

**TASK B:** There are multiple processes involved in hosting a web application on AWS (Amazon Web Services) EC2 and creating a Virtual Private Connection (VPC). This may be achieved by following the step-by-step instructions below, which cover setting up subnets, launching an EC2 instance, building a VPC, and deploying a basic web application.

1. Go to the AWS (Amazon Web Services) Management Console and sign in to your AWS account.

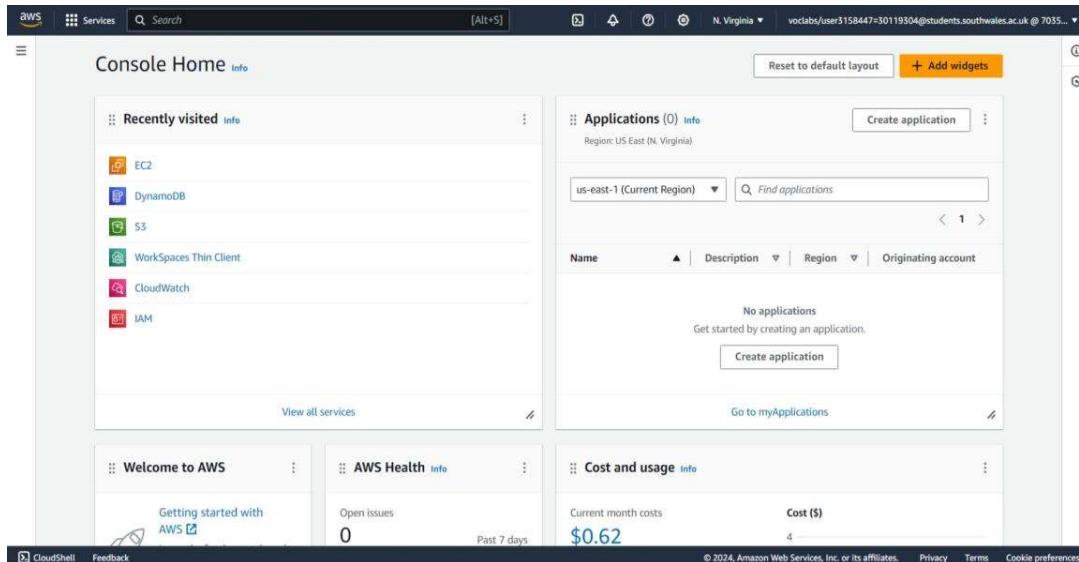
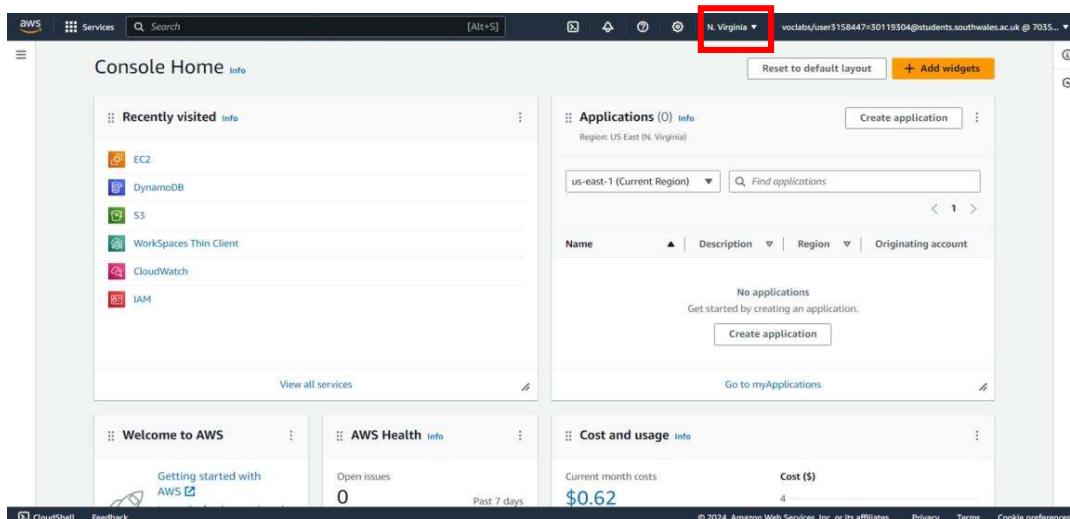


Fig 1: AWS home page

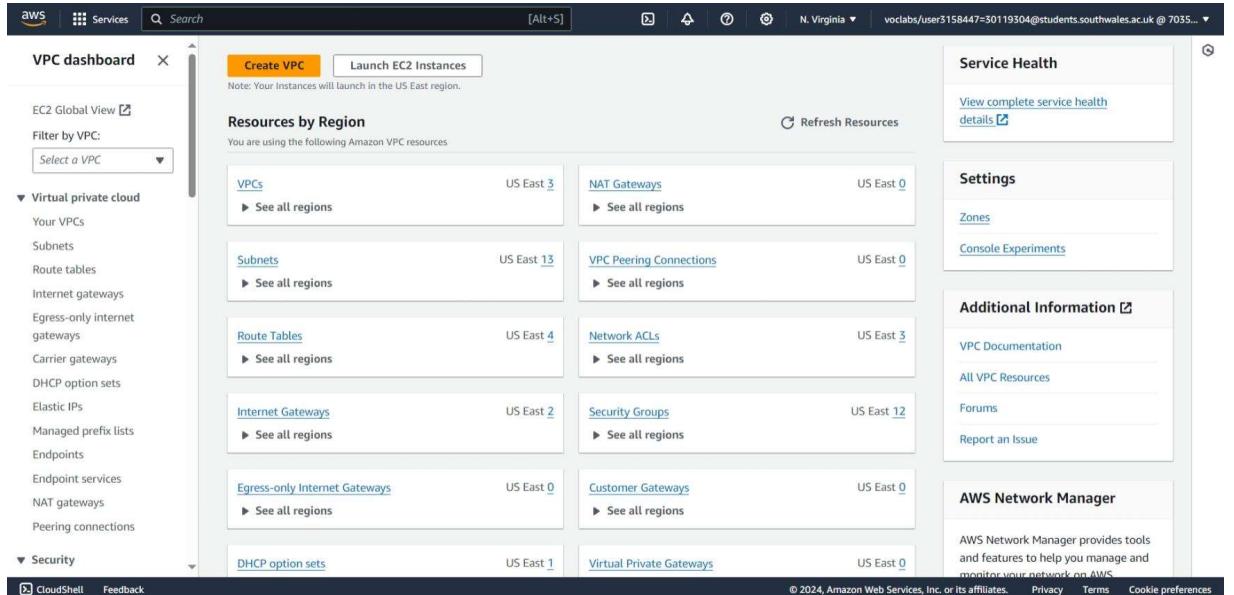
2. Check that the **N. Virginia (us-east-1)** region's resources are currently being managed by your EC2 console. Checking the drop-down option to the left of your username at the top of the screen will allow you to confirm this. Before moving on to the following stage, select the N. Virginia region from the region selection if it is not already shown.



**Fig 2:** AWS home page

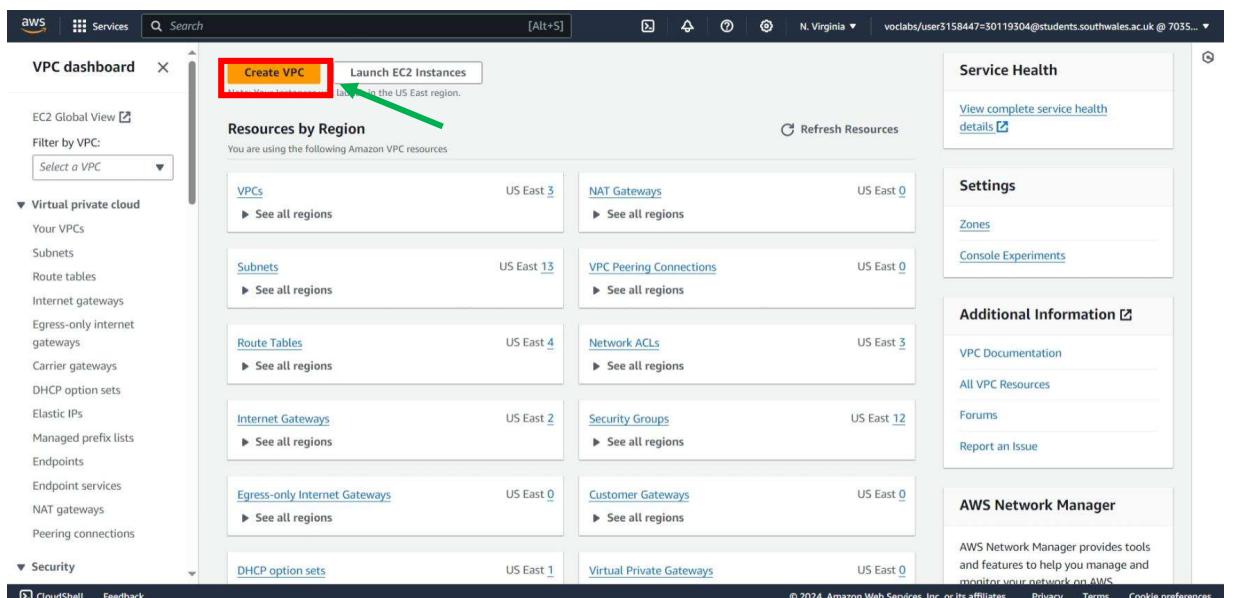
**3.** Navigate to the VPC Dashboard.

- Click on "**Services**" and then select "**VPC**" or you can search in the search bar.

**Fig 3:** VPC home page

**4.** Create a VPC.

- Click on "**Your VPCs (Virtual Private Connection)**" in the left-hand menu.
- Click on "**Create VPC**"

**Fig 4:** VPC home page

**5.** Begin to configure the VPC details:

- Name: 30119304-box-stream-vpc
- IPv4 CIDR block: 10.0.0.0/16
- Leave other settings as default and click on "**Create VPC**" button.

The screenshot shows the AWS VPC settings page. On the left, under 'VPC settings', there are fields for 'Name tag auto-generation' (set to 'Auto-generate' with value '30119304-box-stream-vpc') and 'IPv4 CIDR block' (set to '10.0.0.0/16'). On the right, the 'Preview' section shows the resulting VPC structure: a single VPC containing four subnets (two in us-east-1a and two in us-east-1b) and three route tables. At the bottom right of the preview area, the 'Create VPC' button is highlighted with a red box.

