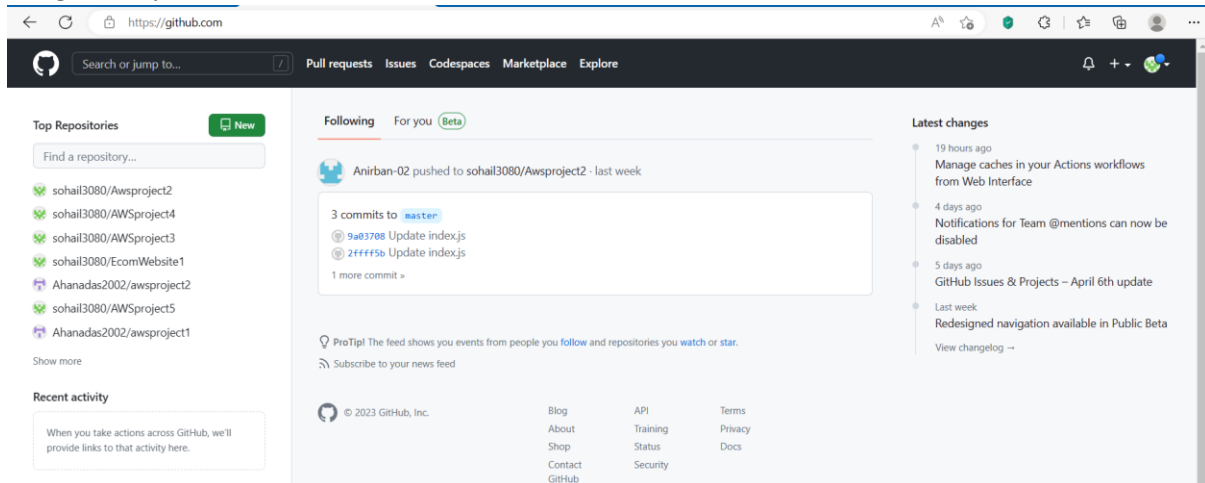


ASSIGNMENT 9

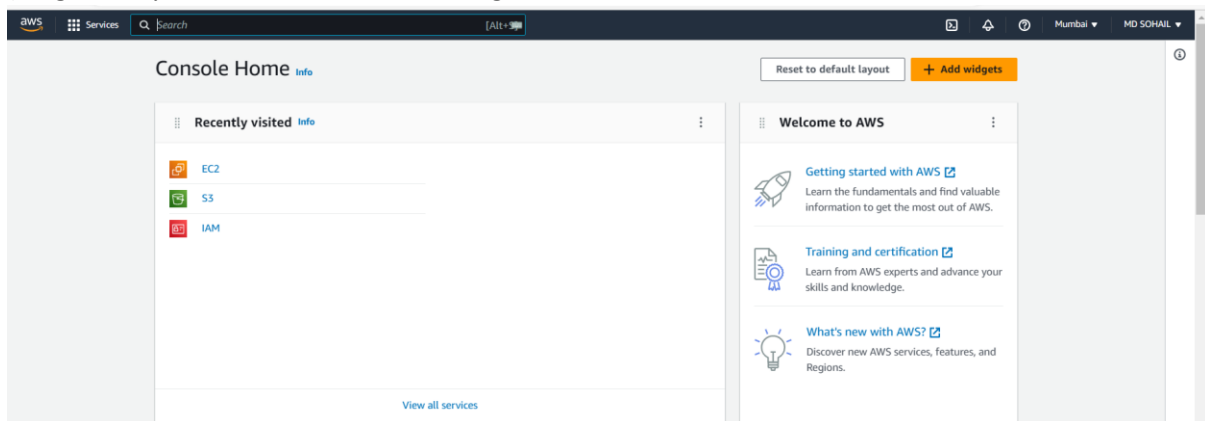
Problem Statement: Deploy a project from Github to EC2.

