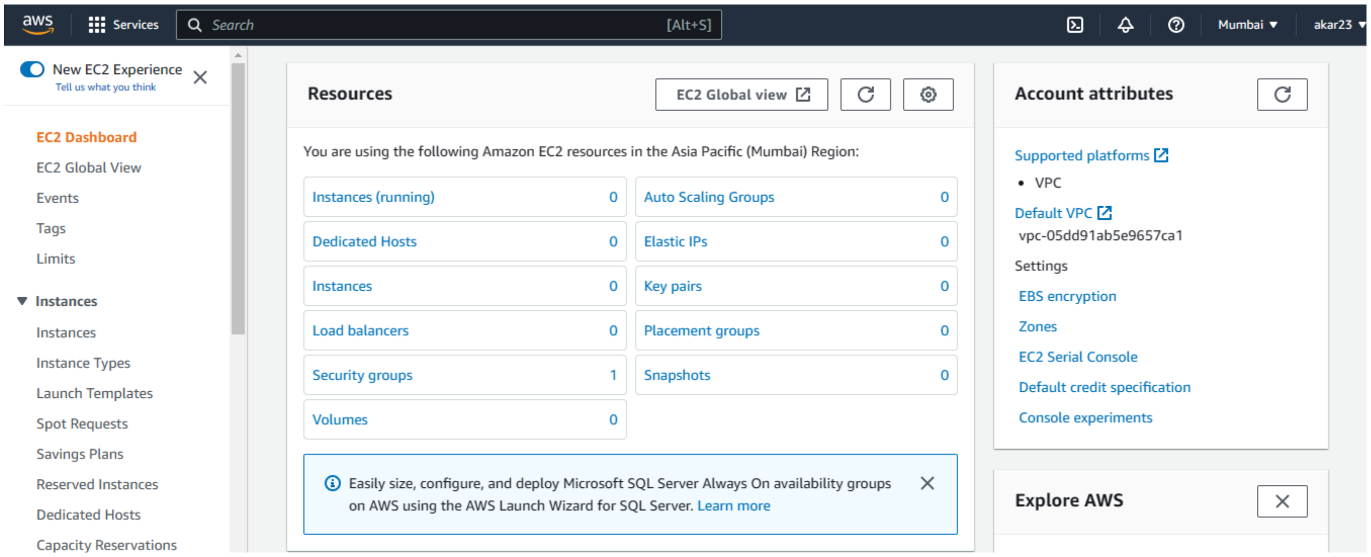
ASSIGNMENT-9

Deploy a project from GitHub to EC2.

Step 1: Sign in to your AWS Account as Root User. Go to EC2 dashboard .



You are using the following Amazon EC2 resources in the Asia Pacific (Mumbai) Region:

Instances (running)	0	Auto Scaling Groups	0
Dedicated Hosts	0	Elastic IPs	0
Instances	0	Key pairs	0
Load balancers	0	Placement groups	0
Security groups	1	Snapshots	0
Volumes	0		

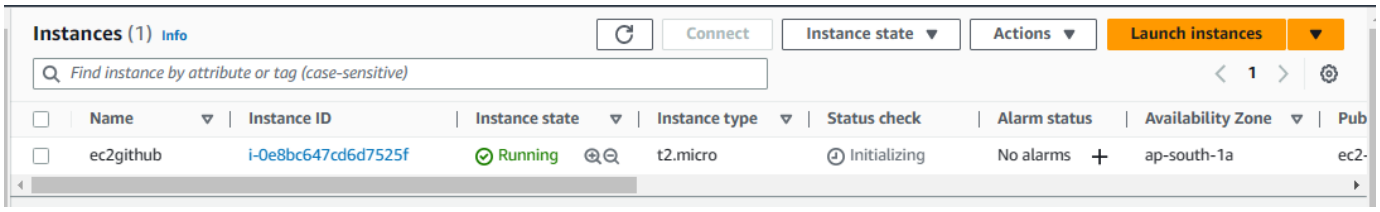
Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more

Supported platforms
- VPC

Default VPC
vpc-05dd91ab5e9657ca1

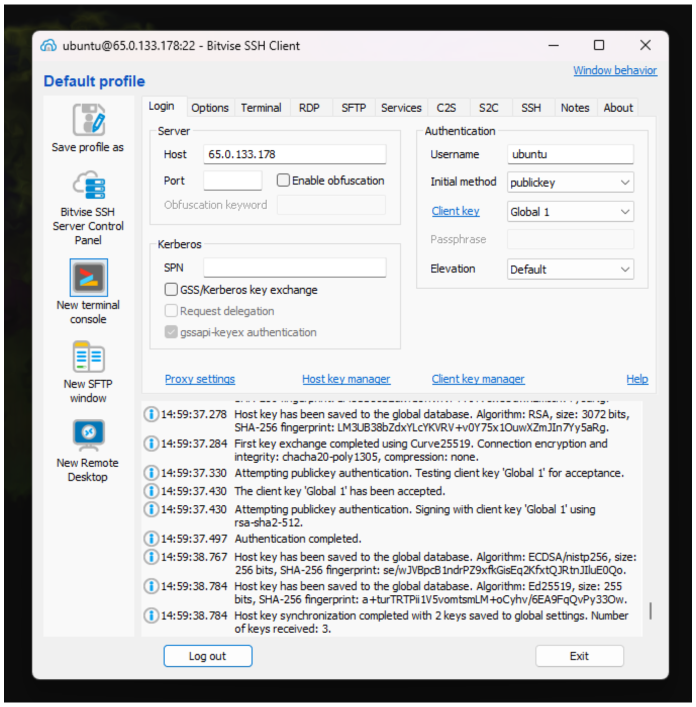
Settings
EBS encryption
Zones
EC2 Serial Console
Default credit specification
Console experiments

Step 2: Launch an instance in EC2 (Refer to Assignment no. 7).



