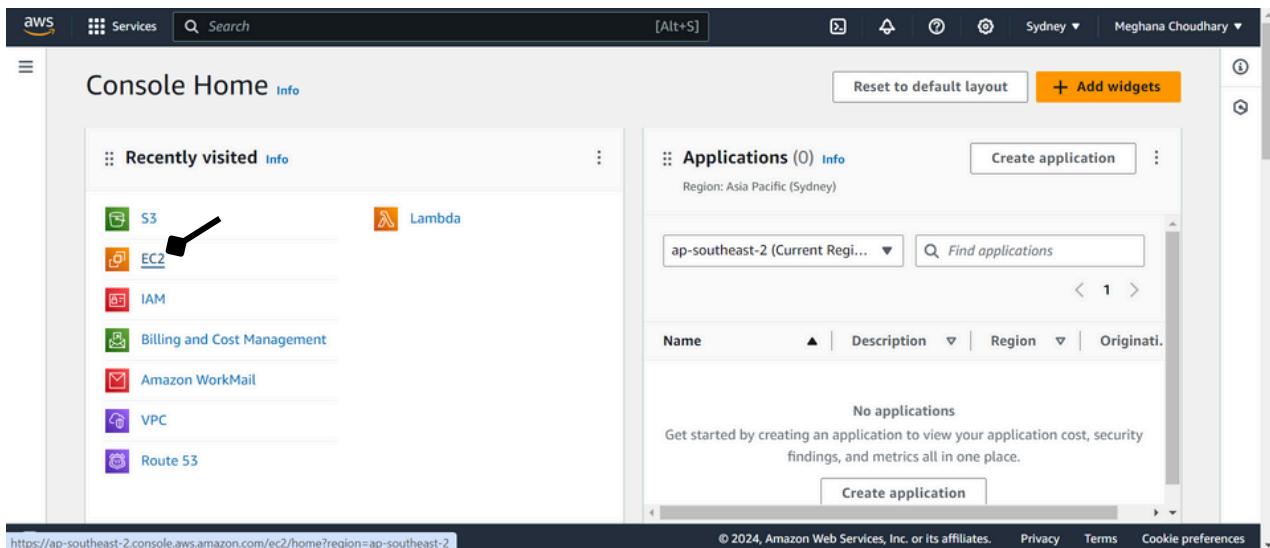


10. PROBLEM STATEMENT:

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

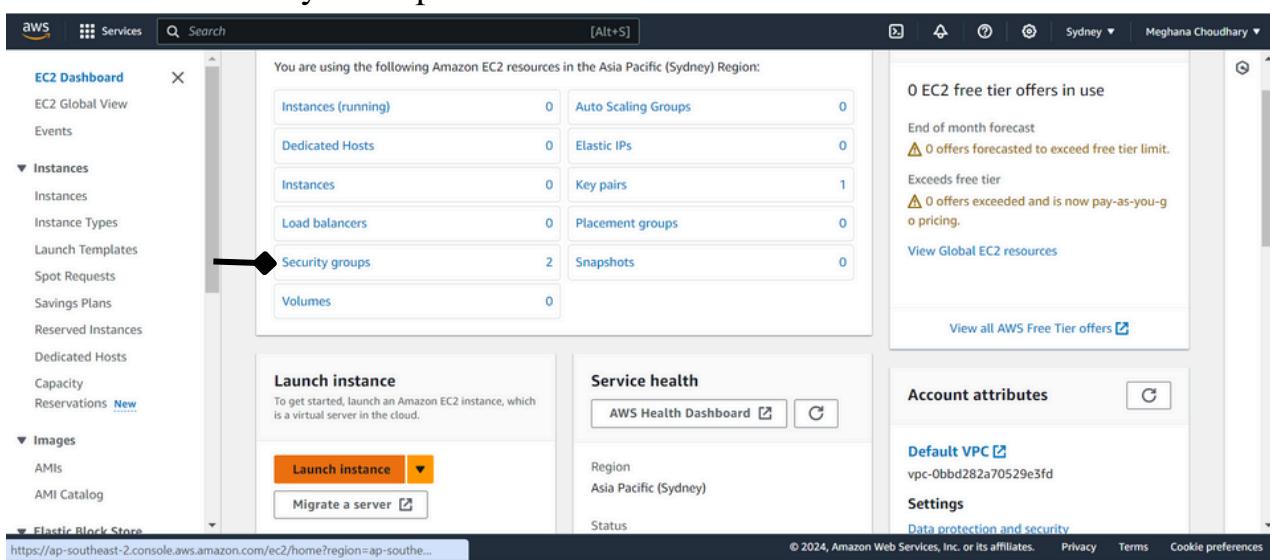
Creating a EC2 Instance with port 4000 open and deploying a project with user data:-