1. Sign in to your Github account.

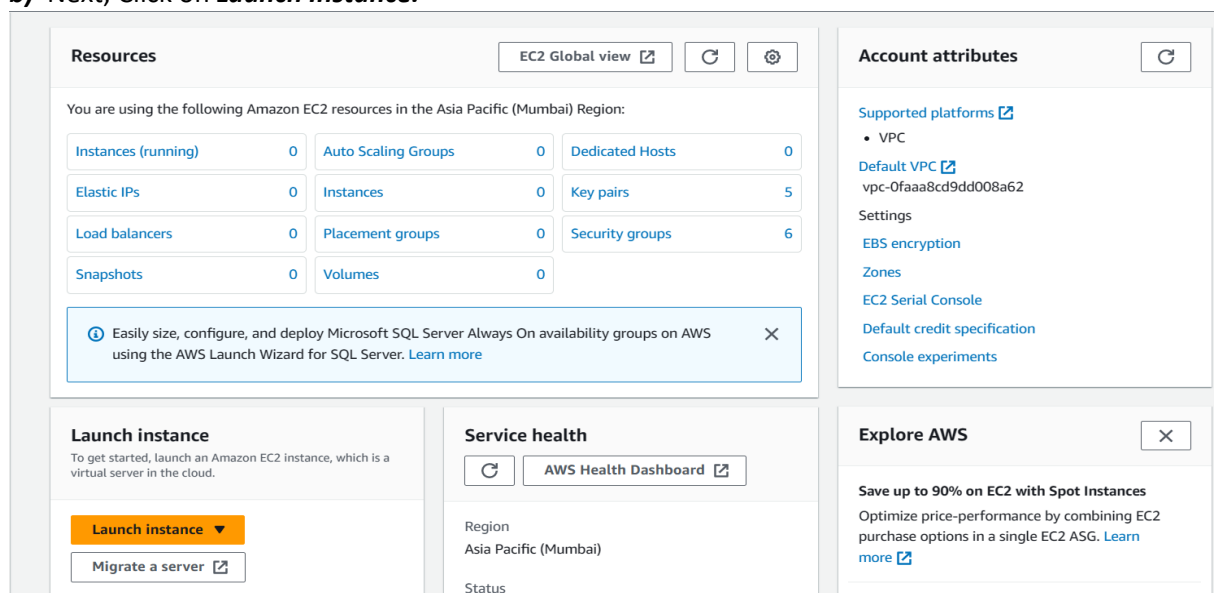


2. Create an EC2 instance in AWS.

a) Sign in to your AWS account and then go to EC2.



b) Next, Click on **Launch Instance**.



c) Write the instance name. Then, Select an OS. Here, we have selected **Ubuntu**.

Launch an instance [Info](#)

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 [Info](#)

Name

sohailec23080


[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

 Search our full catalog including 1000s of application and OS images

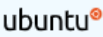
Quick Start




Amazon Linux




macOS




Ubuntu



Windows



Red Hat



[Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

d) We can see the Instance type set here is t2.micro

▼ Instance type [Info](#)

Instance type

☐ All generations

t2.micro

Free tier eligible

Family: t2 1 vCPU 1 GiB Memory
On-Demand Linux pricing: 0.0124 USD per Hour
On-Demand Windows pricing: 0.017 USD per Hour
On-Demand RHEL pricing: 0.0724 USD per Hour
On-Demand SUSE pricing: 0.0124 USD per Hour

[Compare instance types](#)

e) We have given a Key pair.

Key pair name - *required*

keypair5 ▼

↻ Create new key pair

f) Allow the SSH, HTTPS, HTTP traffic from the internet.

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-6**' 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

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

g) Now, Click on **Launch Instance**.

▼ **Summary**

Number of instances [Info](#)

1

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...[read more](#)
ami-02eb7a4783e7e9317

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

[i](#) Free tier: In your first year includes 750 ✕ ▼

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

h) Next, copy the *Public IPv4 address*.

