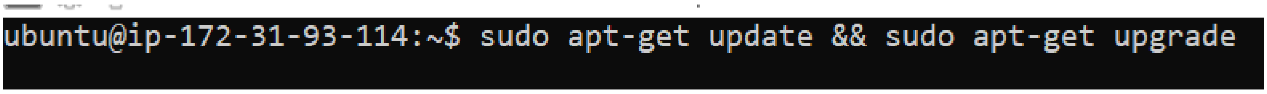
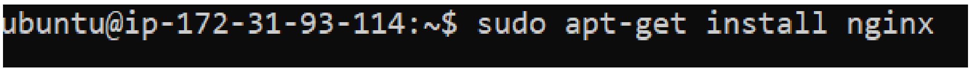
Assignment 9: Deploy a project from GitHub to EC2.

1. Create a new EC2 instance and log in to the bitvise SSH client with the public IP address and key-value pair.
2. After log in run the command ‘sudo apt-get update’ after that run ‘sudo apt-get upgrade’



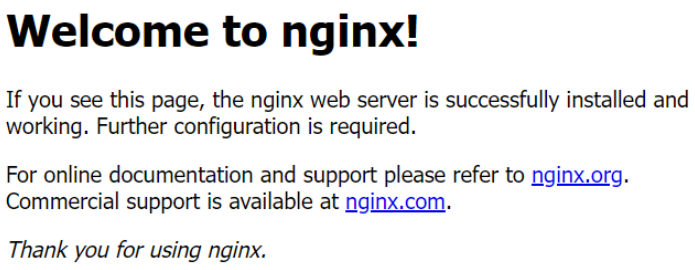
```
ubuntu@ip-172-31-93-114:~$ sudo apt-get update && sudo apt-get upgrade
```

3. After that install the nginx web server in the machine by running the command ‘sudo apt-get install nginx’.



```
ubuntu@ip-172-31-93-114:~$ sudo apt-get install nginx
```

4. Then paste the public IP address of the EC2 instance in some browser and check if the page is coming or not.



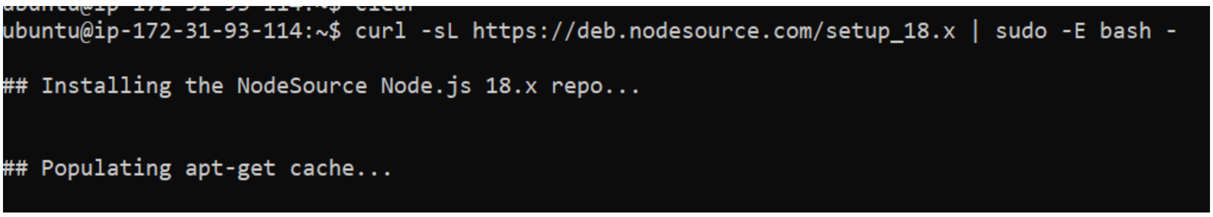
Welcome to nginx!

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

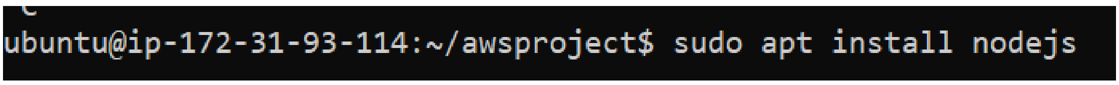
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

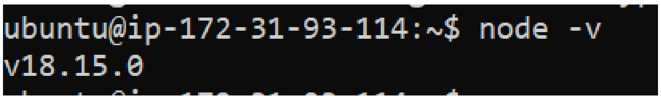
5. In order to run the project that we will be importing from GitHub we need to install node js in our machine for that run the command
   - ‘curl -sL https://deb.nodesource.com/setup_18.x|sudo -E bash-‘ .
   - ‘sudo apt-get install node js’
   - ‘node -v’



```
ubuntu@ip-172-31-93-114:~$ 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-172-31-93-114:~/awsproject$ sudo apt install nodejs
```