1. Log into AWS console and click on “EC2”.



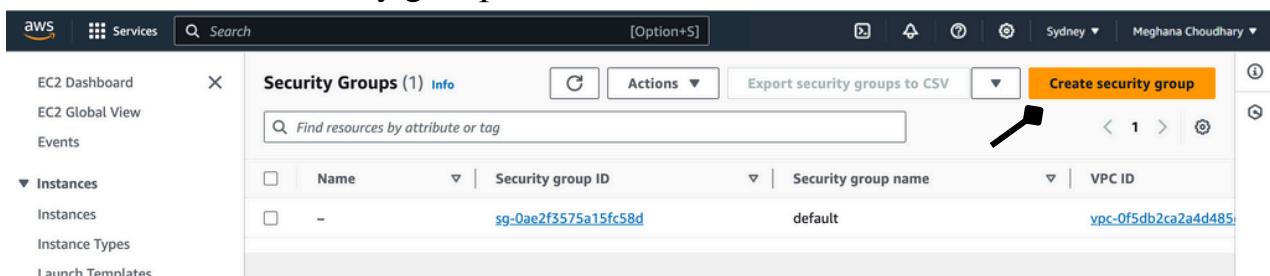
The screenshot shows the AWS Console Home page. On the left, there is a sidebar with 'Recently visited' services: S3, Lambda, EC2 (which has a black arrow pointing to it), IAM, Billing and Cost Management, Amazon WorkMail, VPC, and Route 53. On the right, there is a section titled 'Applications (0)' with a 'Create application' button and a note about getting started with application cost, security findings, and metrics. At the bottom, there is a navigation bar with links for privacy, terms, and cookie preferences.

2. Click on ‘Security Groups’.



The screenshot shows the EC2 Dashboard. On the left, there is a sidebar with 'Instances' (highlighted with a black arrow) and 'Security groups'. The main area displays statistics for various EC2 resources like Instances (running), Auto Scaling Groups, Dedicated Hosts, Elastic IPs, Instances, Key pairs, Load balancers, Placement groups, Security groups (highlighted with a black arrow), Snapshots, and Volumes. On the right, there are sections for 'Service health' (AWS Health Dashboard), 'Account attributes' (Default VPC set to vpc-0bbd282a70529e3fd), and 'Data protection and security'. At the bottom, there is a navigation bar with links for privacy, terms, and cookie preferences.

3. Go to ‘Create security groups’.



The screenshot shows the 'Security Groups' page. On the left, there is a sidebar with 'Instances' (highlighted with a black arrow). The main area shows a table of security groups. One row is selected, showing the 'Name' as 'sg-0ae2f3575a15fc58d', 'Security group ID' as 'sg-0ae2f3575a15fc58d', 'Security group name' as 'default', and 'VPC ID' as 'vpc-0f5db2ca2a4d485'. At the top right, there is a 'Create security group' button. At the bottom right, there is a navigation bar with links for privacy, terms, and cookie preferences.

4. Add basic details.

The screenshot shows the 'Create security group' wizard. In the 'Basic details' section, the 'Security group name' field contains ':4000 Open'. The 'Description' field contains 'Allows traffic through port 4000'. The 'VPC info' dropdown shows 'vpc-0f5db2ca2a4d485ed'. A note at the top states: 'A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.'

5. In 'Inbound rules' go to 'Add rule'.

The screenshot shows the 'Inbound rules' page. It displays a message: 'This security group has no inbound rules.' Below this, there is an 'Add rule' button with a black arrow pointing to it.

6. Add all the necessary rules like SSH, HTTP and HTTPS from anywhere; along with a Custom TCP rule allowing traffic on port 4000 from anywhere.