Fig 5: VPC page

The screenshot shows the AWS VPC settings page. It includes sections for 'Number of private subnets' (set to 2), 'NAT gateways' (set to 'None'), 'VPC endpoints' (set to 'S3 Gateway'), and 'DNS options' (checkboxes for 'Enable DNS hostnames' and 'Enable DNS resolution' are checked). A green arrow points to the 'Create VPC' button at the bottom right of the page.

Fig 6: VPC page

6. Go to your VPC in the left now you can see your VPC called 30119304-box-stream

The screenshot shows the AWS VPC dashboard with the 'Your VPCs' page selected. The left sidebar includes links for EC2 Global View, Filter by VPC, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, NAT gateways, and Peering connections. The main area displays a table titled 'Your VPCs (4) Info' with columns for Name, VPC ID, State, IPv4 CIDR, IPv6 CIDR, and DHCP options. The '30119304-box-stream-vpc-vpc' row is highlighted with a red box. The bottom of the screen shows standard AWS navigation links.

Fig 7: Your VPCs page

## 7. Now create a Public Subnet.

- Click on "**Subnets**" in the left-hand menu.
- Click on "**Create subnet**".

The screenshot shows the AWS VPC dashboard with the 'Subnets' page selected. The left sidebar includes links for EC2 Global View, Filter by VPC, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, NAT gateways, and Peering connections. The main area displays a table titled 'Subnets (1/14) Info' with columns for Name, Subnet ID, State, VPC, and IPv4 CIDR. A specific subnet entry, 'subnet-02b2110736da83d1b / 0000-public-subnet', is expanded to show its details. The bottom of the screen shows standard AWS navigation links.

Fig 8: Subnet page

## 8. Begin to create public subnet configuration:

- Select your VPC 30119304-box-stream-vpc.
- Name: 30119304-public-subnet
- IPv4 CIDR block: 10.0.32.0/20
- Availability Zone: us-east-1a

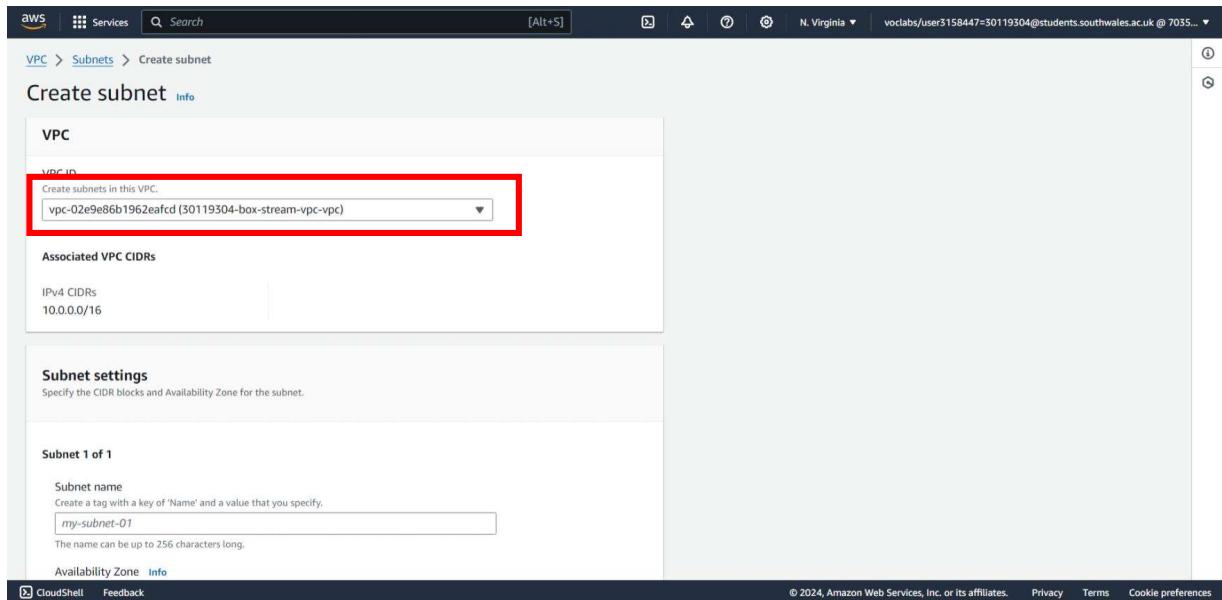


Fig 9: Subnet page

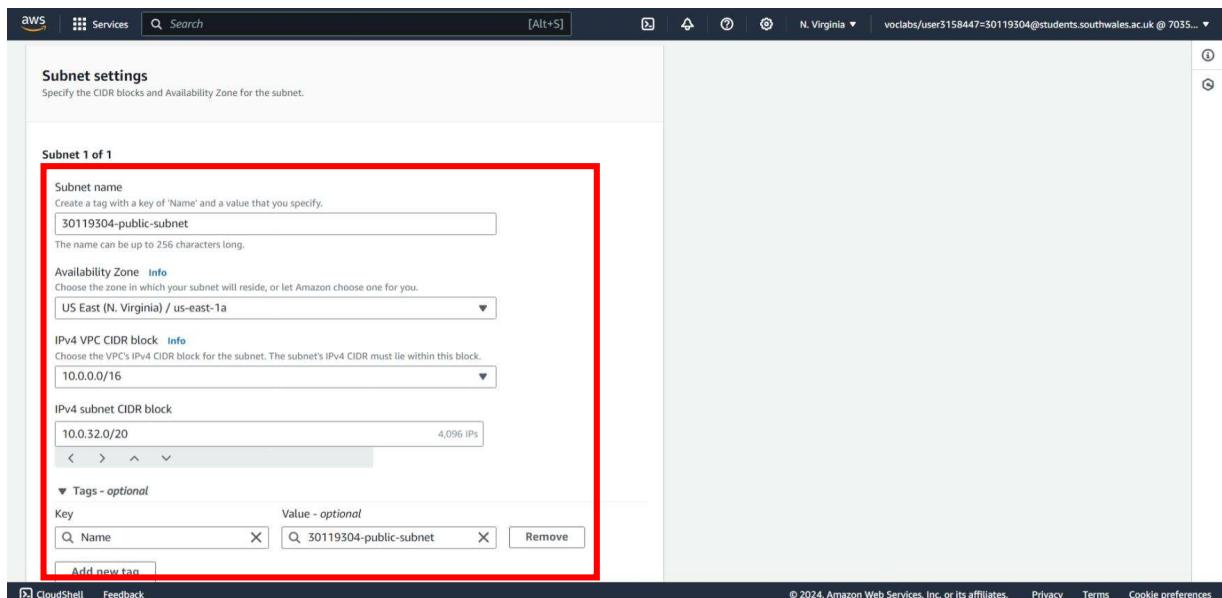


Fig 10: Subnet page

- Review the public subnet configurations then click on the “Create subnet” button.

Create a tag with a key of 'Name' and a value that you specify.  
30119304-public-subnet  
The name can be up to 256 characters long.

**Availability Zone** info  
Choose the zone in which your subnet will reside, or let Amazon choose one for you.  
US East (N. Virginia) / us-east-1a

**IPv4 VPC CIDR block** info  
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.  
10.0.0.0/16

**IPv4 subnet CIDR block**  
10.0.32.0/20 4,096 IPs

**Tags - optional**

| Key  | Value - optional       |
|------|------------------------|
| Name | 30119304-public-subnet |

Add new tag  
You can add 49 more tags.  
Remove  
Add new subnet

Cancel **Create subnet**

**Fig 11:** Subnet page

10. Click on the **Subnet** on the left-hand menu you can see your **30119304-public-subnet**.

You have successfully created 1 subnet: subnet-04b07dab282db10d

**Subnets (1) Info**

| Name                   | Subnet ID               | State     | VPC                              | IPv4 CIDR    |
|------------------------|-------------------------|-----------|----------------------------------|--------------|
| 30119304-public-subnet | subnet-04b07dab282db10d | Available | vpc-02e9e86b1962eafcd   30119304 | 10.0.32.0/20 |

Select a subnet

**Fig 12:** Subnet page

11. Now create a Private Subnet.

- Click on "**Subnets**" in the left-hand menu.
- Click on "**Create subnet**".

The screenshot shows the AWS VPC Subnets page. On the left, there's a sidebar with 'Subnets' selected. The main area displays a table of subnets with columns for Name, Subnet ID, State, VPC, and IPv4 CIDR. A green arrow points to the 'Create subnet' button at the top right of the table.

| Name                                  | Subnet ID                | State     | VPC                             | IPv4 CIDR      |
|---------------------------------------|--------------------------|-----------|---------------------------------|----------------|
| -                                     | subnet-09b925d68c49e07b2 | Available | vpc-046e67d6501b369bd           | 172.31.16.0/20 |
| 30119304-box-stream-vpc-subnet-public | subnet-061a22382d9e87dc6 | Available | vpc-02e9e86b1962eafcd   3011... | 10.0.16.0/20   |
| 0000-VPC-subnet-public2-us-east-1b    | subnet-0b6dc9e8b63a9359b | Available | vpc-01ff227b4c57e588a   0000... | 10.0.16.0/20   |
| 30119304-box-stream-vpc-subnet-public | subnet-0d7600b368a7d7bae | Available | vpc-02e9e86b1962eafcd   3011... | 10.0.0.0/20    |
| -                                     | subnet-06a796d89f544aa8f | Available | vpc-046e67d6501b369bd           | 172.31.80.0/20 |
| 0000-VPC-subnet-private1-us-east-1a   | subnet-063327cf737416fab | Available | vpc-01ff227b4c57e588a   0000... | 10.0.128.0/20  |
| -                                     | subnet-09b563a5787093042 | Available | vpc-046e67d6501b369bd           | 172.31.32.0/20 |
| -                                     | subnet-0104fe8da8c28ed55 | Available | vpc-046e67d6501b369bd           | 172.31.64.0/20 |

**Fig 13:** Subnet page

## 12. Begin to create private subnet configuration:

- Select your VPC 30119304-box-stream-vpc.
- Name: 30119304-private-subnet
- IPv4 CIDR block: 10.0.64.0/20
- Availability Zone: us-east-1a

The screenshot shows the 'Create subnet' page. In the 'VPC' section, the 'VPC ID' dropdown is highlighted with a red box and contains the value 'vpc-02e9e86b1962eafcd (30119304-box-stream-vpc-vpc)'. The 'Associated VPC CIDRs' section shows 'IPv4 CIDRs' as '10.0.0.0/16'. The 'Subnet settings' section includes a 'Subnet 1 of 1' configuration where the 'Subnet name' is set to '30119304-private-subnet'.