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Pub
ec2github	i-0e8bc647cd6d7525f	Running	t2.micro	Initializing	No alarms	ap-south-1a	ec2-

Step 3: Open the software "Bitvise SSH Client" on your machine. Provide Host Server(public IPv4 address of your EC2 instance) and log in (Refer to Assignment no. 7).


ubuntu@65.0.133.178:22 - Bitvise SSH Client

Default profile

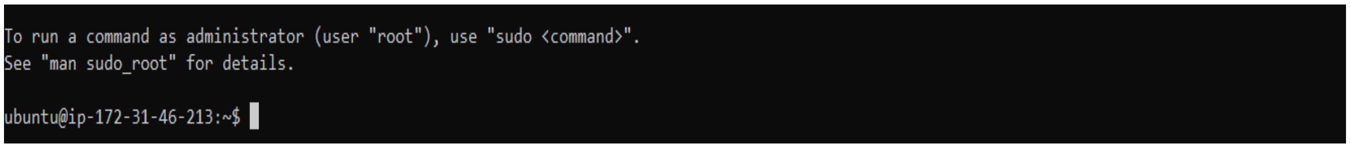
Server
Host: 65.0.133.178
Port:
[ ] Enable obfuscation

Authentication
Username: ubuntu
Initial method: publickey
Client key: Global 1
Elevation: Default

Kerberos
SPN:
[ ] GSS/Kerberos key exchange
[ ] Request delegation
[✓] gssapi-keyex authentication

```
14:59:37.278 Host key has been saved to the global database. Algorithm: RSA, size: 3072 bits,
             SHA-256 fingerprint: LM3UB38bZdxYLcYKVRV+v0Y75x1OuwXZmJIn7Yy5aRg.
14:59:37.284 First key exchange completed using Curve25519. Connection encryption and
             integrity: chacha20-poly1305, compression: none.
14:59:37.330 Attempting publickey authentication. Testing client key 'Global 1' for acceptance.
14:59:37.430 The client key 'Global 1' has been accepted.
14:59:37.430 Attempting publickey authentication. Signing with client key 'Global 1' using
             rsa-sha2-512.
14:59:37.497 Authentication completed.
14:59:38.767 Host key has been saved to the global database. Algorithm: ECDSA/nistp256, size:
             256 bits, SHA-256 fingerprint: se/wJVBpcB1ndrPZ9xfkGisEq2KfxtQJRtnJIluE0Qo.
14:59:38.784 Host key has been saved to the global database. Algorithm: Ed25519, size: 255
             bits, SHA-256 fingerprint: a+turTRTPii1V5vomtsmLM+oCyhv/6EA9FqQvPy33Ow.
14:59:38.784 Host key synchronization completed with 2 keys saved to global settings. Number
             of keys received: 3.
```

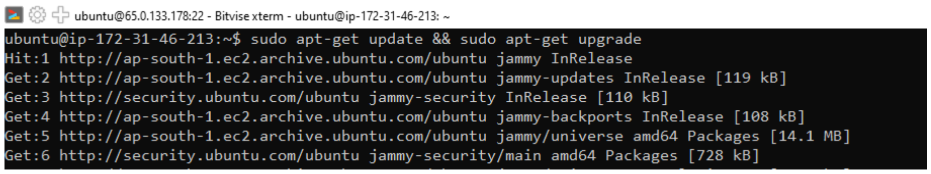
Step 4: Now go to "New Terminal console".



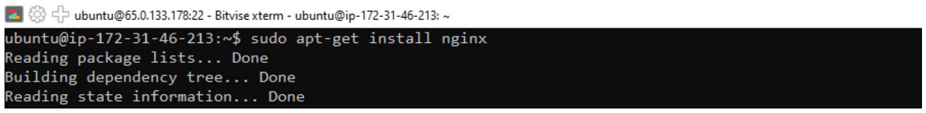
```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-46-213:~$
```

Step 5: Update and upgrade our server. Then install "nginx" and you can check its version using the following commands (as done in Assignment no. 7):

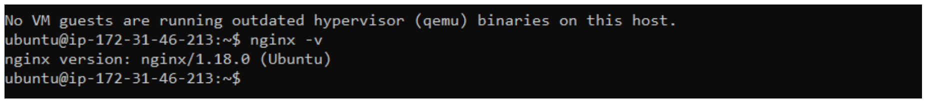
- **sudo apt-get update && sudo apt-get upgrade** – To update and upgrade.
- **sudo apt-get install nginx** – To install nginx.
- **nginx -v** – To check the version of nginx.


ubuntu@65.0.133.178:22 - Bitvise xterm - ubuntu@ip-172-31-46-213: ~

```
ubuntu@ip-172-31-46-213:~$ sudo apt-get update && sudo apt-get upgrade
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [728 kB]
```

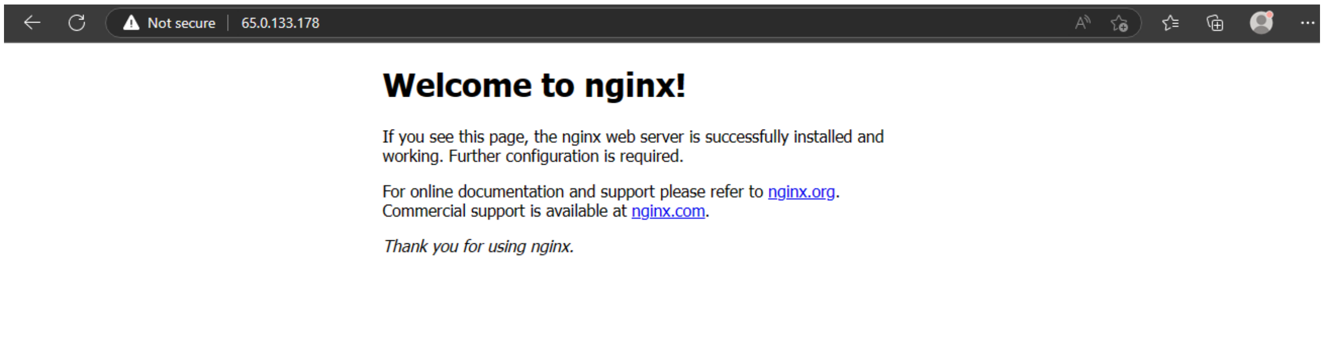

ubuntu@65.0.133.178:22 - Bitvise xterm - ubuntu@ip-172-31-46-213: ~

```
ubuntu@ip-172-31-46-213:~$ sudo apt-get install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```