The screenshot shows the 'Inbound rules' page with five rules listed:

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

Each rule has a 'Delete' button to its right. An 'Add rule' button is located at the bottom left of the list.

7. Then finally ‘Create security group’.

The screenshot shows the 'Create security group' wizard in the AWS Management Console. The top navigation bar includes 'Services', 'Search', and account information for 'Meghana Choudhary'. The main content area is titled 'Create security group' with a sub-section 'Basic details'. It contains fields for 'Security group name' (':4000 Open'), 'Description' ('Allows traffic through port 4000'), and 'VPC' ('vpc-0f5db2ca2a4d485ed'). The 'Inbound rules' section lists four rules: SSH (TCP port 22), HTTP (TCP port 80), HTTPS (TCP port 443), and a custom rule for port 4000. The 'Outbound rules' section shows a single rule for 'All traffic' to 'All' destination. The 'Tags - optional' section indicates no tags are associated with the resource. A large orange button at the bottom right is labeled 'Create security group'.

Basic details

Security group name [Info](#)
:4000 Open

Description [Info](#)
Allows traffic through port 4000

VPC [Info](#)
vpc-0f5db2ca2a4d485ed

Inbound rules [Info](#)

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

Add rule

⚠️ 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. X

Outbound rules [Info](#)

Type Info	Protocol	Port range Info	Destination Info	Description - optional Info
All traffic	All	All	Cu... ▾	0.0.0.0/0 X

Add rule

⚠️ 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. X

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags

Create security group

7. After the security group is created, go to ‘EC2 Dashboard’.

The screenshot shows the AWS EC2 Security Groups page. At the top, a green banner displays a success message: “Security group (sg-0441b4b625b98e5ff | :4000 Open) was created successfully”. Below the banner, the page title is “sg-0441b4b625b98e5ff - :4000 Open”. The “Details” section contains the following information:

Security group name	Security group ID	Description	VPC ID
:4000 Open	sg-0441b4b625b98e5ff	Allows traffic through port 4000	VPC: 0f5db2ca2a4d485ed
Owner	381491890643	Inbound rules count 4 Permission entries	Outbound rules count 1 Permission entry

8. Click on ‘Launch instance’.

The screenshot shows the AWS EC2 Dashboard. On the left sidebar, the “Instances” section is expanded, showing options like Instances, Instance Types, Launch Templates, and Spot Requests. In the main content area, there is a summary of resources in the Asia Pacific (Sydney) Region:

Instances (running)	0	Auto Scaling Groups	0
Dedicated Hosts	0	Elastic IPs	0
Instances	0	Key pairs	1
Load balancers	0	Placement groups	0
Security groups	2	Snapshots	0
Volumes	0		

Below this, there is a “Launch instance” button with a black arrow pointing to it. To the right, there is a “Service health” section and “Account attributes” settings.

9. Fill the required details.

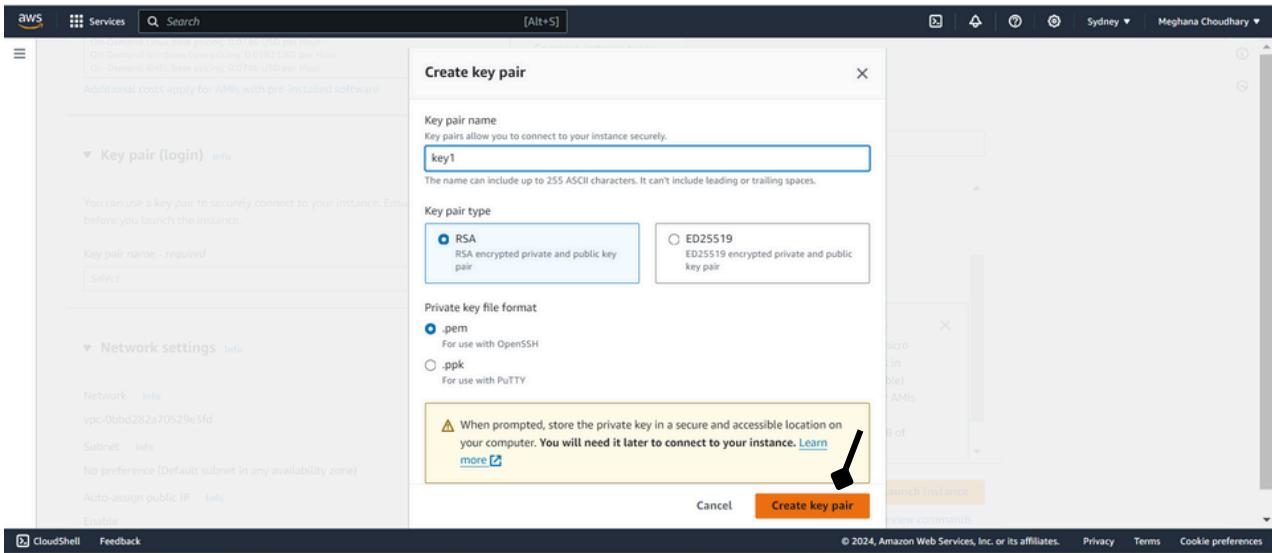
Give a name to your server and choose ubuntu Server as the OS Image.

