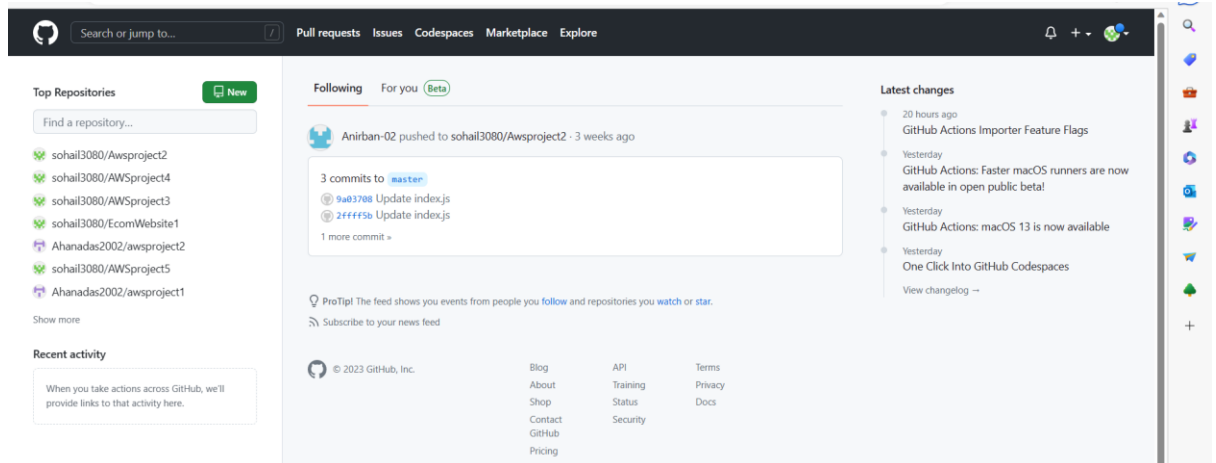


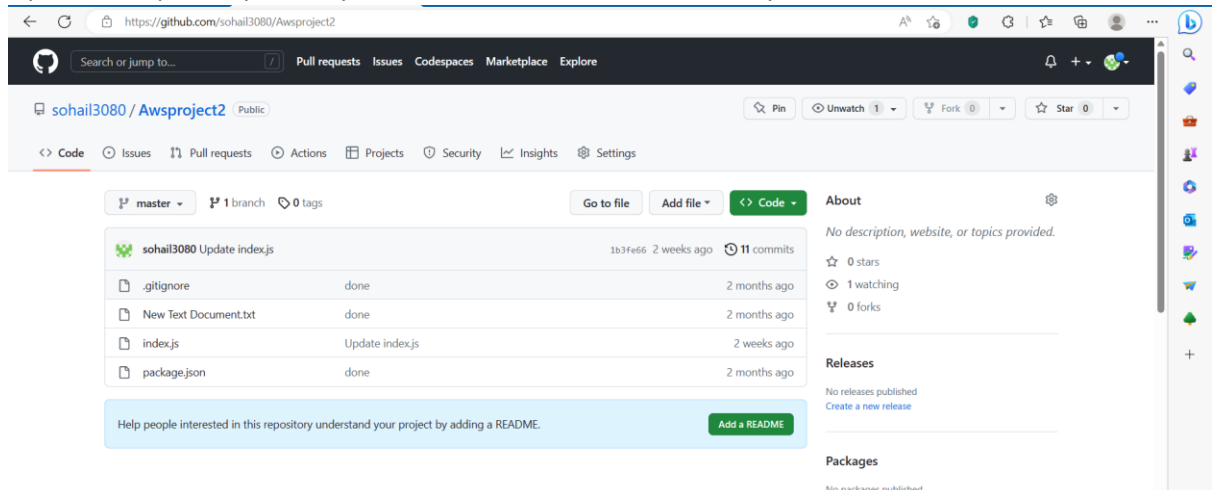
ASSIGNMENT 10

Problem Statement: Deploy a project from GitHub to EC2 by creating a new security group and user data.