**Fig 14:** Subnet page

Subnet 1 of 1

**Subnet name**  
Create a tag with a key of 'Name' and a value that you specify.  
  
The name can be up to 256 characters long.

**Availability Zone** [Info](#)  
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

**IPv4 VPC CIDR block** [Info](#)  
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

**IPv4 subnet CIDR block**  
 4,096 IPs

**Tags - optional**

|                                   |  |
|-----------------------------------|--|
| Key                               | Value - optional                                     |
| <input type="text" value="Name"/> | <input type="text" value="30119304-private-subnet"/> |

[Add new tag](#)  
You can add 49 more tags.  
[Remove](#)

[Add new subnet](#)

**Fig 15:** Subnet page

13. Review the private subnet configurations then click on the “**Create subnet**” button.

Subnet 1 of 1

**Subnet name**  
Create a tag with a key of 'Name' and a value that you specify.  
  
The name can be up to 256 characters long.

**Availability Zone** [Info](#)  
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

**IPv4 VPC CIDR block** [Info](#)  
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

**IPv4 subnet CIDR block**  
 4,096 IPs

**Tags - optional**

|                                   |  |
|-----------------------------------|--|
| Key                               | Value - optional                                     |
| <input type="text" value="Name"/> | <input type="text" value="30119304-private-subnet"/> |

[Add new tag](#)  
You can add 49 more tags.  
[Remove](#)

[Add new subnet](#)

[Cancel](#) **Create subnet**

**Fig 16:** Subnet page

14. Now click on subnet in the left-hand menu and you can see both 30119304-public-subnet and 30119304-private-subnet.

The screenshot shows the AWS VPC dashboard with the 'Subnets' section selected. A message at the top indicates 'You have successfully created 1 subnet: subnet-05820ec13a51ac668'. The table lists several subnets, including:

| Name                                   | Subnet ID                       | State            | VPC                                    | IPv4 CIDR           |
|--|---------------------------------|------------------|--|---------------------|
| 0000-VPC-subnet-private2-us-east-1b    | subnet-0ca71c3405bcfc935        | Available        | vpc-01ff227b4c57e588a   0000...        | 10.0.144.0/20       |
| 30119304-box-stream-vpc-subnet-priv... | subnet-09d0b603fbfa15394f       | Available        | vpc-02e9e86b1962eafcd   3011...        | 10.0.144.0/20       |
| -                                      | subnet-0c07712886bb785c0        | Available        | vpc-046e67d6501b369bd                  | 172.31.48.0/20      |
| 30119304-box-stream-vpc-subnet-priv... | subnet-0f0e6788a1f5f322e        | Available        | vpc-02e9e86b1962eafcd   3011...        | 10.0.128.0/20       |
| -                                      | subnet-060bf58fb761adb1         | Available        | vpc-046e67d6501b369bd                  | 172.31.0.0/20       |
| <b>30119304-public-subnet</b>          | <b>subnet-04b07dafb282db10d</b> | <b>Available</b> | <b>vpc-02e9e86b1962eafcd   3011...</b> | <b>10.0.32.0/20</b> |
| <b>30119304-private-subnet</b>         | <b>subnet-05820ec13a51ac668</b> | <b>Available</b> | <b>vpc-02e9e86b1962eafcd   3011...</b> | <b>10.0.64.0/20</b> |

Fig 17: Subnet page

## 15. Create an Internet Gateway.

- Click on "**Internet Gateways**" in the left-hand menu.
- Click on "**Create internet gateway**".
- Name the Internet Gateway as "**30119304-igw**".
- Click on the "**Create internet gateway**" button.

The screenshot shows the AWS VPC dashboard with the 'Internet gateways' section selected. A message at the top indicates 'You have successfully created 3 internet gateways'. The table lists three internet gateways:

| Name                        | Internet gateway ID   | State    | VPC ID                                | Owner        |
|-----------------------------|-----------------------|----------|---------------------------------------|--------------|
| -                           | igw-017a4907e898755cd | Attached | vpc-046e67d6501b369bd                 | 703538516449 |
| 0000-new-igw                | igw-0ac17cfac57448c91 | Attached | vpc-01ff227b4c57e588a   0000-VPC-vpc  | 703538516449 |
| 30119304-box-stream-vpc-igw | igw-0cdd25e560142b253 | Attached | vpc-02e9e86b1962eafcd   30119304-b... | 703538516449 |

Fig 18: Internet gateway

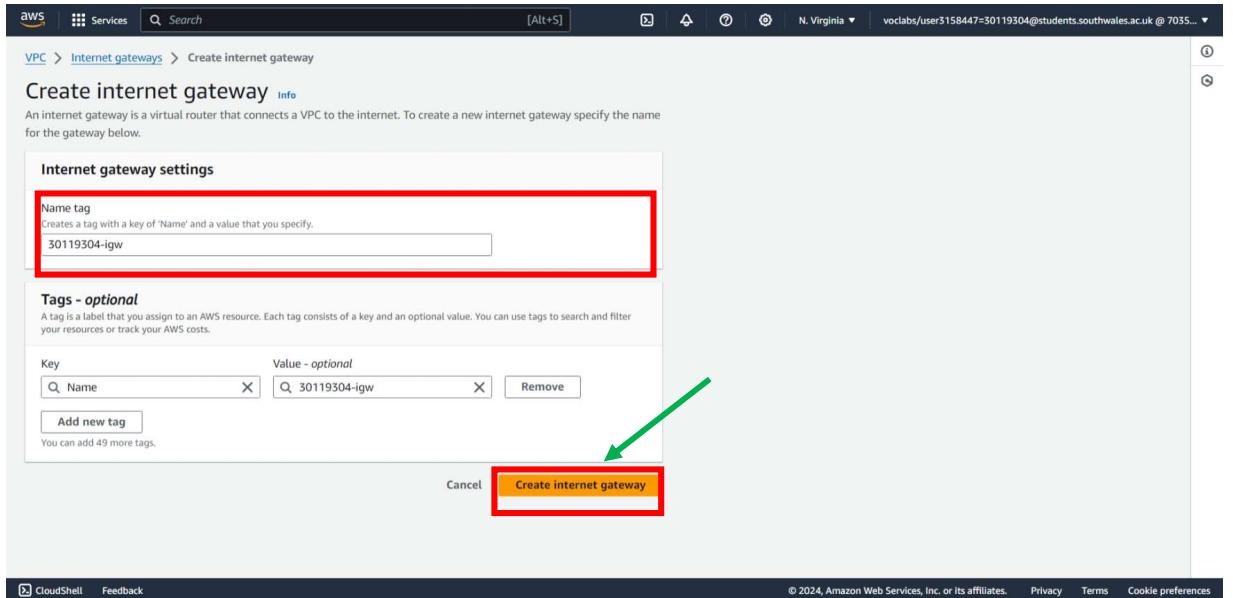


Fig 19: Internet gateway

## 16. Select the newly created Internet Gateway.

- Click on "Actions" and select "Attach to VPC".
- Select your "VPC 30119304-box-stream-vpc" and click on "Attach internet gateway".