EC2 > Instances > i-019b2063afea2a178

Instance summary for i-019b2063afea2a178 (sohailec23080) [Info](#)

Updated less than a minute ago

[Refresh](#) [Connect](#) [Instance state](#) ✓ Public IPv4 address copied

Instance ID i-019b2063afea2a178 (sohailec23080)	Public IPv4 address 13.235.68.63 open address	Private IPv4 addresses 172.31.0.51
IPv6 address -	Instance state Pending	Public IPv4 DNS ec2-13-235-68-63.ap-south-1.compute.amazonaws.com open address
Hostname type IP name: ip-172-31-0-51.ap-south-1.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-0-51.ap-south-1.compute.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	

3. Now open Bitvise SSH.

a) Within the Login tab, In the *host* field paste the Public IPv4 address which you have copied earlier. Set the Username as *ubuntu*, Initial method as *public key* and Client key as *Global 1*.

Bitvise SSH Client 9.27

[Window behavior](#)

Default profile

Load profile | Save profile as | New profile | Reset profile

Login | Options | Terminal | RDP | SFTP | Services | C2S | S2C | SSH | Notes | About

Server

Host: 13.235.68.63

Port: ☐ Enable obfuscation

Obfuscation keyword:

Kerberos

SPN:

☐ GSS/Kerberos key exchange

☐ Request delegation

☒ gssapi-keyex authentication

Authentication

Username: ubuntu

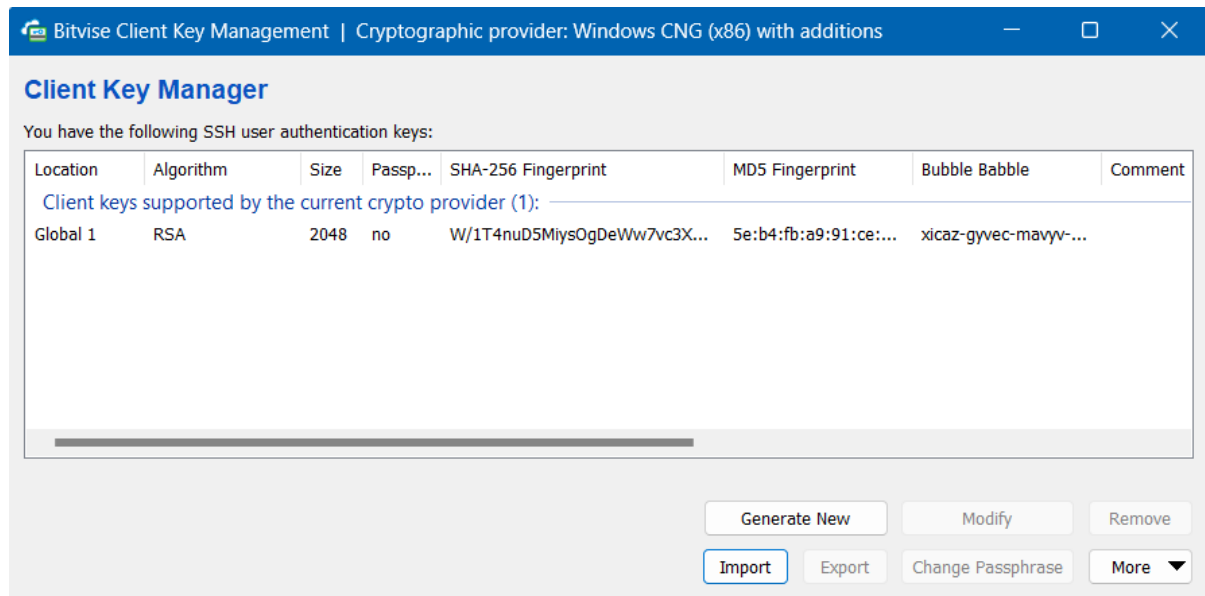
Initial method: publickey

[Client key](#): Global 1

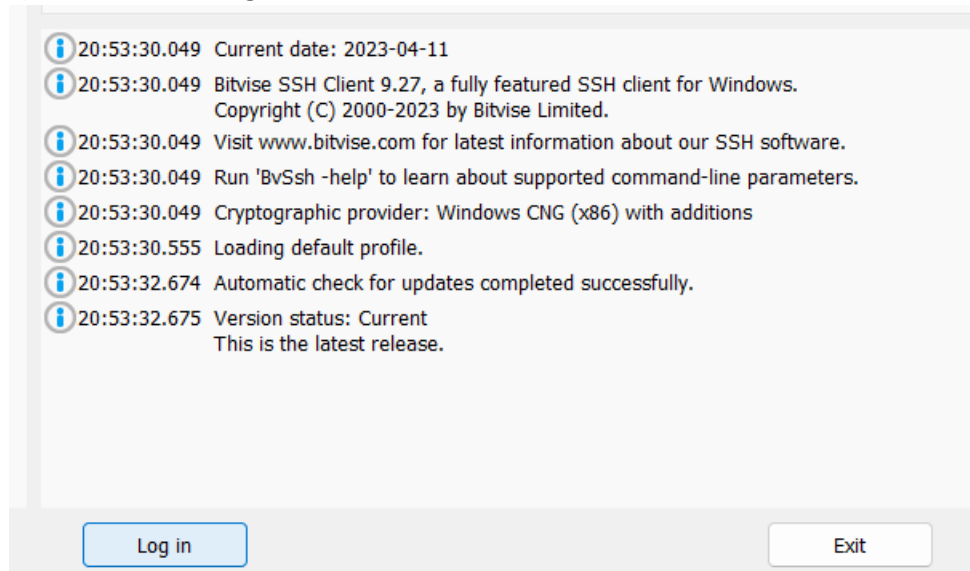
Passphrase:

Elevation: Default

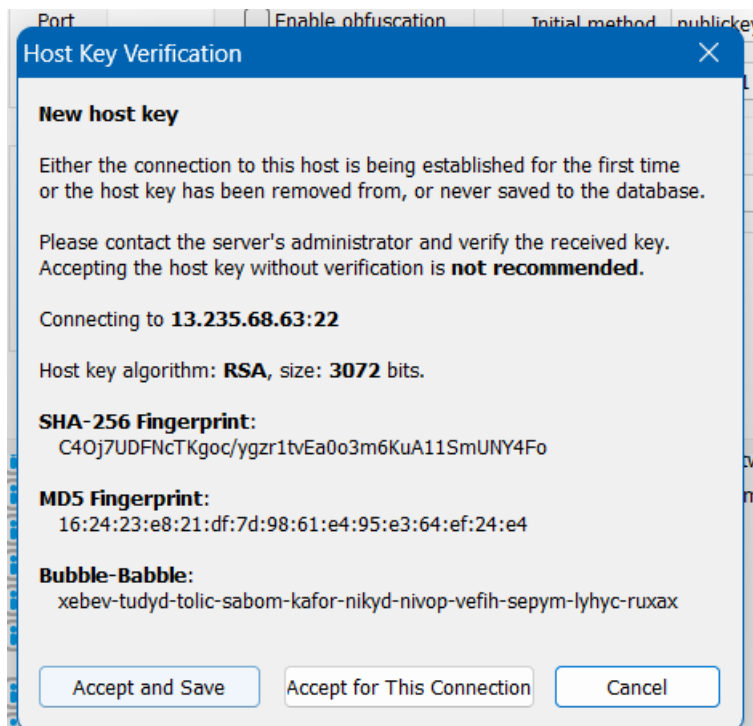
b) Now, Import the key pair.



c) Now, Click on **Log in**.



d) Click on *Accept and Save*.



e) Next, Open the **Terminal**.