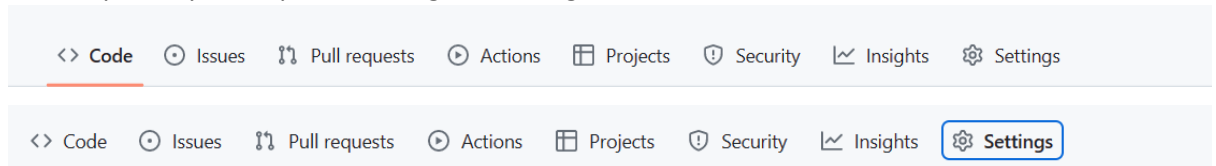
1. Sign in to your Github account.



2. Open the Repository which you want to use and make sure it is public.



3. If the repository is not public, then go to Settings.

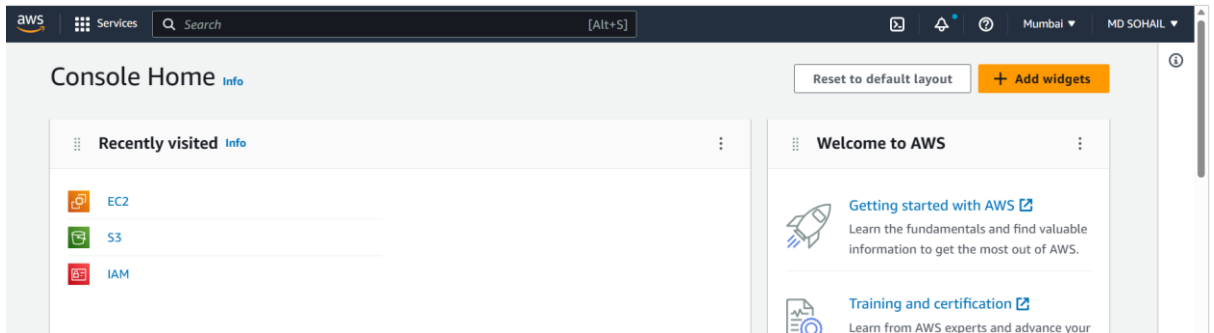


General

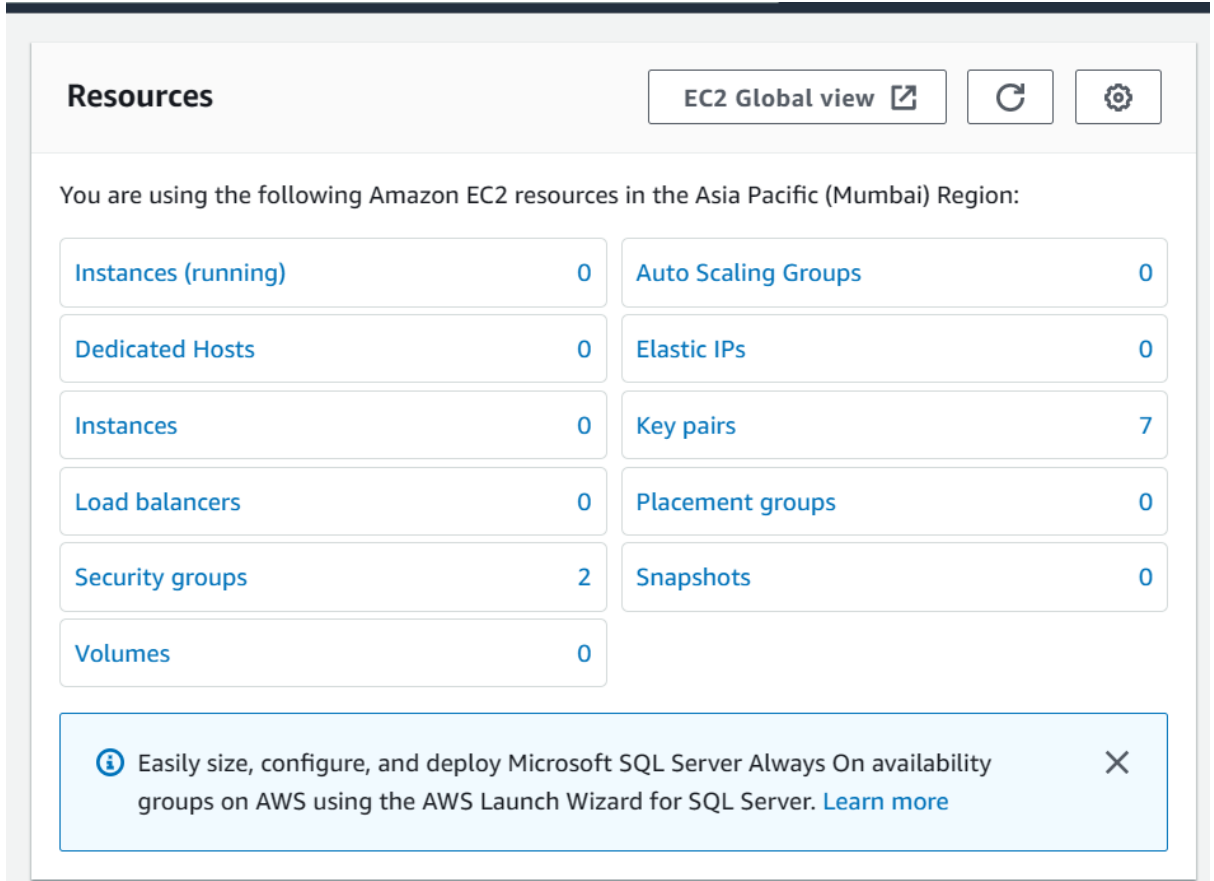
General

Next, scroll to the bottom in the Danger Zone. Click on Change visibility -> Change to Public -> I want to make this repository Public->I have read and understand these effects-> Make this repository