The screenshot shows the 'Launch an instance' wizard in the AWS Management Console. The first step, 'Name and tags', has 'nginx-server' entered in the 'Name' field. The second step, 'Application and OS Images (Amazon Machine Image)', shows the 'Ubuntu' AMI selected from a grid of options. The third step, 'Instance type', shows the 't2.micro' instance type selected. The fourth step, 'Summary', displays the configuration: 1 instance, Canonical Ubuntu 22.04 LTS AMI, t2.micro instance type, and 1 volume (8 GiB). A 'Launch instance' button is visible at the bottom right of the summary step.

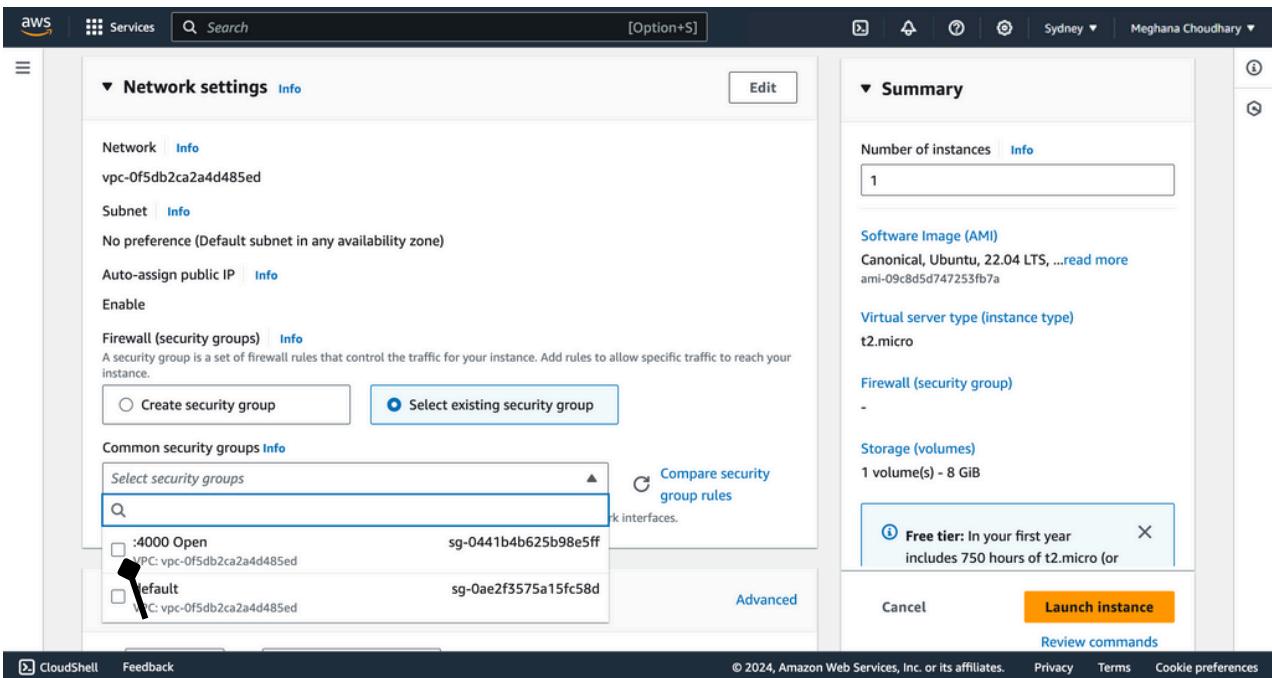
10. For Key pair, if you have a existing key pair select that else 'Create e new key pair'.

