

VPC_TASK1

1. Create VPC with 2 private and 2 public subnets.

The screenshot shows the 'Launch an instance' wizard in the AWS EC2 console. The 'Name and tags' section has 'Name' set to 'VPC_first'. The 'Software Image (AMI)' section shows 'Amazon Linux 2023 AMI 2023.9.2...' with the ID 'ami-068c0051b15cdb816'. The 'Virtual server type (instance type)' is set to 't3.micro'. The 'Firewall (security group)' is 'New security group'. The summary on the right shows 1 instance. The bottom right contains 'Cancel', 'Launch instance', and 'Preview code' buttons.

- Open AWS console and create instance by clicking on Launch instance.
- Select AMI

The screenshot shows the 'Network settings' section of the 'Launch an instance' wizard. It includes fields for 'VPC - required' (set to 'vpc-04bf5b2b4aa81ec1e'), 'Subnet' (set to 'No preference'), 'Availability Zone' (set to 'No preference'), 'Auto-assign public IP' (set to 'Enable'), and 'Firewall (security groups)' (set to 'Create security group'). The summary on the right shows 1 instance. The bottom right contains 'Cancel', 'Launch instance', and 'Preview code' buttons.

- Go to Network Settings
- Edit VPC

VPC_TASK1

The screenshot shows the AWS VPC console home page. At the top, there are buttons for 'Create VPC' and 'Launch EC2 Instances'. A note says 'Note: Your Instances will launch in the United States region.' Below this is a section titled 'Resources by Region' with the following items:

- VPCs**: N. Virginia 1 (with a 'See all regions' link)
- Endpoint Services**: N. Virginia 0 (with a 'See all regions' link)
- Subnets**: N. Virginia 1 (with a 'See all regions' link)
- NAT Gateways**: N. Virginia 0 (with a 'See all regions' link)
- Route Tables**: N. Virginia 1 (with a 'See all regions' link)
- VPC Peering Connections**: N. Virginia 0 (with a 'See all regions' link)
- Internet Gateways**: N. Virginia 1 (with a 'See all regions' link)
- Network ACLs**: N. Virginia 1 (with a 'See all regions' link)

On the right side, there are sections for 'Service Health', 'Settings', and 'Additional Information'. The 'Settings' section includes links for 'Block Public Access', 'Zones', 'Console Experiments', 'VPC Documentation', 'All VPC Resources', 'Forums', and 'Report an Issue'. At the bottom, there are links for 'CloudShell', 'Feedback', and 'Console Mobile App', along with copyright information and links for 'Privacy', 'Terms', and 'Cookie preferences'.

- Search for VPC and click on Create on VPC.
- Click on VPCs

The screenshot shows the 'Create VPC' configuration page. The URL is us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateVpc:createMode=vpcOnly. The page has a breadcrumb navigation: VPC > Your VPCs > Create VPC.

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.

VPC only **VPC and more**

Name tag - optional
Creates a tag with a key of 'Name' and a value that you specify.
my_vpc_01

IPv4 CIDR block Info
 IPv4 CIDR manual input IPAM-allocated IPv4 CIDR block

IPv4 CIDR
172.168.0.0/16
CIDR block size must be between /16 and /28.

At the bottom, there are links for 'CloudShell', 'Feedback', and 'Console Mobile App', along with copyright information and links for 'Privacy', 'Terms', and 'Cookie preferences'.

- VPC → your VPCs → Create VPC
- VPC settings click on VPC only and give Name tag
- IPv4 CIDR block, select IPv4 CIDR manual input
- IPv4 CIDR

VPC_TASK1

The screenshot shows the 'Create VPC' wizard in the AWS VPC console. The first step, 'Select CIDR block', is completed with 'No IPv6 CIDR block' selected. The second step, 'Tenancy', is set to 'Default'. The third step, 'VPC encryption control', shows 'None' selected. The fourth step, 'Tags', shows a tag 'VPC_01' with value 'my_vpc_01' added. The fifth step, 'Review', is not yet reached.

- Select No IPV6 CIDR block
- Click on Create VPC.

The screenshot shows the 'Details' page for a newly created VPC with ID 'vpc-0bc432ffa77faf53e'. The VPC is in 'Available' state and has 'Enabled' DNS resolution. It uses 'Main network ACL' 'acl-09455e140f5f3d045' and 'IPv6 CIDR (Network border group)' '—'. The 'Encryption control ID' is also listed as '—'. The 'Actions' button is visible at the top right.

- VPC created successfully.

VPC_TASK1

The screenshot shows the AWS VPC Subnets page. At the top, there is a search bar and navigation links for 'VPC' and 'Subnets'. Below this, a table displays a single subnet entry:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
-	subnet-07d15c2b263f996aa	Available	vpc-04fb52b4aa81ec1e	Off	172.31.32.0/20

Below the table, a section titled 'Select a subnet' is visible. At the bottom of the page, there is a footer with links for 'CloudShell', 'Feedback', 'Console Mobile App', and copyright information.

- TO create our own subnet click on create subnet.

The screenshot shows the 'Create subnet' page. The top navigation bar includes 'VPC' and 'Subnets'. The main form has two sections: 'VPC' and 'Subnet settings'.

VPC
VPC ID: vpc-0bc432ffa77faf53e

Associated VPC CIDs
IPv4 CIDRs: 172.168.0.0/16

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

At the bottom, there is a footer with links for 'CloudShell', 'Feedback', 'Console Mobile App', and copyright information.

- Create subnet for required VPC.

VPC_TASK1

The screenshot shows the AWS VPC console interface for creating a new subnet. The top navigation bar includes the AWS logo, a search bar, and account information (Account ID: 8145-8843-2081, Region: United States (N. Virginia), User: Ramesh). The main page title is "Subnet settings". A sub-header "Specify the CIDR blocks and Availability Zone for the subnet." is present. The "Subnet 1 of 2" section is active. Under "Subnet name", the value "public_subnet_1" is entered. The "Availability Zone" dropdown is set to "United States (N. Virginia) / use1-az1 (us-east-1a)". The "IPv4 VPC CIDR block" dropdown is set to "172.168.0.0/16". The "IPv4 subnet CIDR block" input field contains "172.168.1.0/24" with a note "256 IPs". The bottom navigation bar includes CloudShell, Feedback, and Console Mobile App, along with copyright information (© 2026, Amazon Web Services, Inc. or its affiliates.) and links for Privacy, Terms, and Cookie preferences.