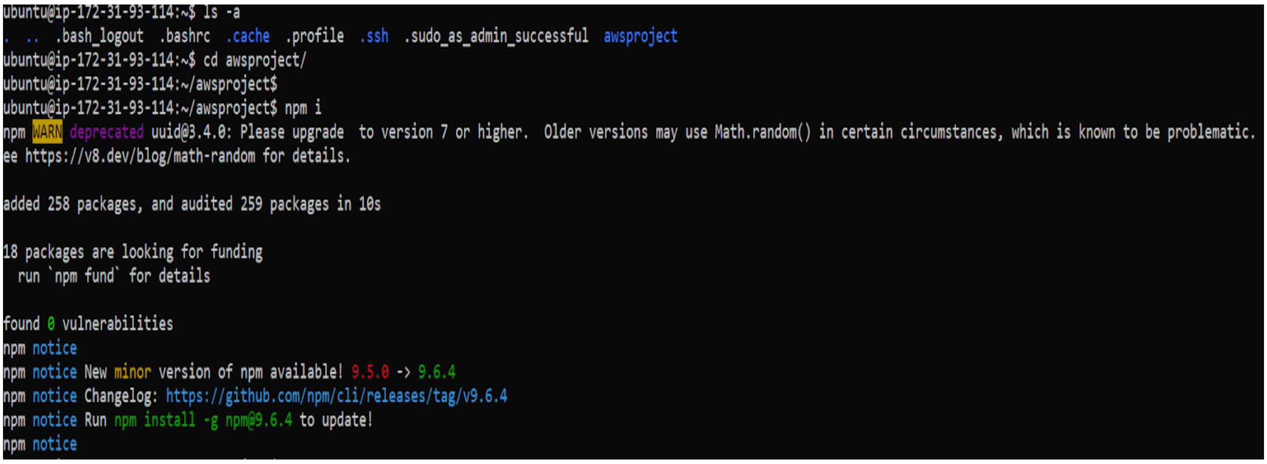


```
ubuntu@ip-172-31-93-114:~$ node -v
v18.15.0
```

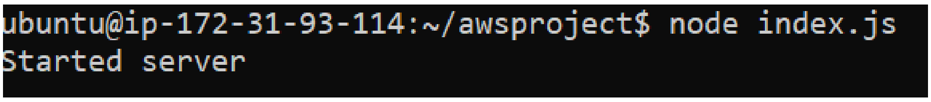
6. Now we need to clone the project repo in our own machine so run the command ‘git clone <origin link or the project>’ in our case ‘git clone
Now it will ask for a GitHub username and password.

Give your GitHub username there and in the password give the access token got from GitHub.

7. Now run the command ‘ls’ then go to the cloned repo and run ‘npm i’ to install all the necessary packages needed to run the project. Then to start the project run the command ‘node index.js’.

