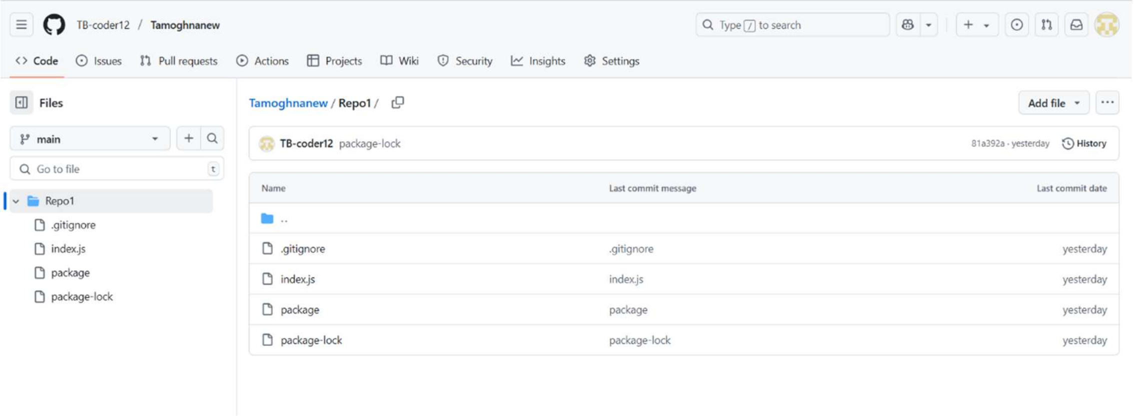
Assignment No: 09

Problem Definition : Deploy a project from GitHub to EC2.

Solution:

Step 1:

Upload required files to GitHub



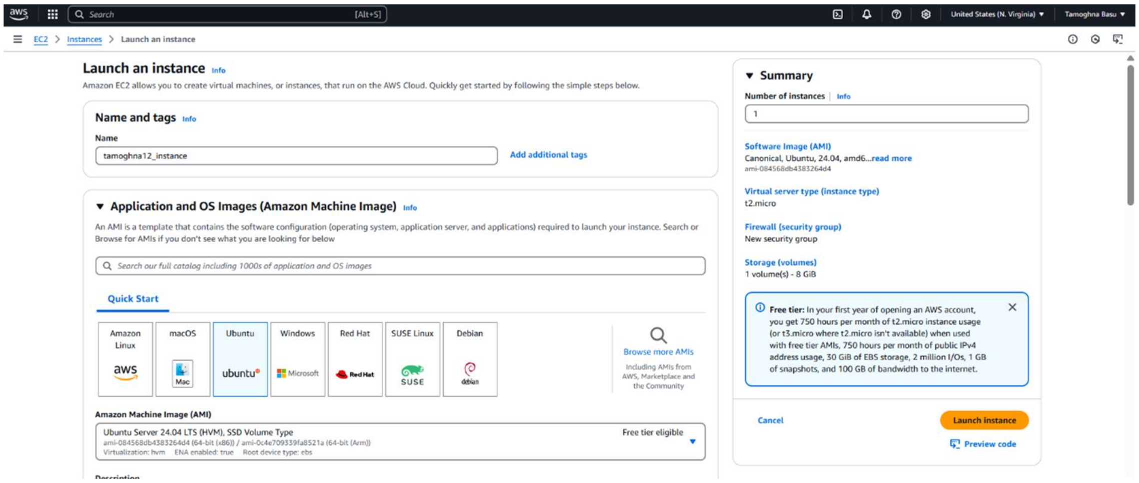
Tamoghnanew / Repo1 /

TB-coder12 package-lock 81a392a · yesterday

Name	Last commit message	Last commit date
..		
.gitignore	.gitignore	yesterday
index.js	index.js	yesterday
package	package	yesterday
package-lock	package-lock	yesterday

Step 2:

Log into AWS and open EC2. Click on launch instance. Give a name to the instance and select the operating system as Ubuntu.



aws
Search [Alt+S]
United States (N. Virginia)
Tamoghna Basu
EC2 > Instances > Launch an instance
Info
Add additional tags
Search our full catalog including 1000s of application and OS images
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Linux
Debian
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Cancel
Launch instance
Preview code

Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name: tamoghna12_instance

Application and OS Images (Amazon Machine Image)

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

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-084568db4383264d4 (64-bit (x86)) / ami-0c4e709339fa8521a (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Summary