4. In the *Bitvise SSH Terminal*, Type the following commands in the terminal one by one.

a) `pwd`

```
ubuntu@ip-172-31-0-51:~$ pwd
/home/ubuntu
```

b) `sudo apt-get update && sudo apt-get upgrade`

```
ubuntu@ip-172-31-0-51:~$ 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 [731 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [147 kB]
```

Continuing...

```
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-0-51:~$
```

c) `sudo apt-get install nginx`

```
ubuntu@ip-172-31-0-51:~$ sudo apt-get install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Continuing...

```
Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-0-51:~$
```

Page 6 of 6

d) `nginx -v`

```
ubuntu@ip-172-31-0-51:~$ nginx -v
nginx version: nginx/1.18.0 (Ubuntu)
ubuntu@ip-172-31-0-51:~$
```

e) `curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -`

```
ubuntu@ip-172-31-0-51:~$ curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
## Installing the NodeSource Node.js 18.x repo...
```

Continuing...

f) `sudo apt install nodejs`

```
ubuntu@ip-172-31-0-51:~$ sudo apt install nodejs
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  nodejs
```

Continuing...

g) `node -v`

```
ubuntu@ip-172-31-0-51:~$ node -v
v18.15.0
```

h) Now, copy the HTTPS link of the Github Repository.

The screenshot shows a GitHub repository page for 'sohail3080 / Awsproject2'. The repository is private and has 1 branch and 0 tags. The file list includes: Anirban-02 Update index.js, .gitignore (done), New Text Document.txt (done), index.js (Update index.js), and package.json (done). A 'Clone' dropdown menu is open, showing the 'HTTPS' option selected. The URL 'https://github.com/sohail3080/Awsproject2.git' is displayed and marked as 'Copied!'. Other options in the dropdown include 'SSH', 'GitHub CLI', 'Open with GitHub Desktop', and 'Download ZIP'.

Again, go to Bitwise SSH terminal and type the following command to clone the repository in the EC2 server.

`git clone https://github.com/sohail3080/Awsproject2.git`

Now, give the Username and Password(Token)

```
ubuntu@ip-172-31-0-51:~$ git clone https://github.com/sohail3080/Awsproject2.git
Cloning into 'Awsproject2'...
Username for 'https://github.com': sohail3080
Password for 'https://sohail3080@github.com':
remote: Enumerating objects: 29, done.
remote: Counting objects: 100% (29/29), done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 29 (delta 14), reused 14 (delta 6), pack-reused 0
Receiving objects: 100% (29/29), 6.23 KiB | 6.23 MiB/s, done.
Resolving deltas: 100% (14/14), done.
```

i) dir

```
ubuntu@ip-172-31-0-51:~$ dir
Awsproject2
```

j) cd Awsproject2

```
ubuntu@ip-172-31-0-51:~$ cd Awsproject2/
ubuntu@ip-172-31-0-51:~/Awsproject2$
```

k) dir

```
ubuntu@ip-172-31-0-51:~/Awsproject2$ dir
New\ Text\ Document.txt  index.js  package.json
```

l) npm install

```
ubuntu@ip-172-31-0-51:~/Awsproject2$ 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 known to be problematic. See https://v8.dev/blog/math-
random for details.