public. At last, give the Password. [My repository is already public so I have not followed these steps]

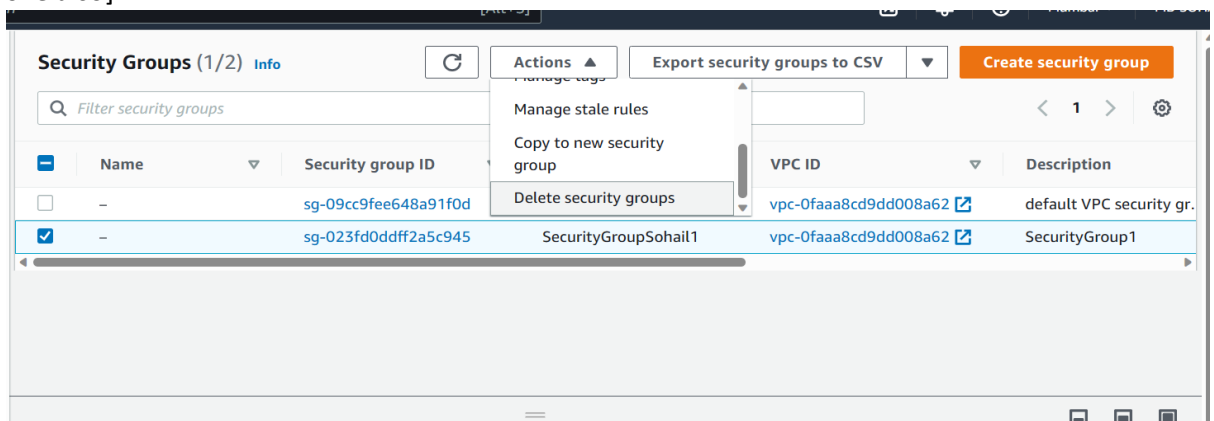
4. Sign in to your AWS account.



5. Go to EC2 Dashboard and Click Security Groups there.



6. Delete all the Security Groups except the default. If you have already created one, then you can keep that one and create a new one. *[But here we are deleting the previously created one also]*



Delete security groups

Are you sure that you want to delete this security group?