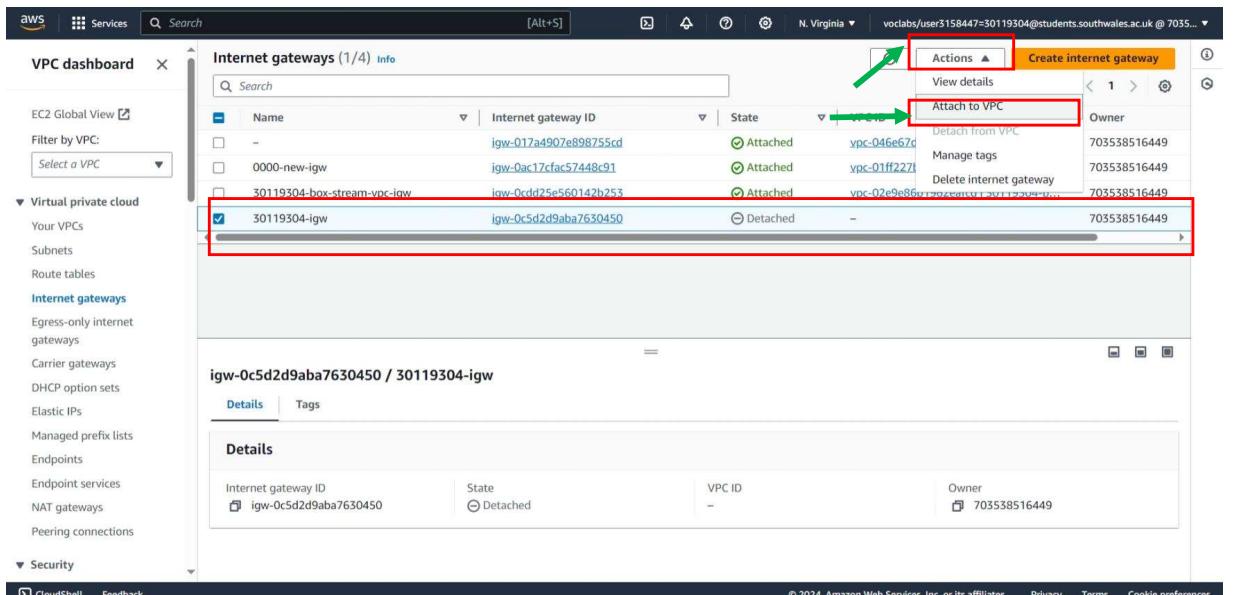


Fig 20: Internet gateway

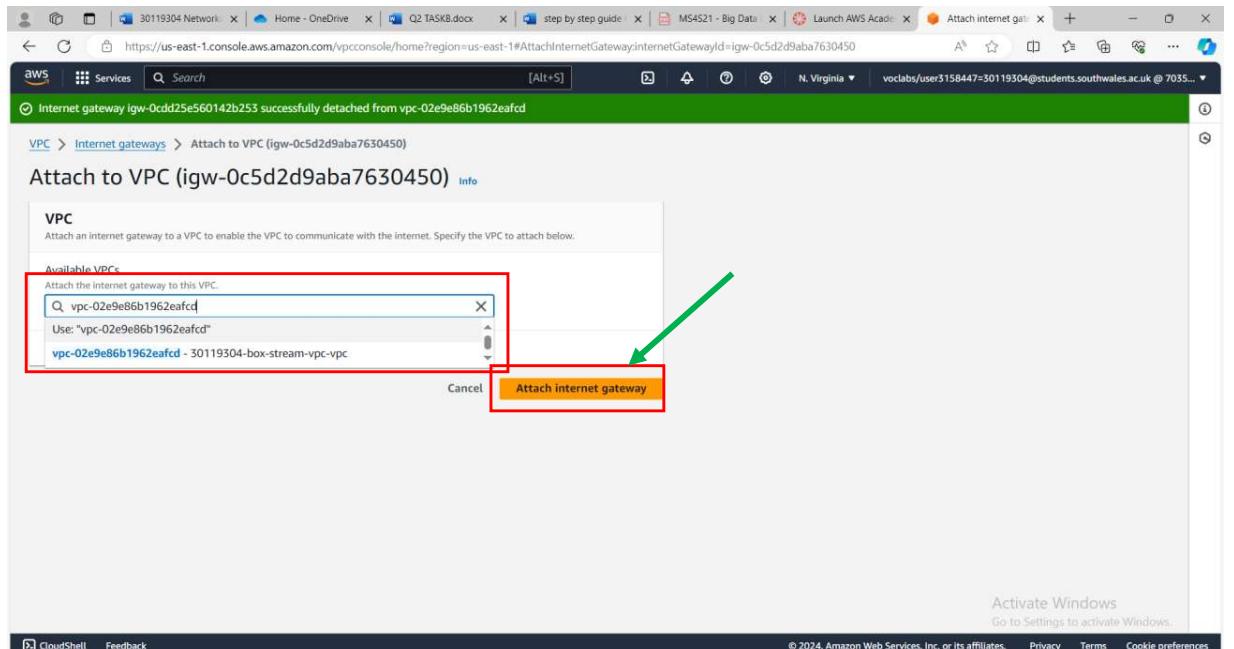
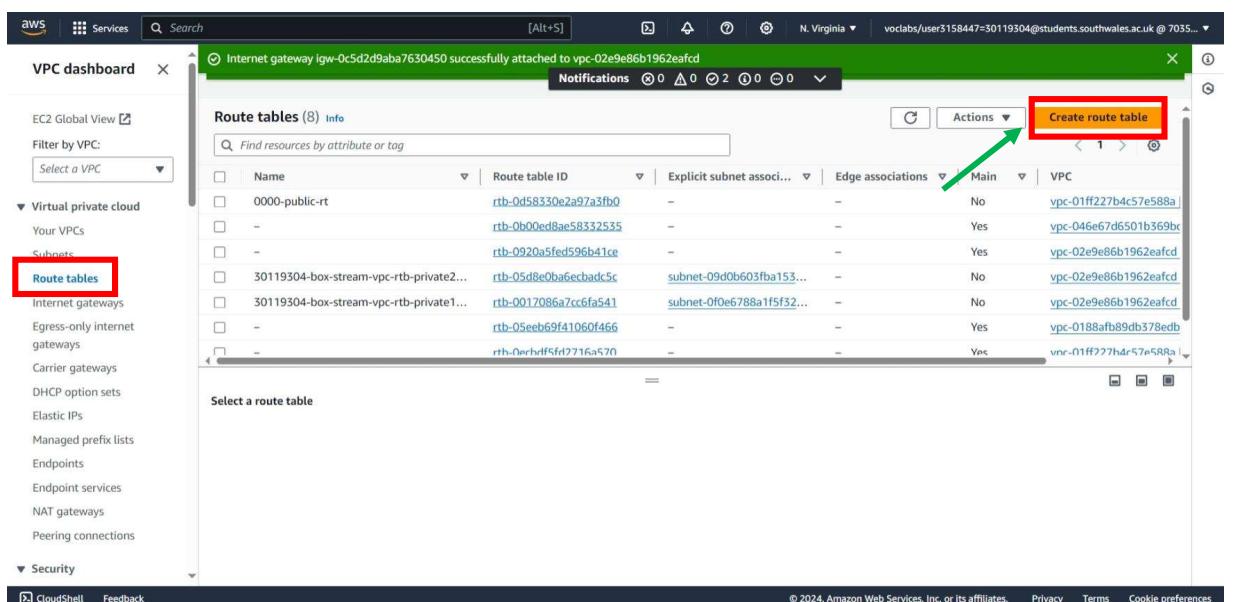
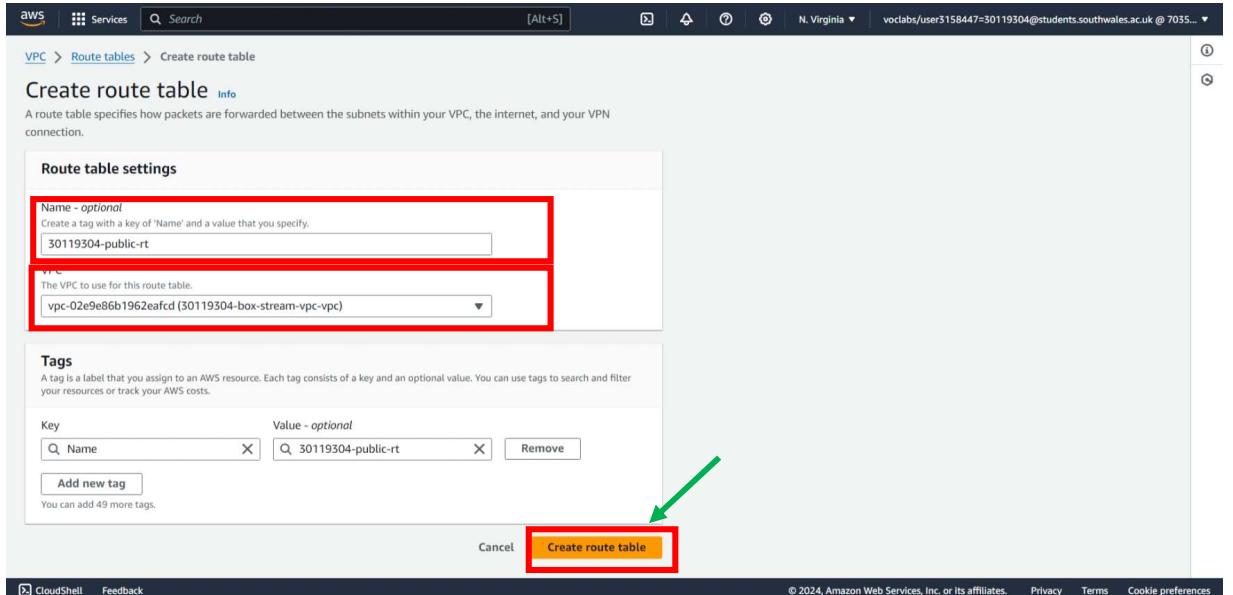


Fig 21: Internet gateway

### 17. Create a Route Table for Public Subnet.

- Click on "**Route Tables**" on the left-hand menu.
- Click on "**Create route table**".
- Name it as "**30119304-public-rt**".
- Select your VPC "**30119304-box-stream-vpc**".
- Leave other setting as default
- Then Click on "**Create route table**"



**Fig 21:** Route tables**Fig 22:** Route tables

**18. Add a Route to the Internet Gateway\*\*.**

- Select the newly created route table “**30119304-public-rt**”.
- Click on the “**Routes**” tab.
- Click on “**Edit routes**”.
- Click on “**Add route**”.
- Set Destination as “**0.0.0.0/0**”.
- Set Target as your Internet Gateway “**30119304-igw**”.
- Click on “**Save changes**”.

The screenshot shows the AWS VPC dashboard with the 'Route tables' section selected. A red box highlights the 'Actions' dropdown menu, and another red box highlights the 'Edit routes' link within that menu. The table lists various route tables, and the row for '30119304-public-rt' is selected.

| Name               | Route table ID        | Explicit subnet associations | Edge associations |
|--------------------|-----------------------|------------------------------|-------------------|
| 30119304-public-rt | rtb-0445420d7a654e9c3 | -                            | -                 |

Fig 23: Route tables

The screenshot shows the 'Edit routes' page for the route table 'rtb-0445420d7a654e9c3'. It displays a table with columns: Destination, Target, Status, and Propagated. A new route is being added with the destination '10.0.0.0/16' and target 'local'. The 'Add route' button is highlighted with a red box. At the bottom right, the 'Save changes' button is also highlighted with a red box.

| Destination | Target           | Status | Propagated |
|-------------|------------------|--------|------------|
| 10.0.0.0/16 | local            | Active | No         |
| 0.0.0.0/0   | Internet Gateway |        | No         |

Fig 24: Route tables

### 19. Associate Public Subnet with Route Table.

- Click on the "**Subnet Associations**" tab.
- Click on "**Edit subnet associations**".
- Select your public subnet "**30119304-public-subnet**".
- Click on "**Save associations**"

VPC dashboard > Updated routes for rtb-0445420d7a654e9c3 / 30119304-public-rt successfully

rtb-0445420d7a654e9c3 / 30119304-public-rt

**Details**

|  |                          |                                   |                        |
|--|--------------------------|-----------------------------------|------------------------|
| Route table ID<br>rtb-0445420d7a654e9c3                    | Main<br>No               | Explicit subnet associations<br>- | Edge associations<br>- |
| VPC<br>vpc-02e9e86b1962eafcd   30119304-box-stream-vpc-vpc | Owner ID<br>703538516449 |                                   |                        |

Routes Subnet associations Edge associations Route propagation Tags

**Explicit subnet associations (0)**

Find subnet association

| Name   | Subnet ID | IPv4 CIDR | IPv6 CIDR |
|--|-----------|-----------|-----------|
| No subnet associations<br>You do not have any subnet associations. |           |           |           |

Edit subnet associations

Fig 25: Route tables

VPC > Route tables > rtb-0445420d7a654e9c3 > Edit subnet associations

### Edit subnet associations

Change which subnets are associated with this route table.