- In subnet settings
- Subnet name, Availability Zone
- IPv4 CIDR block - VPC
- IPv4 subnet CIDR block
- Above image shows subnet-1

The screenshot shows the AWS VPC console interface for creating a new subnet, specifically for Subnet 2 of 2. The top navigation bar and account information are identical to the previous screenshot. The main page title is "Subnet settings". A sub-header "Specify the CIDR blocks and Availability Zone for the subnet." is present. The "Subnet 2 of 2" section is active. Under "Subnet name", the value "Public_subnet_2" is entered. The "Availability Zone" dropdown is set to "United States (N. Virginia) / use1-az1 (us-east-1a)". The "IPv4 VPC CIDR block" dropdown is set to "172.168.0.0/16". The "IPv4 subnet CIDR block" input field contains "172.168.2.0/24" with a note "256 IPs". Below this, there is a "Tags - optional" section with "Key" and "Value" fields. The bottom navigation bar includes CloudShell, Feedback, and Console Mobile App, along with copyright information (© 2026, Amazon Web Services, Inc. or its affiliates.) and links for Privacy, Terms, and Cookie preferences.

- We have to create 2 subnet of private
- Above image shows of subnet-2

VPC_TASK1

The screenshot shows the AWS VPC console interface for creating a subnet. The top navigation bar includes the AWS logo, a search bar, and account information (Account ID: 8145-8843-2081, Ramesh). The main heading is "Subnet 3 of 4". The form fields are as follows:

- Subnet name:** my-subnet-01
- Availability Zone:** United States (N. Virginia) / use1-az1 (us-east-1a)
- IPv4 VPC CIDR block:** 172.168.0.0/16
- IPv4 subnet CIDR block:** 172.168.3.0/25 (128 IPs)

At the bottom, there are links for CloudShell, Feedback, Console Mobile App, and standard footer links.

- We have to create 2 private subnet within VPC range
- Above image is private subnet-1

The screenshot shows the AWS VPC console interface for creating a subnet. The top navigation bar includes the AWS logo, a search bar, and account information (Account ID: 8145-8843-2081, Ramesh). The main heading is "Subnet 4 of 4". The form fields are as follows:

- Subnet name:** my-subnet-01
- Availability Zone:** United States (N. Virginia) / use1-az1 (us-east-1a)
- IPv4 VPC CIDR block:** 172.168.0.0/16
- IPv4 subnet CIDR block:** 172.168.4.0/24 (256 IPs)

Below the form, there is a section for "Tags - optional" with a key and value field. At the bottom, there are links for CloudShell, Feedback, Console Mobile App, and standard footer links.

- It is private subnet-2
- By Clicking create subnet we can save it.

VPC_TASK1

The screenshot shows the AWS VPC Subnets page. It lists five subnets under the 'Subnets (5)' section. The subnets are:

Subnet Name	Subnet ID	State	VPC
public_subnet	subnet-0c73e8bb53c37239bf	Available	vpc-0bc432ffa77faf53e
Public_subnet_2	-	Available	vpc-0bc432ffa77faf53e
public_subnet_1	-	Available	vpc-04fb52b4aa81ec1e
-	-	Available	vpc-0bc432ffa77faf53e
private_subnet1	subnet-0a5cbd124a108a546	Available	vpc-0bc432ffa77faf53e
private_subnet2	subnet-04200f7f69924724a	Available	vpc-0bc432ffa77faf53e

- Created 2 public subnets and 2 private subnets successfully.

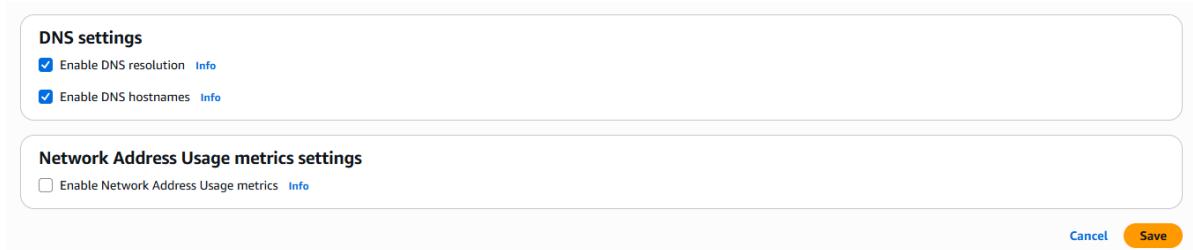
2. Enable DNS Hostname in VPC.

- Search for VPC and click on it
- Select your VPCs and
- Edit VPC settings
- DNS settings and click on DNS Hostname and enable it.
- Click on save.

The screenshot shows the 'Edit VPC settings' page for a VPC with ID vpc-0bc432ffa77faf53e. The page is divided into several sections:

- VPC details:** Shows the VPC ID (vpc-0bc432ffa77faf53e) and Name (empty).
- DHCP settings:** Shows a dropdown menu for the DHCP option set, currently set to dopt-068fb342b37621beb.
- DNS settings:** Contains two checked checkboxes:
 - Enable DNS resolution
 - Enable DNS hostnames

VPC_TASK1



- Click on save

You have successfully modified the settings for vpc-0bc432ffa77faf53e.

Details	
VPC ID	vpc-0bc432ffa77faf53e
DNS resolution	Enabled
Main network ACL	acl-09455e140f5f3d045
IPv6 CIDR (Network border group)	-
Encryption control ID	-
State	Available
Tenancy	default
Default VPC	No
Network Address Usage metrics	Disabled
Encryption control mode	-
Block Public Access	Off
DHCP option set	dopt-068fb342b37621beb
IPv4 CIDR	172.168.0.0/16
Route 53 Resolver DNS Firewall rule groups	-
DNS hostnames	Enabled
Main route table	rtb-0696df46b8705c337
IPv6 pool	-
Owner ID	814588432081

- We can successfully modified setting for VPC message.

3. Enable Auto Assign Public IP in 2 public subnets.

Last updated 18 minutes ago

Actions ▾

Create subnet

Edit subnet settings

View details

Create flow log

Edit IPv6 CIDs

Edit network ACL association

Edit route table association

Edit CIDR reservations

Share subnet

Manage tags

Delete subnet

- Go to VPC and click on subnets
- Click on public subnet we create already.