```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-46-213:~$ nginx -v
nginx version: nginx/1.18.0 (Ubuntu)
ubuntu@ip-172-31-46-213:~$
```

Step 6: Now paste the public IPv4 address of your EC2 instance in a new browser to check whether nginx server is working or not.



Welcome to nginx!

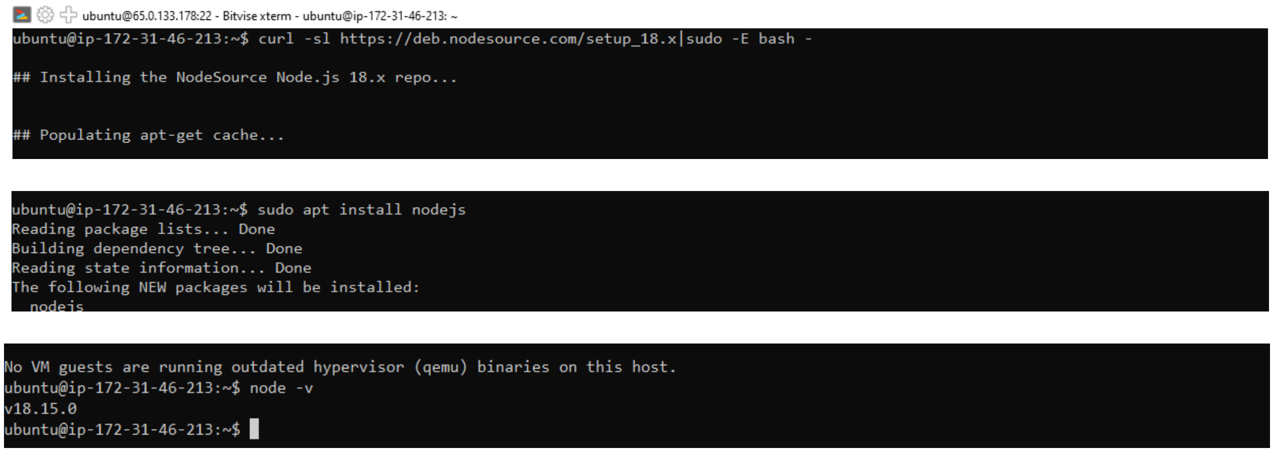
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Step 7: Download .exe file for nodejs. Install it and check its version using the following commands:

- **curl -sl https://deb.nodesource.com/setup_18.x|sudo -E bash** – To download nodejs files with all dependencies in our server system.
- **sudo apt install nodejs** – To install nodejs in our server system.
- **node -v** – To check its version.

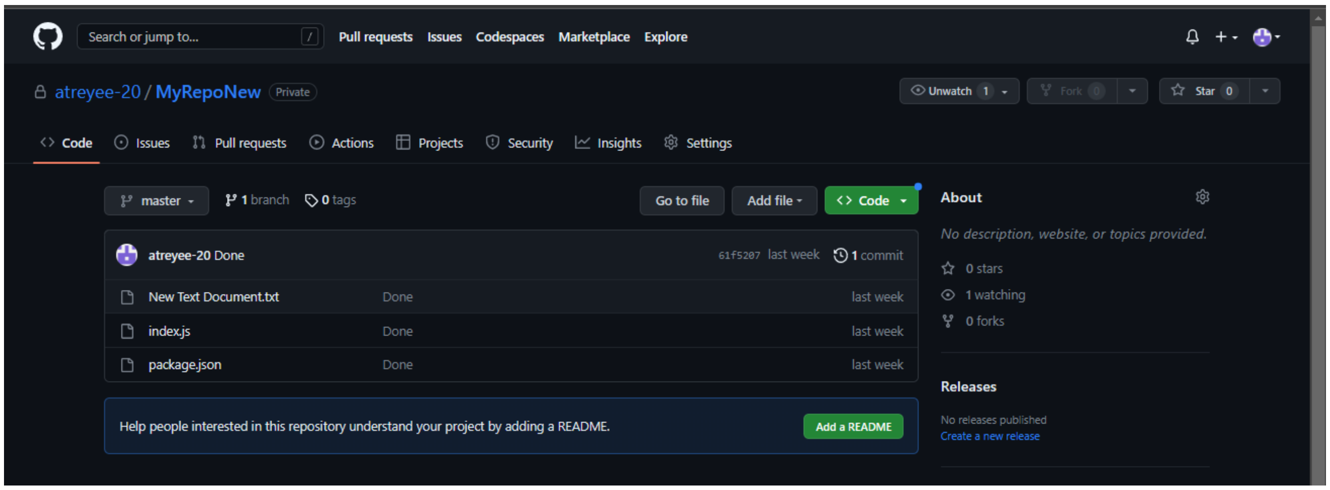

ubuntu@65.0.133.178:22 - Bitvise xterm - ubuntu@ip-172-31-46-213: ~

```
ubuntu@ip-172-31-46-213:~$ curl -sl https://deb.nodesource.com/setup_18.x|sudo -E bash -