**Available subnets (1/6)**

| Name   | Subnet ID                 | IPv4 CIDR     | IPv6 CIDR | Route table ID                        |
|--|---------------------------|---------------|-----------|---------------------------------------|
| 30119304-box-stream-vpc-subnet-...                         | subnet-061a22382d9e87dc6  | 10.0.16.0/20  | -         | rtb-0c89484b87a4758f0 / 30119304...   |
| 30119304-box-stream-vpc-subnet-...                         | subnet-0d7600b368a7d7bae  | 10.0.0.0/20   | -         | rtb-0c89484b87a4758f0 / 30119304...   |
| 30119304-box-stream-vpc-subnet-...                         | subnet-09d0b603fb15394f   | 10.0.144.0/20 | -         | rtb-05d8e0ba6ecbadc5c / 30119304-b... |
| 30119304-box-stream-vpc-subnet-...                         | subnet-0f06e788a1f5f322e  | 10.0.128.0/20 | -         | rtb-0017086a7c6f541 / 30119304-b...   |
| <input checked="" type="checkbox"/> 30119304-public-subnet | subnet-04b07dafb282db10d  | 10.0.32.0/20  | -         | Main (rtb-0920a5fed596b4fce)          |
| 30119304-private-subnet                                    | subnet-05820eet13a51ac668 | 10.0.64.0/20  | -         | Main (rtb-0920a5fed596b4fce)          |

**Selected subnets**

subnet-04b07dafb282db10d / 30119304-public-subnet

Cancel Save associations

Fig 26: Route tables

## 20. Launch an EC2 Instance in the Public Subnet

- Click on "**Services**" and select "**EC2**" under Compute section.
- Click on the "**Launch Instance**" button.

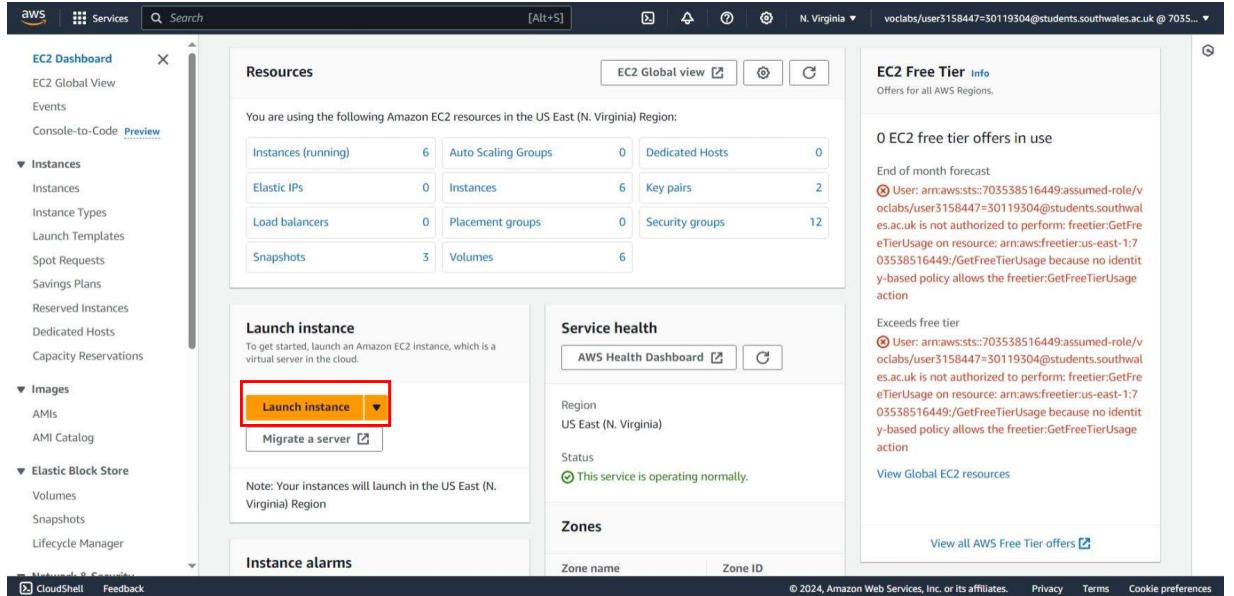


Fig 27: EC2 page

## 21. Begin to configure the instances details:

- Name: 30119304-web-server
- Amazon Machine Image: Amazon Linux 2023 AMI
- Instances type: t2.micro
- Key pair: vockey

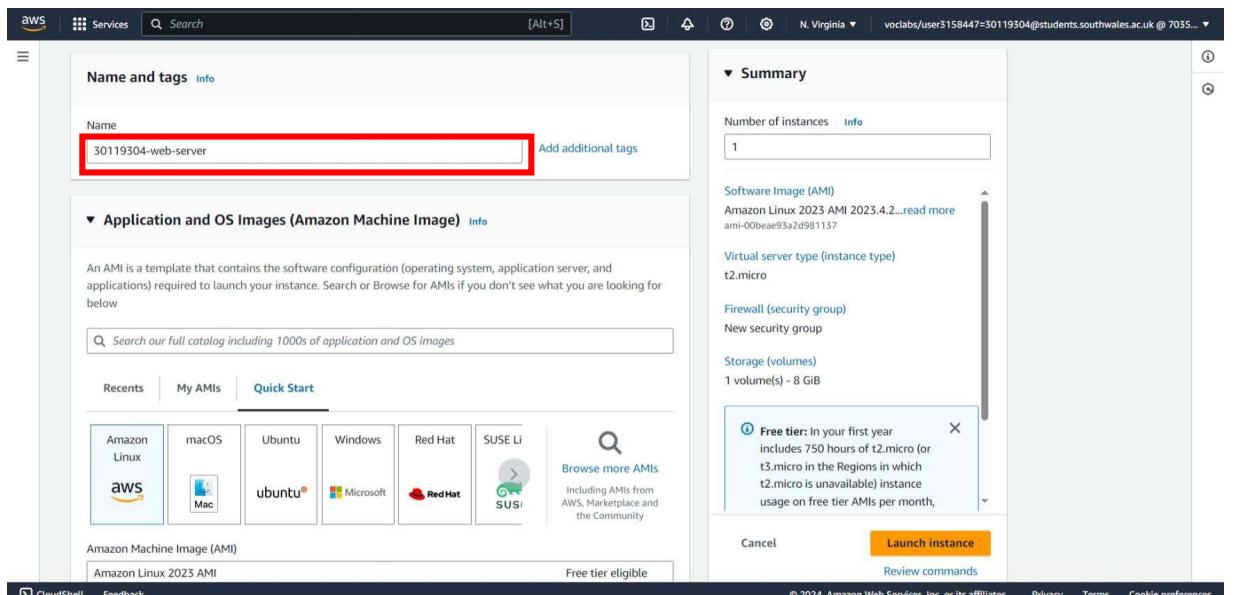


Fig 28: Instance page

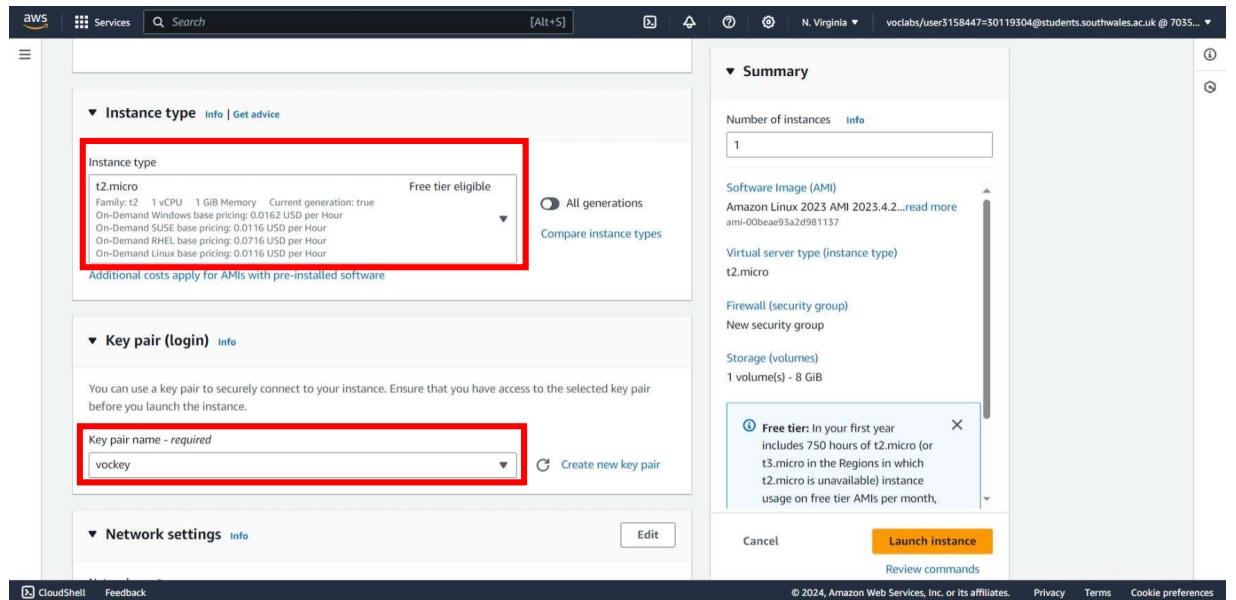


Fig 29: Instance page

## 22. Next configure the **network settings details**:

- VPC: 30119304-box-stream-vpc
- Subnet: 30119304-public-subnet
- Auto-assign public IP: Enable

## 23. Next configure the **security group details**:

- Security group name: 30119304-sg
- Description: Allow ssh access to developers
- Click on “**Add security group rule**” and change the following
  - Type: HTTP
  - Protocol: TCP
  - Port Range: 80
  - Source: Anywhere (0.0.0.0/0)

