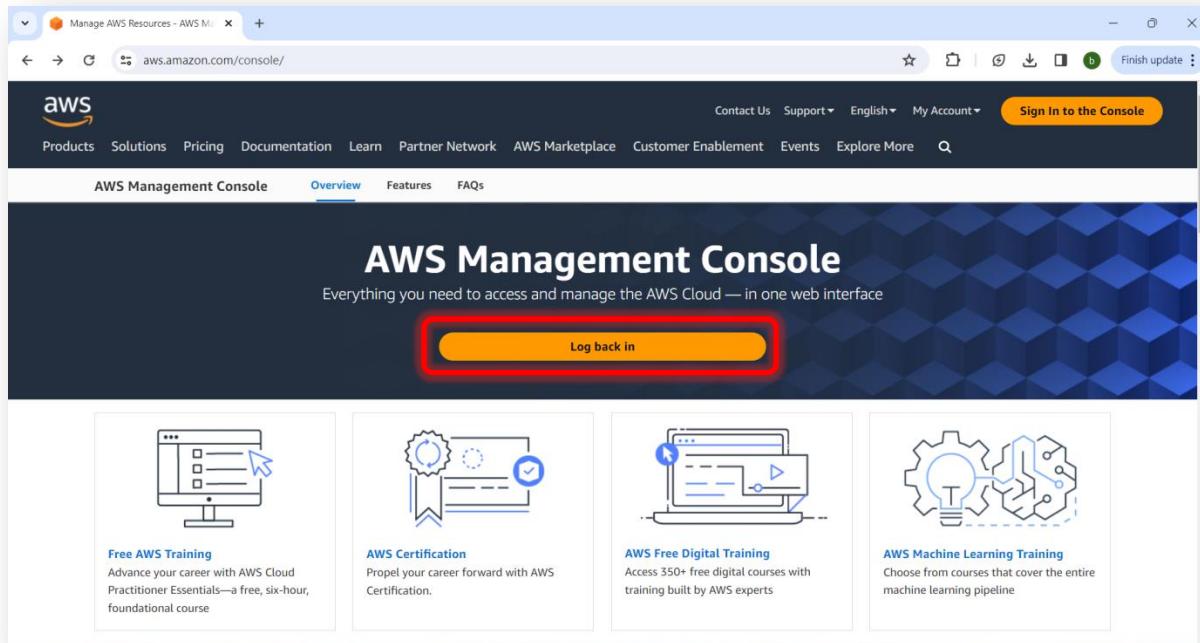
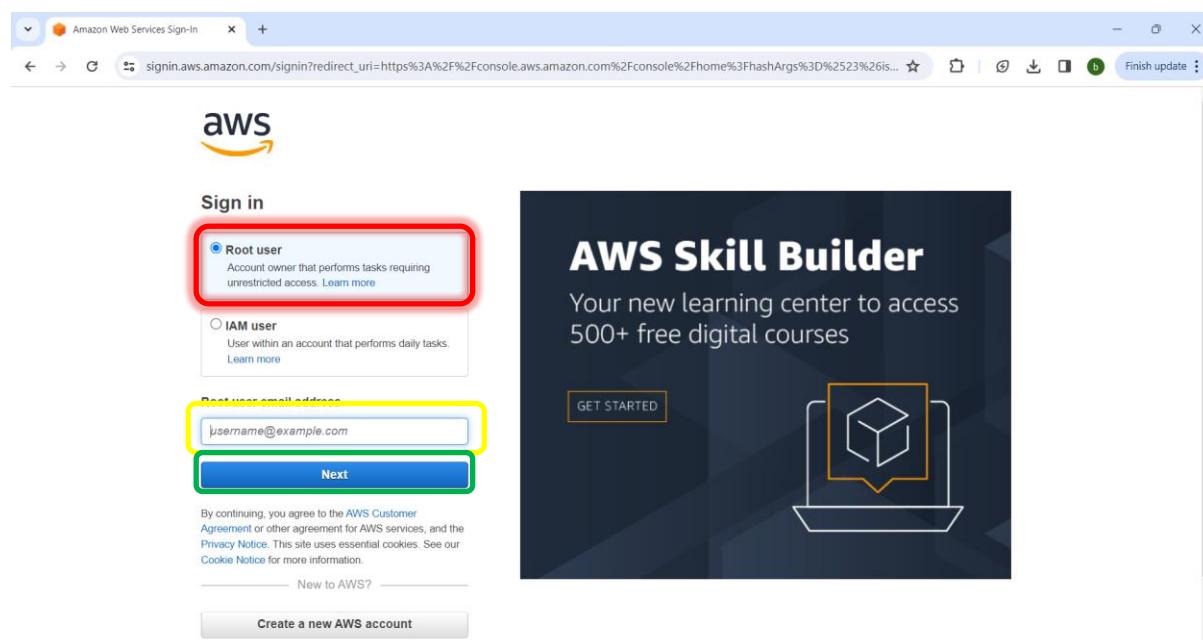


VPC (virtual private cloud) to EC2 creation.