## Installing the NodeSource Node.js 18.x repo...


## Populating apt-get cache...
```

```
ubuntu@ip-172-31-46-213:~$ sudo apt install nodejs
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  nodejs
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-46-213:~$ node -v
v18.15.0
ubuntu@ip-172-31-46-213:~$
```

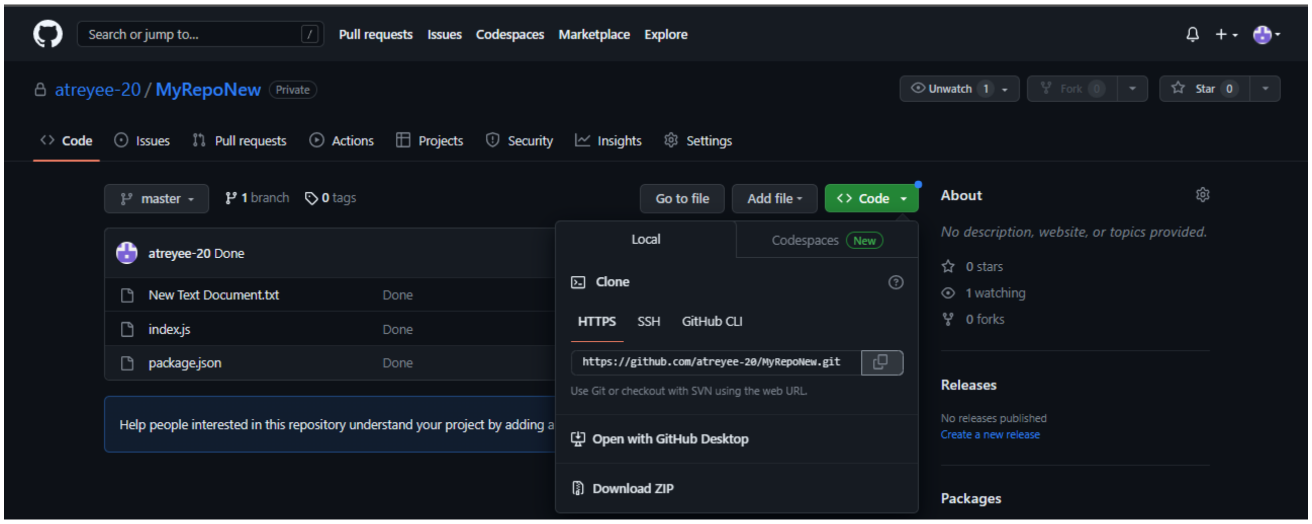
Step 8: Now sign in to your GitHub Account and open your repository.



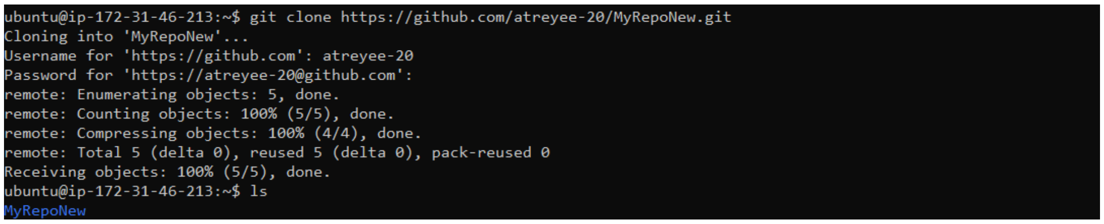
atreyee-20 Done	61f5207 last week	1 commit
New Text Document.txt	Done	last week
index.js	Done	last week
package.json	Done	last week

Help people interested in this repository understand your project by adding a README.

Step 9: Now click on "Code" and copy the HTTPS address.

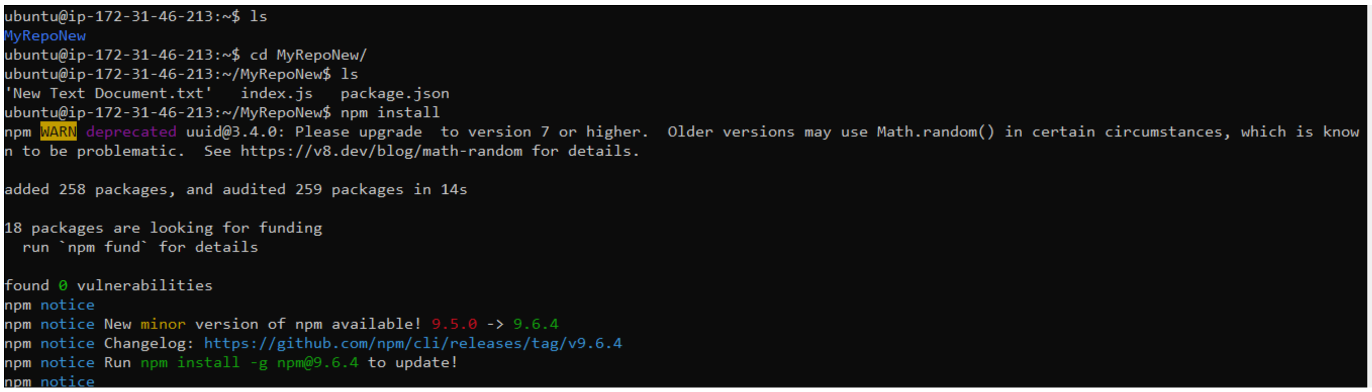


Step 10: Now we have to retrieve the project from GitHub using the URL. The command used for doing it is : **git clone <The URL of the project>.** Provide your Username and Password (Account Token which was created earlier). We can see a directory is created by our repository name.



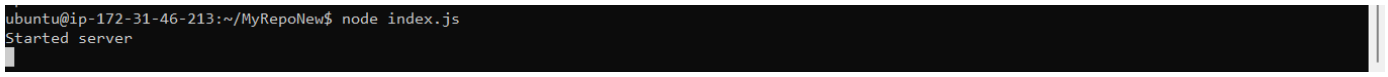
```
ubuntu@ip-172-31-46-213:~$ git clone https://github.com/atreyee-20/MyRepoNew.git
Cloning into 'MyRepoNew'...
Username for 'https://github.com': atreyee-20
Password for 'https://atreyee-20@github.com':
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 5 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
ubuntu@ip-172-31-46-213:~$ ls
MyRepoNew
```

Step 11: Now go to the directory which has been cloned and install the npm package manager using the command : **npm install**.



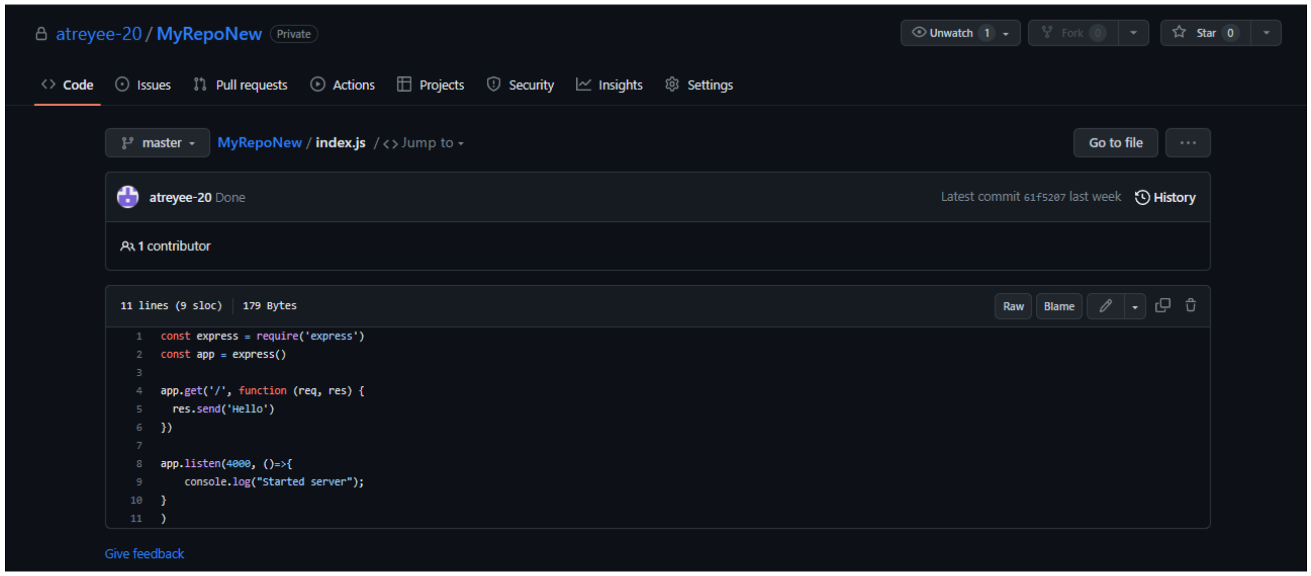
```
ubuntu@ip-172-31-46-213:~$ ls
MyRepoNew
ubuntu@ip-172-31-46-213:~$ cd MyRepoNew/
ubuntu@ip-172-31-46-213:~/MyRepoNew$ ls
'New Text Document.txt'   index.js   package.json
ubuntu@ip-172-31-46-213:~/MyRepoNew$ npm install
npm WARN deprecated uuid@3.4.0: Please upgrade  to version 7 or higher.  Older versions may use Math.random() in certain circumstances, which is know
n to be problematic.  See https://v8.dev/blog/math-random for details.