- sg-023fd0ddff2a5c945 - SecurityGroupSohail1

Cancel
Delete

7. Now, Select **Create security group**.

Services
Search
[Alt+S]
Mumbai
MD SOHA

Security Groups (1/1) Info
Actions
Export security groups to CSV
Create security group

Filter security groups

<input checked="" type="checkbox"/>	Name	Security group ID	Security group name	VPC ID	Description	Owner
<input checked="" type="checkbox"/>	-	sg-09cc9fee648a91f0d	default	vpc-0faaa8cd9dd008a62	default VPC security gr...	181319814378

8. Now, Enter the Security group name.

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a

Basic details

Security group name Info

SecurityGroupSohail2

Name cannot be edited after creation.

Description Info

SecurityGroup2

VPC Info

vpc-0faaa8cd9dd008a62

9. Go to **Inbound Rules**. Click on **Add rule** to create them.

Inbound rules Info

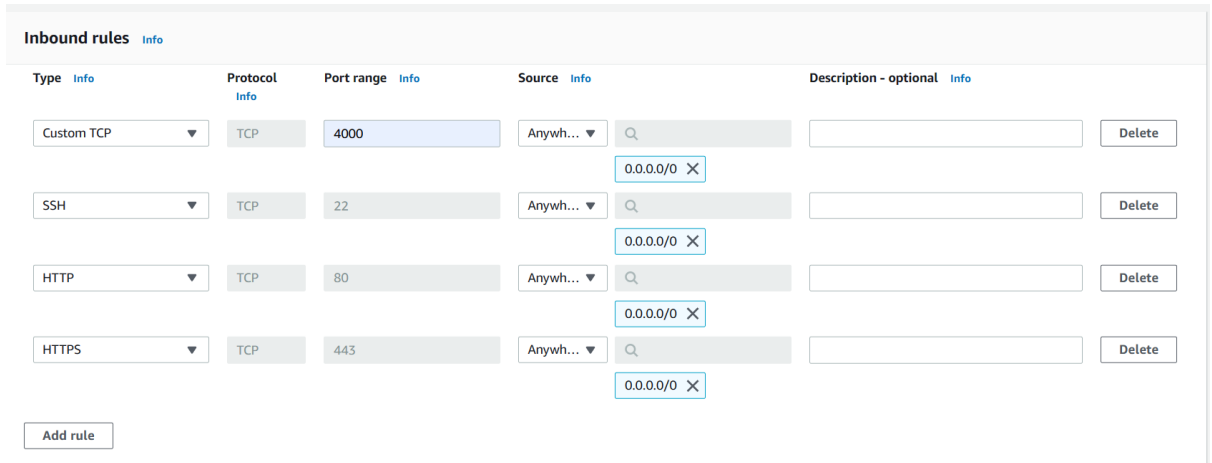
This security group has no inbound rules.

Add rule

10. Create the following.

- Custom TCP with Port range 4000. Give Source 0.0.0.0/0.
- SSH with default Port range 22 and Source 0.0.0.0/0.
- HTTP with default Port range 80 and Source 0.0.0.0/0.

iv)HTTPS with default Port range 443 and Source 0.0.0.0/0.

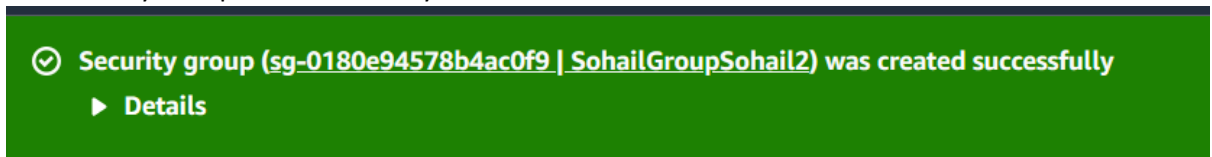


The screenshot shows the 'Inbound rules' configuration page in the AWS Management Console. It displays a table of rules with columns: Type, Protocol, Port range, Source, and Description - optional. There are four rules listed: Custom TCP (Port 4000), SSH (Port 22), HTTP (Port 80), and HTTPS (Port 443). Each rule has a source of 'Anywhere...' and a description of '0.0.0.0/0'. A 'Delete' button is next to each rule. An 'Add rule' button is at the bottom left.