1. Open AWS console → <https://aws.amazon.com/console/>
Click on Login Button

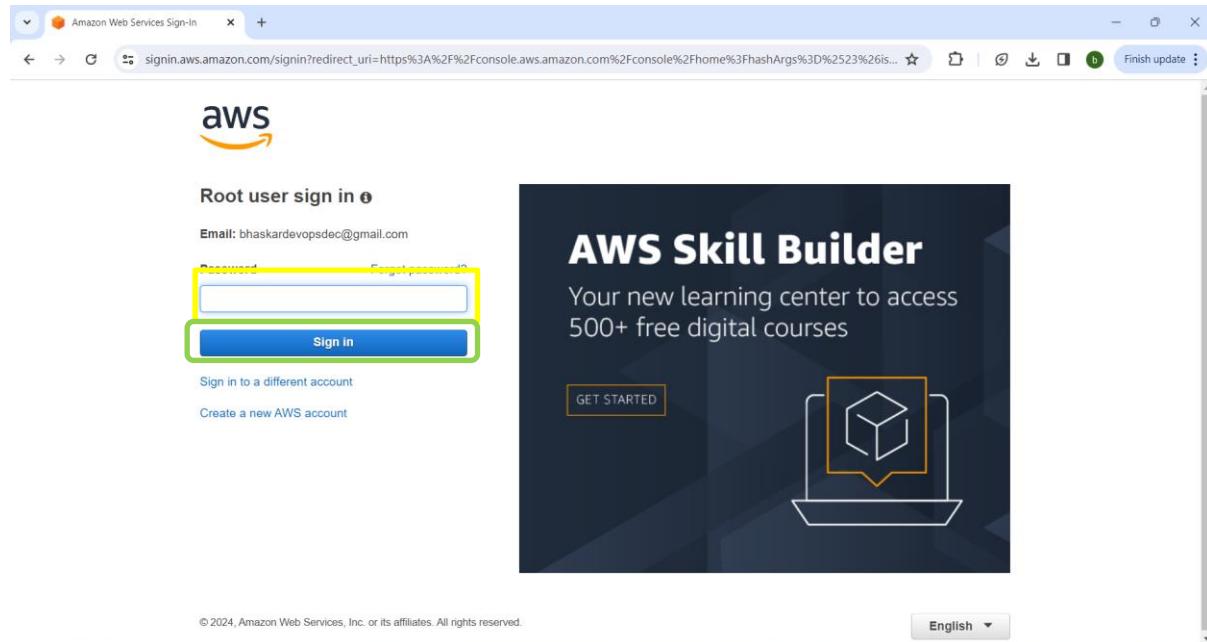


2. Select Root user → Enter your Root user email address
Click on Next.

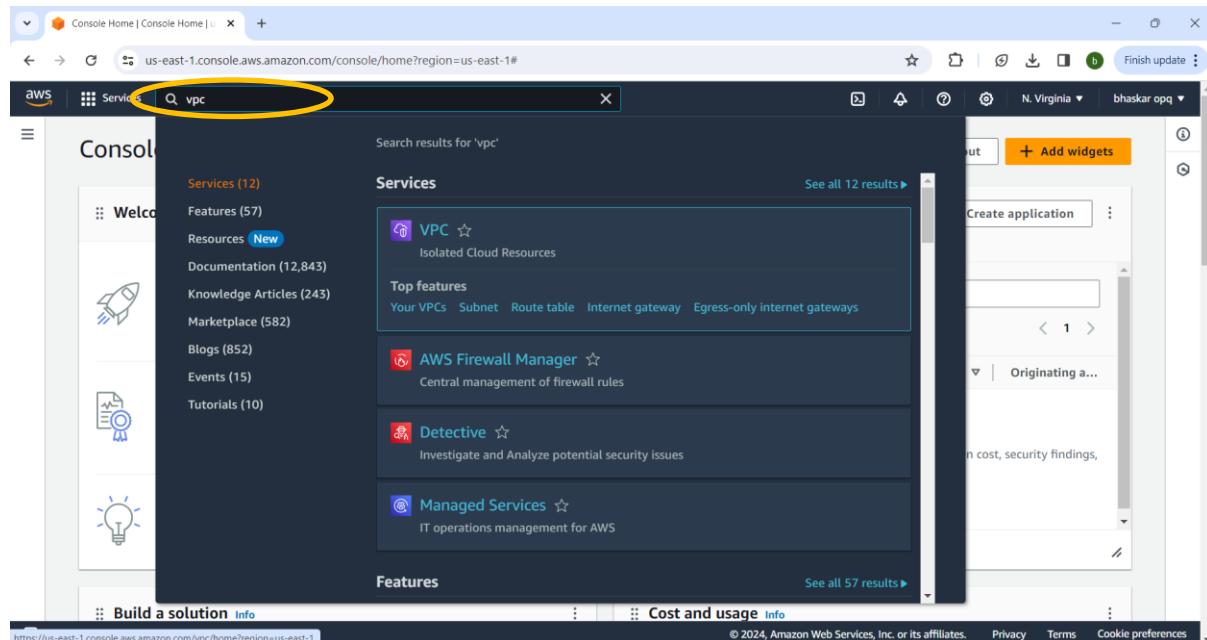


3. Enter your Root user email Password

Click on Sign in Button



4. In AWS search bar search on → Enter VPC → Select VPC→ Enter.



5. You will get this VPC UI (User interface) Page

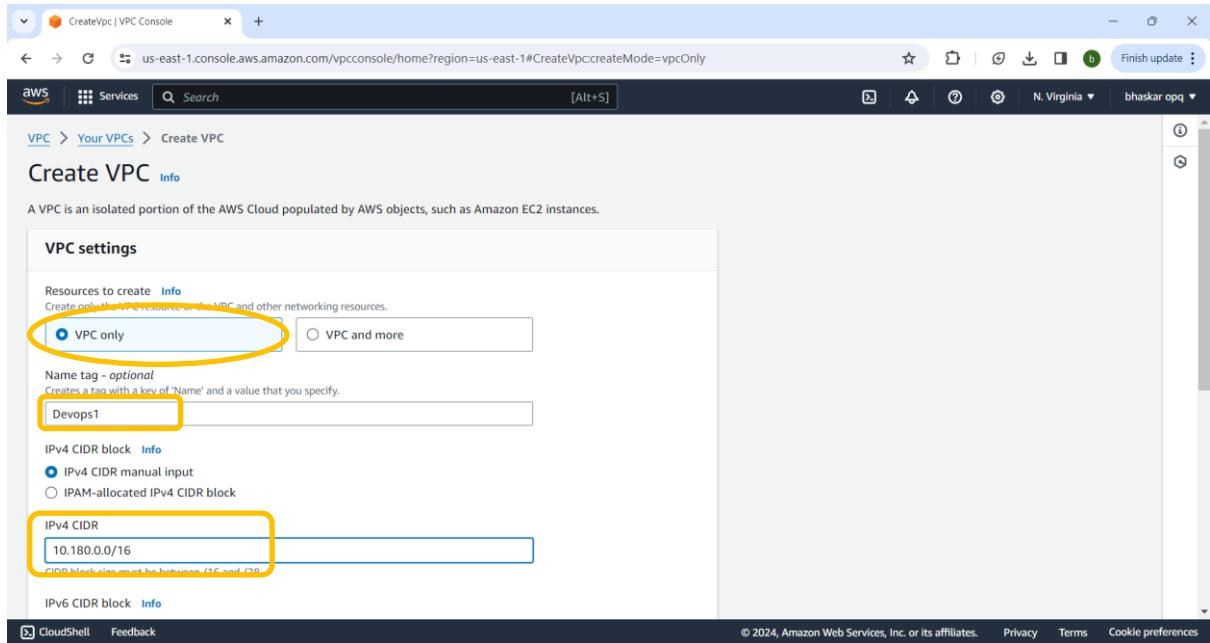
The screenshot shows the AWS VPC Console Home page. The left sidebar has a tree view under 'Virtual private cloud' with 'Your VPCs' selected and highlighted with a yellow circle. The main content area is titled 'Resources by Region' and lists various Amazon VPC resources with counts: VPCs (2), Subnets (10), Route Tables (4), Internet Gateways (2), NAT Gateways (0), and Security Groups (7). At the top right of the main content area, there is a 'Create VPC' button.

6. You can see Default VPCs

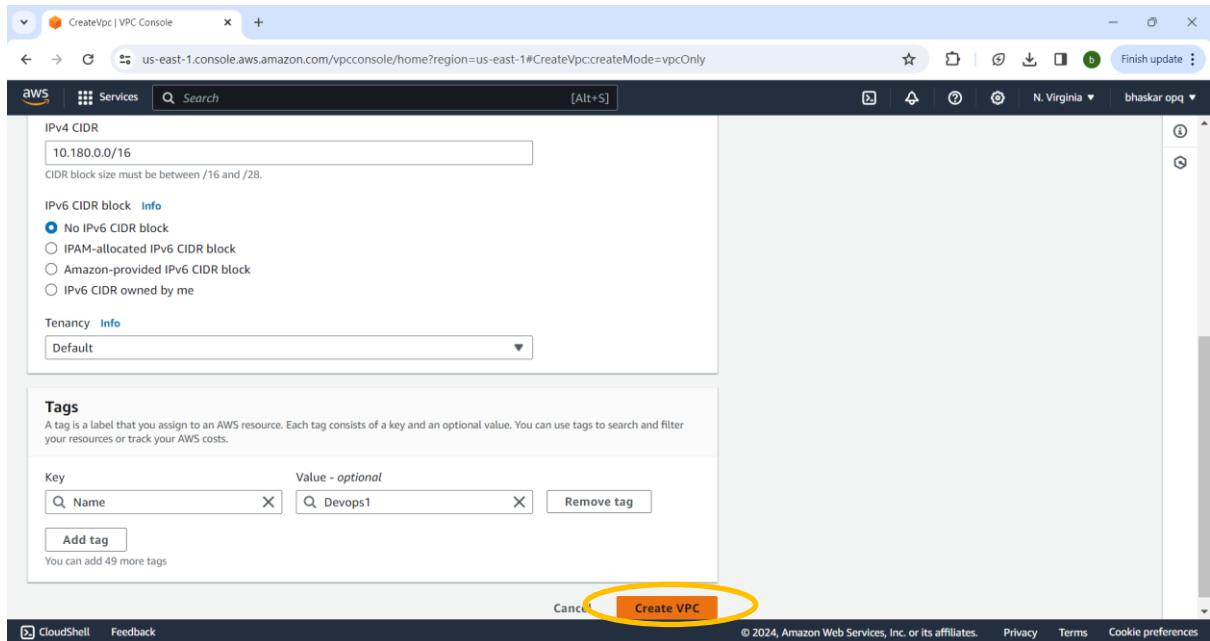
Click on Create VPC .RHs (top right corner)

The screenshot shows the AWS VPCs page. The left sidebar has a tree view under 'Virtual private cloud' with 'Your VPCs' selected and highlighted with a green box. The main content area displays a table of 'Your VPCs' with 2 entries: 'Devops' (VPC ID: vpc-0b6b61ab36ff1c384, State: Available, IPv4 CIDR: 10.180.0.0/16) and another entry (VPC ID: vpc-0a491cb5a5a3f1384, State: Available, IPv4 CIDR: 172.31.0.0/16). At the top right of the main content area, there is a large orange 'Create VPC' button.