VPC_TASK1

- Click on Actions and Edit subnet settings.

The screenshot shows the AWS VPC console with the URL us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1>EditSubnetSettings:SubnetId=subnet-0c73e8b53c37239bf. The page title is "Edit subnet settings". The subnet ID is `subnet-0c73e8b53c37239bf` and the name is `Public_subnet_2`. Under "Auto-assign IP settings", the "Enable auto-assign public IPv4 address" checkbox is checked. Under "Resource-based name (RBN) settings", the "Enable resource name DNS A record on launch" checkbox is unchecked. At the bottom, there are links for CloudShell, Feedback, and Console Mobile App, along with copyright information and privacy terms.

- Search for auto-assign IP settings and Enable auto-assign Public IP4 address and save it.

The screenshot shows the AWS VPC dashboard with the URL us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#subnets. A green success message box says "You have successfully changed subnet settings: Enable auto-assign public IPv4 address". Below it, the "Subnets (1/5)" table lists one subnet: `Public_subnet_2` (Subnet ID: `subnet-0c73e8b53c37239bf`, State: Available, VPC: vpc-0). The table has columns for Name, Subnet ID, State, and VPC. At the bottom, the subnet ID is shown as `subnet-0c73e8b53c37239bf / Public_subnet_2`.

- We see message that successfully changed subnet settings.

VPC_TASK1

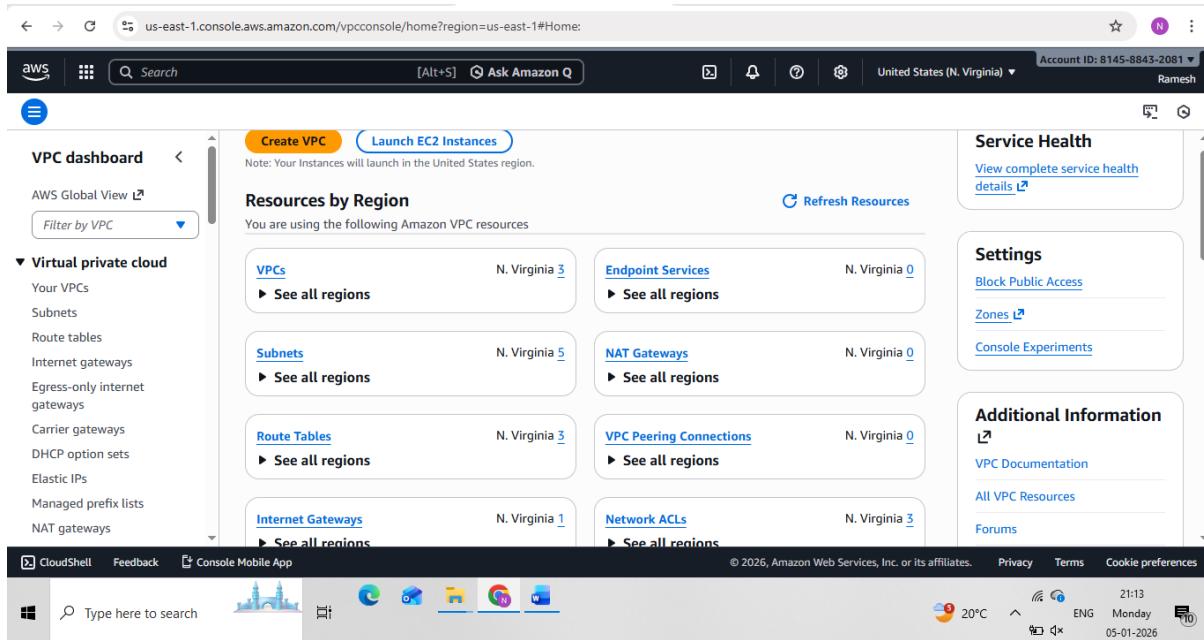
The screenshot shows the AWS VPC Subnets console. On the left, there's a navigation sidebar with 'VPC dashboard' and 'Virtual private cloud' sections. The main area displays a table of subnets with columns for 'Subnet ID', 'State', and 'VPC'. A modal window titled 'Subnets (1/5) Info' is open over the table. In the top right corner of the main area, a context menu is open under the 'Actions' button. The menu items include 'View details', 'Create flow log', 'Edit subnet settings' (which is highlighted with a blue border), 'Edit IPv6 CIDRs', 'Edit network ACL association', 'Edit route table association', 'Edit CIDR reservations', 'Share subnet', 'Manage tags', and 'Delete subnet'. The URL in the browser bar is 'us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#subnets'.

This screenshot is from the same AWS VPC Subnets console as the previous one. It shows the same interface with the 'Edit subnet settings' option selected in the Actions menu. However, a green success message at the top of the page states: 'You have successfully changed subnet settings: Enable auto-assign public IPv4 address'. The rest of the interface, including the table of subnets and the Actions menu, appears identical to the first screenshot. The URL in the browser bar is 'us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#subnets'.

- We follow same steps for public subnet 2 also and save it.

