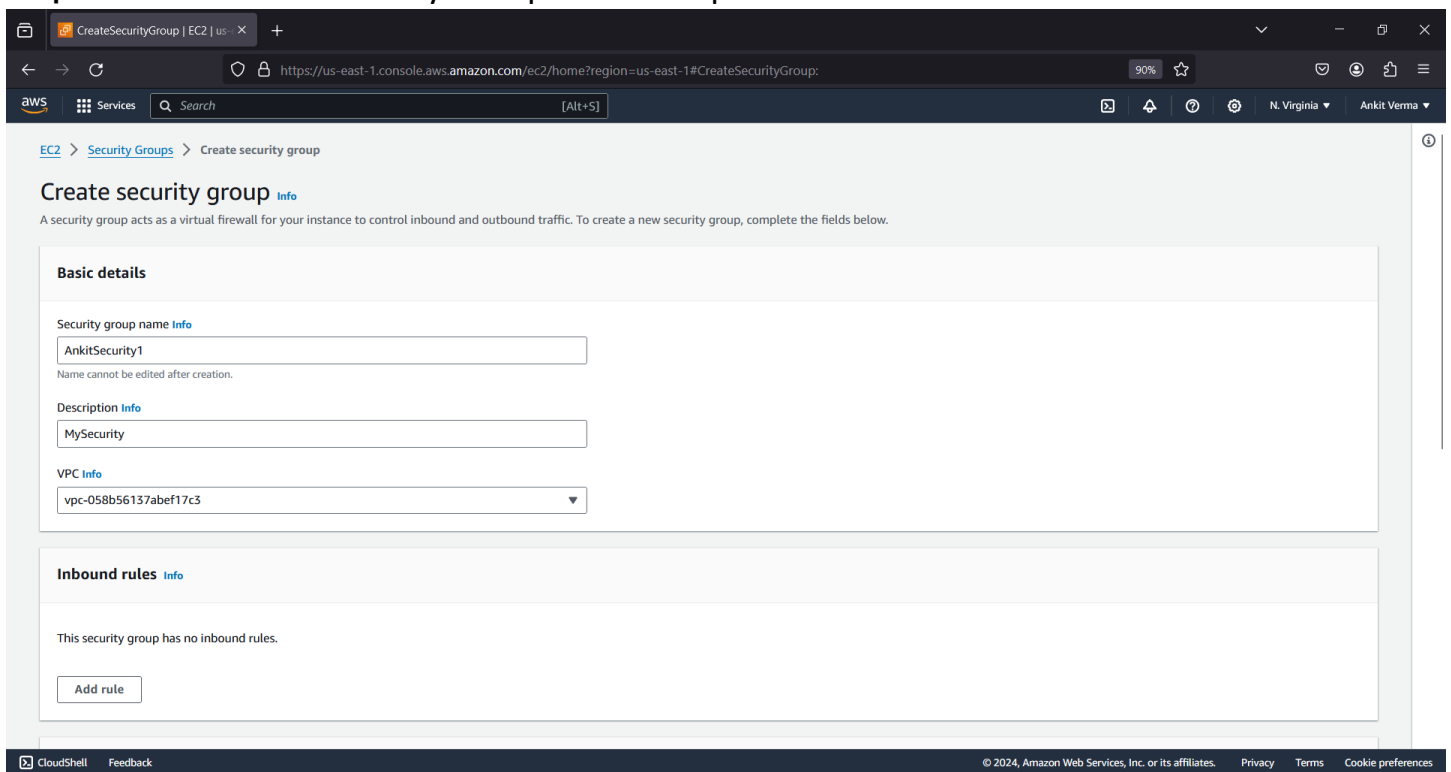
Step 1: Go to EC2 then Security groups and click on Create Security Group option.