Number of instances: 1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-084568db4383264d4

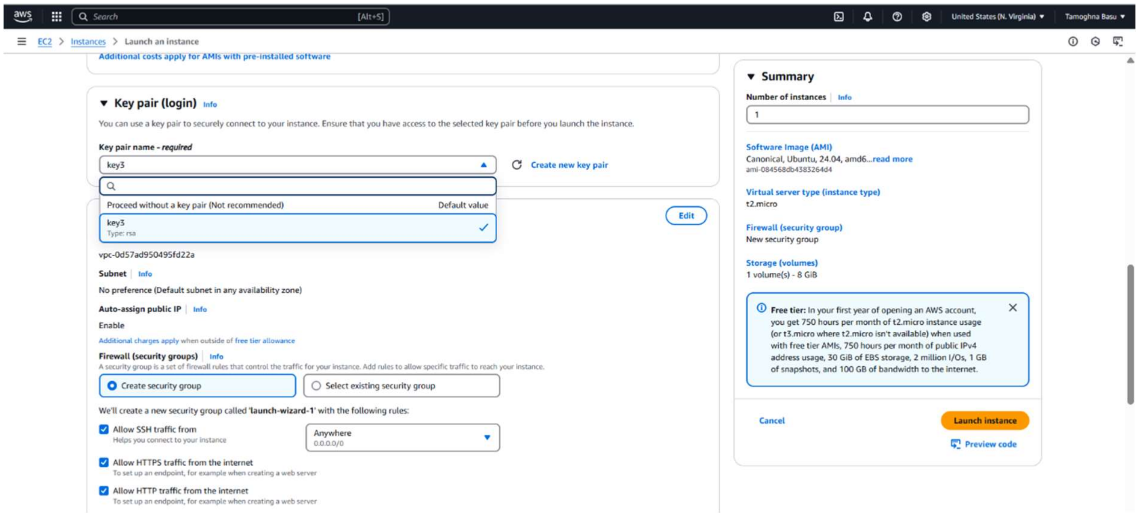
Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Step 3: Create Key Pair.



Search [Alt+S]

United States (N. Virginia) ▾ Tamoghna Basu ▾

EC2 > Instances > Launch an instance

Additional costs apply for AMIs with pre-installed software

▼ Key pair (login) Info

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.

Key pair name - *required*

key3

Create new key pair

Proceed without a key pair (Not recommended) Default value

key3
Type: rsa ✓

Edit

vpc-0d57ad950495fd22a

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

We'll create a new security group called '**launch-wizard-1**' with the following rules:

☑ Allow SSH traffic from
Helps you connect to your instance
Anywhere
0.0.0.0/0

☑ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

☑ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

▼ **Summary**

Number of instances Info

1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-084568db4383264d4

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

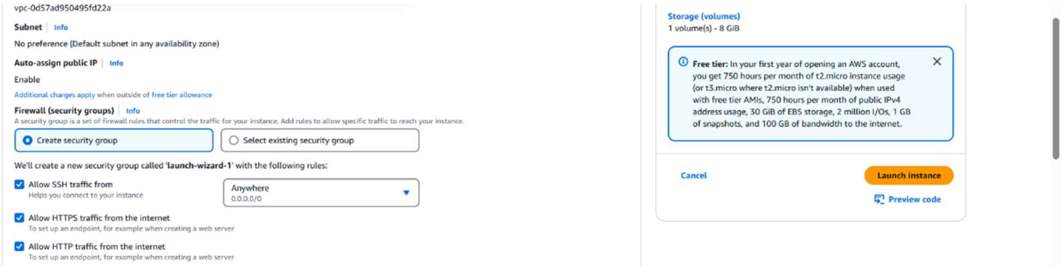
Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel Launch instance

Preview code

Step 4: Check the 3 boxes to allow SSH, HTTP and HTTPS traffics and click launch instance.



vpc-0d57ad950495fd22a

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

We'll create a new security group called '**launch-wizard-1**' with the following rules:

☑ Allow SSH traffic from
Helps you connect to your instance
Anywhere
0.0.0.0/0

☑ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

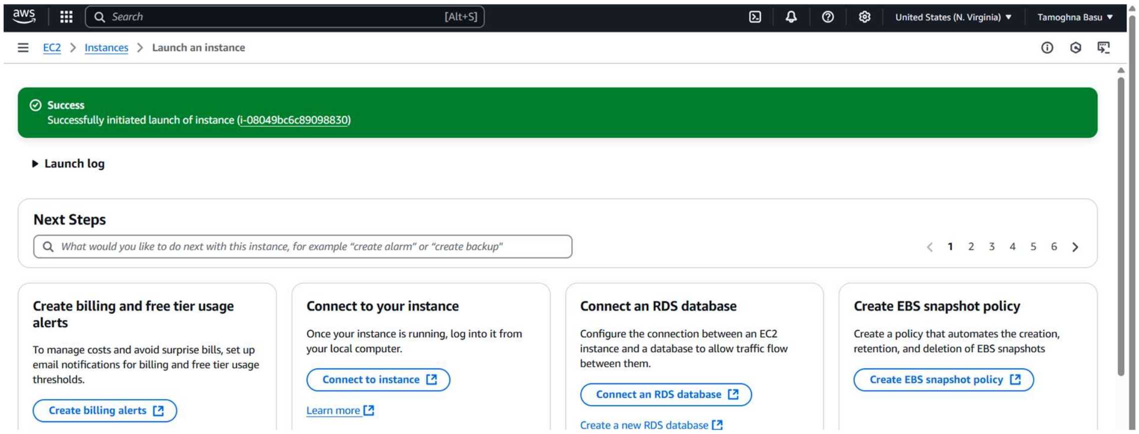
☑ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

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

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel Launch instance

Preview code



Search [Alt+S]

United States (N. Virginia) ▾ Tamoghna Basu ▾

EC2 > Instances > Launch an instance

Success
Successfully initiated launch of instance (i-08049bc6c89098830)

▶ Launch log

Next Steps

What would you like to do next with this instance, for example "create alarm" or "create backup"

< 1 2 3 4 5 6 >

Create billing and free tier usage alerts

To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.

Create billing alerts

Connect to your instance

Once your instance is running, log into it from your local computer.

Connect to instance

Learn more

Connect an RDS database

Configure the connection between an EC2 instance and a database to allow traffic flow between them.

Connect an RDS database

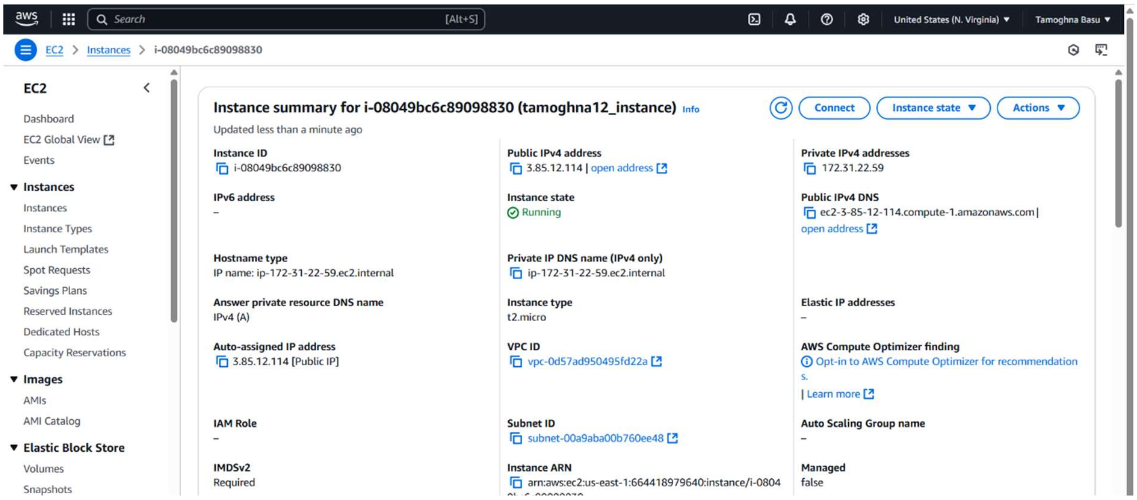
Create a new RDS database

Create EBS snapshot policy

Create a policy that automates the creation, retention, and deletion of EBS snapshots

Create EBS snapshot policy

Step 5: Open instances and click of the instance Id of the instance. Copy the IPv4 address.