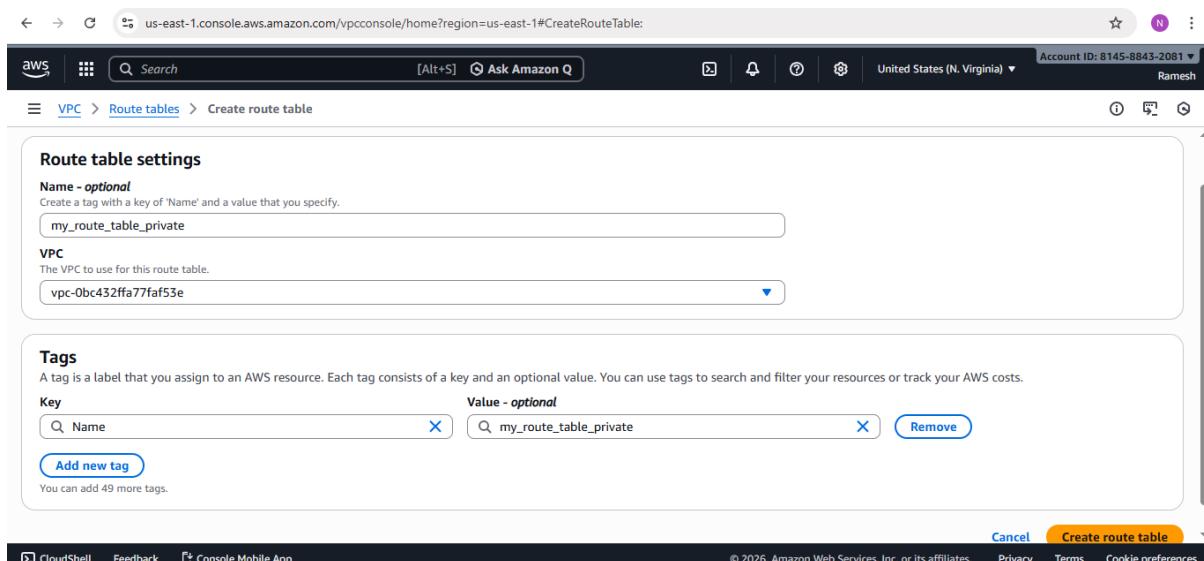
VPC_TASK1

4. Add 2 private subnets in private route table.



The screenshot shows the AWS VPC Dashboard. On the left, there's a sidebar with options like 'Your VPCs', 'Subnets', 'Route tables', 'Internet gateways', 'Egress-only internet gateways', 'Carrier gateways', 'DHCP option sets', 'Elastic IPs', 'Managed prefix lists', and 'NAT gateways'. The main area displays 'Resources by Region' for N. Virginia, including 'VPCs' (3), 'Endpoint Services' (0), 'Subnets' (5), 'NAT Gateways' (0), 'Route Tables' (3), 'VPC Peering Connections' (0), and 'Internet Gateways' (1). A note at the top says 'Note: Your Instances will launch in the United States region.' On the right, there are sections for 'Service Health', 'Settings' (with 'Block Public Access' and 'Zones'), and 'Additional Information' (with links to 'VPC Documentation', 'All VPC Resources', and 'Forums'). The bottom of the screen shows the AWS navigation bar and some system status icons.

- Click on VPC in that we have Route tables



The screenshot shows the 'Create route table' wizard. Step 1: Route table settings. It asks for a 'Name - optional' (input: 'my_route_table_private') and a 'VPC' (input: 'vpc-0bc432fa77faf53e'). Below these, there's a 'Tags' section where a single tag 'my_route_table_private' is added with key 'Name' and value 'my_route_table_private'. At the bottom, there are 'Cancel' and 'Create route table' buttons.

- Go to Route table settings
- Select VPC what you need.
- Click on create route table.

VPC_TASK1

The screenshot shows the AWS VPC console route table creation interface. A success message at the top states "Route table rtb-043c4f1f0db026eee | my_route_table_private was created successfully." The main section displays the route table details, including its ID (rtb-043c4f1f0db026eee), name (my_route_table_private), and owner (814588432081). The "Routes" tab is selected, showing one route entry: Destination 172.168.0.0/16, Target local, Status Active, Propagated No, and Route Origin Create Route Table. On the right, an "Actions" menu is open with options like Set main route table, Edit subnet associations, and Delete.

- After Creating Route table then click on edit subnet associations.

The screenshot shows the "Edit subnet associations" page for the previously created route table. The title is "rtb-043c4f1f0db026eee > Edit subnet associations". It lists available subnets (Public_subnet_2, public_subnet_1, subnet-0a5cbd124a108a546, subnet-04200f7f69924724a) and selected subnets (subnet-0a5cbd124a108a546 and subnet-04200f7f69924724a). The "Save associations" button is visible at the bottom right.

- In edit subnet associations we find all subnets in that select what ever we required subnets(private) and save associations.

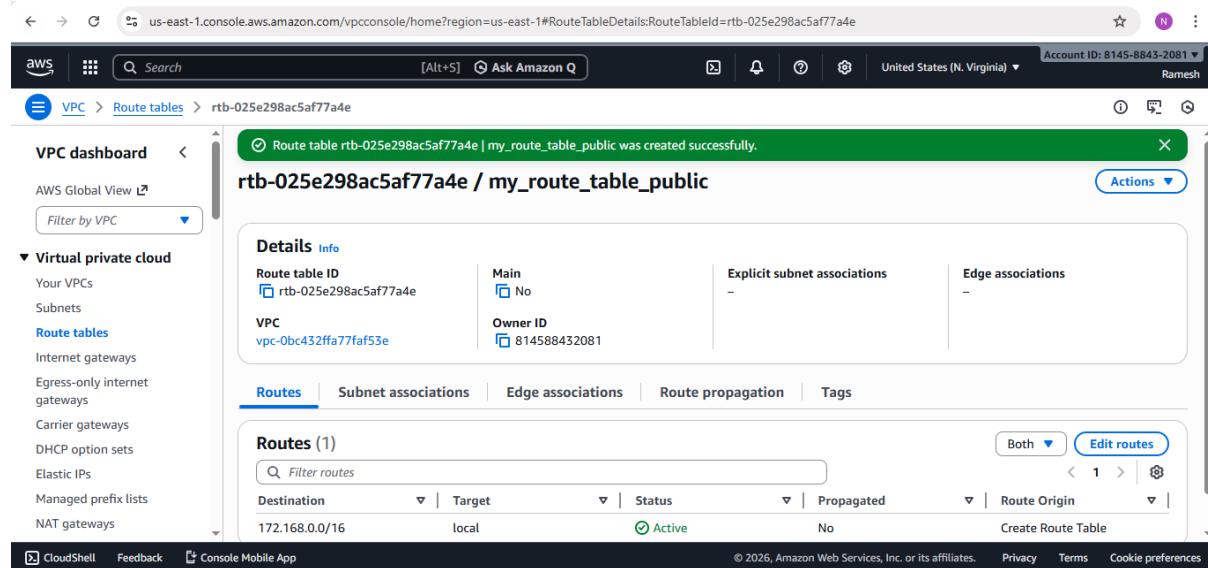
VPC_TASK1

The screenshot shows the AWS VPC console interface. The URL is us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#RouteTableDetails:RouteTableId=rtb-043c4f1f0db026eee. The top navigation bar includes the AWS logo, search bar, and account information (Account ID: 8145-8843-2081, Ramesh). The left sidebar shows the VPC dashboard and lists various VPC components under 'Virtual private cloud'. The main content area displays the details for a specific route table named 'rtb-043c4f1f0db026eee / my_route_table_private'. A green success message at the top states: 'You have successfully updated subnet associations for rtb-043c4f1f0db026eee / my_route_table_private.' Below this, the 'Details' section shows the route table ID, owner ID, and association status. The 'Routes' tab is selected, showing one route entry: Destination 172.168.0.0/16, Target local, Status Active, Propagated No, and Route Origin Create Route Table.