The screenshot shows the AWS Management Console for the 'us-east-1' region. The left-hand navigation pane is open, showing the 'Security Groups' link under the 'Network & Security' section. The main content area displays the 'Security Groups (2)' page. At the top, there is a search bar and a 'Create security group' button. Below this is a table listing the existing security groups:

	Name	Security group ID	Security group name	VPC ID	Description
<input type="checkbox"/>	-	sg-020fd4366ab57636c	launch-wizard-1	vpc-058b56137abef17c3	launch-wizard-1 created 2024-05-20T...
<input type="checkbox"/>	-	sg-0dec1216d6abb8f75	default	vpc-058b56137abef17c3	default VPC security group

Step 2: Give name of Security Group and description.



The screenshot shows the 'Create security group' page in the AWS Management Console. The breadcrumb navigation at the top indicates the path: 'EC2 > Security Groups > Create security group'. The page title is 'Create security group'. Below the title, a brief description states: 'A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.'

The form is divided into two sections:

- Basic details:** This section contains three input fields:
 - Security group name:** The field contains 'AnkitSecurity1'. A note below the field states: 'Name cannot be edited after creation.'
 - Description:** The field contains 'MySecurity'.
 - VPC:** A dropdown menu showing 'vpc-058b56137abef17c3'.
- Inbound rules:** This section contains a message: 'This security group has no inbound rules.' and an 'Add rule' button.

Step 3: In Inbound rules click on Add rule. Here, we add all 4 rules: Custom TCP, SSH, HTTP, HTTPS and in Source select 0.0.0.0/0 In port range of Custom TCP give 4000. Rest has default port number.

The screenshot shows the AWS Management Console interface for configuring a Security Group. The 'Inbound rules' section is active, displaying a table of rules. The rules are:

Type	Protocol	Port range	Source	Description - optional
Custom TCP	TCP	4000	Anywher... 0.0.0.0/0	
SSH	TCP	22	Anywher... 0.0.0.0/0	
HTTP	TCP	80	Anywher... 0.0.0.0/0	
HTTPS	TCP	443	Anywher... 0.0.0.0/0	

Below the table, there is a yellow warning banner: "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."

Step 4: Go back to instance and click on Launch instance.

The screenshot shows the AWS Management Console interface for the 'Instances' page. The 'Launch instances' button is highlighted in orange. Below the table, there is a large grey area with a black arrow pointing to the right, indicating the next step in the process.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public
AnkitInstance	i-027e53b3d5b4e347c	Running	t2.micro	2/2 checks passed	View alarms	us-east-1d	ec2-100-28-36-90.com...	100.28

Step 5: Give name of instance and in Application and OS Images select Ubuntu.

Name and tags Info

Name: [Add additional tags](#)

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Recents **Quick Start**

Amazon Linux macOS **Ubuntu** Windows Red Hat SUSE Linux

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
ami-04b70fa74e45c3917 (64-bit (x86)) / ami-0eac975a54dfee8cb (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs [Free tier eligible](#)

Summary

Number of instances: Info

Software Image (AMI)
Canonical, Ubuntu, 24.04 LTS, ...[read more](#)
ami-04b70fa74e45c3917

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

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 ...

[Cancel](#) [Launch instance](#) [Review commands](#)

Step 6: Select existing key pair and then click on Common Security group dropdown and select the created security group. Then click on Launch Instance.

Key pair name - required

[Create new key pair](#)

Network settings Info [Edit](#)

Network Info
vpc-058b56137abef17c3

Subnet Info
No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

☐ launch-wizard-1 sg-020fd4366ab57636c
VPC: vpc-058b56137abef17c3

☒ **AnkitSecurity1** sg-0df82616bd8c0c29
VPC: vpc-058b56137abef17c3

☐ default sg-0dec1216d6abb8f75
VPC: vpc-058b56137abef17c3

[Select security groups](#)

[Compare security group rules](#)

Summary

Number of instances: Info

Software Image (AMI)
Canonical, Ubuntu, 24.04 LTS, ...[read more](#)
ami-04b70fa74e45c3917