7. Select VPC only and give IPv4 CIDR value 10.180.0.0/16



8. Remaining others Leave it As it is Default values → click on Create VPC



9. Your VPC will be created and you will get the You successfully created

The screenshot shows the AWS VPC Details page for a newly created VPC. A yellow circle highlights the green success message at the top: "You successfully created vpc-07e6f77b7f8d0cb6b / Devops1". The main table displays various details about the VPC, such as its ID, state, and CIDR range. The table has columns for VPC ID, State, DNS hostnames, DNS resolution, Tenancy, DHCP option set, Main route table, Main network ACL, Default VPC, IPv4 CIDR, IPv6 pool, Network Address Usage metrics, Route 53 Resolver DNS Firewall rule groups, Owner ID, and IPv6 CIDR.

10. Then you will come back Your VPCs

The screenshot shows the AWS VPC dashboard. A yellow circle highlights the "Your VPCs" link in the left sidebar under the "Virtual private cloud" section. The main area displays a table of three existing VPCs. The table includes columns for Name, VPC ID, State, IPv4 CIDR, and IPv6 CIDR. The VPCs listed are Devops, Devops1, and another unnamed VPC. The "Devops1" row is also highlighted with a yellow circle.

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
Devops	vpc-0b6b61ab36ff1c384	Available	10.180.0.0/16	-
Devops1	vpc-0a491cb5a5a3f1384	Available	172.31.0.0/16	-
	vpc-07e6f77b7f8d0cb6b	Available	10.180.0.0/16	-

11. Select Internet gateways(left hand side bar).

Select Create internet gateway (Right hand side Top corner)

The screenshot shows the AWS VPC Console interface. On the left, there's a sidebar with various VPC-related options like 'Your VPCs', 'Subnets', 'Route tables', and 'Internet gateways'. The 'Internet gateways' option is highlighted with a blue circle. The main content area displays a table titled 'Internet gateways (2) Info' with two entries: 'devops1GW' and another entry with a placeholder '—'. At the top right of this table, there's a 'Create internet gateway' button, which is also circled in blue. The browser address bar shows the URL 'https://us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#igws:'.

12. Enter Name in Name tag Field → devops1IGW

Click on→ create internet gateway

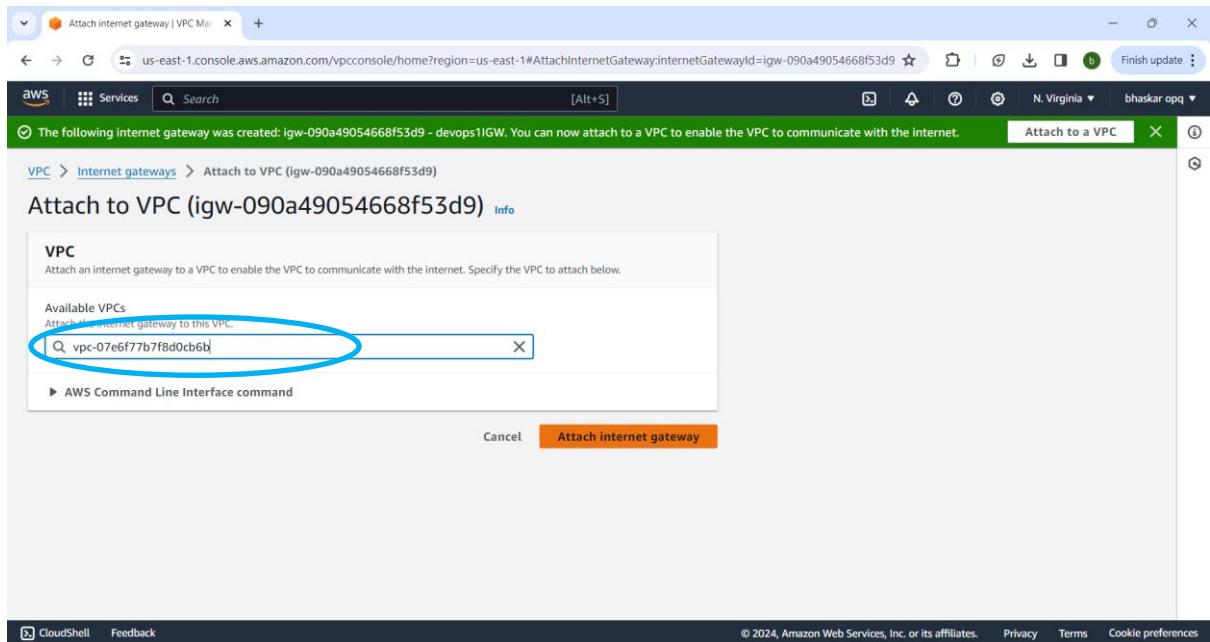
This screenshot shows the 'Create internet gateway' wizard. In the 'Internet gateway settings' section, the 'Name tag' field is filled with 'devops1IGW', which is circled in blue. Below this, there's a 'Tags - optional' section where a single tag 'Name: devops1IGW' is defined. At the bottom right of the wizard, there's a prominent 'Create internet gateway' button, which is also circled in blue. The browser address bar shows the URL 'https://us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateInternetGateway:'.

**13. You will get the this ui page and
RHs Top corner → select Attach to a VPC**

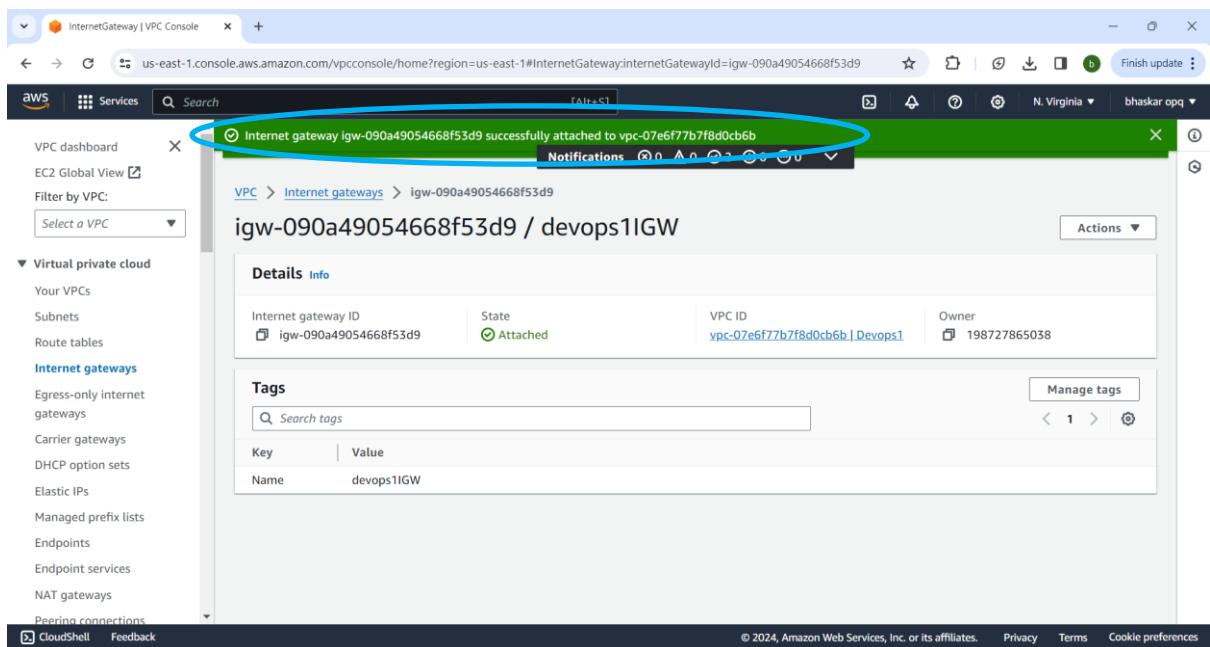
The screenshot shows the AWS VPC Internet Gateway console. At the top, there is a green banner message: "The following internet gateway was created: igw-090a49054668f53d9 - devops1IGW. You can now attach to a VPC to enable the VPC to communicate with the internet." Below this, the main interface shows details for the internet gateway, including its ID (igw-090a49054668f53d9), state (Detached), and owner (198727865038). A section for tags is also present. On the right side of the interface, there is a prominent blue button labeled "Attach to a VPC". This button is circled in red in the screenshot.