Instance summary for i-08049bc6c89098830 (tamoghna12_instance) Info

Updated less than a minute ago

Instance ID
i-08049bc6c89098830

Public IPv4 address
3.85.12.114 | open address

Private IPv4 addresses
172.31.22.59

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-3-85-12-114.compute-1.amazonaws.com | open address

Hostname type
IP name: ip-172-31-22-59.ec2.internal

Private IP DNS name (IPv4 only)
ip-172-31-22-59.ec2.internal

Answer private resource DNS name
IPv4 (A)

Instance type
t2.micro

Elastic IP addresses
–

Auto-assigned IP address
3.85.12.114 [Public IP]

VPC ID
vpc-0d57ad950495fd22a

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

IAM Role
–

Subnet ID
subnet-00a9aba00b760ee48

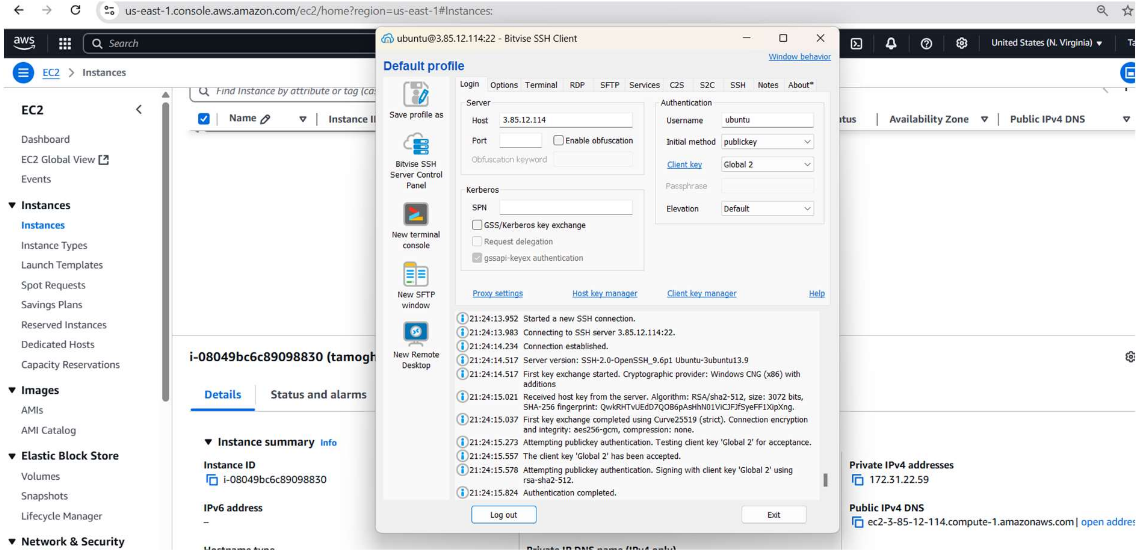
Auto Scaling Group name
–

IMDSv2
Required

Instance ARN
arn:aws:ec2:us-east-1:664418979640:instance/i-0804

Managed
false

Step 6: Open Bitwise SSH Client and paste the IP address in host and set up authentication using the key pair then login and after logging in open new terminal window.



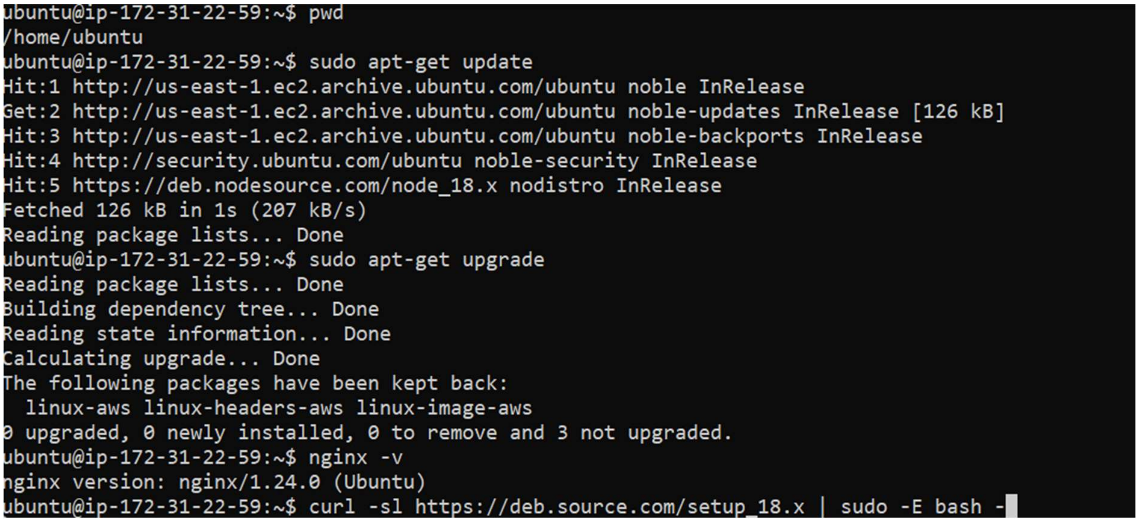
Server
Host: 3.85.12.114
Port:
□ Enable obfuscation
Obfuscation keyword