added 258 packages, and audited 259 packages in 15s

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
```

m) node index.js

```
ubuntu@ip-172-31-0-51:~/Awsproject2$ node index.js
Started server
```

n) Copy the Public IPv4 address and paste it in a browser.

A screenshot of a web browser window. The address bar shows "Not secure | 13.235.68.63". The page content includes the heading "Welcome to nginx!", a paragraph stating "If you see this page, the nginx web server is successfully installed and working. Further configuration is required.", and links to "nginx.org" and "nginx.com". It also says "Thank you for using nginx."

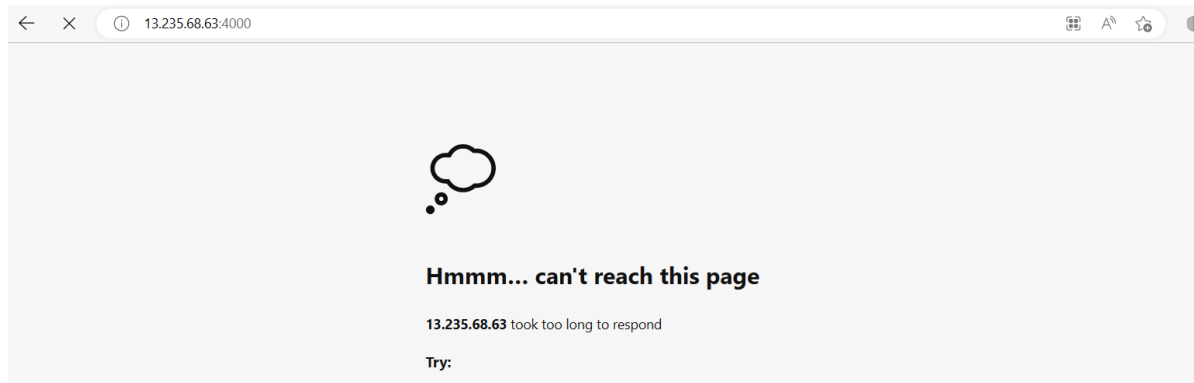
Welcome to nginx!

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.

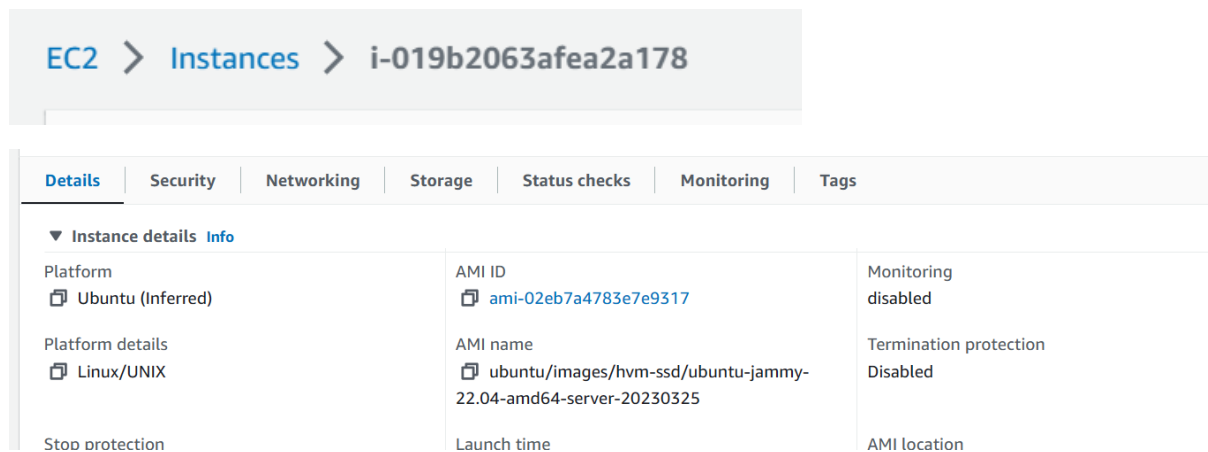
Thank you for using nginx.

o) We cannot see our website in any port.

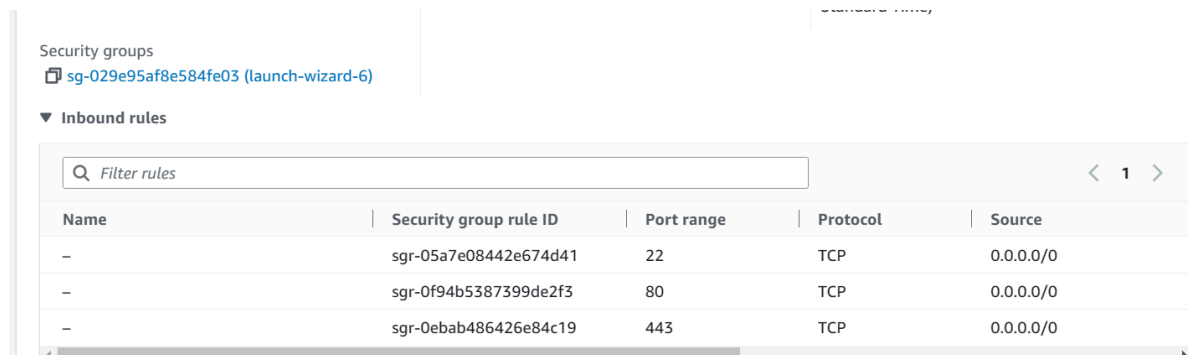


5. Now, to see our website in port 4000, we have to follow the steps.

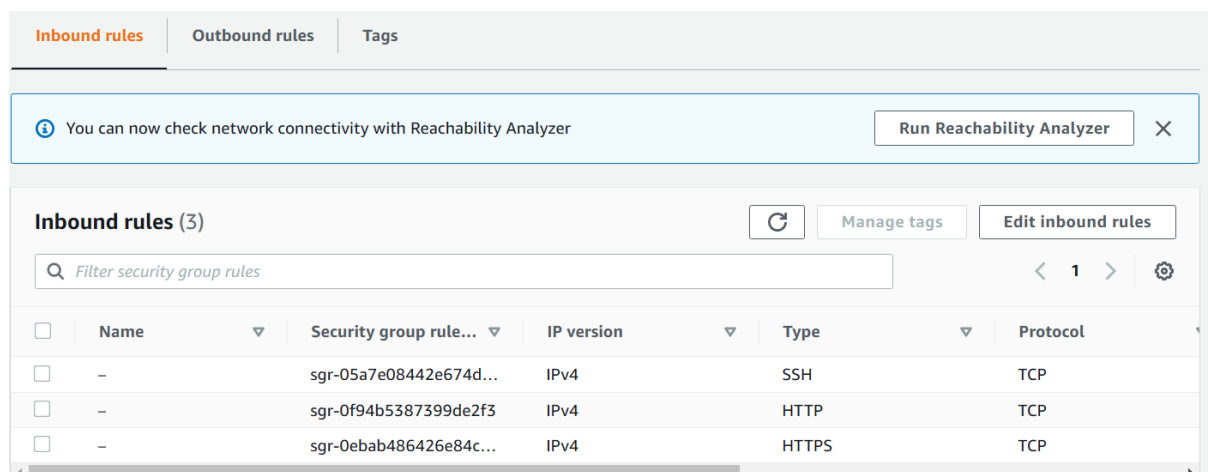
a) Within the instance you create, got to the Security tab.



b) Next, Go to the Security groups.



c) Next, Click on Edit inbound rules.