- In Route tables Details we can see explicit subnet associations to verify.

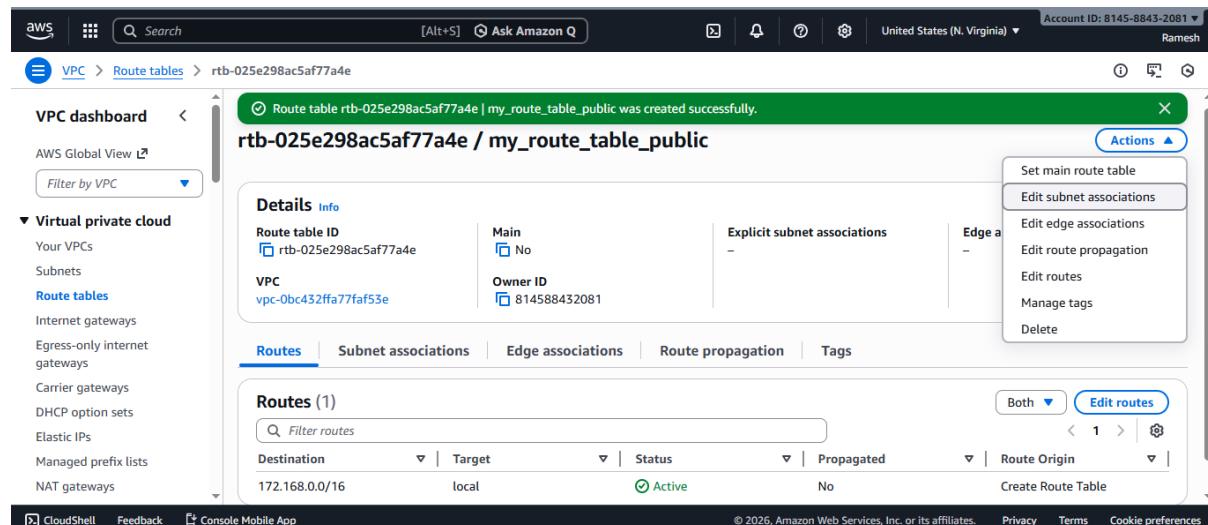
VPC_TASK1

5. Add 2 public subnets in public route table.



The screenshot shows the AWS VPC Route Table Details page. A green success message at the top states: "Route table rtb-025e298ac5af77a4e | my_route_table_public was created successfully." Below this, the route table information is displayed: Route table ID (rtb-025e298ac5af77a4e), Main status (No), Owner ID (814588432081). The "Routes" tab is selected, showing one route entry: Destination (172.168.0.0/16), Target (local), Status (Active), Propagated (No). The "Actions" dropdown menu is visible on the right.

- Go to Route table settings
- Select VPC what you need.
- Click on create route table.



The screenshot shows the same AWS VPC Route Table Details page as before. The "Actions" dropdown menu is open, and the "Edit subnet associations" option is highlighted. The rest of the page content is identical to the first screenshot, showing the route table details and the single route entry.

- Click on Edit subnet associations

VPC_TASK1

Available subnets (2/4)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
Public_subnet_2	subnet-0c73e8b53c37239bf	172.168.2.0/24	-	Main (rtb-0696df46b8705c337)
public_subnet_1	subnet-0ca3fe419325d50fb	172.168.1.0/24	-	Main (rtb-0696df46b8705c337)
	subnet-0a5cbd124a108a546	172.168.4.0/24	-	rtb-043c4f1f0db026eee / my_route_table_public
	subnet-04200f7f69924724a	172.168.3.0/25	-	rtb-043c4f1f0db026eee / my_route_table_public

Selected subnets

subnet-0c73e8b53c37239bf / Public_subnet_2 subnet-0ca3fe419325d50fb / public_subnet_1

Cancel Save associations

- In edit subnet associations select subnet what we required(public) and save associations.

You have successfully updated subnet associations for rtb-025e298ac5af77a4e / my_route_table_public.

rtb-025e298ac5af77a4e / my_route_table_public

Details

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-025e298ac5af77a4e	No	2 subnets	-
VPC	Owner ID		
vpc-0bc432ffa77faf53e	814588432081		

Routes (1)

Destination	Target	Status	Propagated	Route Origin
172.168.0.0/16	local	Active	No	Create Route Table

- We can see successfully updated subnet associations for route table.

VPC_TASK1

The screenshot shows the AWS VPC Route Tables page. The left sidebar has a 'Route tables' section under 'Virtual private cloud'. The main area displays a table titled 'Route tables (5) Info' with columns: Name, Route table ID, Explicit subnet assoc..., Edge associations, Main, and VPC. The table lists five entries:

Name	Route table ID	Explicit subnet assoc...	Edge associations	Main	VPC
-	rtb-0696df46b8705c337	-	-	Yes	vpc
-	rtb-0ece973234e8882d	-	-	Yes	vpc
my_route_table_private	rtb-043c4f1f0db026eee	2 subnets	-	No	vpc
my_route_table_public	rtb-025e298ac5af77a4e	2 subnets	-	No	vpc

A search bar at the top says 'Find route tables by attribute or tag'. Below the table is a section titled 'Select a route table'.

- In Route tables we can see route list what have we created.

VPC_TASK1

6. Public route table will have the routes to internet and local.

- Yes public route table will have the routes to internet and local.

