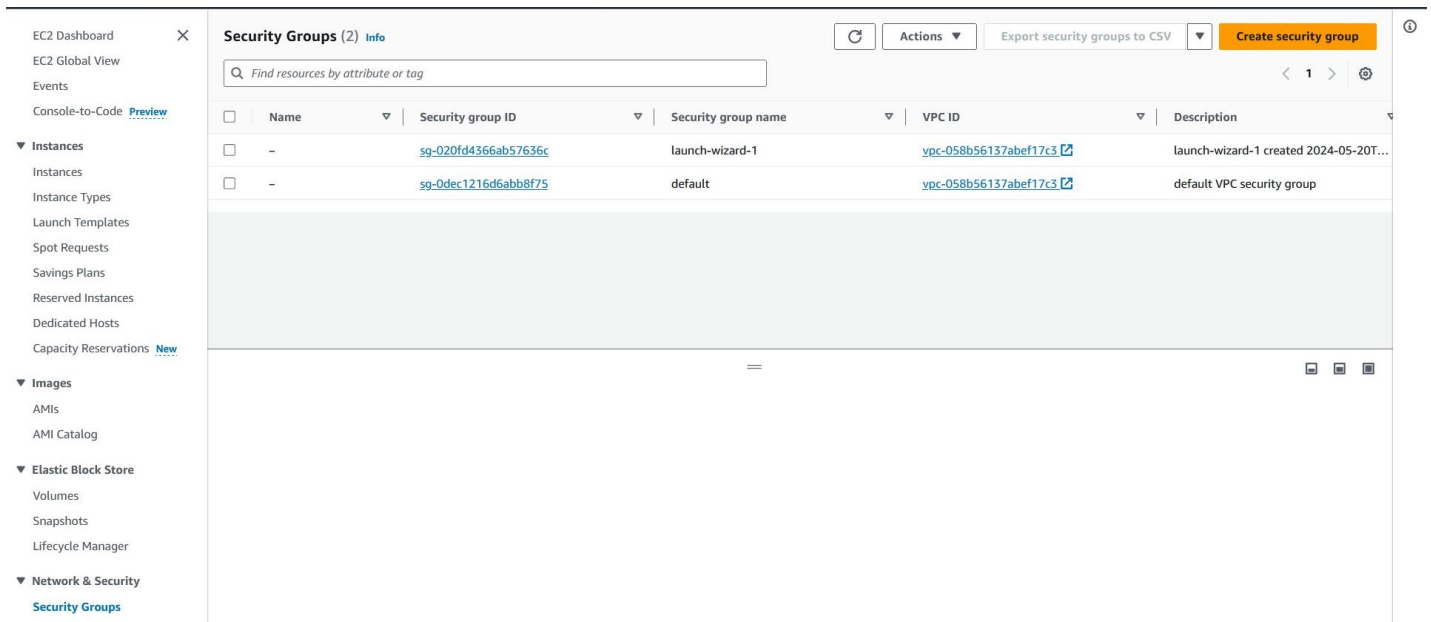


# Assignment: 12

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

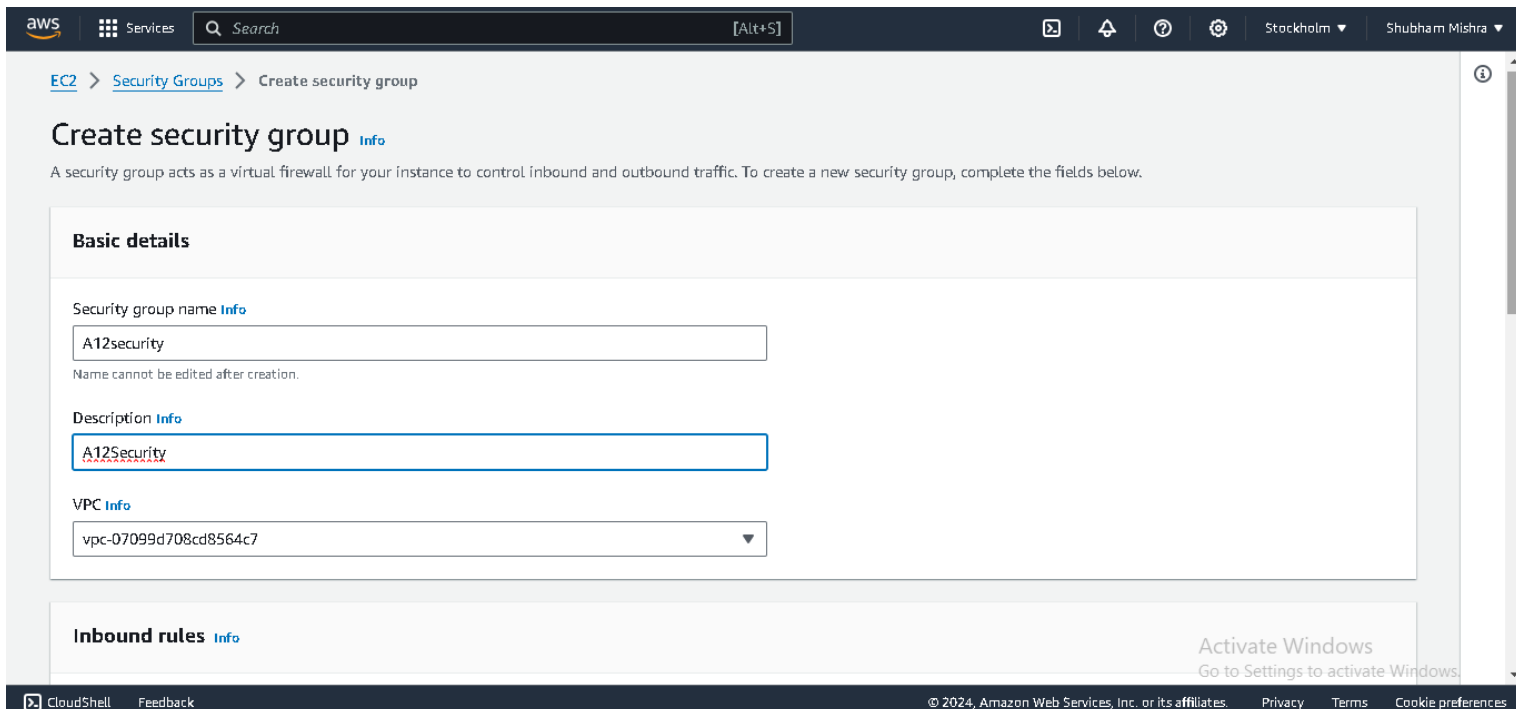
**Step 1:** Go to EC2 then Security groups and click on Create Security Group option.



The screenshot shows the AWS Management Console interface for the 'Security Groups' page. The left-hand navigation pane is visible, with 'Security Groups' selected under the 'Network & Security' category. The main content area displays a table of existing security groups. The table has columns for Name, Security group ID, Security group name, VPC ID, and Description. Two security groups are listed: 'launch-wizard-1' and 'default'. Above the table, there is a search bar and a 'Create security group' button.

	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' form in the AWS Management Console. The breadcrumb navigation at the top indicates the path: EC2 > Security Groups > Create security group. The form is titled 'Create security group' and includes a brief description of a security group's function. The 'Basic details' section contains three fields: 'Security group name' (with the value 'A12security'), 'Description' (with the value 'A12Security'), and 'VPC' (with the value 'vpc-07099d708cd8564c7'). The 'Inbound rules' section is partially visible at the bottom. The bottom of the screen shows the AWS footer with copyright information and links to Privacy, Terms, and Cookie preferences.

**Create security group**

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.