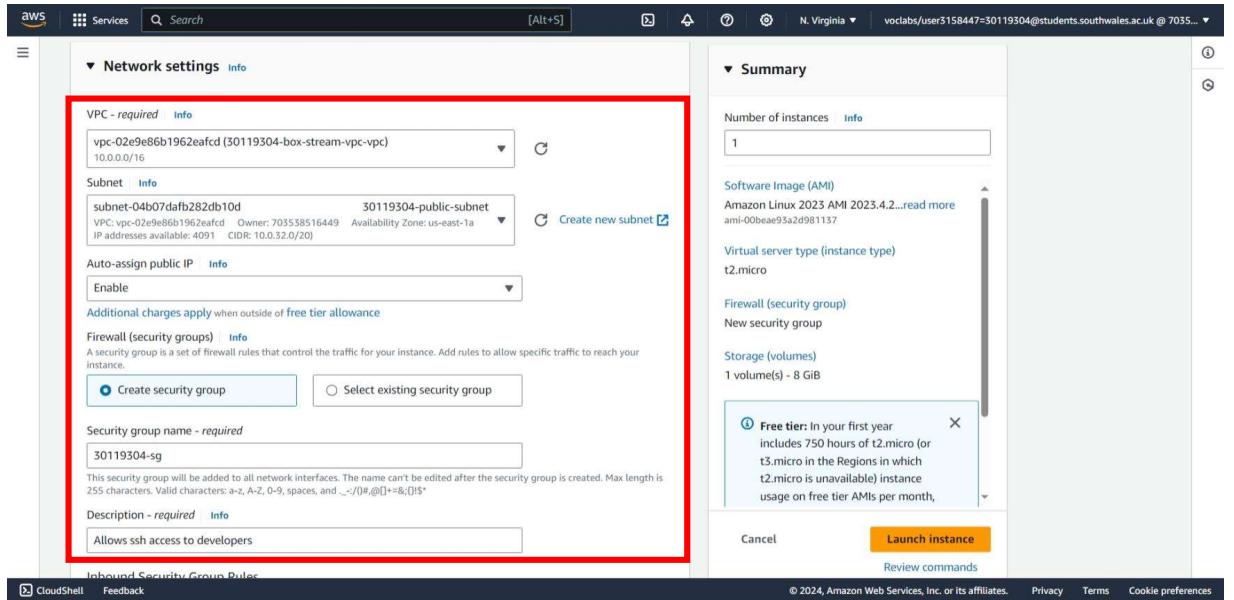


Fig 30: Instance page

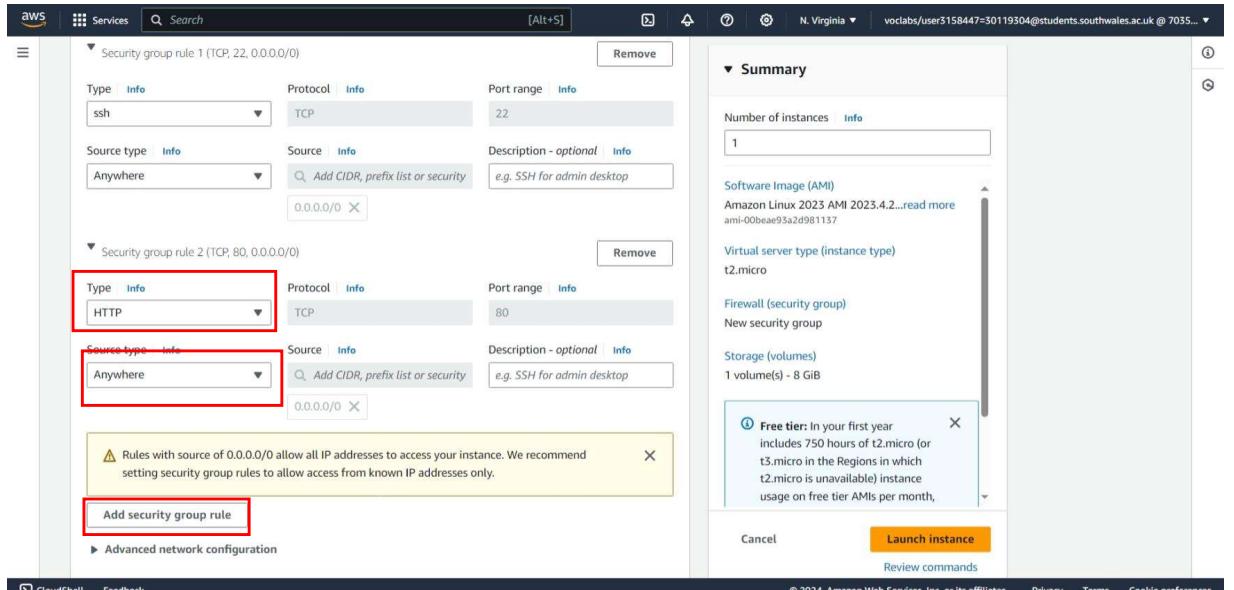


Fig 31: Instance page

24. Review all the configuration carefully and click on the “**Launch instance**” button.

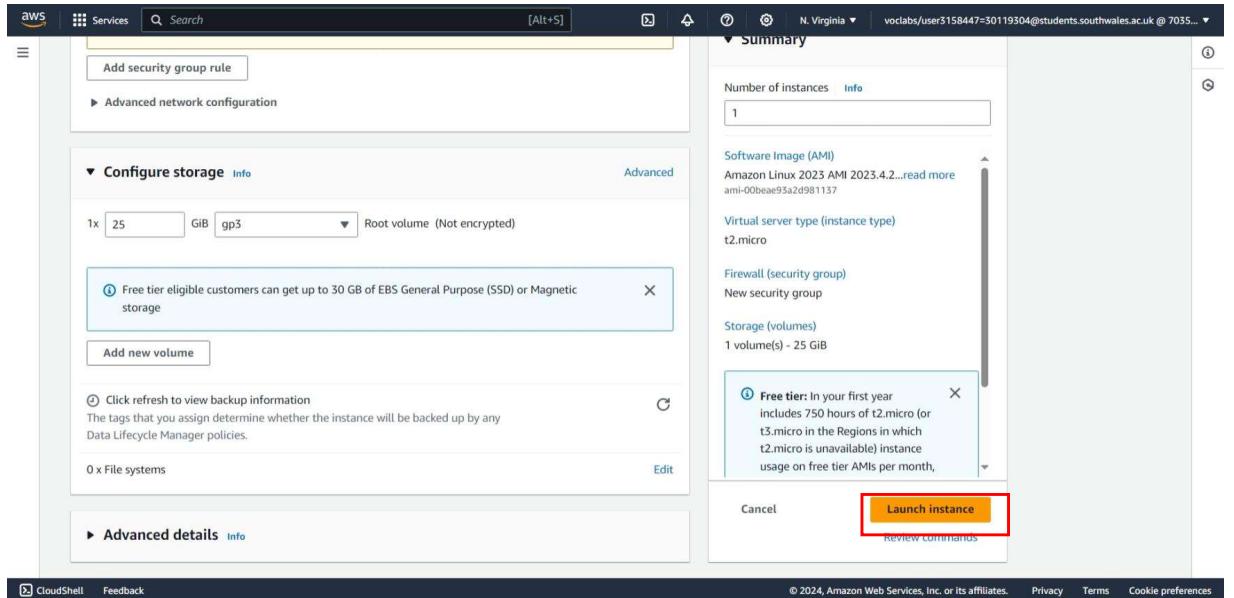


Fig 32: Instance page

25. Go to instances select your “30119304-web-server” then click on “Connect” button.

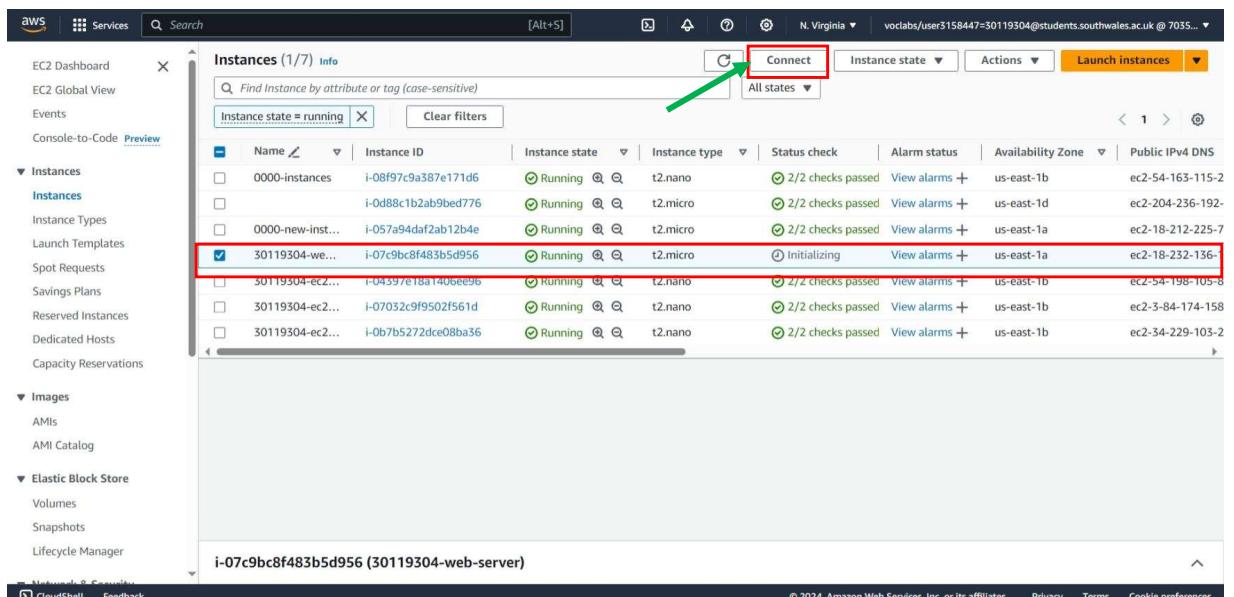


Fig 33: Instances page

26. Do not make any changes just click on the “Connect” button.

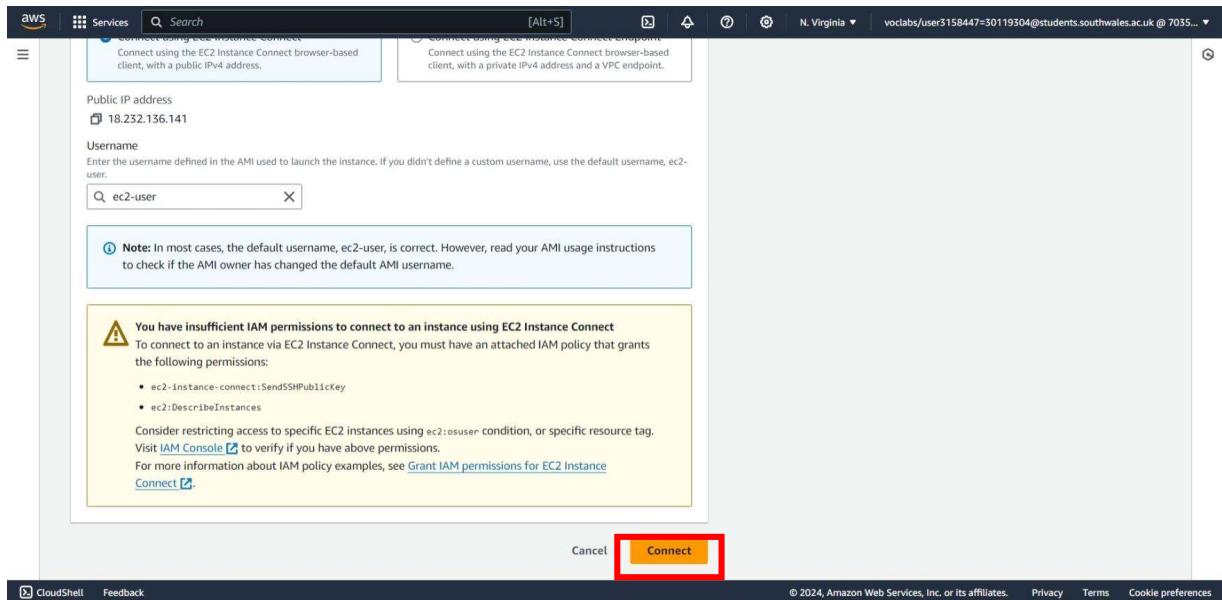


Fig 34: connect page

27. The new terminal will open and run the following command in the new terminal.

➤ sh

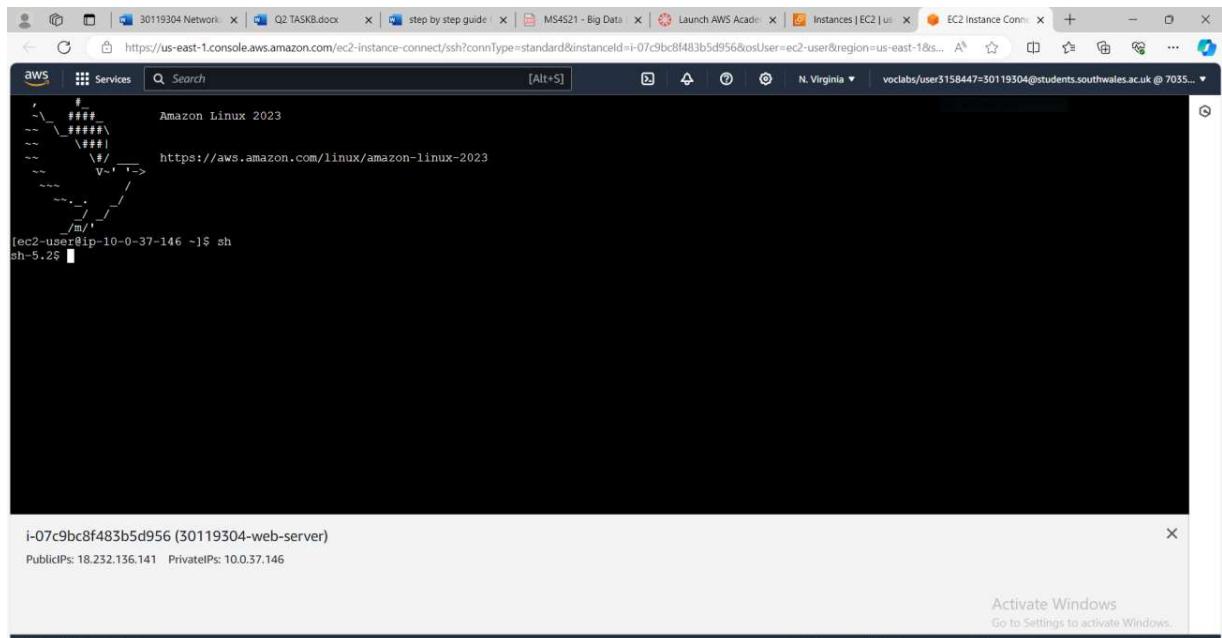


Fig 35: Terminal page

➤ sudo su -

```
[ec2-user@ip-10-0-37-146 ~]$ sh
sh-5.2$ sudo su -
[root@ip-10-0-37-146 ~]#
```

i-07c9bc8f483b5d956 (30119304-web-server)  
PublicIPs: 18.232.136.141 PrivateIPs: 10.0.37.146

Activate Windows  
Go to Settings to activate Windows.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**Fig 36:** Terminal page

➤ `yum update -y`

```
[ec2-user@ip-10-0-37-146 ~]$ sh
sh-5.2$ sudo su -
[root@ip-10-0-37-146 ~]# yum update -y
Last metadata expiration check: 0:04:12 ago on Wed Jun 5 15:09:22 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-10-0-37-146 ~]#
```

i-07c9bc8f483b5d956 (30119304-web-server)  
PublicIPs: 18.232.136.141 PrivateIPs: 10.0.37.146

Activate Windows  
Go to Settings to activate Windows.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**Fig 37:** Terminal page

➤ `yum install -y httpd`

```

Installing : libbrotli-1.0.9-4.amzn2023.0.2.x86_64
Running scriptlet: httpd-filesystem-2.4.59-2.amzn2023.noarch
Installing : httpd-filesystem-2.4.59-2.amzn2023.noarch
Installing : httpd-core-2.4.59-2.amzn2023.x86_64
Installing : mod_http2-2.0.27-1.amzn2023.0.2.x86_64
Installing : mod_lua-2.4.59-2.amzn2023.x86_64
Installing : generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch
Installing : httpd-2.4.59-2.amzn2023.x86_64
Running scriptlet: httpd-2.4.59-2.amzn2023.x86_64
Verifying : apr-1.7.2-2.amzn2023.0.2.x86_64
Verifying : apr-util-1.6.3-1.amzn2023.0.1.x86_64
Verifying : generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch
Verifying : httpd-2.4.59-2.amzn2023.x86_64
Verifying : httpd-filesystem-2.4.59-2.amzn2023.x86_64