added 258 packages, and audited 259 packages in 14s

18 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 9.5.0 -> 9.6.4
npm notice Changelog: https://github.com/npm/cli/releases/tag/v9.6.4
npm notice Run npm install -g npm@9.6.4 to update!
npm notice
```

Step 12: Now start the server using this command: **node <.js file name>**



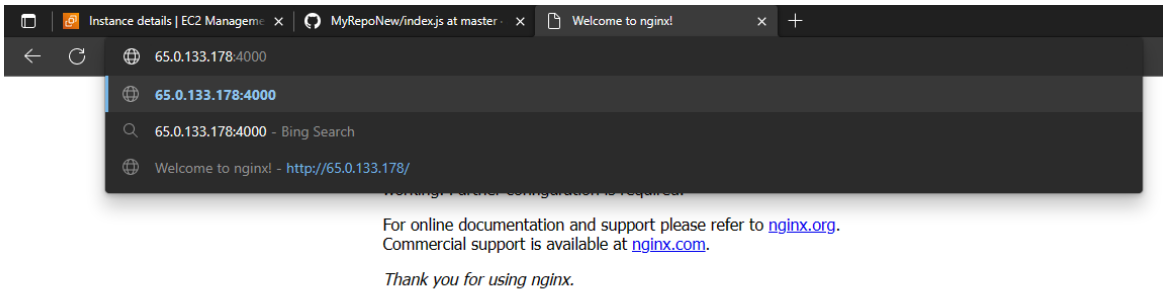
```
ubuntu@ip-172-31-46-213:~/MyRepoNew$ node index.js
Started server
```

Step 13: Go to your repository in your GitHub and open the .js file. Here, it is "index.js". Check out the post number (The number specified in app.listen()). Here it is "4000".



```
const express = require('express')
const app = express()

app.get('/', function (req, res) {
  res.send('Hello')
})