**Basic details**

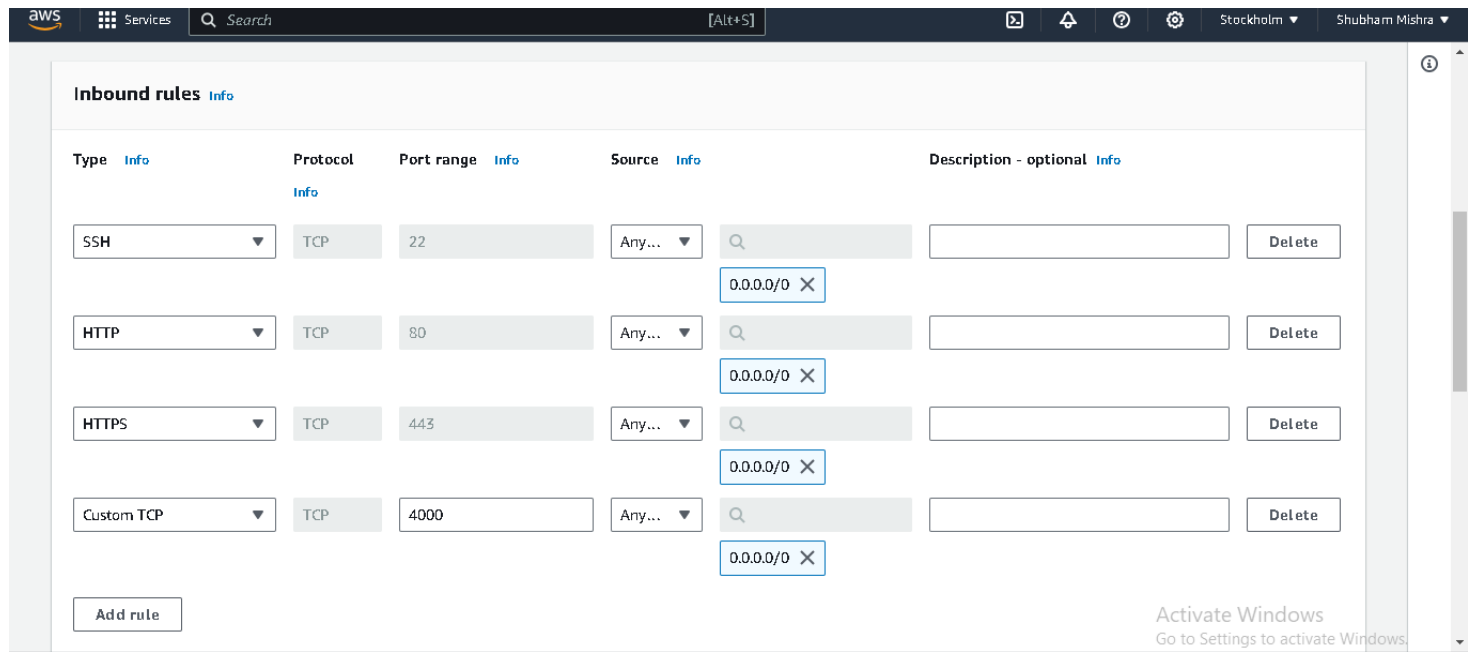
Security group name: A12security  
Name cannot be edited after creation.