Verifying : httpd-tools-2.4.59-2.amzn2023.x86_64
Verifying : libbrotli-1.0.9-4.amzn2023.0.2.x86_64
Verifying : mailcap-2.1.49-3.amzn2023.0.3.noarch
Verifying : mod_http2-2.0.27-1.amzn2023.0.2.x86_64
Verifying : mod_lua-2.4.59-2.amzn2023.x86_64

Installed:
apr-1.7.2-2.amzn2023.0.2.x86_64
generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch
httpd-filesystem-2.4.59-2.amzn2023.noarch
mailcap-2.1.49-3.amzn2023.0.3.noarch

Complete!
[root@ip-10-0-37-146 ~]#

```

i-07c9bc8f483b5d956 (30119304-web-server)  
PublicIPs: 18.232.136.141 PrivateIPs: 10.0.37.146

**Fig 38:** Terminal page

- echo "Hey 30119304! Thanks for dropping by BoxStream, our game of the day is Valorant. Check it out!" > /var/www/html/index.html

```

Installing : apr-1.7.2-2.amzn2023.0.2.x86_64
installing : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
Installing : apr-util-1.6.3-1.amzn2023.0.1.x86_64
Installing : mailcap-2.1.49-3.amzn2023.0.3.noarch
Installing : httpd-filesystem-2.4.59-2.amzn2023.x86_64
Installing : libbrotli-1.0.9-4.amzn2023.0.2.x86_64
Running scriptlet: httpd-filesystem-2.4.59-2.amzn2023.noarch
Installing : httpd-filesystem-2.4.59-2.amzn2023.noarch
Installing : httpd-core-2.4.59-2.amzn2023.x86_64
Installing : mod_http2-2.0.27-1.amzn2023.0.2.x86_64
Installing : mod_lua-2.4.59-2.amzn2023.x86_64
Installing : generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch
Installing : httpd-2.4.59-2.amzn2023.x86_64
Running scriptlet: httpd-2.4.59-2.amzn2023.x86_64
Verifying : apr-1.7.2-2.amzn2023.0.2.x86_64
Verifying : apr-util-1.6.3-1.amzn2023.0.1.x86_64
Verifying : generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch
Verifying : httpd-2.4.59-2.amzn2023.x86_64
Verifying : httpd-filesystem-2.4.59-2.amzn2023.noarch
Verifying : httpd-tools-2.4.59-2.amzn2023.x86_64
Verifying : libbrotli-1.0.9-4.amzn2023.0.2.x86_64
Verifying : mailcap-2.1.49-3.amzn2023.0.3.noarch
Verifying : mod_http2-2.0.27-1.amzn2023.0.2.x86_64
Verifying : mod_lua-2.4.59-2.amzn2023.x86_64

Installed:
apr-1.7.2-2.amzn2023.0.2.x86_64
generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch
httpd-filesystem-2.4.59-2.amzn2023.noarch
mailcap-2.1.49-3.amzn2023.0.3.noarch

Complete!
[root@ip-10-0-37-146 ~]# echo "hey 30119304! Thanks for dropping by BoxStream,our game of the day is valorant,check it out!">/var/www/html/index.html
[root@ip-10-0-37-146 ~]#