Virtual server type (instance type)
t2.micro

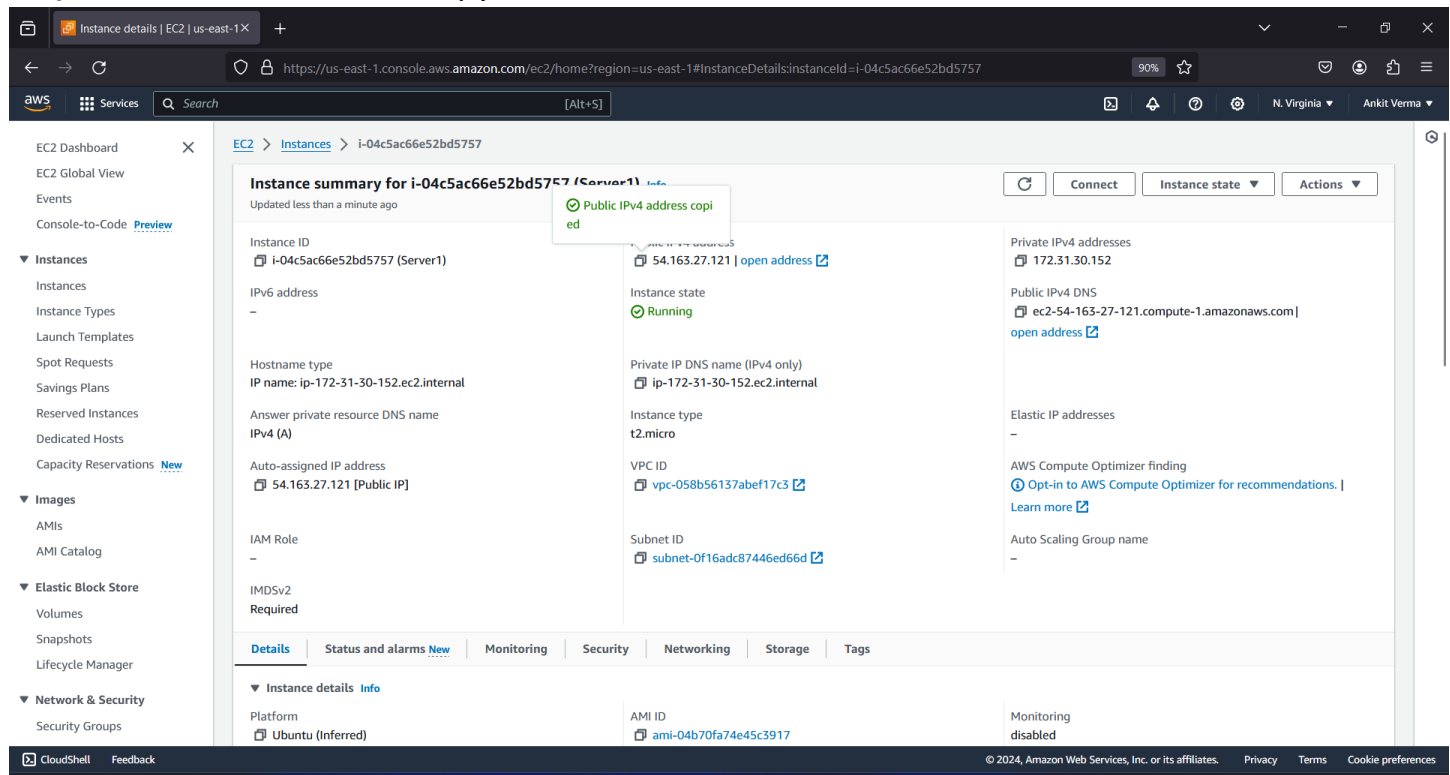
Firewall (security group)
AnkitSecurity1

Storage (volumes)
1 volume(s) - 8 GiB

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 ...