Description: A12Security

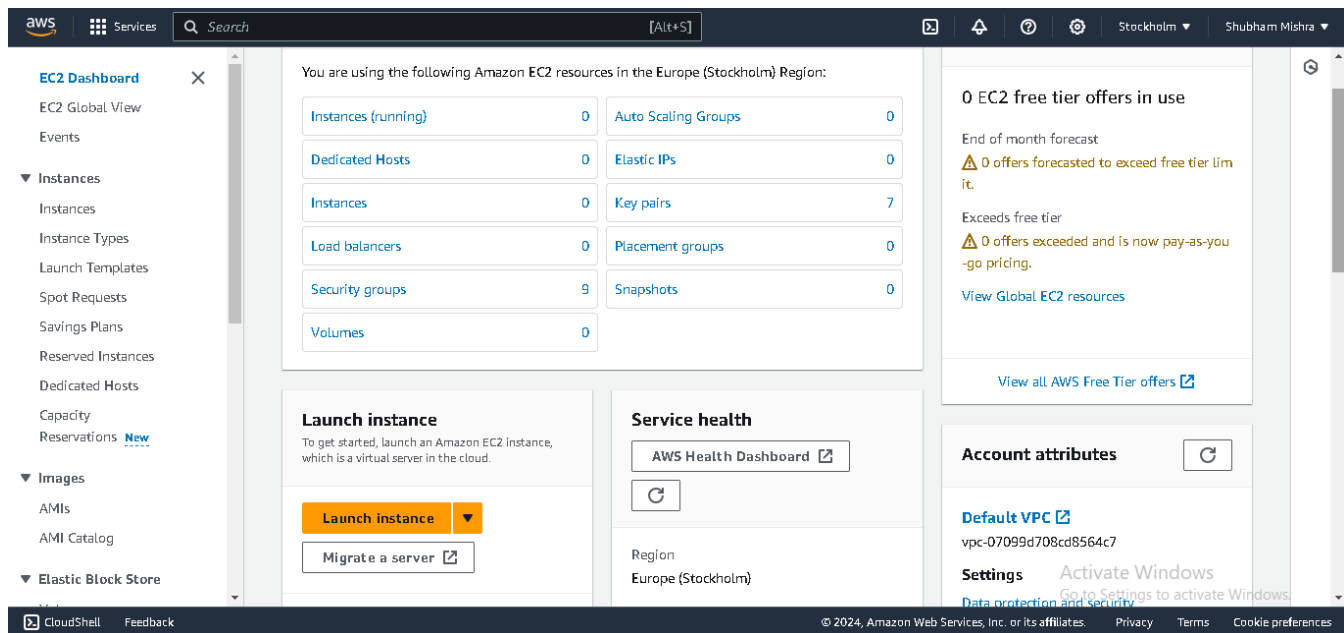
VPC: vpc-07099d708cd8564c7

**Inbound rules**

**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.



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



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

**Name and tags** [Info](#)

Name  
A12server [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

**Quick Start**

Amazon Linux

macOS

**Ubuntu**

Windows

Red Hat

SUSE Li

[Browse more AMIs](#)  
Including AMIs from AWS, Marketplace and the Community

**Summary**

Number of instances [Info](#)  
1

**Software Image (AMI)**  
Canonical, Ubuntu, 24.04 LTS, ...[read more](#)  
ami-0705384c0b33c194c

**Virtual server type (instance type)**  
t3.micro

**Firewall (security group)**  
New security group

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

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

[Go to Settings to activate Windows](#)  
[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.

**Network settings** [Info](#) [Edit](#)

**Network** [Info](#)  
vpc-07099d708cd8564c7

**Subnet** [Info](#)  
No preference (Default subnet in any availability zone)

**Auto-assign public IP** [Info](#)  
Enable  
[Additional charges apply](#) when outside of [free tier allowance](#)

**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

**Common security groups** [Info](#)

Shubhamsecurity sg-028d555def910712a [X](#)  
VPC: vpc-07099d708cd8564c7

[Compare security group rules](#)

Security groups that you add or remove here will be added to or removed from all your network interfaces.