14. Select hear Available VPCs field → your created VPC → click on Attach internet gateway

The screenshot shows the "Attach to VPC" dialog box. It starts with a message: "The following internet gateway was created: igw-090a49054668f53d9 - devops1IGW. You can now attach to a VPC to enable the VPC to communicate with the internet." Below this, the title is "Attach to VPC (igw-090a49054668f53d9)". The main area is titled "VPC" with the sub-instruction: "Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below." A "Available VPCs" section follows, containing a dropdown menu with the placeholder "Select a VPC". The dropdown is circled in red. Below the dropdown, a list shows "vpc-07e6f77b7f8d0cb6b - Devops1" and a "AWS Command Line Interface command" option. At the bottom of the dialog, there is a "Cancel" button and a large orange "Attach internet gateway" button, which is also circled in red.



15. Internet gateway → Successfully attached to vpc (message will get)



16. You will LHS vpc dashboard → select Subnets → there will Default subnets are there before you creating.

Select RHS Top corner → Create subnet → click on that link button

The screenshot shows the AWS VPC Subnets page. On the left, there's a sidebar with 'Virtual private cloud' options, including 'Subnets' which is currently selected and highlighted with a red circle. The main area displays a table of subnets with columns for Name, Subnet ID, State, VPC, and IPv4 CIDR. There are 10 subnets listed. In the top right corner of the table, there's a 'Actions' dropdown menu with a 'Create subnet' option, which is also highlighted with a red circle. Below the table, there's a section titled 'Select a subnet'.

17. VPC ID filed search your vpc and select (Devops1)→ Click on Create subnet.

The screenshot shows the 'Create subnet' wizard. At the top, it says 'VPC' and 'VPC ID'. Below that is a section labeled 'Create subnets in this VPC.' with a dropdown menu titled 'Select a VPC'. Inside the dropdown, three VPCs are listed: 'vpc-0b6b61ab36ff1c384 (Devops)', 'vpc-07e6f77b7f8d0cb6b (Devops1)', and 'vpc-0a491cb5a5a3f1384 (default)'. The 'vpc-07e6f77b7f8d0cb6b (Devops1)' entry is highlighted with a red circle and has a yellow arrow pointing to it from the left. At the bottom right of the wizard, there's a 'Create subnet' button, which is also highlighted with a red circle.

18. Hear we create subnet settings.

Subnet name field → PubSubNet1(create public subnet)

Next → IPv4 VPC CIDR block → Enter value 10.180.0.0/16

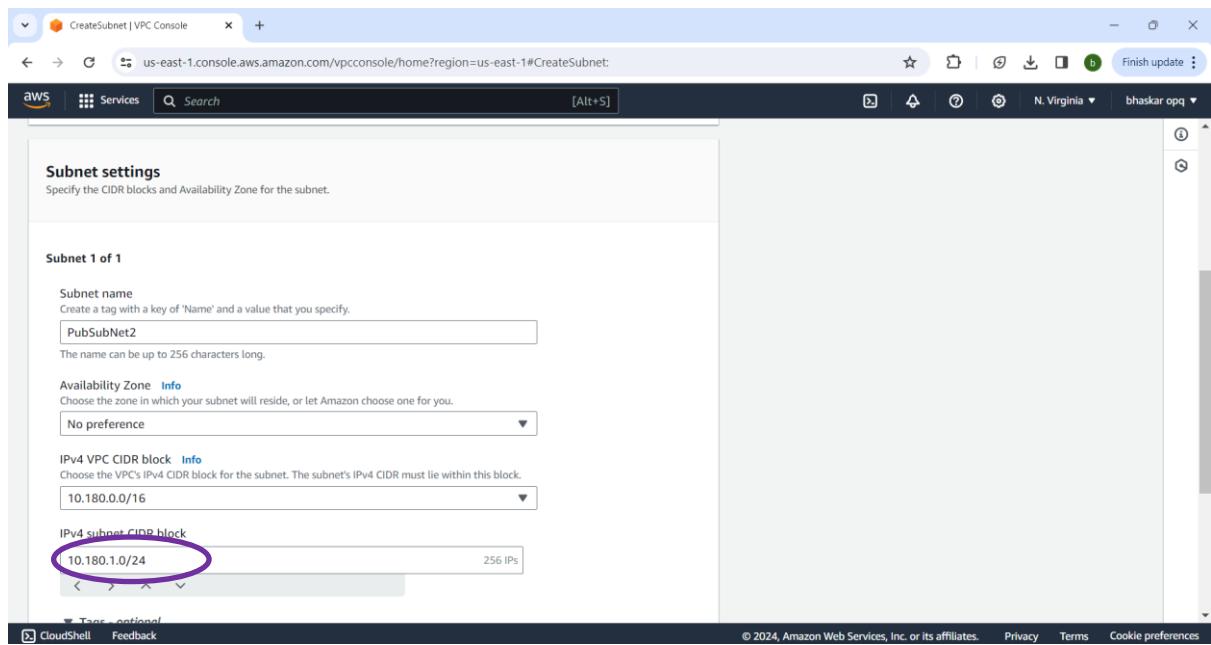
Next → IPv4 subnet CIDR block → 10.180.0.0/24

The screenshot shows the 'CreateSubnet' page in the AWS VPC Console. The 'Subnet settings' section is displayed. The 'Subnet name' field contains 'PubSubNet1'. The 'IPv4 VPC CIDR block' dropdown is set to '10.180.0.0/16'. The 'IPv4 subnet CIDR block' dropdown is set to '10.180.0.0/24'. The 'Tags - optional' section shows a single tag named 'Name' with the value 'PubSubNet1'. The 'CreateSubnet' button is visible at the bottom right.

19. click on Add new subnet.

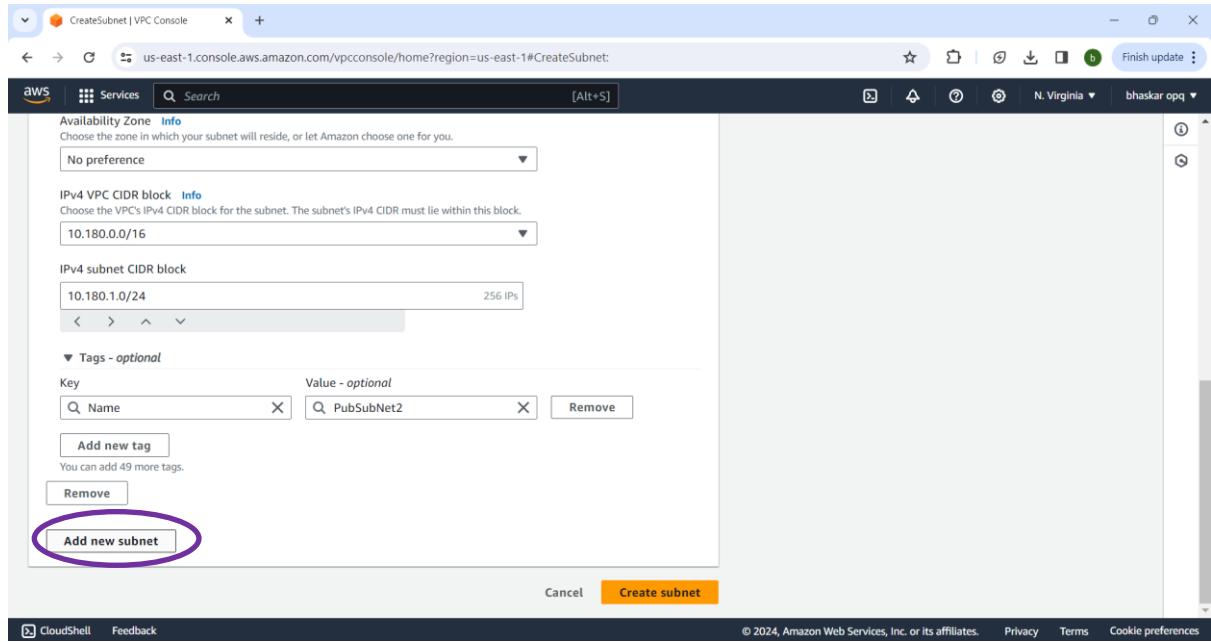
The screenshot shows the 'CreateSubnet' page in the AWS VPC Console. The 'Availability Zone' dropdown is set to 'No preference'. The 'IPv4 VPC CIDR block' dropdown is set to '10.180.0.0/16'. The 'IPv4 subnet CIDR block' dropdown is set to '10.180.0.0/24'. The 'Tags - optional' section shows a single tag named 'Name' with the value 'PubSubNet1'. At the bottom left, there is a button labeled 'Add new subnet' which is circled in purple. The 'Create subnet' button is visible at the bottom right.