The screenshot shows the AWS VPC Route Tables page. On the left, there's a navigation sidebar for 'Virtual private cloud' with options like 'Route tables'. The main area displays a table titled 'Route tables (1/5)'. It lists three entries:

Name	Route table ID	Explicit subnet associations	Main	VPNs
-	rtb-0cef973234e8882d	-	Yes	vpc
my_route_table_private	rtb-043c4f1f0db026eee	2 subnets	No	vpc
my_route_table_public	rtb-025e298ac5af77a4e	2 subnets	No	vpc

Below the table, a specific route table is selected: 'rtb-025e298ac5af77a4e / my_route_table_public'. The 'Details' tab is active, showing information like the route table ID, Main status (unchecked), and explicit subnet associations (2 subnets). The 'Routes' tab is also visible.

- Go to route tables and click on route table which is created for public subnets.

The screenshot shows the 'rtb-025e298ac5af77a4e / my_route_table_public' route table details. The 'Details' tab is active, displaying information such as the route table ID, Main status (unchecked), owner ID, and explicit subnet associations (2 subnets). The 'Routes' tab is selected, showing one route entry:

Destination	Target	Status	Propagated	Route Origin
172.168.0.0/16	local	Active	No	Create Route Table

- Click on edit routes

VPC_TASK1

The screenshot shows the 'Edit routes' page for a route table. There is one existing route: a local route (Status: Active) pointing to the destination 172.168.0.0/16. A new route is being added: an Internet Gateway route (Status: -) pointing to the destination 0.0.0.0/0. The 'Add route' button is visible at the bottom left, and the 'Save changes' button is at the bottom right.

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- In that we can see local Status is active and we have add route for Internet, which nothing but internet gateway.

The screenshot shows the 'Edit routes' page for a route table. A new route is being added: an Internet Gateway route (Status: -) pointing to the destination 0.0.0.0/0. Both the target dropdown and the search input field for 'igw-' contain the value 'Internet Gateway'. An error message 'A valid resource id has to be specified.' is displayed next to both fields. The 'Add route' button is visible at the bottom left, and the 'Save changes' button is at the bottom right.

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- Before adding Internet gate way we have create one internet gateway for the particular VPC what we required.

VPC_TASK1

The screenshot shows the AWS VPC Internet gateways page. A green success message at the top states "Internet gateway successfully deleted - igw-04b24eb91d022eef1". Below it, a table lists one internet gateway:

Name	Internet gateway ID	State	VPC ID
mylgm	igw-0acbc724e84e81de	Attached	vpc-04bf5b52b4aa81ec1e

Below the table, a message says "Select an internet gateway above". At the bottom of the page, there are links for CloudShell, Feedback, and Console Mobile App.

- Steps for creating Internet gateway search for internet gateways
- Click on internet gateway.

The screenshot shows the "Create internet gateway" settings page. It has two main sections: "Internet gateway settings" and "Tags - optional".

Internet gateway settings
Name tag: my_Internet_gateway_1

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.
Key: Name, Value: my_Internet_gateway_1
Add new tag

At the bottom right, there are "Cancel" and "Create internet gateway" buttons.

- In Internet gateway settings
- Click on Create internet gateway.

VPC_TASK1

The screenshot shows the AWS VPC Internet Gateways console. A green notification bar at the top states: "The following internet gateway was created: igw-0da6253974c65398c - my_Internet_gateway_1. You can now attach to a VPC to enable the VPC to communicate with the internet." Below this, the internet gateway details are shown: ID igw-0da6253974c65398c, State Detached, VPC ID -, Owner 814588432081, and a single tag Name: my_Internet_gateway_1.

- After creating internet gateway we must attach to a VPC.

The screenshot shows the "Attach to VPC" dialog box. It displays a list of available VPCs: "vpc-0bc432ffa77faf53e", "vpc-0bc432ffa77faf53e", and "vpc-00914b2d91d53fed7". The user has selected "vpc-0bc432ffa77faf53e". At the bottom right are "Cancel" and "Attach internet gateway" buttons.

- Click on actions attach internet gateway for VPC

The screenshot shows the AWS VPC Internet Gateways console again. A green notification bar at the top states: "Internet gateway igw-0da6253974c65398c successfully attached to vpc-0bc432ffa77faf53e". Below this, the internet gateway details are shown: ID igw-0da6253974c65398c, State Attached, VPC ID vpc-0bc432ffa77faf53e, and a single tag Name: my_Internet_gateway_1.

- We can see internet gateway details along with VPC-ID.

VPC_TASK1

The screenshot shows the 'Edit routes' page in the AWS VPC console. A route is being added to a route table. The 'Destination' field contains '172.168.0.0/16'. The 'Target' dropdown shows 'local' (selected) and 'Internet Gateway' (available). The 'Status' is 'Active'. The 'Propagated' field is 'No'. The 'Route Origin' is 'CreateRouteTable'. The 'Route table' dropdown shows 'rtb-025e298ac5af77a4e' (selected) and 'igw-0da6253974c65398c' (available). The 'Add route' button is visible at the bottom left. At the bottom right are 'Cancel', 'Preview', and 'Save changes' buttons.

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- After creating Internet gateway add to route tables in edit routes and save changes.

The screenshot shows the 'Route tables' page in the AWS VPC console. A modal window displays a success message: 'Updated routes for rtb-025e298ac5af77a4e / my_route_table_public successfully'. Below the modal, the 'Details' section shows the 'Route table ID' as 'rtb-025e298ac5af77a4e', 'Main' status as 'No', 'Explicit subnet associations' as '2 subnets', and 'Edge associations' as '—'. The 'Routes' tab is selected, showing two routes: one to '0.0.0.0/0' via 'igw-0da6253974c65398c' (status Active, propagated No, route origin Create Route) and another to '172.168.0.0/16' via 'local' (status Active, propagated No, route origin Create Route Table). Other tabs include 'Subnet associations', 'Edge associations', 'Route propagation', and 'Tags'. At the bottom are 'CloudShell', 'Feedback', and 'Console Mobile App' links, along with copyright and legal information.

VPC_TASK1

7. Create EC2 in public subnet with t2.micro and install PHP.

The screenshot shows the AWS EC2 Instances launch wizard. The top navigation bar includes the AWS logo, search bar, and account information (Account ID: 8145-8843-2081, Ramesh). The main page title is "Launch an instance". A blue banner at the top says "It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices". Below it are two buttons: "Take a walkthrough" and "Do not show me this message again." The main form is titled "Name and tags" and contains a "Name" field with "Php_server" and a "Add additional tags" button. The "Application and OS Images (Amazon Machine Image)" section shows a search bar and a list of recent AMIs: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and a "Del" button. To the right, the "Summary" section shows "Number of instances" set to 1, "Software Image (AMI)" as Amazon Linux 2023 AMI 2023.9.2..., "Virtual server type (instance type)" as t3.micro, and "Firewall (security group)" as New security group. At the bottom are "CloudShell", "Feedback", "Console Mobile App", and copyright information.