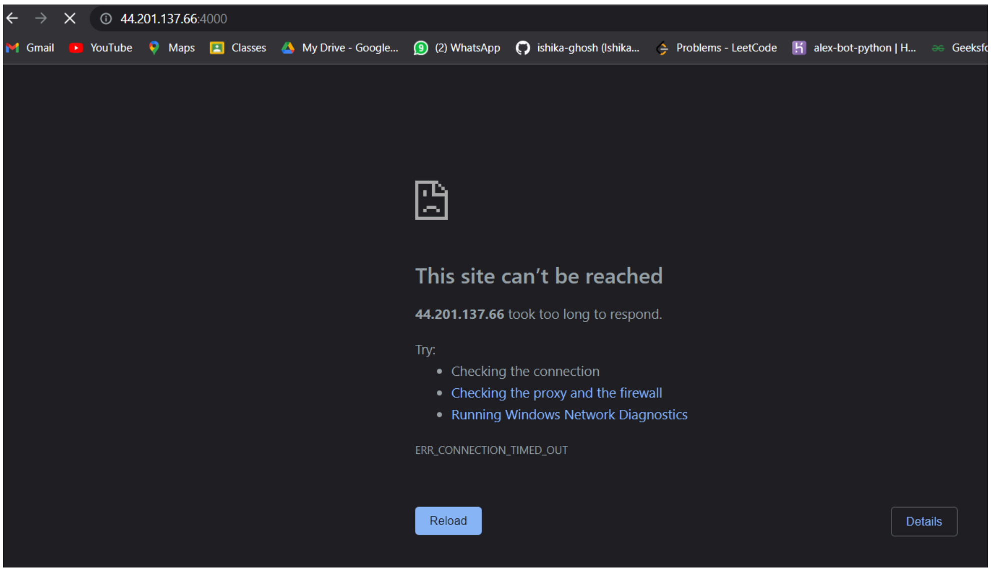


```
ubuntu@ip-172-31-93-114:~$ ls -a
.  ..  .bash_logout  .bashrc  .cache  .profile  .ssh  .sudo_as_admin_successful  awsproject
ubuntu@ip-172-31-93-114:~$ cd awsproject/
ubuntu@ip-172-31-93-114:~/awsproject$
ubuntu@ip-172-31-93-114:~/awsproject$ npm i
npm WARN deprecated uuid@3.4.0: Please upgrade  to version 7 or higher.  Older versions may use Math.random() in certain circumstances, which is known to be problematic.
ee https://v8.dev/blog/math-random for details.

added 258 packages, and audited 259 packages in 10s

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
```



```
ubuntu@ip-172-31-93-114:~/awsproject$ node index.js
Started server
```

8. Now, if we refresh the browser, we will not see the response. We have to add the port number in the instance to see the response.



This site can’t be reached

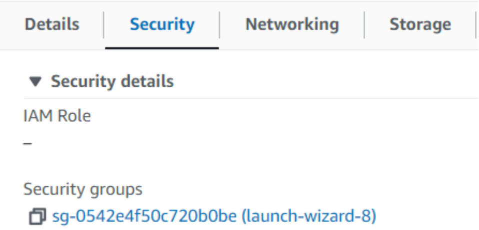
44.201.137.66 took too long to respond.

Try:

- Checking the connection
- Checking the proxy and the firewall
- Running Windows Network Diagnostics

ERR_CONNECTION_TIMED_OUT

- Go to instance security.



▼ Security details

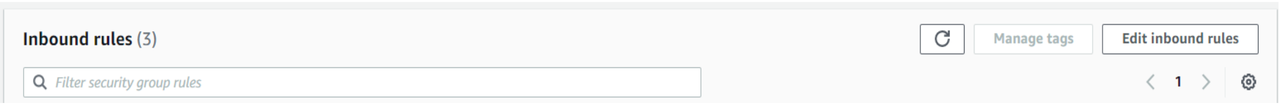
IAM Role

–

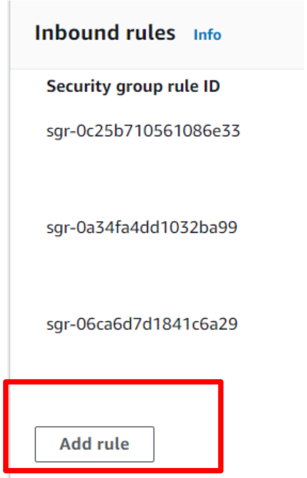
Security groups

sg-0542e4f50c720b0be (launch-wizard-8)