20. create another one PubSubNet2(Public SubNet).



The screenshot shows the AWS VPC CreateSubnet interface. Under 'Subnet settings', it specifies the CIDR block and Availability Zone for the subnet. The 'Subnet name' is 'PubSubNet2'. The 'Availability Zone' dropdown shows 'No preference'. The 'IPv4 VPC CIDR block' dropdown is set to '10.180.0.0/16'. The 'IPv4 subnet CIDR block' dropdown is set to '10.180.1.0/24', which is highlighted with a purple circle. Below these fields, there is a section for 'Tags - optional' where a tag 'Name' is associated with the value 'PubSubNet2'. At the bottom right, there are 'Cancel' and 'Create subnet' buttons.

21.



This screenshot shows the same AWS VPC CreateSubnet interface as the previous one, but with additional configuration options visible. The 'Tags - optional' section is expanded, showing a key 'Name' with a value 'PubSubNet2'. Below this, there is a 'Remove' button and an 'Add new tag' button. At the bottom left, there is a prominent 'Add new subnet' button, which is circled in purple. The overall layout is identical to the first screenshot, with the addition of the expanded tags section and the highlighted 'Add new subnet' button.

22. Create PrivSubNet1 and PrivSubNet2 (Private Subnet).

Click on Create Subnet.

The screenshot shows the 'Create Subnet' page in the AWS VPC console. The 'Subnet name' field contains 'PrivSubNet1'. The 'Availability Zone' dropdown is set to 'No preference'. The 'IPv4 CIDR block' dropdown shows '10.180.0.0/16'. The 'IPv4 subnet CIDR block' dropdown shows '10.180.2.0/24'. Under 'Tags - optional', there is a tag 'Name: PrivSubNet1'. The 'Create subnet' button is highlighted with an orange box.

23. You will get ui page all 4 subnets created.

The screenshot shows the 'Subnets' page in the AWS VPC Management Console. A success message at the top says 'You have successfully created 3 subnets: subnet-058d7c99a3dfc2509, subnet-08401741f23606e0c, subnet-0abad09ec094b165e'. The main table lists three subnets:

Name	Subnet ID	State	VPC	IPv4 CIDR
PrivSubNet1	subnet-08401741f23606e0c	Available	vpc-07e6f77b7f8d0cb6b Devo...	10.180.2.0/24
PrivSubNet2	subnet-0abad09ec094b165e	Available	vpc-07e6f77b7f8d0cb6b Devo...	10.180.3.0/24
PubSubNet2	subnet-058d7c99a3dfc2509	Available	vpc-07e6f77b7f8d0cb6b Devo...	10.180.1.0/24

24. LHS select Route tables → click on

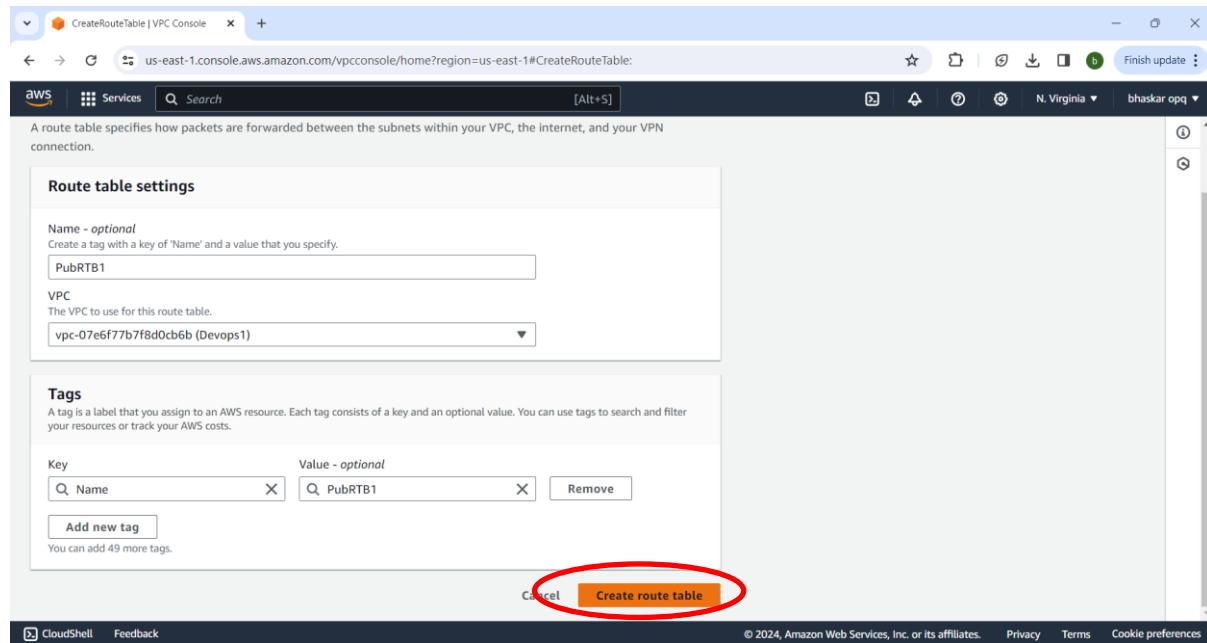
Select RHs (top corner) → click on → Create route table 1. Public RTB 2. Private RTB.

The screenshot shows the AWS VPC Console interface. On the left, there's a navigation pane with sections like 'VPC dashboard', 'EC2 Global View', 'Filter by VPC', 'Virtual private cloud' (with 'Your VPCs' and 'Subnets' listed), and 'Route tables' (which is selected and highlighted with a red box). The main area displays a table titled 'Route tables (5) Info' with columns for 'Name', 'Route table ID', 'Explicit subnet assoc...', 'Edge associations', 'Main', and 'VPC'. A specific row is selected, showing 'rtb-06f83f54150e884c4'. At the top right of the table, there's an 'Actions' menu with a 'Create route table' option, which is also highlighted with a red circle. Below the table, there's a section titled 'Select a route table'.