Kerberos
SPN:
□ GSS/Kerberos key exchange
□ Request delegation
☑ gssapi-keyex authentication

Authentication
Username: ubuntu
Initial method: publickey
Client key: Global 2
Passphrase
Elevation: Default

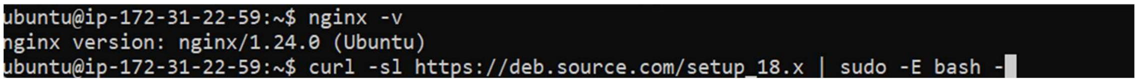
```
21:24:13.952 Started a new SSH connection.
21:24:13.983 Connecting to SSH server 3.85.12.114:22.
21:24:14.234 Connection established.
21:24:14.517 Server version: SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.9
21:24:14.517 First key exchange started. Cryptographic provider: Windows CNG (x86) with
             additions
21:24:15.021 Received host key from the server. Algorithm: RSA/sha2-512, size: 3072 bits,
             SHA-256 fingerprint: QwkRHTvUEdD7QOB6pAsHhN01ViCJFJfSyeFF1XipXng.
21:24:15.037 First key exchange completed using Curve25519 (strict). Connection encryption
             and integrity: aes256-gcm, compression: none.
21:24:15.273 Attempting publickey authentication. Testing client key 'Global 2' for acceptance.
21:24:15.557 The client key 'Global 2' has been accepted.
21:24:15.578 Attempting publickey authentication. Signing with client key 'Global 2' using
             rsa-sha2-512.
21:24:15.824 Authentication completed.
```

Step 7: Then Execute command sudo apt-get update and sudo apt-get upgrade in the Bitvise SSH Client Terminal. And Now Execute sudo apt -get install nginx Cmd for setting up the web server.

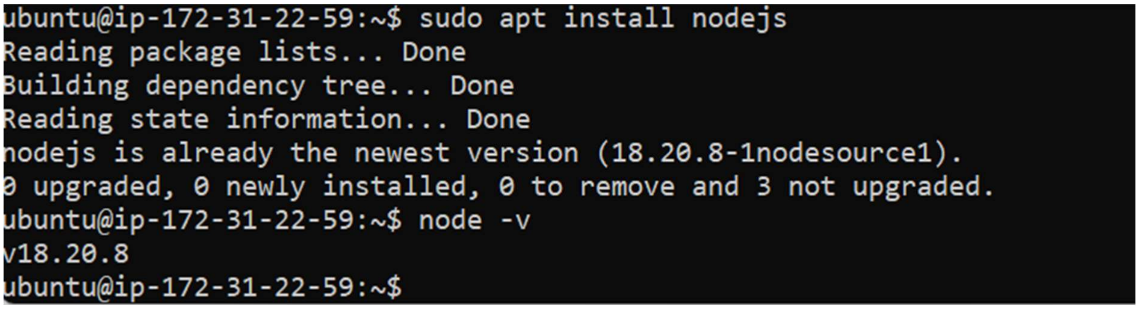


```
ubuntu@ip-172-31-22-59:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-22-59:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://deb.nodesource.com/node_18.x nodistro InRelease
Fetched 126 kB in 1s (207 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-22-59:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  linux-aws linux-headers-aws linux-image-aws
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
ubuntu@ip-172-31-22-59:~$ nginx -v
nginx version: nginx/1.24.0 (Ubuntu)
ubuntu@ip-172-31-22-59:~$ curl -sl https://deb.source.com/setup_18.x | sudo -E bash -
```

Step 8: After that, then open the directory that is cloned using command cd name of the directory. Then install NodeJS using commands curl: https://deb.nodesource.com/setup_18.x | sudo -E bash – and sudo apt-get install nodejs.