app.listen(4000, ()=>{
    console.log("Started server");
}
)
```

Step 14: Open the nginx window. Go to the IP address, put a colon and give the port number.

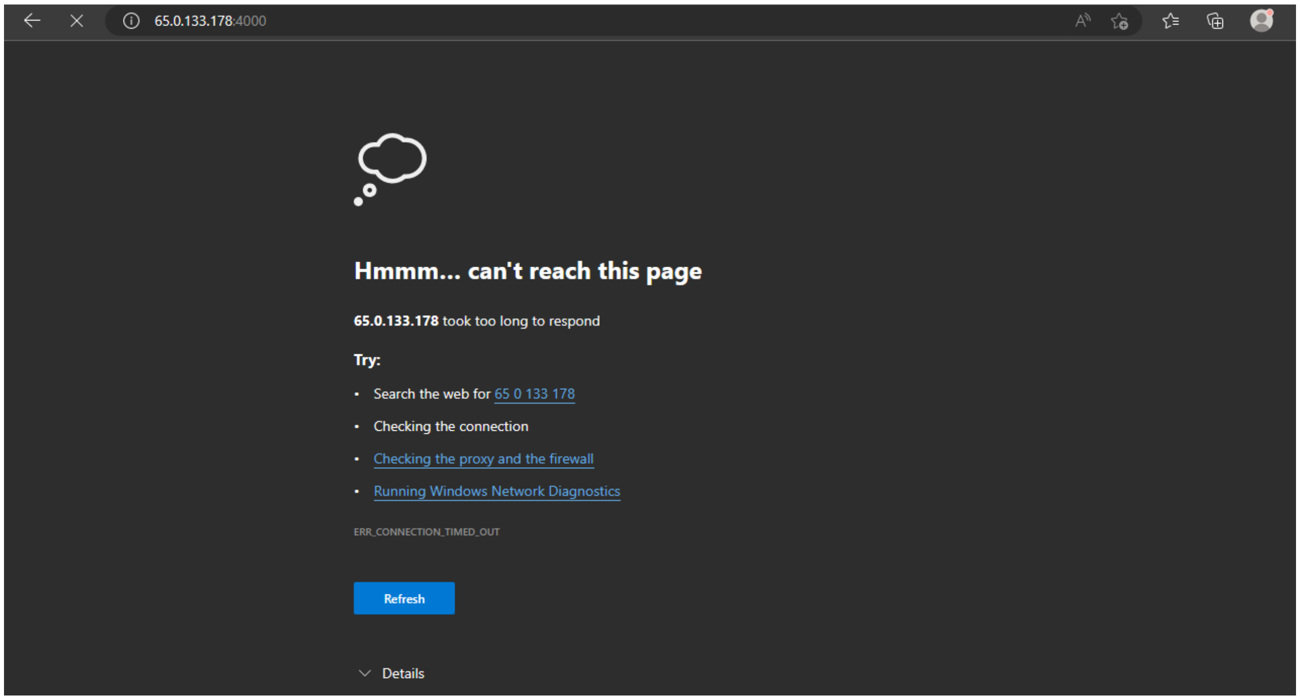


Instance details | EC2 Manageme
MyRepoNew/index.js at master
Welcome to nginx!
65.0.133.178:4000
65.0.133.178:4000
65.0.133.178:4000 - Bing Search
Welcome to nginx! - http://65.0.133.178/

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Step 15: We cannot reach the page.


65.0.133.178:4000

Hmmm... can't reach this page

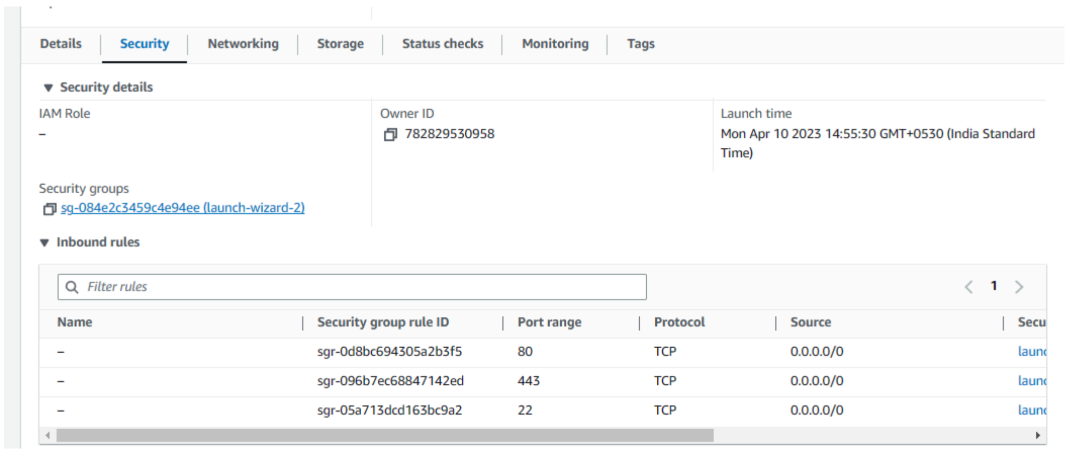
65.0.133.178 took too long to respond

Try:

- Search the web for 65 0 133 178
- Checking the connection
- Checking the proxy and the firewall
- Running Windows Network Diagnostics

ERR_CONNECTION_TIMED_OUT

Step 16: Now open your EC2 instance and go to Security. In Security we can see there is a Security group . Click on it.



Security details

IAM Role: –

Owner ID: 782829530958

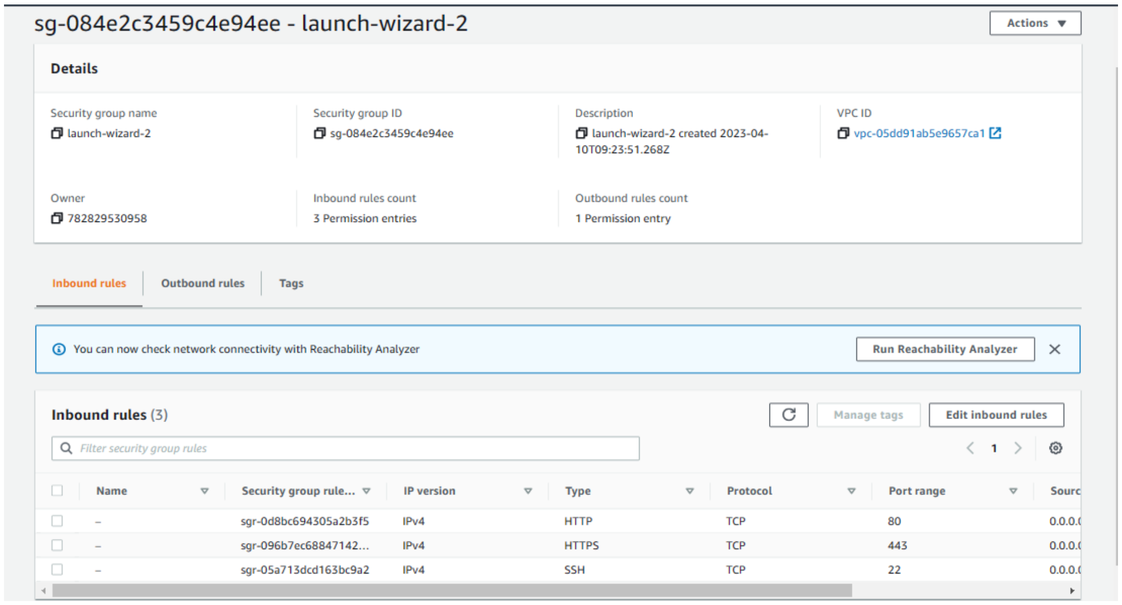
Launch time: Mon Apr 10 2023 14:55:30 GMT+0530 (India Standard Time)

Security groups: sg-084e2c3459c4e94ee (launch-wizard-2)

Inbound rules

Name	Security group rule ID	Port range	Protocol	Source	Secu
–	sgr-0d8bc694305a2b3f5	80	TCP	0.0.0.0/0	launc
–	sgr-096b7ec68847142ed	443	TCP	0.0.0.0/0	launc
–	sgr-05a713dcd163bc9a2	22	TCP	0.0.0.0/0	launc

Step 17: Now go to "Edit Inbound rules".



sg-084e2c3459c4e94ee - launch-wizard-2

Details

Security group name	Security group ID	Description	VPC ID
launch-wizard-2	sg-084e2c3459c4e94ee	launch-wizard-2 created 2023-04-10T09:23:51.268Z	vpc-05dd91ab5e9657ca1

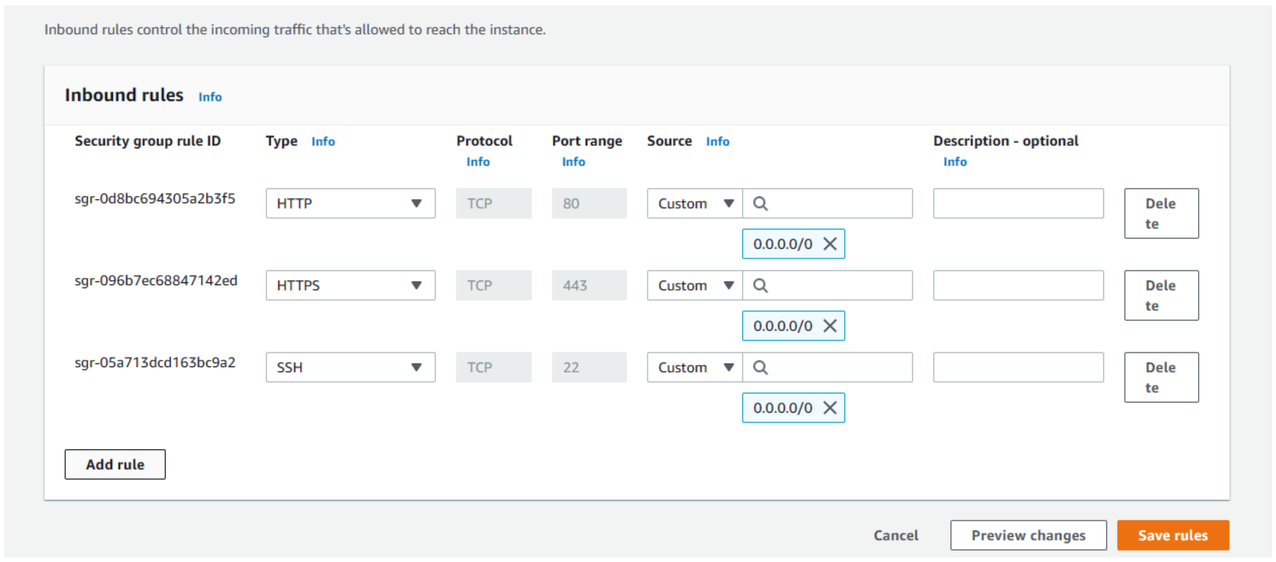
Owner	Inbound rules count	Outbound rules count
782829530958	3 Permission entries	1 Permission entry