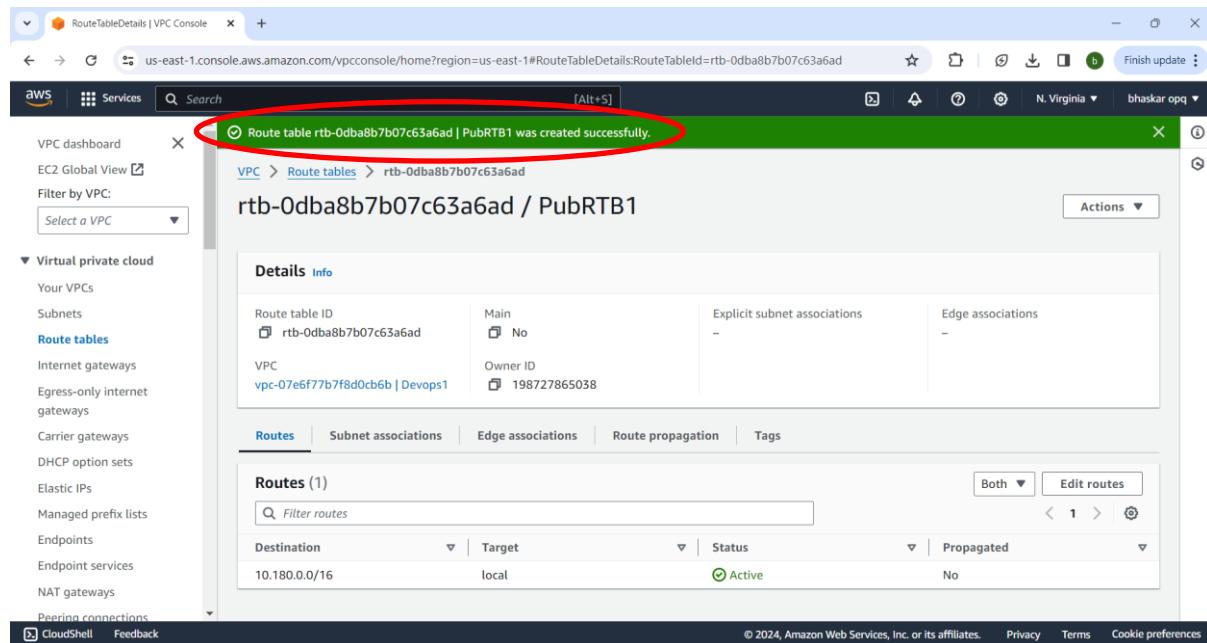
25. Name field give name PubRTB1(Public RTB) → select your VPC(Devops1).

The screenshot shows the 'Create route table' wizard. The first step, 'Route table settings', is displayed. It has fields for 'Name - optional' (containing 'PubRTB1', which is highlighted with a red box) and 'VPC' (containing 'vpc-07e6f77b7f8d0cb6b (Devops1)', which is also highlighted with a red box). Below these, there's a 'Tags' section where a single tag 'Name: PubRTB1' is defined. At the bottom of the page, there are 'CloudShell' and 'Feedback' links, along with standard copyright and legal links at the bottom right.

26. click on → Create route table button.



27. you will get ui page PubRTB1 was created successfully.



28. Create Private RTB → PrivRTB1 → select your VPC → click on Create route table.

The screenshot shows the 'Create route table' wizard in the AWS VPC console. In the 'Route table settings' section, the 'Name' field contains 'PrivRTB1', which is circled in red. Below it, the 'VPC' dropdown is set to 'vpc-07e6f77b7f8d0cb6b (Devops1)'. In the 'Tags' section, there is one tag named 'Name' with the value 'PrivRTB1'. At the bottom right of the page, there is a 'Create route table' button.

29. you will get ui page PrivRTB1 was created successfully.

The screenshot shows the 'Route tables (1/7)' list in the AWS VPC console. The table includes columns for Name, Route table ID, Explicit subnet associations, Edge associations, Main, and VPC. One row is selected, showing 'PubRTB1' with Route table ID 'rtb-0db8b7b07c63a6ad'. Below the table, the details for 'rtb-0db8b7b07c63a6ad / PubRTB1' are shown, including tabs for Details, Routes, Subnet associations, Edge associations, Route propagation, and Tags. The 'Subnet associations' tab is selected, and the 'Edit subnet associations' button is circled in red. The table also shows other route tables: 'rtb-0b5af3687c9ed2ea6' and 'rtb-0915dc32044fef875'.

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC
-	rtb-0b5af3687c9ed2ea6	-	-	Yes	vpc-07e6f77b7f8d0cb6b
<input checked="" type="checkbox"/> PubRTB1	rtb-0db8b7b07c63a6ad	-	-	No	vpc-07e6f77b7f8d0cb6b
-	rtb-0915dc32044fef875	-	-	Yes	vpc-07e6f77b7f8d0cb6b

30. select in Route tables click on check box PubRTB1 then you got the → below → select Subnet associations → click on Edit subnet associations.

The screenshot shows the AWS VPC console with the 'Route tables' section selected. A route table named 'PubRTB1' is highlighted. Below it, the 'rtb-0dba8b7b07c63a6ad' entry is selected. The 'Subnet associations' tab is active, and its 'Edit subnet associations' button is circled in red.

31. Available subnets (before we come this page, we created the Subnets (1. PubSubNet1 , 2. PubSubNet2) select the two check boxes.

Click on save associations.

The screenshot shows the 'Edit subnet associations' dialog box. It lists available subnets: 'PrivSubNet1', 'PrivSubNet2', 'PubSubNet1', and 'PubSubNet2'. The 'PubSubNet1' and 'PubSubNet2' checkboxes are checked and circled in red. The 'Selected subnets' section shows 'PubSubNet1' and 'PubSubNet2' selected. The 'Save associations' button at the bottom right is also circled in red.

32. you will get the → you have successfully updated subnet associations for rtb(Round table)

Reape same process for the PrivRTB1 → Subnet associations → Edit subnet associations → click on that link button.

The screenshot shows the AWS VPC Management Console. On the left, there's a navigation pane with options like 'VPC dashboard', 'EC2 Global View', 'Route tables', and 'Subnets'. The 'Route tables' section is selected. In the main area, a table lists three route tables: 'PrivRTB1' (selected), 'PrivateRTB', and another unnamed entry. A success message at the top of the page reads: 'You have successfully updated subnet associations for rtb-067da2ab7e4550e61 / PubRTB1.' Below the table, a sub-section titled 'rtb-067da2ab7e4550e61 / PrivRTB1' shows the 'Subnet associations' tab selected. It displays a table with the heading 'Explicit subnet associations (0)' and a note: 'No subnet associations. You do not have any subnet associations.'

33. Add select the two we created PrivSubNets (1. PrivSubNet1 , 2. PrivSubNet2) → select the two check boxes only → click on Save associations button.

This screenshot shows the 'Edit subnet associations' dialog for route table 'rtb-067da2ab7e4550e61'. The 'Available subnets (2/4)' section lists four subnets: 'PrivSubNet1' (selected), 'PrivSubNet2' (selected), 'PubSubNet1', and 'PubSubNet2'. Both 'PrivSubNet1' and 'PrivSubNet2' are circled in red. Below this, the 'Selected subnets' section shows 'subnet-08401741f23606e0c / PrivSubNet1' and 'subnet-0abad09ec094b165e / PrivSubNet2'. At the bottom right, a large orange button labeled 'Save associations' is circled in red.

34. In Route tables Page → Select only PubRTB1 → come down → select Routes → click on Edit routes.

The screenshot shows the AWS VPC Route Tables page. On the left sidebar, under 'Route tables', 'PubRTB1' is selected and highlighted with a red circle. In the main content area, the 'Routes' tab is selected for the route table 'rtb-0dba8b7b07c63a6ad / PubRTB1'. The 'Edit routes' button is also circled in red at the top right of the routes table.

35.

This screenshot is identical to the one above, showing the AWS VPC Route Tables page. The 'PubRTB1' route table is selected in the sidebar, and the 'Edit routes' button is circled in red in the main content area.

36. you will in Edit routes page. We Add the route. Click on → Add route

The screenshot shows the 'Edit routes' page for a specific route table. A single route entry is listed:

Destination	Target	Status	Propagated
10.180.0.0/16	local	Active	No

A red circle highlights the 'Add route' button at the bottom left of the table.

At the bottom right of the page are 'Cancel', 'Preview', and 'Save changes' buttons.

37. click on → Add route → Target first column → select Internet Gateway

The screenshot shows the 'Edit routes' page with the target dropdown open. The list includes:

- Carrier Gateway
- Core Network
- Egress Only Internet Gateway
- Gateway Load Balancer Endpoint
- Instance
- Internet Gateway** (highlighted with a red circle)
- local
- NAT Gateway
- Network Interface

The 'Internet Gateway' option is currently selected.