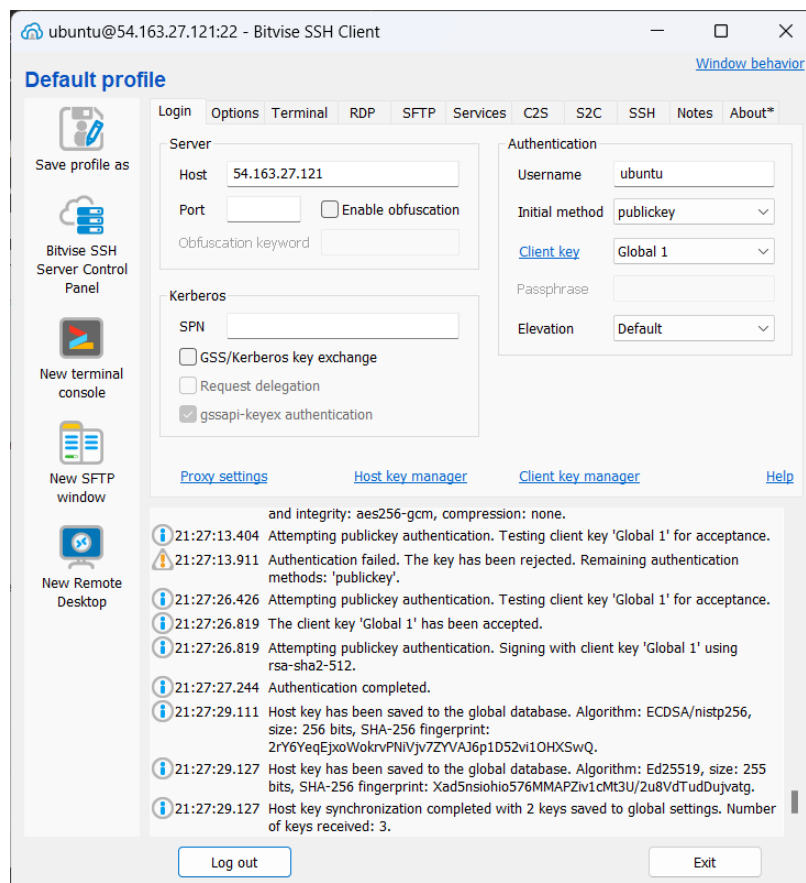
[Cancel](#) [Launch instance](#) [Review commands](#)

Step 7: Go to Instances and copy Public IPv4 address.



The screenshot shows the AWS Management Console interface. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Console-to-Code, and various instance and image management options. The main content area displays the 'Instance details' for an EC2 instance with ID 'i-04c5ac66e52bd5757'. A tooltip indicates that the 'Public IPv4 address' has been copied. The instance is in a 'Running' state. Key details include: Instance ID: i-04c5ac66e52bd5757 (Server1), IP address: 54.163.27.121, Hostname type: IP name: ip-172-31-30-152.ec2.internal, Answer private resource DNS name: IPv4 (A), Auto-assigned IP address: 54.163.27.121 [Public IP], IAM Role: -, IMDSv2: Required, Platform: Ubuntu (Inferred), AMI ID: ami-04b70fa74e5c3917, Monitoring: disabled. The bottom of the console shows the 'CloudShell' and 'Feedback' buttons.

Step 8: In Bitwise SSH Client, paste Copied IPv4 address, import the required key and click on Log In.