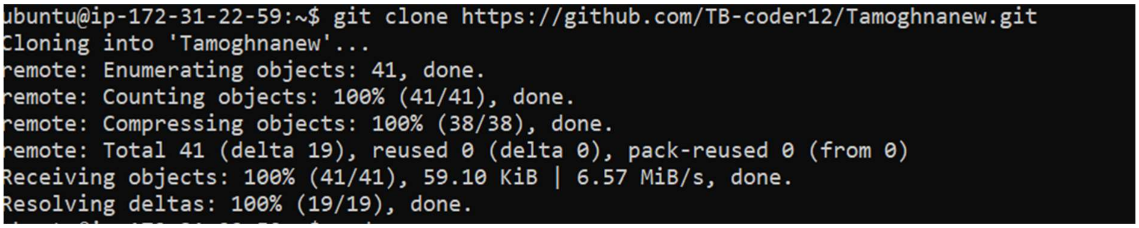


```
ubuntu@ip-172-31-22-59:~$ nginx -v
nginx version: nginx/1.24.0 (Ubuntu)
ubuntu@ip-172-31-22-59:~$ curl -sl https://deb.source.com/setup_18.x | sudo -E bash -
```

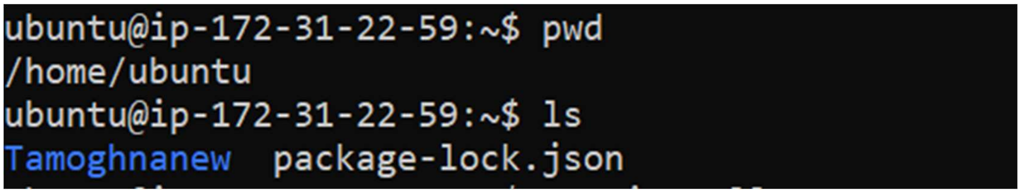


```
ubuntu@ip-172-31-22-59:~$ sudo apt install nodejs
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nodejs is already the newest version (18.20.8-1nodesource1).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
ubuntu@ip-172-31-22-59:~$ node -v
v18.20.8
ubuntu@ip-172-31-22-59:~$
```

Step 9: Now clone the GitHub repository from GitHub using command git clone paste the link of the repository.

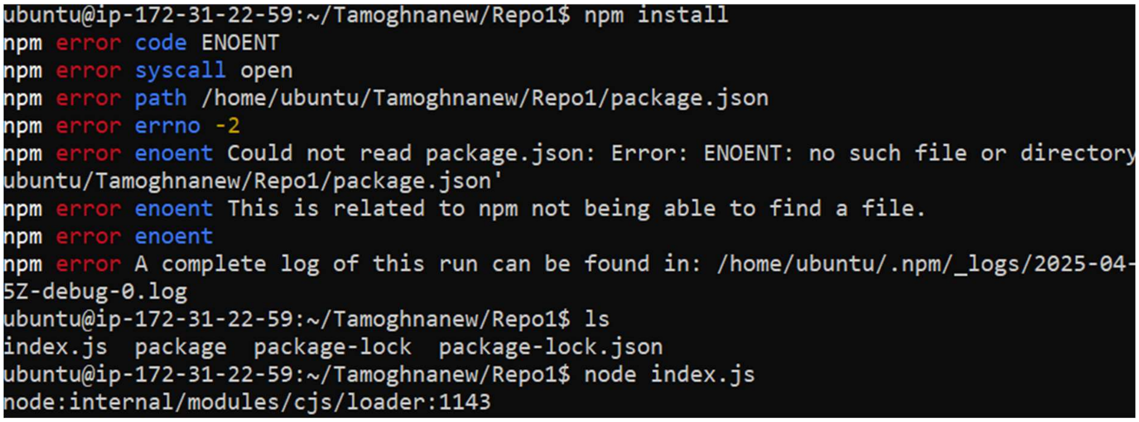


```
ubuntu@ip-172-31-22-59:~$ git clone https://github.com/TB-coder12/Tamoghnanew.git
Cloning into 'Tamoghnanew'...
remote: Enumerating objects: 41, done.
remote: Counting objects: 100% (41/41), done.
remote: Compressing objects: 100% (38/38), done.
remote: Total 41 (delta 19), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (41/41), 59.10 KiB | 6.57 MiB/s, done.
Resolving deltas: 100% (19/19), done.
```