d) Now, add a custom TCP with port 4000 and source 0.0.0.0/0 and Save rules.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional	
sgr-05a7e08442e674d41	SSH	TCP	22	Custom 0.0.0.0/0		Delete
sgr-0f94b5387399de2f3	HTTP	TCP	80	Custom 0.0.0.0/0		Delete
sgr-0ebab486426e84c19	HTTPS	TCP	443	Custom 0.0.0.0/0		Delete
-	Custom TCP	TCP	4000	Anywh... 0.0.0.0/0		Delete

[Add rule](#) [Cancel](#) [Preview changes](#) [Save rules](#)

e) We can see that the new rule is added.

Name	Security group rule...	IP version	Type	Protocol
-	sgr-05a7e08442e674d...	IPv4	SSH	TCP
-	sgr-0393f8e768bb40b8a	IPv4	Custom TCP	TCP
-	sgr-0f94b5387399de2f3	IPv4	HTTP	TCP
-	sgr-0ebab486426e84c...	IPv4	HTTPS	TCP

f) Now, add the port number 4000 after the Public IPv4 address you have in the browser URL.



Hello

We can see the Text “Hello”. Thus, it is working.

6. Making changes in the webpage.

a) Go to index.js file in your Github Repository. Then, Click on the pen icon.

sohail3080 / Awsproject2 Private Unwatch 1 Fork 0

<> Code Issues Pull requests Actions Projects Security Insights Settings

master Awsproject2 / index.js / <> Jump to Go to file

sohail3080 Update index.js Latest commit ac7d3c9 38 minutes ago History

3 contributors

11 lines (9 sloc) 179 Bytes Raw Blame