- Click on launch instance for creating instance

The screenshot shows the AWS EC2 Instances launch wizard. The top navigation bar includes the AWS logo, search bar, and account information (Account ID: 8145-8843-2081, Ramesh). The main page title is "Launch an instance". The "Application and OS Images (Amazon Machine Image)" section shows a search bar and a list of recent AMIs: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and a "Del" button. Below this is a "Quick Start" tab and a "Recent" tab. The "Recent" tab is active, showing icons for Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and a "Del" button. To the right, the "Summary" section shows "Number of instances" set to 1, "Software Image (AMI)" as Amazon Linux 2023 AMI 2023.9.2..., "Virtual server type (instance type)" as t3.micro, and "Firewall (security group)" as New security group. At the bottom are "CloudShell", "Feedback", "Console Mobile App", and copyright information. A large orange "Launch instance" button is prominently displayed.

- Select AMI (Amazon Machine Image)

VPC_TASK1

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Instance type' section, 't2.micro' is selected. It lists various On-Demand Windows and Linux base pricing options. A note states 'Additional costs apply for AMIs with pre-installed software'. In the 'Summary' panel on the right, it shows 1 instance, the AMI as 'Amazon Linux 2023 AMI 2023.9.2...', the instance type as 't2.micro', and a new security group. Buttons for 'Cancel', 'Launch instance', and 'Preview code' are visible.

- Select instance type as t2.micro

The screenshot shows the 'Network settings' step. It shows a key pair named 'Neelima-Jenkins'. Under 'Network', 'vpc-04bfb52b4aa81ec1e' is selected. Under 'Subnet', 'No preference (Default subnet in any availability zone)' is chosen. Under 'Auto-assign public IP', 'Enable' is selected. In the 'Firewall (security group)' section, there's a note about creating a new security group. Buttons for 'Edit', 'Create security group', and 'Select existing security group' are shown. The 'Summary' panel on the right remains the same as the previous step.

- Click on Network setting

VPC_TASK1

The screenshot shows the 'Network settings' step of the 'Launch an instance' wizard. It includes fields for VPC (selected: 'vpc-0bc432ffa77faf53e'), Subnet (selected: 'subnet-0c73e8b53c37239bf'), Auto-assign public IP (set to 'Enable'), and a security group ('Create security group' selected). The right panel displays the 'VPC' section with a brief description and a link to 'Get more networking guidance'.

- Add VPC as required, add subnet and Enable Auto assign public ip.

The screenshot shows the 'Inbound Security Group Rules' step of the 'Launch an instance' wizard. A new rule is being created for port 22 (SSH) from anywhere. A warning message at the bottom left advises against using 0.0.0.0/0 as a source. The right panel displays the 'VPC' section with a brief description and a link to 'Get more networking guidance'.

- Edit Inboud rules for connection.

VPC_TASK1

The screenshot shows the AWS EC2 Instances Launch an instance page. The user has selected '1' instance, chosen 't2.micro' as the virtual server type, and created a new security group. The storage is set to 1 volume(s) - 8 GiB. The 'Launch instance' button is highlighted in orange at the bottom right.

- Click on launch instance

The screenshot shows the AWS EC2 Instances page after a successful launch. A green success message box displays: "Success Successfully initiated launch of instance (i-0d867aa6e19b0850b)". Below the message, there is a "Launch log" link and a "Next Steps" section with several options: "Create billing usage alerts", "Connect to your instance", "Connect an RDS database", and "Create EBS snapshot policy".

- This shows successfully created instance

VPC_TASK1

- Check for connectivity of EC2 instance
- ssh -l ".pem" ec2-user@ip

```
user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ ssh -i "Neelima-Jenkins.pem" ec2-user@ec2-34-201-13-64.compute-1.amazonaws.com
, #
~\_ #####
~~ \##### Amazon Linux 2023
~~ \|##|
~~ \#/ ____ https://aws.amazon.com/linux/amazon-linux-2023
~~ V~' '-->
~~ . . / \
~~ / \ / \
~~ /m/
last login: Mon Jan  5 17:19:50 2026 from 115.96.62.22
[ec2-user@ip-172-168-2-143 ~]$ sudo su -
[root@ip-172-168-2-143 ~]# yum update -y
last metadata expiration check: 0:00:52 ago on Mon Jan  5 17:21:31 2026.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-168-2-143 ~]# yum install -y httpd php
last metadata expiration check: 0:01:08 ago on Mon Jan  5 17:21:31 2026.
Dependencies resolved.
=====
Package           Arch      Version       Repository      Size
=====
Installing:
httpd            x86_64    2.4.65-1.amzn2023.0.2   amazonlinux   47 k
php8.4           x86_64    8.4.14-1.amzn2023.0.1   amazonlinux   16 k
Installing dependencies:
apr               x86_64    1.7.5-1.amzn2023.0.4   amazonlinux   129 k
apr-util          x86_64    1.6.3-1.amzn2023.0.2   amazonlinux   97 k
apr-util-lmdb     x86_64    1.6.3-1.amzn2023.0.2   amazonlinux   13 k
generic-logos-httpd noarch   18.0.0-12.amzn2023.0.3  amazonlinux   19 k
httpd-core        x86_64    2.4.65-1.amzn2023.0.2   amazonlinux   1.4 M
httpd-filesystem  noarch   2.4.65-1.amzn2023.0.2   amazonlinux   13 k
httpd-tools        x86_64    2.4.65-1.amzn2023.0.2   amazonlinux   81 k
libbrotli         x86_64    1.0.9-4.amzn2023.0.2   amazonlinux   315 k
libsodium          x86_64    1.0.19-4.amzn2023   amazonlinux   176 k
libxml2            x86_64    2.9.9-1.amzn2023.0.3  amazonlinux   183 k
mailcap            noarch   2.1.49-3.amzn2023.0.3  amazonlinux   33 k
nginx-filesystem  noarch   1:1.28.0-1.amzn2023.0.2  amazonlinux   9.6 k
php8.4-cli         x86_64    8.4.14-1.amzn2023.0.1  amazonlinux   3.8 M
```