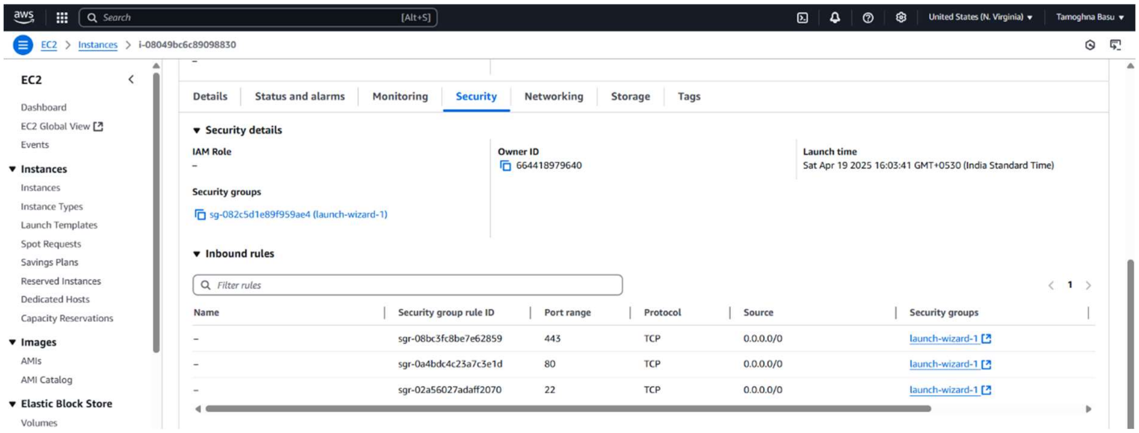
```
ubuntu@ip-172-31-22-59:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-22-59:~$ ls
Tamoghnanew  package-lock.json
```

Step 10: After installing nodejs use npm install command to install required dependencies.



```
ubuntu@ip-172-31-22-59:~/Tamoghnanew/Repo1$ npm install
npm error code ENOENT
npm error syscall open
npm error path /home/ubuntu/Tamoghnanew/Repo1/package.json
npm error errno -2
npm error enoent Could not read package.json: Error: ENOENT: no such file or directory
ubuntu/Tamoghnanew/Repo1/package.json'
npm error enoent This is related to npm not being able to find a file.
npm error enoent
npm error A complete log of this run can be found in: /home/ubuntu/.npm/_logs/2025-04-
5Z-debug-0.log
ubuntu@ip-172-31-22-59:~/Tamoghnanew/Repo1$ ls
index.js  package  package-lock  package-lock.json
ubuntu@ip-172-31-22-59:~/Tamoghnanew/Repo1$ node index.js
node:internal/modules/cjs/loader:1143
```

Step 11: Now open AWS and open the current instance then scroll down and go to security Section. Then click on the link in security groups then click on edit inbound rules.



▾ Security details

IAM Role: –

Owner ID: 664418979640

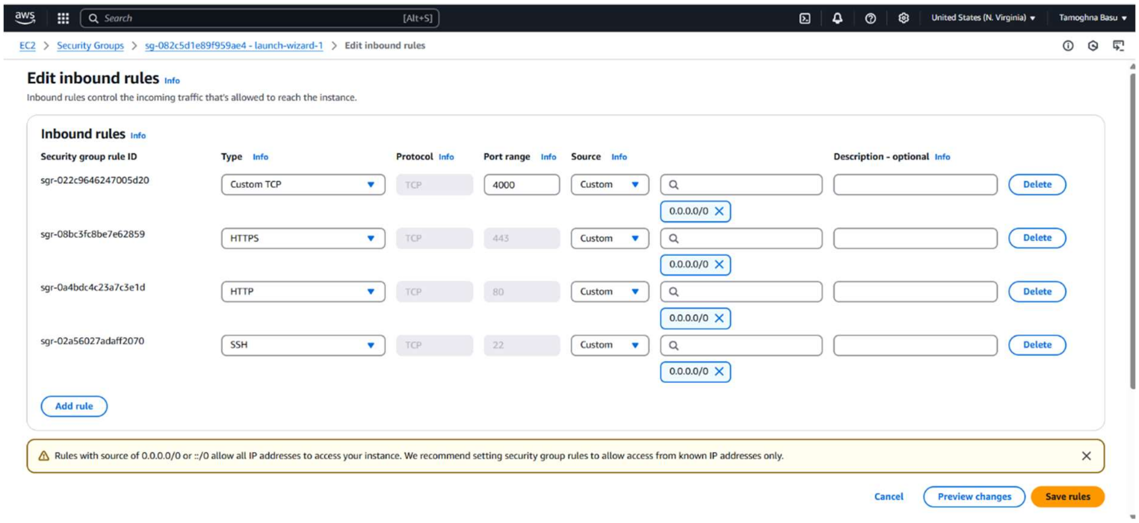
Launch time: Sat Apr 19 2025 16:03:41 GMT+0530 (India Standard Time)