```
1 const express = require('express')
2 const app = express()
3
4 app.get('/', function (req, res) {
5   res.send('Hello')
6 })
7
8 app.listen(4000, ()=>{
9   console.log("Started server");
10 })
11 )
```

Give feedback

b) Edit the code.

Awsproject2 / index.js in master

<> Edit file Preview changes

```
1 const express = require('express')
2 const app = express()
3
4 app.get('/', function (req, res) {
5   res.send('Hello Sohail')
6 })
7
8 app.listen(4000, ()=>{
9   console.log("Started server");
10 })
11 )
12
```

c) Click **Commit changes**.

Commit changes

Update index.js

Add an optional extended description...

☒ Commit directly to the master branch.

☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes Cancel

d) Now, go to the Bitwise terminal and run the following commands respectively.

- Stop the server first.

```
Started server
^C
ubuntu@ip-172-31-0-51:~/Awsproject2$
```

- git pull

```
ubuntu@ip-172-31-0-51:~/Awsproject2$ git pull
Username for 'https://github.com':
```

Give *username* and *password*.

```
ubuntu@ip-172-31-0-51:~/Awsproject2$ git pull
Username for 'https://github.com': sohail3080
Password for 'https://sohail3080@github.com':
```

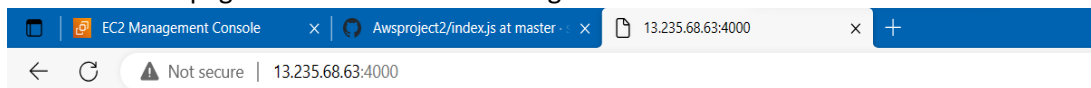
You can see the change is done. i.e. 1 file changed, 1 insertion(+), 1 deletion(-)

```
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), 673 bytes | 673.00 KiB/s, done.
From https://github.com/sohail3080/Awsproject2
   ac7d3c9..1b3fe66  master    -> origin/master
Updating ac7d3c9..1b3fe66
Fast-forward
 index.js | 2 + -
 1 file changed, 1 insertion(+), 1 deletion(-)
```

- Start the server again.

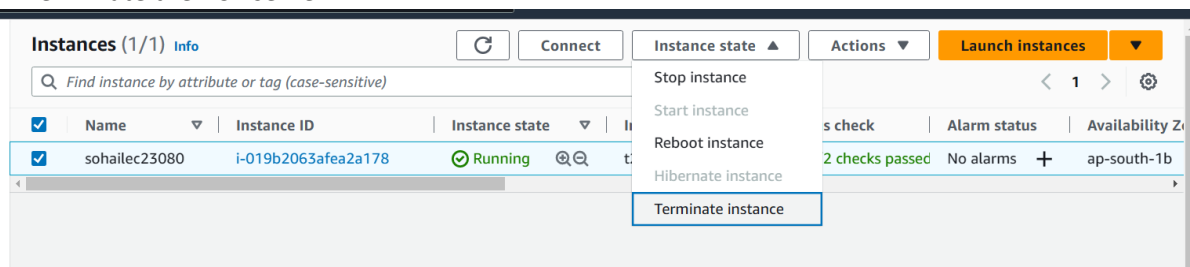
```
ubuntu@ip-172-31-0-51:~/Awsproject2$ node index.js
Started server
```

- Refresh the page and we can see the changes.



Hello Sohail

7. Terminate the EC2 server.



Next, Logout of **Bitwise SSH Client** by clicking *Abort*.