The screenshot shows the 'Key pair (login)' step of the wizard. It asks for a key pair name, with a dropdown menu showing 'Select' and a 'Create new key pair' button. To the right, the 'Summary' section is partially visible, showing the same configuration as the previous step: 1 instance, Canonical Ubuntu 22.04 LTS AMI, t2.micro instance type, and 1 volume (8 GiB). A 'Launch instance' button is also present here.

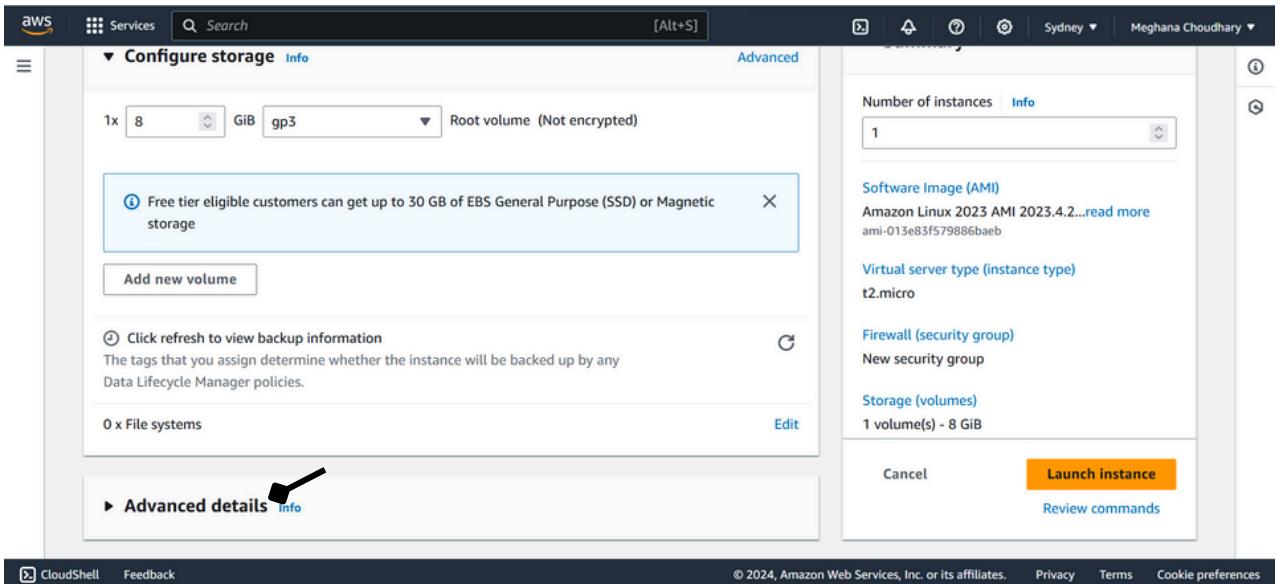
11. Give a name to the key pair. Make sure the key pair type is ‘RSA’ and the file extension is ‘.pem’.