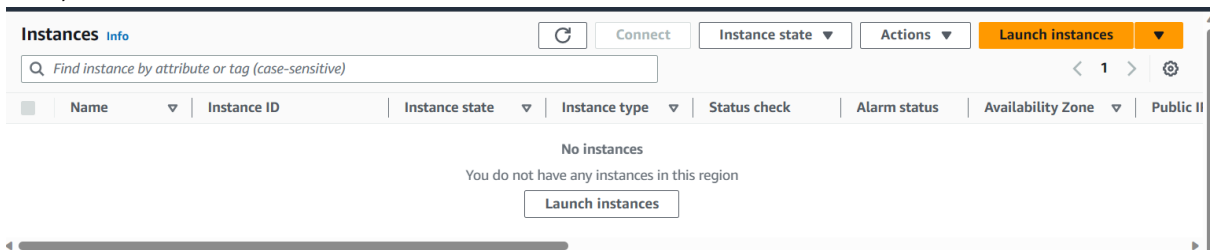
Type	Protocol	Port range	Source	Description - optional
Custom TCP	TCP	4000	Anywhere... 0.0.0.0/0	
SSH	TCP	22	Anywhere... 0.0.0.0/0	
HTTP	TCP	80	Anywhere... 0.0.0.0/0	
HTTPS	TCP	443	Anywhere... 0.0.0.0/0	

Go to the bottom and click on **Create Security Group**.

11. The Security Group was successfully created.

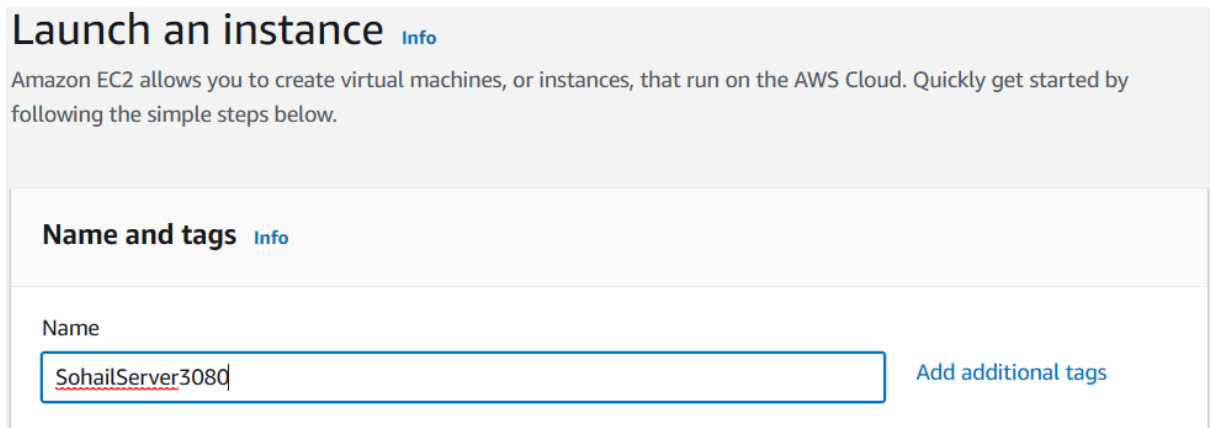


12. Now, Go to Instances and Click on **Launch instances**.



The screenshot shows the 'Instances' page in the AWS Management Console. It features a search bar, a table of instances (currently empty), and a 'Launch instances' button. The table has columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IP. A message states 'No instances. You do not have any instances in this region.' A 'Launch instances' button is at the bottom.

13. Give a name to the new Instance.



The screenshot shows the 'Launch an instance' page in the AWS Management Console. It includes a heading 'Launch an instance', a brief description of Amazon EC2, and a section titled 'Name and tags'. In the 'Name' field, the text 'SohailServer3080' is entered. There is an 'Add additional tags' link next to the field.

14. Select Ubuntu as the Operating System of the EC2 server.

Recents

Quick Start

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

SUSE

SUSE

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type

Free tier eligible

ami-02eb7a4783e7e9317 (64-bit (x86)) / ami-0a5dcff6fb7af3fc9 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

15. Select instance type as t2.micro if not already in default.

▼ Instance type [Info](#)

Instance type

t2.micro

Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true

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

All generations

Compare instance types

16. Select a key pair you have previously created and if not, create a new pair.

▼ 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*

keypair6

Create new key pair

17. In Network settings field, Click on **Select existing security group** and select the security Group you previously created (SohailGroupSohail2).