At the bottom right of the page are 'Cancel', 'Preview', and 'Save changes' buttons.

38. Second column select → Internet Gateway(Our created devops1IGW).

The screenshot shows the 'Edit routes' page for a specific route table. A red circle highlights the 'Target' dropdown menu, which is open and displays 'Internet Gateway' as the selected option. Below the dropdown, a search bar contains the text 'igw-'.

Destination	Target	Status	Propagated
10.180.0.0/16	local	Active	No
	Internet Gateway	-	No
	igw-090a49054668f53d9 (devops1IGW)	-	No

Buttons at the bottom include 'Cancel', 'Preview', and 'Save changes'.

39. select Destination field → select 0.0.0.0/0

The screenshot shows the 'Edit routes' page for a specific route table. A red circle highlights the 'Destination' input field, which contains '0.0.0.0/0'. Another red circle highlights the 'Save changes' button at the bottom right.

Destination	Target	Status	Propagated
10.180.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway	-	No

Buttons at the bottom include 'Cancel', 'Preview', and 'Save changes'.

-→**Click on Save changes.**

40.

The screenshot shows the 'Edit routes' page in the AWS VPC Console. A new route is being added:

Destination	Target	Status	Propagated
10.180.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway	-	No

At the bottom right, the 'Save changes' button is highlighted with a red circle.

41. You will get Updated the routes and we will see the in Routes section we get two routes hear.

The screenshot shows the 'RouteTableDetails' page in the AWS VPC Console. A success message is displayed:

Updated routes for rtb-0dba8b7b07c63a6ad / PubRTB1 successfully

The 'Routes' section shows the following routes:

Destination	Target	Status	Propagated
0.0.0.0/0	igw-090a49054668f53d9	Active	No
10.180.0.0/16	local	Active	No

42. Hear we cant given/Edited the routes.

The screenshot shows the AWS VPC console's Route Table Details page. The route table ID is `rtb-067da2ab7e4550e61`, which is circled in red. The table lists one route entry:

Destination	Target	Status	Propagated
10.180.0.0/16	local	Active	No

43. AWS search bar find EC2 --> click on.

The screenshot shows the AWS Services search results page. The search term `ec2` is highlighted in the search bar and circled in red. The results list the `EC2` service, which is also circled in red. Other results include `EC2 Image Builder`, `Recycle Bin`, and `Amazon Inspector`.

44. Go to Ec2 page →select Instances.

Click on RHs(top corner) Launch instances→ click on button.

The screenshot shows the AWS EC2 Instances page. On the left, a sidebar menu is open under the 'Instances' section, with 'Instances' itself circled in red. In the main content area, there is a table titled 'Instances (3) Info' showing three stopped t2.micro instances. At the top right of the table, there is a yellow 'Launch instances' button, which is also circled in red. The URL in the browser bar is <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances>.

45. Click on→ Launch instances.

This screenshot is identical to the one above, showing the AWS EC2 Instances page with three stopped t2.micro instances. The 'Instances' menu item in the sidebar and the yellow 'Launch instances' button at the top right are both highlighted with yellow boxes, indicating they have been selected or are active.

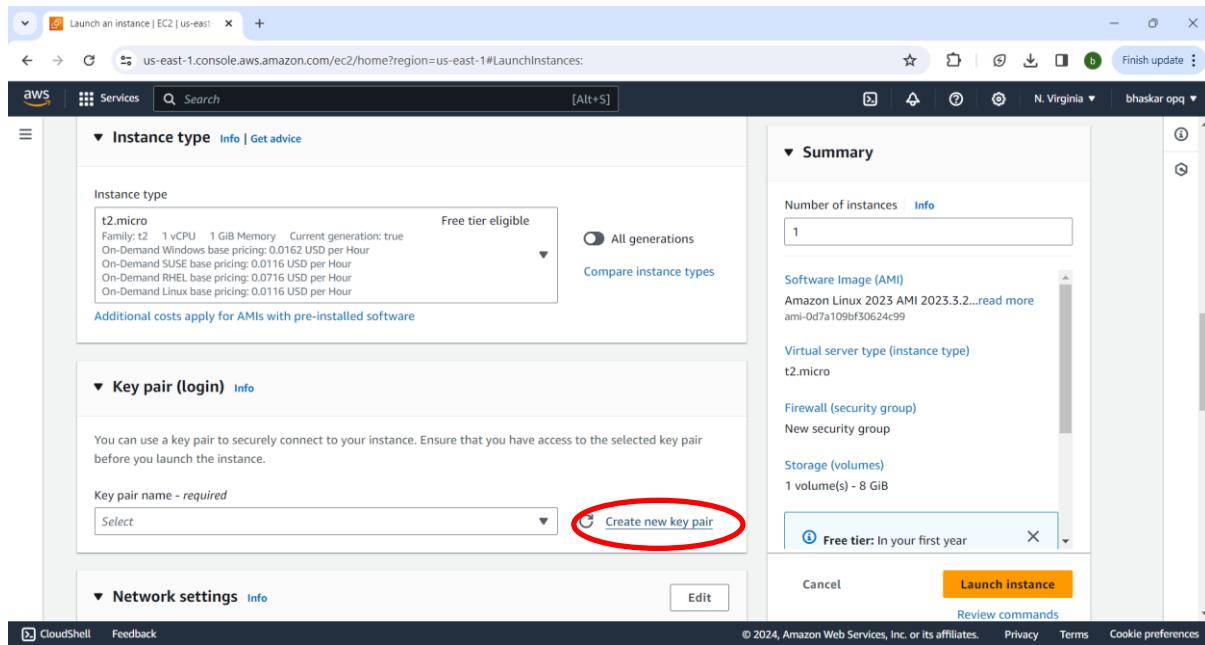
46. Enter Proper Name here → Vpctest2. Remaining will be the Default (Don't change any fields).

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Name and tags' section, the 'Name' input field contains the value 'Vpctest2', which is circled in red. The 'Software Image (AMI)' section shows 'Amazon Linux 2023 AMI 2023.3.2...' as the selected AMI. The 'Virtual server type (instance type)' is set to 't2.micro'. The 'Storage (volumes)' section indicates 1 volume(s) - 8 GiB. On the right, there's a summary panel and a 'Launch instance' button.

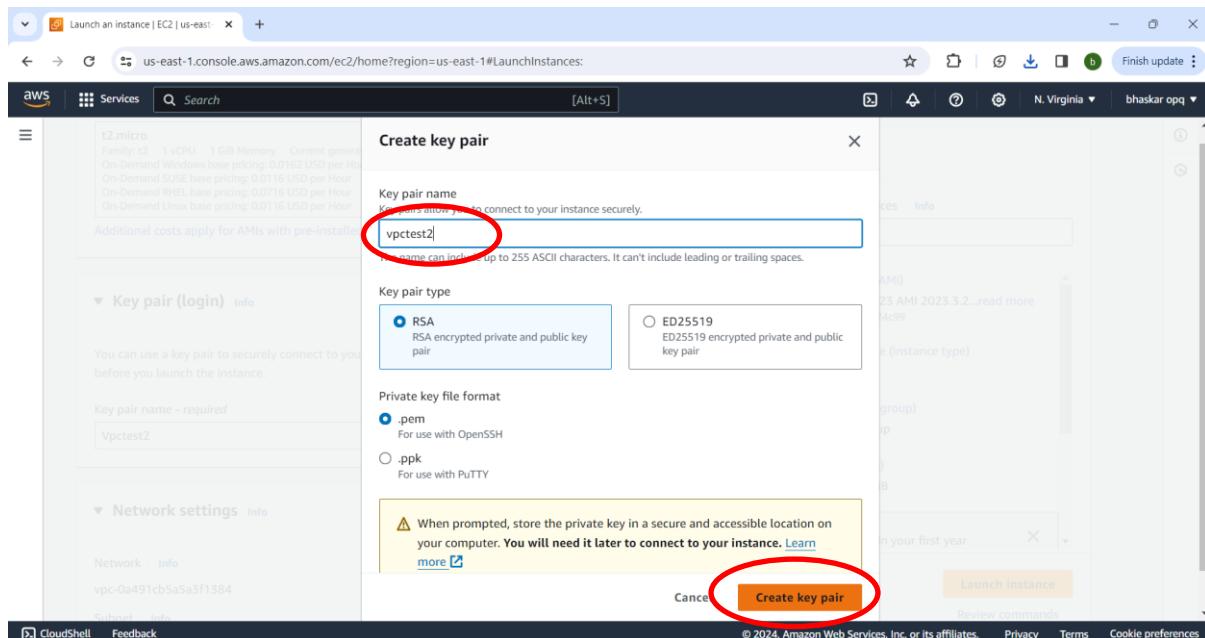
47.