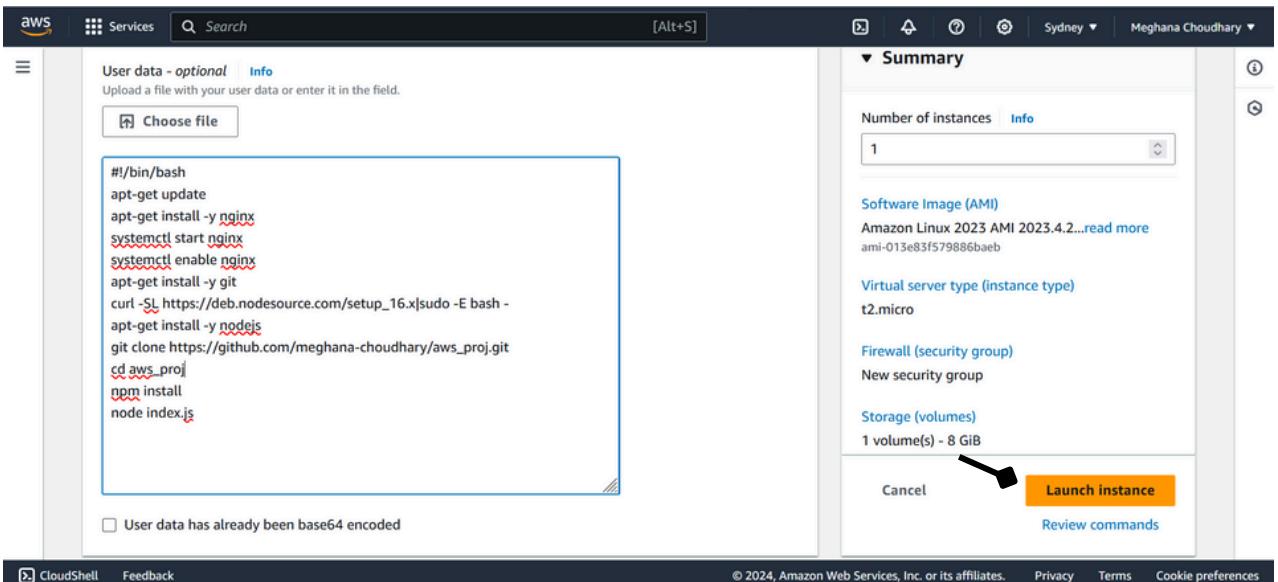
12. For 'Network Settings' from 'Select existing security group' choose the security group you just created.



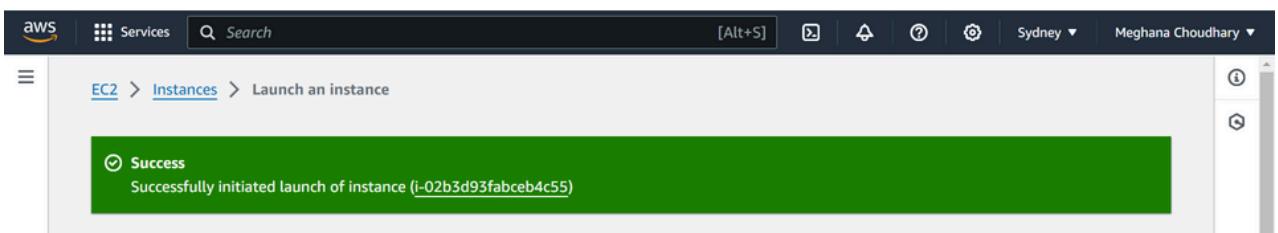
13. Scroll Down and Click on ‘Advanced details’



14. In ‘Advanced details’ go to the last, in advanced data put the following bash script. Then click on ‘Launch instance’



15. Your instance is created successfully.



16. Then Scroll Down and Click on ‘View all instances’.

The screenshot shows the AWS EC2 console with several cards. One card is titled 'Create EBS snapshot policy' with a sub-section about automating EBS snapshots. Another card is titled 'Manage detailed monitoring' with a sub-section about enabling monitoring. A third card is titled 'Create Load Balancer'. At the bottom right, there is an orange button labeled 'View all instances' with a large orange arrow pointing towards it.

17. You can see your instance is created and running. Click on the ‘Instance ID’ of your instance.

The screenshot shows the EC2 Instances page. It lists one instance named 'nginx-server' with the Instance ID 'i-02b3d93fabceb4c55'. The instance is shown as 'Running'. An orange arrow points to the Instance ID 'i-02b3d93fabceb4c55'.

18. Copy the Public IPV4 address of your instance.

The screenshot shows the Instance summary page for the instance 'i-02b3d93fabceb4c55'. It displays various details such as Instance ID, Public IPv4 address (3.27.226.106), Instance state (Running), and VPC ID. The Public IPv4 address '3.27.226.106' is highlighted with a yellow box.

19. Now pasting the Public IPV4 address of your EC2 instance in a web browser gets us to the Nginx default welcome page, through the default HTTP port, port 80.



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.

Adding :4000 after the IPV4 address, we get to our Node.js server through our specified port 4000.