▼ **Network settings** [Info](#)

Edit

Network [Info](#)

vpc-0faaa8cd9dd008a62

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

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

Security groups [Info](#)

Select security groups ▼

SohailGroupSohail2 sg-0180e94578b4ac0f9 ✕

VPC: vpc-0faaa8cd9dd008a62

 [Compare security group rules](#)

18. Go to the **Advance details** field and scroll down to its last until you see the **User data** field.

▼ **Advanced details** [Info](#)

Purchasing option [Info](#)

19. Within the User data field, enter the following codes.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
cd /home/ubuntu
git clone https://github.com/sohail3080/Awsproject2.git
cd Awsproject2
npm install
node index.js
```

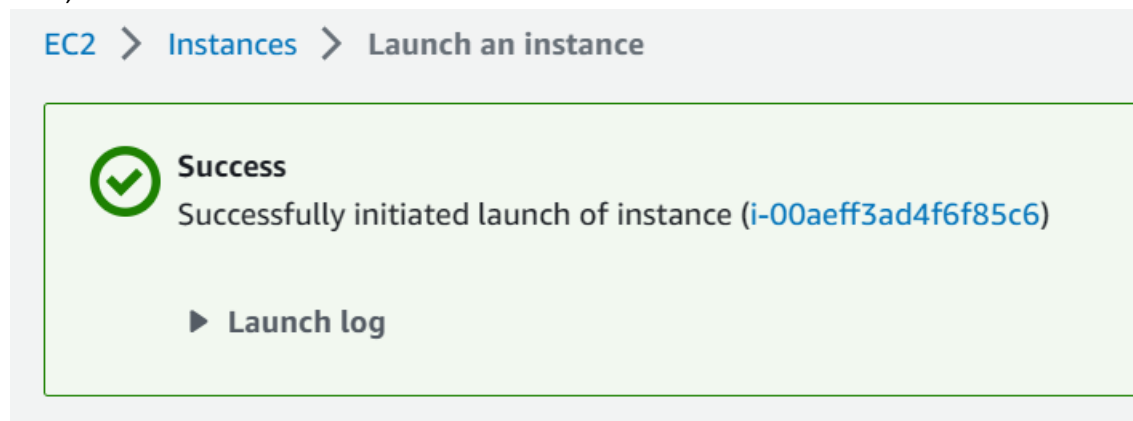
User data - optional [Info](#)

Enter user data in the field.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
cd /home/ubuntu
git clone https://github.com/sohail3080/Awsproject2.git
cd Awsproject2
npm install
node index.js
```

The link after git clone is the HTTPS link of the Github repository.

20. Now, Click on **Launch instance**.



21. Click on the Running Instance.

The screenshot shows the 'Instances (1)' page in the AWS Management Console. It includes a search bar, a filter for 'Instance state = running', and a table of instances. The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. One instance is listed: 'SohailServer3...' with ID 'i-00aeff3ad4f6f85c6', state 'Running', type 't2.micro', and status '2/2 checks passed'.

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/>	SohailServer3...	i-00aeff3ad4f6f85c6	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-13-232-15-15...

22. Copy the IPv4 address.

EC2 > Instances > i-00aeff3ad4f6f85c6

Instance summary for i-00aeff3ad4f6f85c6 (SohailServer3080) Info

Updated less than a minute ago

Instance ID
i-00aeff3ad4f6f85c6 (SohailServer3080)

IPv6 address
-

Public IPv4 address
13.232.15.159 | open address

Instance state
Running

Private IPv4
172.31.1.1

Public IPv4
ec2-13-232-15-159

23. Paste the copied IPv4 address to the URL of the browser.[It might take some time.]

Not secure | 13.232.15.159

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.

24. Add the Port number to the URL. Here, we have taken 4000 as port number. [It might take some time.]

← ↻ Not secure | 13.232.15.159:4000

Hello Sohail

Hence, the Project was successfully deployed.