```

**Fig 39:** Terminal page

- systemctl enable httpd

```

Installing : apr-util-1.6.3-1.amzn2023.0.1.x86_64
Installing : mailcap-2.1.49-3.amzn2023.0.3.noarch
Installing : httpd-tools-2.4.59-2.amzn2023.x86_64
Installing : libbrotli-1.0.9-4.amzn2023.0.2.x86_64
Running scriptlet: httpd-filesystem-2.4.59-2.amzn2023.noarch
Installing : httpd-filesystem-2.4.59-2.amzn2023.noarch
Installing : httpd-core-2.4.59-2.amzn2023.x86_64
Installing : mod_http2-2.0.27-1.amzn2023.0.2.x86_64
Installing : mod_lua-2.4.59-2.amzn2023.x86_64
Installing : generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch
Installing : httpd-2.4.59-2.amzn2023.x86_64
Running scriptlet: httpd-2.4.59-2.amzn2023.x86_64
Verifying   : apr-1.7.2-2.amzn2023.0.2.x86_64
Verifying   : apr-util-1.6.3-1.amzn2023.0.1.x86_64
Verifying   : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
Verifying   : generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch
Verifying   : httpd-2.4.59-2.amzn2023.x86_64
Verifying   : httpd-core-2.4.59-2.amzn2023.x86_64
Verifying   : httpd-filesystem-2.4.59-2.amzn2023.noarch
Verifying   : httpd-tools-2.4.59-2.amzn2023.x86_64
Verifying   : libbrotli-1.0.9-4.amzn2023.0.2.x86_64
Verifying   : mailcap-2.1.49-3.amzn2023.0.3.noarch
Verifying   : mod_http2-2.0.27-1.amzn2023.0.2.x86_64
Verifying   : mod_lua-2.4.59-2.amzn2023.x86_64

Installed:
  apr-1.7.2-2.amzn2023.0.2.x86_64           apr-util-1.6.3-1.amzn2023.0.1.x86_64           apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
  generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch  httpd-2.4.59-2.amzn2023.x86_64           httpd-core-2.4.59-2.amzn2023.x86_64
  httpd-filesystem-2.4.59-2.amzn2023.noarch    httpd-tools-2.4.59-2.amzn2023.x86_64           libbrotli-1.0.9-4.amzn2023.0.2.x86_64
  mailcap-2.1.49-3.amzn2023.0.3.noarch       mod_http2-2.0.27-1.amzn2023.0.2.x86_64           mod_lua-2.4.59-2.amzn2023.x86_64

Complete!
[root@ip-10-0-37-146 ~]# echo "hey 30119304! Thanks for dropping by BoxStream,our game of the day is valorant,check it out!">/var/www/html/index.html
[root@ip-10-0-37-146 ~]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-10-0-37-146 ~]#

```

Fig 40: Terminal page

```

Installing : mailcap-2.1.49-3.amzn2023.0.3.noarch
Installing : httpd-tools-2.4.59-2.amzn2023.x86_64
Installing : libbrotli-1.0.9-4.amzn2023.0.2.x86_64
Running scriptlet: httpd-filesystem-2.4.59-2.amzn2023.noarch
Installing : httpd-filesystem-2.4.59-2.amzn2023.noarch
Installing : httpd-core-2.4.59-2.amzn2023.x86_64
Installing : mod_http2-2.0.27-1.amzn2023.0.2.x86_64
Installing : mod_lua-2.4.59-2.amzn2023.x86_64
Installing : generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch
Installing : httpd-2.4.59-2.amzn2023.x86_64
Running scriptlet: httpd-2.4.59-2.amzn2023.x86_64
Verifying   : apr-1.7.2-2.amzn2023.0.2.x86_64
Verifying   : apr-util-1.6.3-1.amzn2023.0.1.x86_64
Verifying   : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
Verifying   : generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch
Verifying   : httpd-2.4.59-2.amzn2023.x86_64
Verifying   : httpd-core-2.4.59-2.amzn2023.x86_64
Verifying   : httpd-filesystem-2.4.59-2.amzn2023.noarch
Verifying   : httpd-tools-2.4.59-2.amzn2023.x86_64
Verifying   : libbrotli-1.0.9-4.amzn2023.0.2.x86_64
Verifying   : mailcap-2.1.49-3.amzn2023.0.3.noarch
Verifying   : mod_http2-2.0.27-1.amzn2023.0.2.x86_64
Verifying   : mod_lua-2.4.59-2.amzn2023.x86_64

Installed:
  apr-1.7.2-2.amzn2023.0.2.x86_64           apr-util-1.6.3-1.amzn2023.0.1.x86_64           apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
  generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch  httpd-2.4.59-2.amzn2023.x86_64           httpd-core-2.4.59-2.amzn2023.x86_64
  httpd-filesystem-2.4.59-2.amzn2023.noarch    httpd-tools-2.4.59-2.amzn2023.x86_64           libbrotli-1.0.9-4.amzn2023.0.2.x86_64
  mailcap-2.1.49-3.amzn2023.0.3.noarch       mod_http2-2.0.27-1.amzn2023.0.2.x86_64           mod_lua-2.4.59-2.amzn2023.x86_64

Complete!
[root@ip-10-0-37-146 ~]# echo "hey 30119304! Thanks for dropping by Boxstream,our game of the day is valorant,check it out!">/var/www/html/index.html
[root@ip-10-0-37-146 ~]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-10-0-37-146 ~]# systemctl start httpd
[root@ip-10-0-37-146 ~]#

```

Fig 41: Terminal page

28. Now navigate to instance in the left-hand menu and select 30119304-web-server and then click on Action, then select View details.

Instances (1/7) Info

| Name   | Instance ID          | Instance state | Instance type | Status check      | Alarm state              |
|--|----------------------|----------------|---------------|-------------------|--------------------------|
| <input checked="" type="checkbox"/> 30119304-we... | i-07c9bc8f483b5d956  | Running        | t2.micro      | 2/2 checks passed | View alarms              |
| 0000-instances                                     | i-08f197c9a387e171db | Running        | t2.nano       | 2/2 checks passed | View alarms              |
|  | i-0d88c1b2abb9ed776  | Running        | t2.micro      | 2/2 checks passed | View alarms              |
|  | i-057a94da12ab12b4e  | Running        | t2.micro      | 2/2 checks passed | View alarms              |
|  | i-04397e18a1406ee96  | Running        | t2.nano       | 2/2 checks passed | View alarms              |
|  | i-07032c9f9502f561d  | Running        | t2.nano       | 2/2 checks passed | View alarms + us-east-1b |
|  | i-0b7b5272dce08ba36  | Running        | t2.nano       | 2/2 checks passed | View alarms + us-east-1b |

i-07c9bc8f483b5d956 (30119304-web-server)

Fig 42: Instances page

## 29. Copy the Public IP address of 30119304-web-server

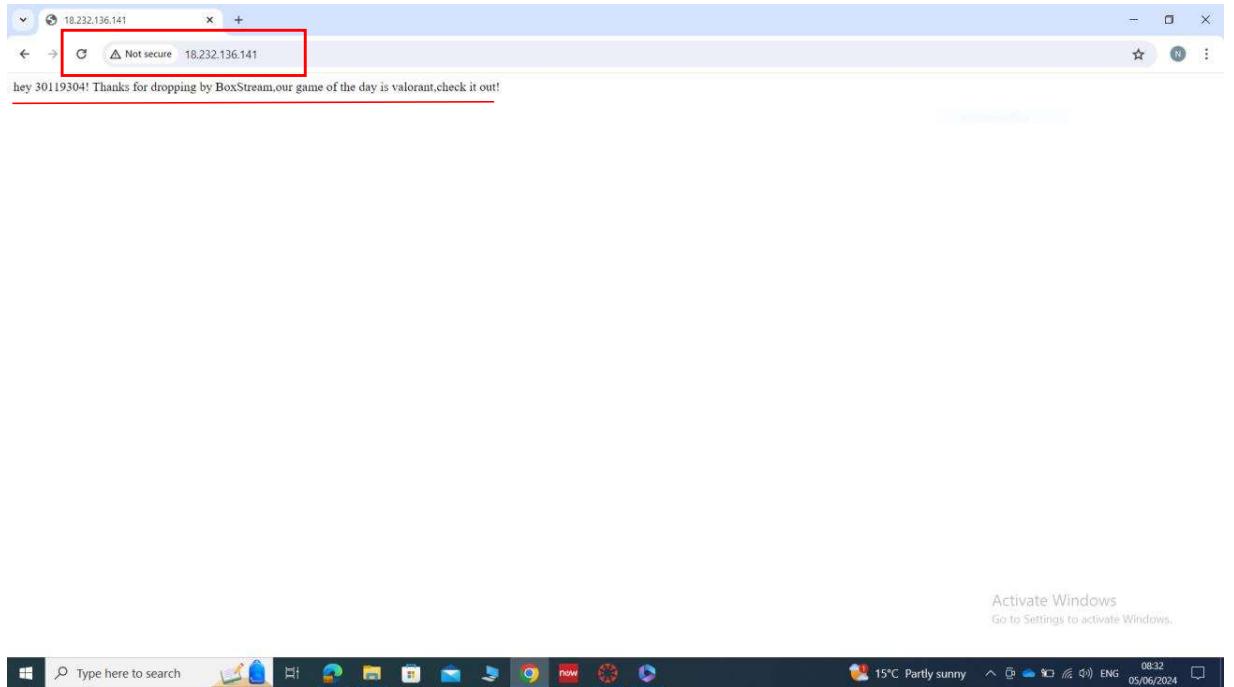
EC2 > Instances > i-07c9bc8f483b5d956

Instance summary for i-07c9bc8f483b5d956 (30119304-web-server) info

|  |  |   |
|--|--|---|
| Instance ID<br>i-07c9bc8f483b5d956 (30119304-web-server) | Public IPv4 address<br>18.232.156.141   open address           | Private IPv4 addresses<br>10.37.146   |
| IPv6 address<br>-  | Instance state<br>Running                                      | Public IPv4 DNS<br>ec2-18-232-136-141.compute-1.amazonaws.com   open address        |
| Hostname type<br>IP name: ip-10-0-37-146.ec2.internal    | Private IP DNS name (IPv4 only)<br>ip-10-0-37-146.ec2.internal | Elastic IP addresses<br>-   |
| Answer private resource DNS name<br>-                    | Instance type<br>t2.micro                                      | AWS Compute Optimizer finding<br>Opt-in to AWS Compute Optimizer for recommendation |
| Auto-assigned IP address<br>18.232.136.141 [Public IP]   | VPC ID<br>vpc-02e9e86b1962efcd (30119304-box-stream-vpc-vpo)   | Auto Scaling Group name<br>-  |
| IAM Role<br>-  | Subnet ID<br>subnet-04b07dafb282db10d (30119304-public-subnet) | Learn more  |
| IMDSv2<br>Required                                       | Instance ARN<br>arn:aws:ec2:us-east-1:703538516449:instance/i- |   |

Fig 43: 30119304-web-server page

## 30. Open the new browser paste that public IP address and press enter now You should see the message: "Hey 30119304! Thanks for dropping by BoxStream, our game of the day is Valorant. Check it out!".



**Fig 44:** New browser page

**Congratulations** you set up VPC is correctly configured with public and private subnets, the web application is deployed, and publicly accessible, and relevant screenshots are provided for verification. And host a web application on AWS EC2 effectively.