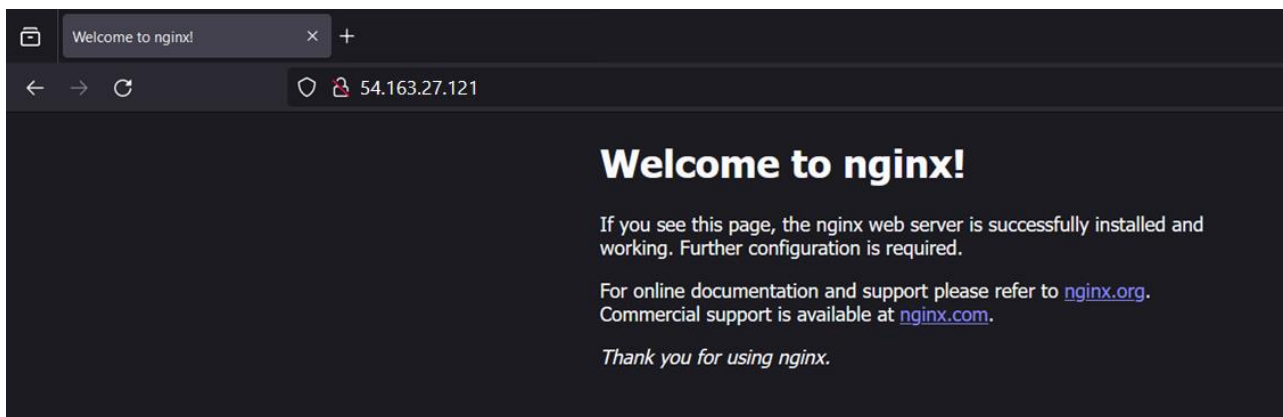
The screenshot shows the Bitwise SSH Client interface. The 'Default profile' tab is active, displaying the 'Login' section. The 'Host' field is set to '54.163.27.121'. The 'Port' field is empty, and the 'Enable obfuscation' checkbox is unchecked. The 'Authentication' section shows the 'Username' as 'ubuntu', the 'Initial method' as 'publickey', and the 'Client key' as 'Global 1'. The 'Passphrase' field is empty, and the 'Elevation' is set to 'Default'. The 'Kerberos' section is expanded, showing the 'SPN' field and checkboxes for 'GSS/Kerberos key exchange', 'Request delegation', and 'gssapi-keyex authentication'. The 'Log In' button is visible at the bottom. The terminal window at the bottom shows the following output:

```
and integrity: aes256-gcm, compression: none.
21:27:13.404 Attempting publickey authentication. Testing client key 'Global 1' for acceptance.
21:27:13.911 Authentication failed. The key has been rejected. Remaining authentication methods: 'publickey'.
21:27:26.426 Attempting publickey authentication. Testing client key 'Global 1' for acceptance.
21:27:26.819 The client key 'Global 1' has been accepted.
21:27:26.819 Attempting publickey authentication. Signing with client key 'Global 1' using rsa-sha2-512.
21:27:27.244 Authentication completed.
21:27:29.111 Host key has been saved to the global database. Algorithm: ECDSA/nistp256, size: 256 bits, SHA-256 fingerprint: 2rY6YeqEjxWokrvPNivjv7ZYVAJ6p1D52vi0HXSwQ.
21:27:29.127 Host key has been saved to the global database. Algorithm: Ed25519, size: 255 bits, SHA-256 fingerprint: Xad5nsiohio576MMApZv1cMt3U/2u8VdTudDujvatg.
21:27:29.127 Host key synchronization completed with 2 keys saved to global settings. Number of keys received: 3.
```

Step 9: Open New Terminal Console and enter the following commands:

```
ubuntu@54.163.27.121:22 - Bitvise xterm - ubuntu@ip-172-31-30-152: ~
ubuntu@ip-172-31-30-152:~$ sudo apt-get update
sudo apt upgrade
sudo apt-get install nginx
curl -SL https://deb.nodesource.com/setup_16.x|sudo -E bash -
sudo apt install nodejs
git clone https://github.com/UnderDevelopment10/new-repo1.git
cd repo2
npm install
node index.js
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates In
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [89.
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main Trans
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe a
```

Step 10: Paste the copied URL in browser. Then stop the server.



Step 11: Now type the following commands:

- i. `cd /`
- ii. `pwd`
- iii. `cd etc/nginx/sites-available/`
- iv. `sudo nano default`