**Summary**

Number of instances [Info](#)  
1

**Software Image (AMI)**  
Canonical, Ubuntu, 24.04 LTS, ...[read more](#)  
ami-0705384c0b33c194c

**Virtual server type (instance type)**  
t3.micro

**Firewall (security group)**  
Shubhamsecurity

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

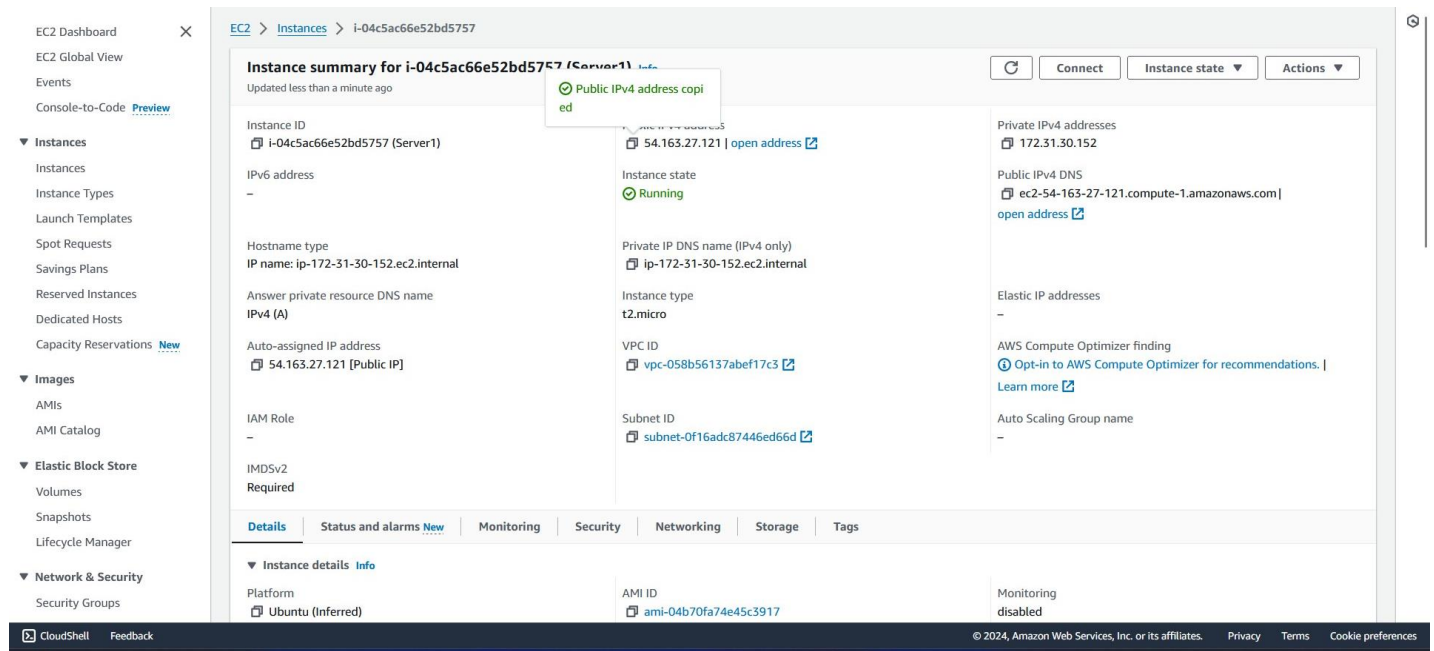
[Cancel](#) [Launch instance](#)

[Go to Settings to activate Windows](#)  
[Review commands](#)

CloudShell Feedback

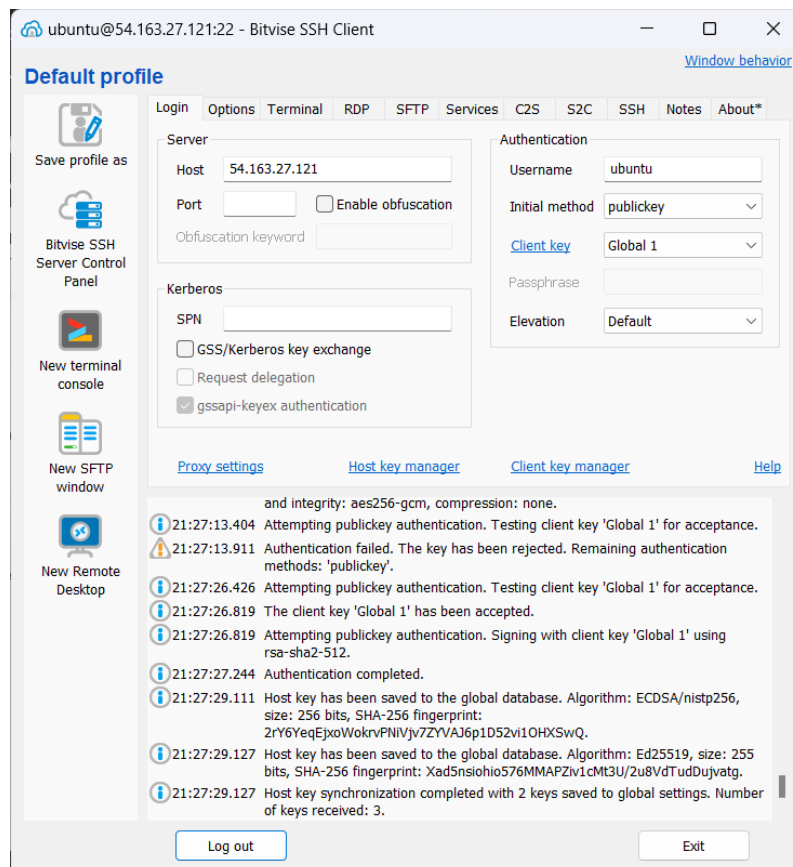
© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