The screenshot shows the AWS EC2 'Quick Start' page. Under the 'Amazon Machine Image (AMI)' section, 'Amazon Linux 2023 AMI' is selected, showing details like AMI ID 'ami-0d7a109bf50624c99', architecture '64-bit (x86)', boot mode 'uefi-preferred', and AMI ID 'ami-0d7a109bf30624c99'. The 'Verified provider' badge is present. The 'Instance type' section is visible at the bottom. The right side of the screen shows the same summary and launch instance steps as in the previous screenshot.

48. Click on Create new keyPair.



49. Give Properame(vpctest2) remaining will be default don't do anything. → click on Create key pair. You will be downloaded the .pem file in (downloads section in your system).



50. Select Network Settings → click on Edit button

The screenshot shows the 'Network settings' section of the AWS EC2 'Launch an instance' wizard. The 'Edit' button is highlighted with a red circle. The 'Network settings' section includes fields for Network (vpc-0a491cb5a5a3f1384), Subnet (No preference), Auto-assign public IP (Enable), Firewall (security groups) (Create security group selected), and a note about additional charges. Below these are options for Allow SSH traffic from Anywhere (selected) and Allow HTTPS traffic from the internet (unchecked). The summary on the right shows 1 instance, Amazon Linux 2023 AMI 2023.3.2..., t2.micro instance type, and 1 volume(s) - 8 GiB storage.

51. Change Default Vpc - required field to Our created VPC (Devops1).

The screenshot shows the 'Network settings' section of the AWS EC2 'Launch an instance' wizard. The 'VPC - required' dropdown is highlighted with a red box, showing options: 'vpc-0a491cb5a5a3f1384 (default)', 'vpc-0b6b61ab36ff1c384 (Devops)', and 'vpc-07e6f7b7f8d0cb6b (Devops1)'. The 'vpc-07e6f7b7f8d0cb6b (Devops1)' option is selected. Other fields include Firewall (security groups) (Create security group selected), Security group name (launch-wizard-4), and a note about security group creation. The summary on the right shows 1 instance, Amazon Linux 2023 AMI 2023.3.2..., t2.micro instance type, and 1 volume(s) - 8 GiB storage.

52. Select Subnet field,, select our created Subnet (PubSubNet1).

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Network settings' section, the 'Subnet' dropdown is open, displaying several subnet options. One option, 'PubSubNet1', is highlighted with a red box. The 'Summary' panel on the right shows basic instance details like 'Number of instances: 1', 'Software Image (AMI): Amazon Linux 2023.3.2...', and 'Virtual server type (instance type): t2.micro'. At the bottom right is a large orange 'Launch instance' button.

53. Auto –assign public IP field → Enable.

This screenshot shows the same EC2 launch wizard. In the 'Network settings' section, the 'Auto-assign public IP' dropdown is highlighted with a red circle, showing 'Enable' as the selected option. The rest of the interface and summary panel are identical to the previous screenshot.

54.

The screenshot shows the 'Launch an instance' wizard in the AWS Management Console. On the left, under 'Network settings', the 'VPC - required' dropdown is set to 'vpc-07e6f77b7f8d0cb6b (Devops1)'. The 'Subnet' dropdown is set to 'subnet-0f5ac61702e4151e6' (PubSubNet1). The 'Auto-assign public IP' dropdown is set to 'Enable'. Below these, there's a note about additional charges for public IP usage outside the free tier. Under 'Firewall (security groups)', there are two options: 'Create security group' (selected) and 'Select existing security group'. A security group name 'launch-wizard-4' is entered. A note states that this group will be added to all network interfaces. On the right, the 'Summary' section shows 'Number of instances' (1), 'Software Image (AMI)' (Amazon Linux 2023.3.2...), 'Virtual server type (instance type)' (t2.micro), 'Firewall (security group)' (New security group), and 'Storage (volumes)' (1 volume(s) - 8 GiB). At the bottom, there are 'Cancel' and 'Launch instance' buttons, with the 'Launch instance' button highlighted by a yellow box.

55. Remaining all as it is leave it don't disturb the values now. Click → Launch instance.

This screenshot shows the same 'Launch an instance' wizard as the previous one, but with a red oval highlighting the 'Launch instance' button at the bottom right of the summary section. The configuration remains the same: 1 instance of Amazon Linux 2023.3.2, t2.micro instance type, and a new security group named 'launch-wizard-4' with an inbound rule for SSH (TCP port 22) from anywhere. A note about allowing all IP addresses through the security group is visible.

56. Successfully we created the instance. After it will take few minutes for creating instance in AWS server.

The screenshot shows the AWS EC2 Launch Instances page. At the top, there is a green success banner stating "Successfully initiated launch of instance (i-0eacba52341e59ab3)". Below the banner, there is a "Launch log" section showing the following steps and their outcomes:

Action	Status
Initializing requests	Succeeded
Creating security groups	Succeeded
Creating security group rules	Succeeded
Launch initiation	Succeeded

Below the log, there is a "Next Steps" section with several options:

- Create billing and free tier usage alerts
- Connect to your instance
- Connect an RDS database
- Create EBS snapshot policy

At the bottom of the page, there is a footer with links to CloudShell, Feedback, and various AWS terms and policies.

57. Pending stage our created Instances.

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed, and the main area displays a table of instances. One instance, "Vpctest2" (ID: i-0ab26509606690b7d), is highlighted with a red circle around its "Pending" status in the "Instance state" column. The table includes columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
feb24	i-09c1af9cddb0e47	Stopped	t2.micro	-	View alarms +	us-east-1b
Vpctest	i-0eacba52341e59ab3	Stopped	t2.micro	-	View alarms +	us-east-1f
FEB_SSH_24	i-0db795488c60f0288	Stopped	t2.micro	-	View alarms +	us-east-1d
<input checked="" type="checkbox"/> Vpctest2	i-0ab26509606690b7d	Pending	t2.micro	-	View alarms +	us-east-1f

Below the table, a detailed view for the instance "Vpctest2" (i-0ab26509606690b7d) is shown. The "Details" tab is selected, displaying information such as Public IPv4 address (3.238.113.252), Instance state (Pending), and Private IP DNS name (10.180.0.207).

58. Our Instance Is Running (Instance state column).

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, and Images. The Instances section is expanded, showing sub-options like Instances, Instance Types, Launch Templates, etc. The main content area has a table titled 'Instances (1/4) Info'. The table has columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. There are five rows: 'feb24' (Stopped), 'Vpctest' (Stopped), 'FEB_SSH_24' (Stopped), 'Vpctest2' (Running), and a new row 'i-0ab26509606690b7d' (Running). A red circle highlights the 'Running' status for the 'Vpctest2' instance. Below the table, a modal window is open for the 'Vpctest2' instance, showing its details like Public IPv4 address (3.238.113.252), Instance state (Running), and Private IP DNS name (10.180.0.207). The modal also has tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags.

59. You will see the Status check(2/2 checks passed).

This screenshot is similar to the previous one but shows four instances instead of one. The 'Status check' column for the 'Vpctest2' instance now displays '2/2 checks passed' with a green checkmark icon, which is highlighted by a red circle. The rest of the interface is identical to the previous screenshot, showing the EC2 dashboard, instance list, and the detailed view for the selected instance.

Now you work on Ec2.... !