Step 12: In the “default” file, comment out all lines under location. Then type the following lines in place of location.

```
location / {

    proxy_pass http://localhost:4000;

    proxy_http_version 1.1;

    proxy_set_header Upgrade $http_upgrade;

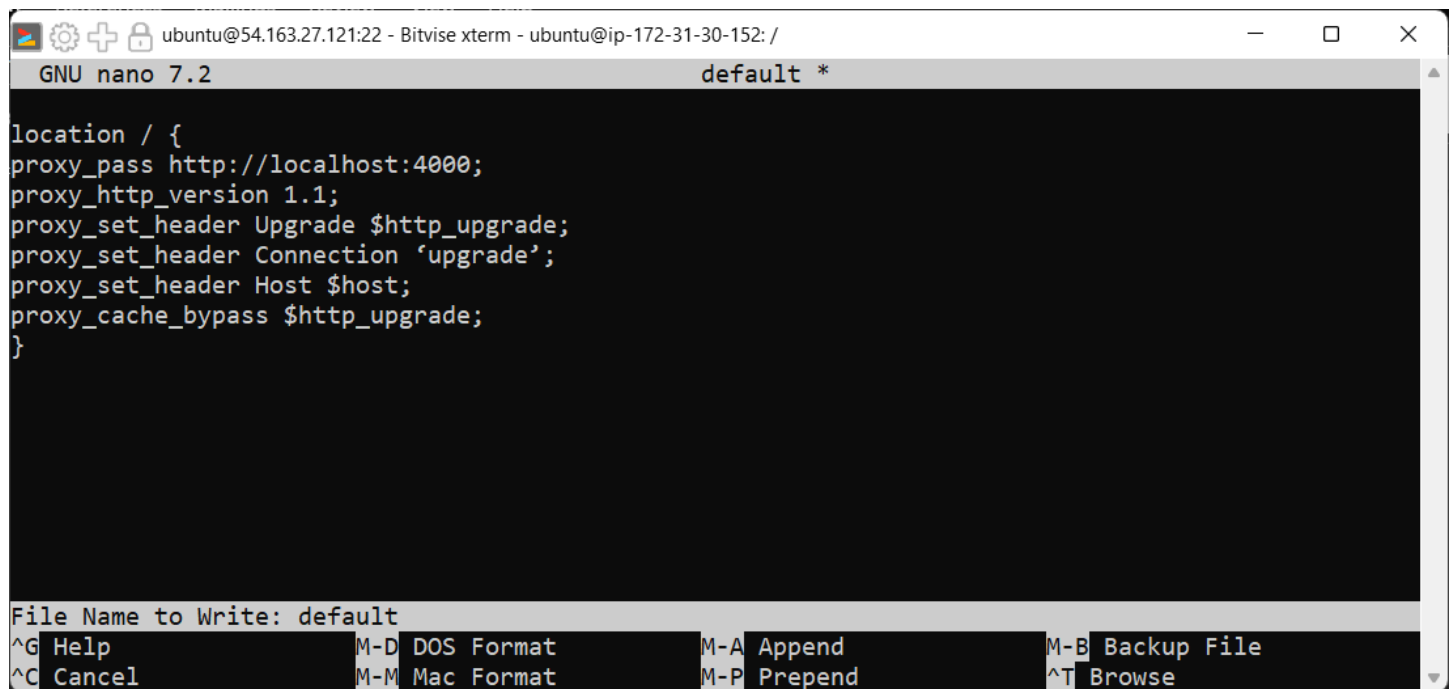
    proxy_set_header Connection 'upgrade';

    proxy_set_header Host $host;

    proxy_cache_bypass $http_upgrade;

}
```

Step 13: To save and exit, press Ctrl+X, then Y and press Enter.



```
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;
}

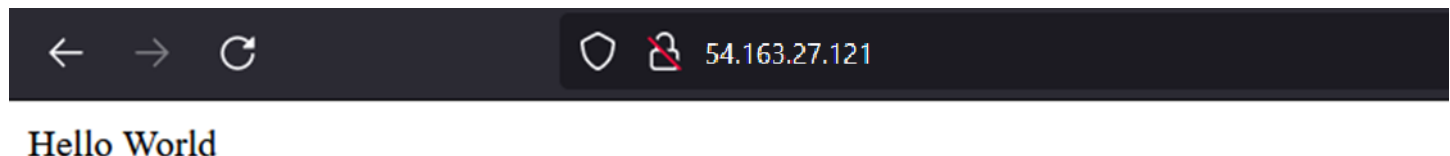
File Name to Write: default
^G Help      ^M-D DOS Format  ^M-A Append     ^M-B Backup File
^C Cancel    ^M-M Mac Format  ^M-P Prepend    ^T Browse
```

Step 14: Open new server terminal and type

```
cd new-repo1
```

```
sudo systemctl restart nginx
```

Step 15: Paste the copied IPv4 address in browser.



Here, the page has been accessed without using any port number with the IPv4 address