You can now check network connectivity with Reachability Analyzer

Inbound rules (3)

	Name	Security group rule...	IP version	Type	Protocol	Port range	Sourc
	–	sgr-0d8bc694305a2b3f5	IPv4	HTTP	TCP	80	0.0.0.(
	–	sgr-096b7ec68847142...	IPv4	HTTPS	TCP	443	0.0.0.(
	–	sgr-05a713dcd163bc9a2	IPv4	SSH	TCP	22	0.0.0.(

Step 18: Click on "Add rule".

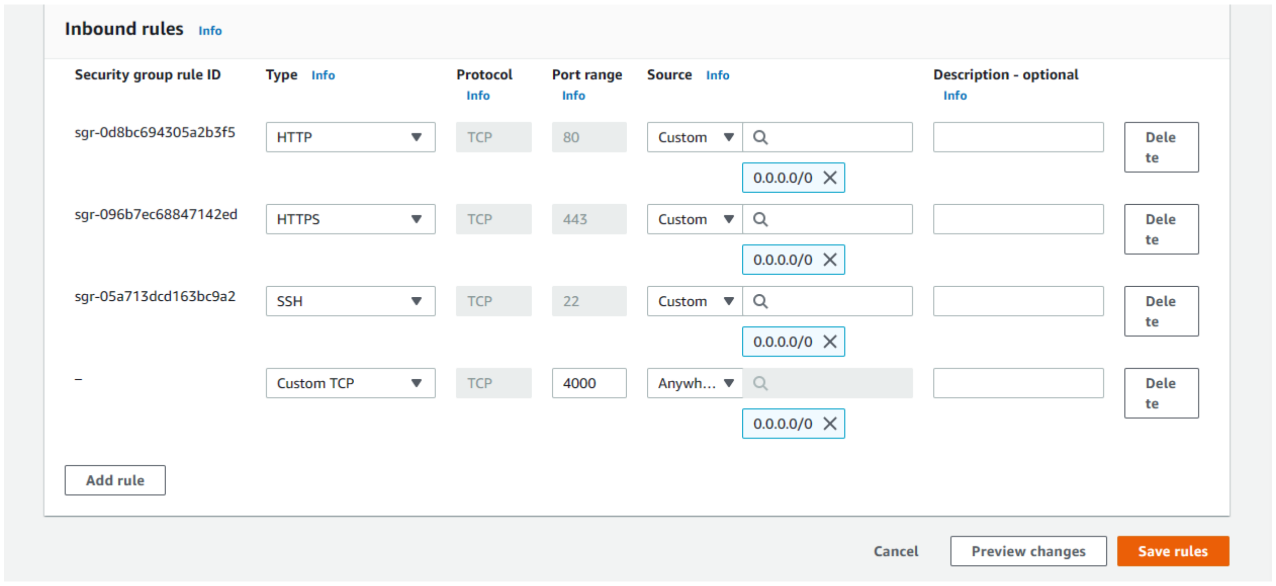


Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type	Protocol	Port range	Source	Description - optional	
sgr-0d8bc694305a2b3f5	HTTP	TCP	80	Custom 0.0.0.0/0		Delete
sgr-096b7ec68847142ed	HTTPS	TCP	443	Custom 0.0.0.0/0		Delete
sgr-05a713dcd163bc9a2	SSH	TCP	22	Custom 0.0.0.0/0		Delete

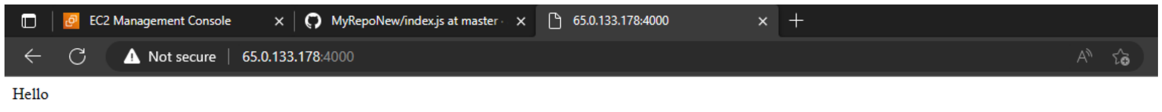
<u>Step 19:</u> Give Type: Custom TCP, Port range: 4000, Source: Anywhere. Then "Save rules".



Inbound rules Info

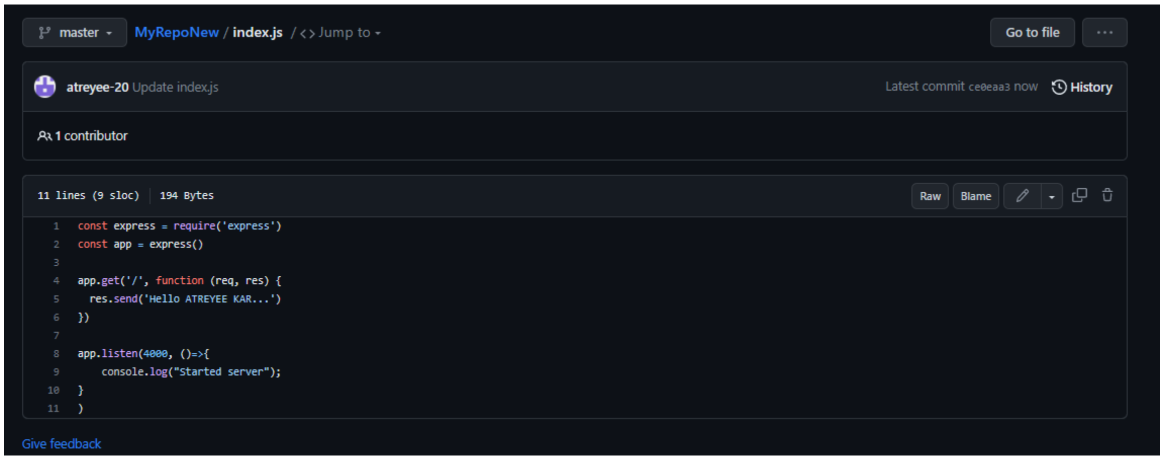
Security group rule ID	Type	Protocol	Port range	Source	Description - optional	
sgr-0d8bc694305a2b3f5	HTTP	TCP	80	Custom 0.0.0.0/0		Delete
sgr-096b7ec68847142ed	HTTPS	TCP	443	Custom 0.0.0.0/0		Delete
sgr-05a713dcd163bc9a2	SSH	TCP	22	Custom 0.0.0.0/0		Delete
–	Custom TCP	TCP	4000	Anywh... 0.0.0.0/0		Delete

<u>Step 20:</u> Now refresh the nginx window. We can see the content.



Hello

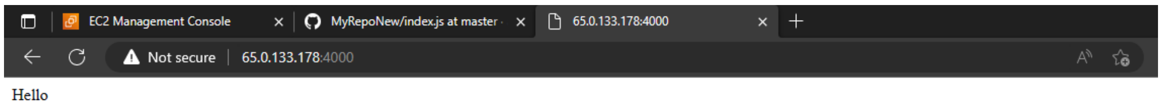
<u>Step 21:</u> Now go to the terminal and terminate the server using **Ctrl+C**. Then open the .js file in your repository and modify the content. Then "Commit changes".



master
MyRepoNew / index.js / <> Jump to
Go to file
atreyee-20 Update index.js
Latest commit ce0eaa3 now
History
1 contributor
11 lines (9 sloc) | 194 Bytes
Raw
Blame
Give feedback

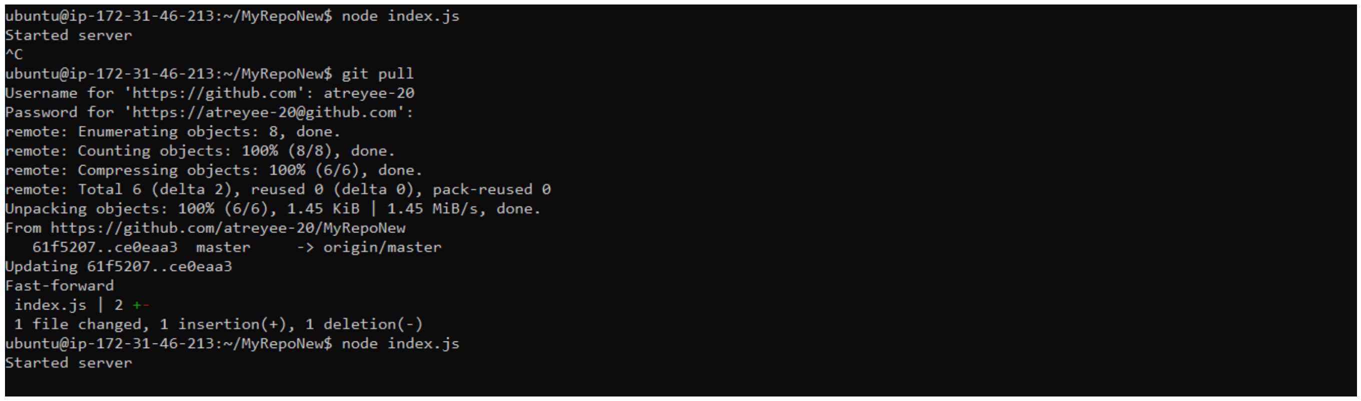
```
const express = require('express')
const app = express()