- Then click **edit inbound rules.**



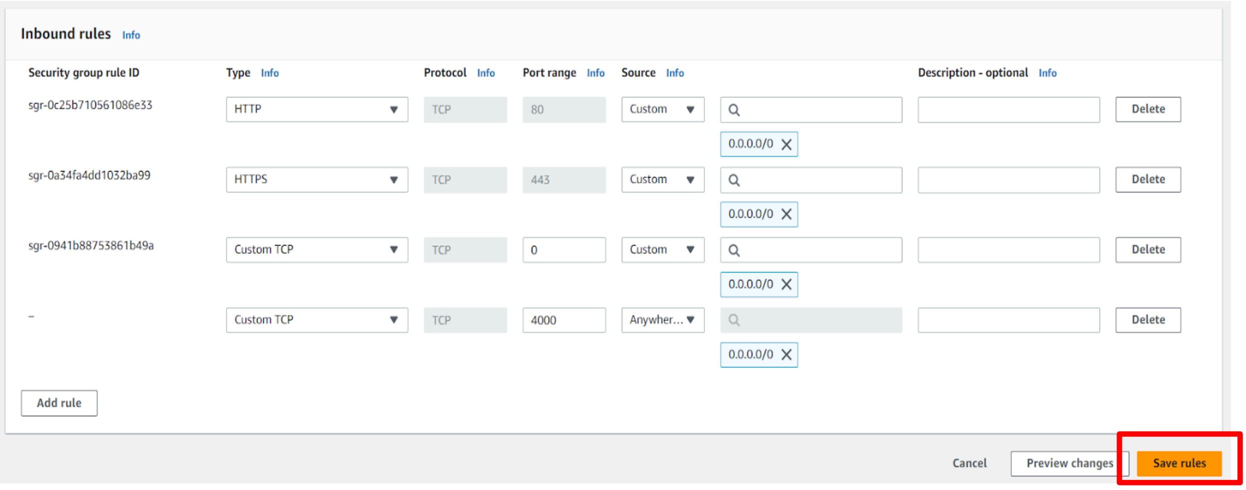
- Then click **ADD RULE.**



Inbound rules Info

Security group rule ID
sgr-0c25b710561086e33
sgr-0a34fa4dd1032ba99
sgr-06ca6d7d1841c6a29

- Give the port range 4000.
- In the field source info select **Anywhere IPv4.**
- Click **save rules.**

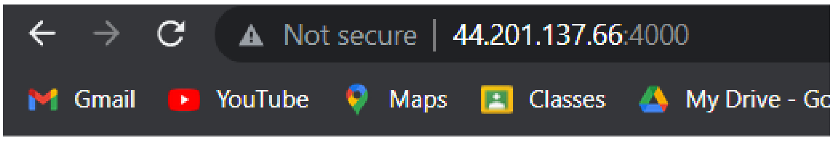


Inbound rules Info

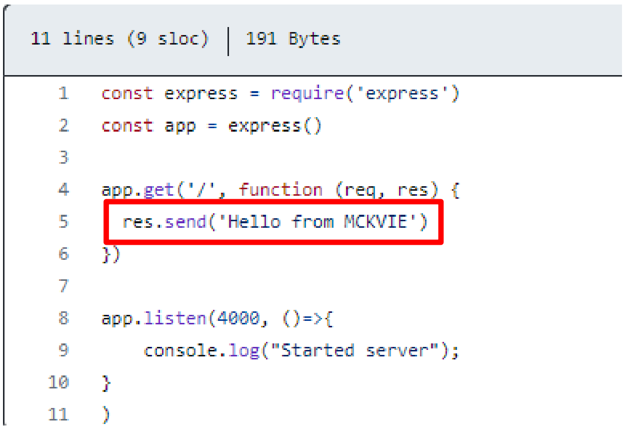
Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-0c25b710561086e33	HTTP	TCP	80	Custom	0.0.0.0/0		Delete
sgr-0a34fa4dd1032ba99	HTTPS	TCP	443	Custom	0.0.0.0/0		Delete
sgr-0941b88753861b49a	Custom TCP	TCP	0	Custom	0.0.0.0/0		Delete
-	Custom TCP	TCP	4000	Anywher...	0.0.0.0/0		Delete

- You will be seeing the port number in the inbound rules.