Security groups: sg-082c5d1e89f959ae4 (launch-wizard-1)

▾ Inbound rules

Name	Security group rule ID	Port range	Protocol	Source	Security groups
–	sgr-08bc3fc8be7e62859	443	TCP	0.0.0.0/0	launch-wizard-1
–	sgr-0a4bdc4c23a7c3e1d	80	TCP	0.0.0.0/0	launch-wizard-1
–	sgr-02a56027adaff2070	22	TCP	0.0.0.0/0	launch-wizard-1

Step 12: Click on add rule and select the type of new rule custom TCP Port range 4000 source 0.0.0.0 or anywhere then click on save rule.



EC2 > Security Groups > sg-082c5d1e89f959ae4 - launch-wizard-1 > Edit inbound rules

Edit inbound rules Info

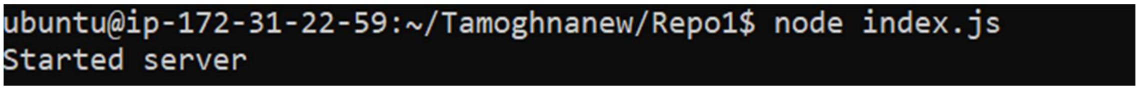
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type	Protocol	Port range	Source		Description - optional	
sgr-022c9646247005d20	Custom TCP	TCP	4000	Custom	0.0.0.0/0		Delete
sgr-08bc3fc8be7e62859	HTTPS	TCP	443	Custom	0.0.0.0/0		Delete
sgr-0a4bdc4c23a7c3e1d	HTTP	TCP	80	Custom	0.0.0.0/0		Delete
sgr-02a56027adaff2070	SSH	TCP	22	Custom	0.0.0.0/0		Delete

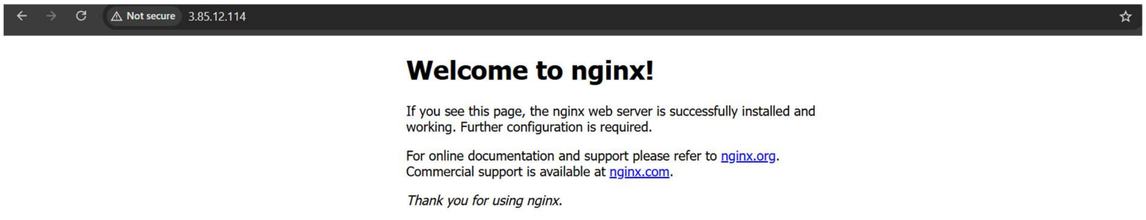
⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Step 13: Open the Bitvise SSH Client terminal console and start the server using command node index.js .



```
ubuntu@ip-172-31-22-59:~/Tamoghnanew/Repo1$ node index.js
Started server
```

Step 14: Now copy the IPv4 address of the instance and paste it on web browser.



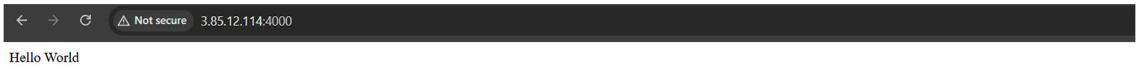
Welcome to nginx!

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.

Step 15: Then, put the port like this IP address: Port(3.85.12.114:4000)



Hello World

** Deploy a project from GitHub to EC2.

All commands:

1. **Sudo apt-get update**
2. **Sudo apt-get upgrade**
3. **nginx -v**
4. **curl -sl https://deb.nodesource.com/setup_18.x | sudo -E bash –**
5. **sudo apt install nodejs**
6. **git clone https://github.com/TB-coder12/Tamoghnanew.git**
7. **npm install**
8. **npm -v**
9. **node index.js**