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



The screenshot shows the AWS Management Console interface for an EC2 instance. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Console-to-Code, and various instance management options. The main content area displays the 'Instance summary for i-04c5ac66e52bd5757 (Server1)'. The instance is in the 'Running' state. The Public IPv4 address is 54.163.27.121. A tooltip indicates that the Public IPv4 address has been copied. The instance details include: Instance ID: i-04c5ac66e52bd5757 (Server1), IPv6 address: -, 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, Instance state: Running, Private IP address: 172.31.30.152, Public IPv4 DNS: ec2-54-163-27-121.compute-1.amazonaws.com, Elastic IP addresses: -, AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations, Auto Scaling Group name: -, Instance type: t2.micro, VPC ID: vpc-058b56137abef17c3, Subnet ID: subnet-0f16adc87446ed66d, AMI ID: ami-04b70fa74e45c3917, Monitoring: disabled.

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



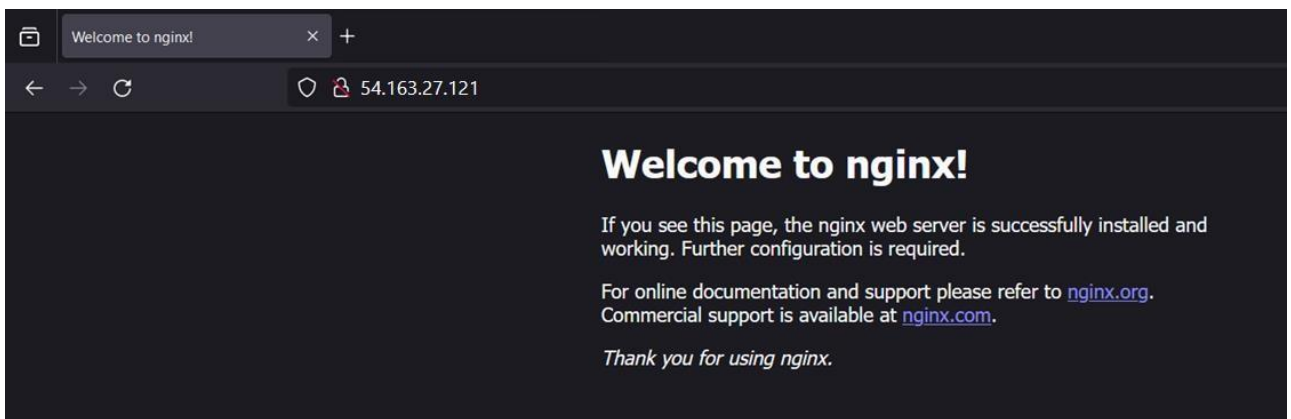
The screenshot shows the Bitwise SSH Client interface. The 'Default profile' is selected. The 'Host' field is set to 54.163.27.121. The 'Authentication' section shows 'Username' as 'ubuntu' and 'Initial method' as 'publickey'. The 'Client key' is set to 'Global 1'. The 'Log In' button is visible at the bottom. The terminal window 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: 2rY6YeqEjxWokrvPNIVjv7ZYVAJ6p1D52vi1OHXSwQ.
21:27:29.127 Host key has been saved to the global database. Algorithm: Ed25519, size: 255 bits, SHA-256 fingerprint: Xad5nsiohio576MMAFZiv1cMt3U/2u8VdTudDuyvatg.
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;