app.get('/', function (req, res) {
  res.send('Hello ATREYEE KAR...')
})

app.listen(4000, ()=>{
    console.log("Started server");
}
)
```

<u>Step 22:</u> Again start the server and refresh the nginx page. No changes can be seen.



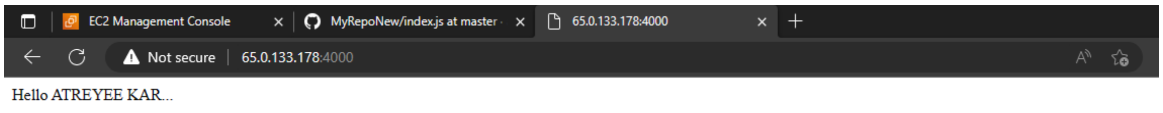
Hello

<u>Step 23:</u> Now go to the terminal and terminate the server. We have to "pull" the new updated files into the repository directory in our Remote Server. To do so, we have to write the command: **git pull.** Then again start the server.



```
ubuntu@ip-172-31-46-213:~/MyRepoNew$ node index.js
Started server
^C
ubuntu@ip-172-31-46-213:~/MyRepoNew$ git pull
Username for 'https://github.com': atreyee-20
Password for 'https://atreyee-20@github.com':
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), 1.45 KiB | 1.45 MiB/s, done.
From https://github.com/atreyee-20/MyRepoNew
   61f5207..ce0eaa3  master     -> origin/master
Updating 61f5207..ce0eaa3
Fast-forward
 index.js | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
ubuntu@ip-172-31-46-213:~/MyRepoNew$ node index.js
Started server
```

<u>Step 24:</u> Now refresh the nginx page. Our modified content can be seen now.



Hello ATREYEE KAR...

Thus, we have successfully deployed a project from GitHub to EC2.