9. Now go to the browser and add a ':4000' at the end of the public IP v4 address and refresh you can see the response.



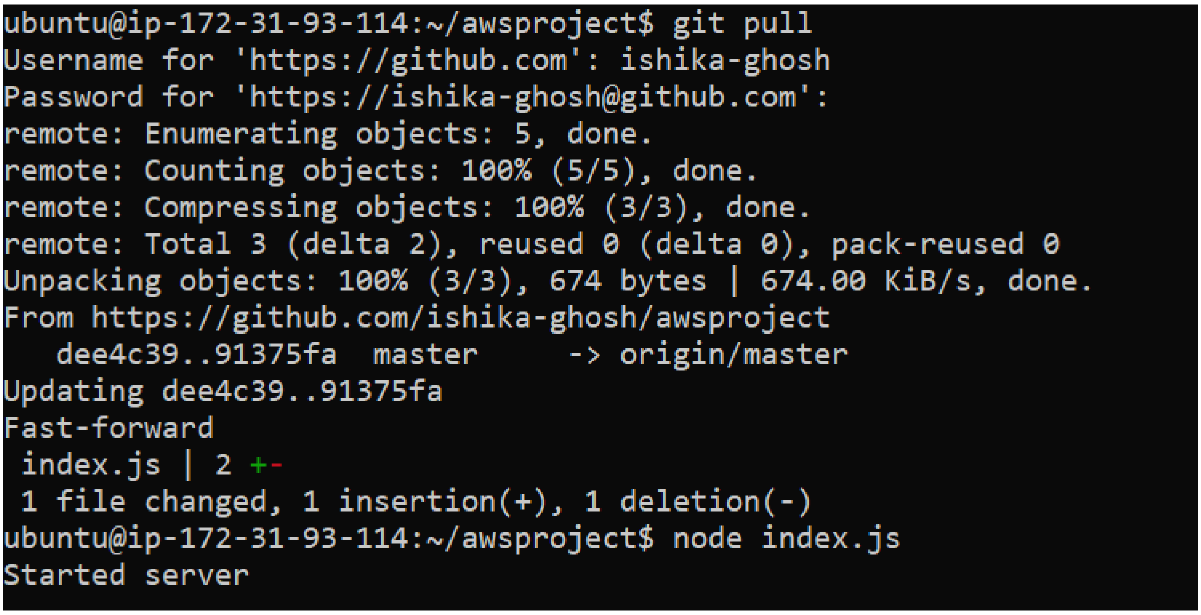
10.Now go to the GitHub repo and change something in the index.js.



```
11 lines (9 sloc) | 191 Bytes
```

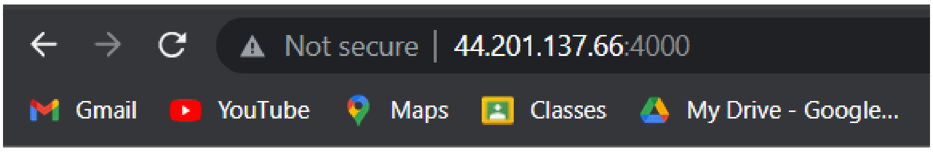
```
const express = require('express')
const app = express()

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

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

Then come back to the terminal and terminate the currently running program. Then run ‘git pull’ and pull the changes from the GitHub then again start the index.js using ‘node index.js’.



```
ubuntu@ip-172-31-93-114:~/awsproject$ git pull
Username for 'https://github.com': ishika-ghosh
Password for 'https://ishika-ghosh@github.com':
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 674 bytes | 674.00 KiB/s, done.
From https://github.com/ishika-ghosh/awsproject
   dee4c39..91375fa  master     -> origin/master
Updating dee4c39..91375fa
Fast-forward
 index.js | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
ubuntu@ip-172-31-93-114:~/awsproject$ node index.js
Started server
```

11. Now go to the browser and refresh to see the change.



Hello from MCKVIE