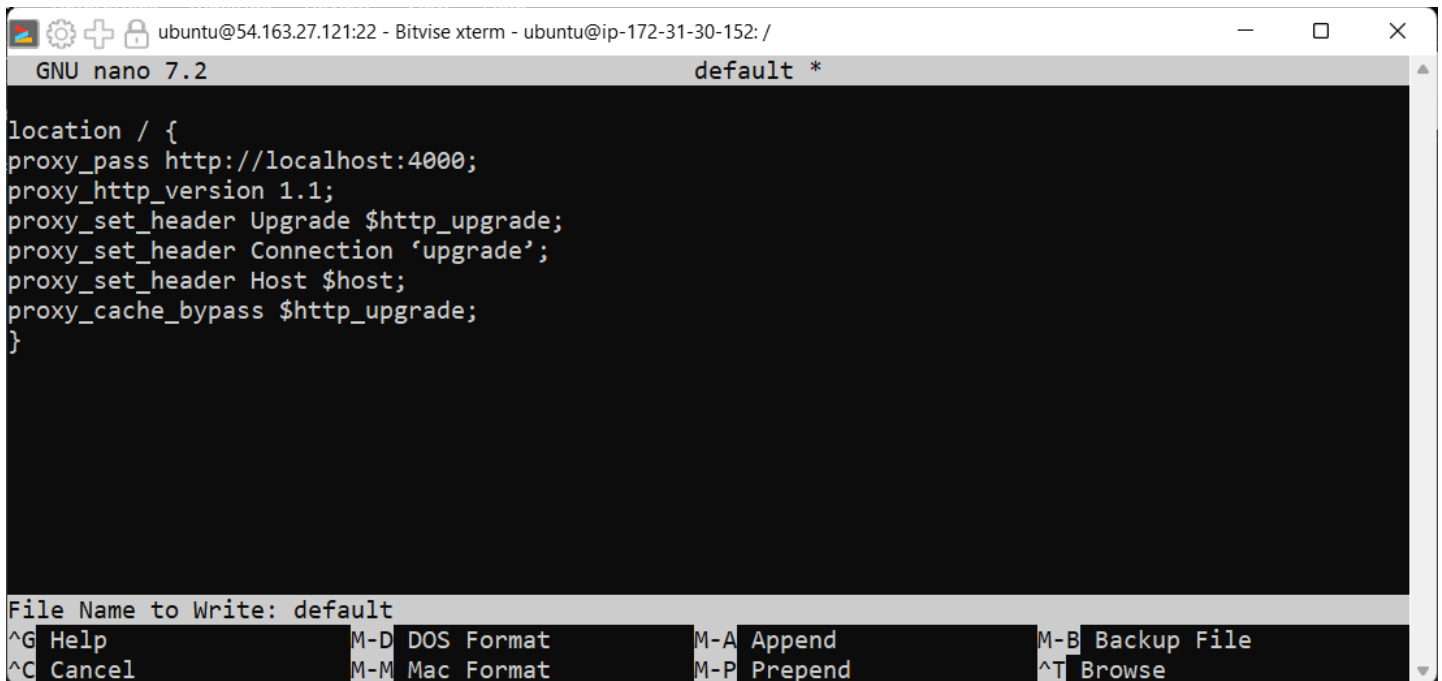
    proxy_set_header Connection 'upgrade';

    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

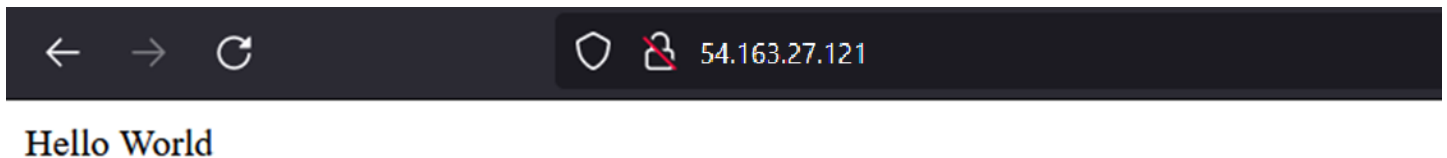
^G Help M-D DOS Format M-A Append M-B Backup File  
^C Cancel M-M Mac Format M-P Prepend ^T Browse

**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