- sudo su – for root login
- yum install -y httpd php

```
Complete!
[root@ip-172-168-2-143 ~]# systemctl start httpd
[root@ip-172-168-2-143 ~]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr
/lib/systemd/system/httpd.service.
[root@ip-172-168-2-143 ~]# sudo vi /var/www/html/index.php
[root@ip-172-168-2-143 ~]# |
```

- systemctl start httpd
- systemctl enable httpd
- sudo vi /var/www/html/index.php

VPC_TASK1

```
root@ip-172-168-2-143:~  
?php  
phpinfo();
```

- phpinfo gives information of version and details

Not secure 34.201.13.64

PHP Version 8.4.14	
System	Linux ip-172-168-2-143 ec2.internal 8.1.158-180.294.amzn2023.x86_64 #1 SMP PREEMPT_DYNAMIC Mon Dec 1 05:36:50 UTC 2025 x86_64
Build Date	Oct 21 2025 19:23:55
Build System	Linux
Build Provider	Amazon Linux
Compiler	gcc (GCC) 11.5.0 20240719 (Red Hat 11.5.0-5)
Architecture	x86_64
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/10-opcache.ini, /etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-dom.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-finfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-mbstring.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-posix.ini, /etc/php.d/20-shmop.ini, /etc/php.d/20-simplexml.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sodium.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-sysvmsg.ini, /etc/php.d/20-sysvsem.ini, /etc/php.d/20-sysvshm.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/20-xml.ini, /etc/php.d/20-xmlwriter.ini, /etc/php.d/20-xsl.ini, /etc/php.d/30-pdo_sqlite.ini, /etc/php.d/30-xmlextender.ini
PHP API	20240924
PHP Extension	20240924
Zend Extension	420240924
Zend Extension Build	API420240924.NTS
PHP Extension Build	API20240924.NTS

- Verify in browser.

VPC_TASK1

8. Configure NAT gateway in public subnet and connect to private instance.

The screenshot shows the 'Create NAT gateway' page in the AWS VPC console. The 'NAT gateway settings' section includes:

- Name**: my-nat-gateway-01
- Availability mode**: Regional - new (selected)
- VPC**: Select a VPC (vpc-0bc432ffa77faf53e selected)
- Connectivity type**: Public (selected)

At the bottom, there are links for CloudShell, Feedback, and Console Mobile App, along with copyright information and privacy terms.

- Click VPC → NAT gateway → create NAT gateway

The screenshot shows the 'Create NAT gateway' page in the AWS VPC console, displaying advanced settings:

- Availability mode**: Regional - new (selected)
- VPC**: Select a VPC (vpc-0bc432ffa77faf53e selected)
- Connectivity type**: Public (selected)
- Method of Elastic IP (EIP) allocation**: Automatic (selected)

At the bottom, there are links for CloudShell, Feedback, and Console Mobile App, along with copyright information and privacy terms.

- Select Regional-new
- VPC which we want
- Connecting type
- Method of Elastic IP as Manual

VPC_TASK1

The screenshot shows the AWS VPC NAT gateway creation interface. A green success message at the top states: "Elastic IP address 54.157.200.46 (eipalloc-0e6a9795f6b7e081a) allocated." Below this, under the "Elastic IP (EIP) association" section, there is a table for associating EIPs with Availability Zones. The "use1-az1 (us-east-1a)" row has "54.157.200.46" selected in the dropdown. An "Allocate EIP" button is visible on the right.

- Click on Elastic IP association and click on allocate IP

The screenshot shows the AWS VPC NAT gateway details page for "nat-13759beec21195186 / nat_gateway_1". A green success message says: "NAT gateway nat-13759beec21195186 | nat_gateway_1 was created successfully." The "Details" section shows the following information:

NAT gateway ID nat-13759beec21195186	Availability mode Regional	State Pending	State message -
NAT gateway ARN arn:aws:ec2:us-east-1:814588432081:natgateway/nat-13759beec21195186	Connectivity type Public	Created Monday, January 5, 2026 at 23:23:45 GMT+5:30	Info
VPC vpc-0bc432ffa77faf53e	Method of EIP allocation Manual	Deleted -	

Below the details, there are tabs for "IP addresses", "Monitoring", "Flow logs", and "Tags". The "IP addresses" tab shows one associated IP address: "Associated IP addresses (1)".

- Successfully created NAT-gateway.

VPC_TASK1

The screenshot shows the 'Edit routes' screen for a specific route table. A route is being added for destination 172.168.0.0/16, targeting 'local'. Another route is being added for destination 0.0.0.0/0, targeting a NAT gateway with ID nat-13759beec21195186. The 'Save changes' button is highlighted.

- In Route tables add NAT-gateway and click on save changes

The screenshot shows the 'Resource map' interface. It displays four subnets (public_subnet_1, Public_subnet_2, subnet-0a5cbd124a108a546, subnet-04200f7f69924724a), four route tables (rtb-0696df46b8705c337, my_route_table_private, rtb-02b5e48005fadbcc, my_route_table_public), and two network connections (my_Internet_gateway_1, nat_gateway_1). The route table my_route_table_private and the connection to nat_gateway_1 are highlighted with orange boxes.

- This resource map it shows how connections are placed

VPC_TASK1

The screenshot shows the 'Launch an instance' wizard in the AWS EC2 console. The 'Network settings' section is open, displaying a VPC selection dropdown containing 'vpc-0bc432ffa77faf53e' (172.168.0.0/16). Below it, a subnet selection dropdown shows 'subnet-0c73e8b53c37239bf' (Public_subnet_2) with details: VPC: vpc-0bc432ffa77faf53e, Owner: 814588432081, Availability Zone: us-east-1a (use1-az1), Zone type: Availability Zone, IP addresses available: 250, CIDR: 172.168.2.0/24. A 'Create new subnet' button is also present. The 'Auto-assign public IP' section is set to 'Enable'. The 'Firewall (security groups)' section shows a radio button for 'Create security group' selected, with a text input field below containing 'launch-wizard-27'. The top navigation bar includes CloudShell, Feedback, and Console Mobile App.

The screenshot shows the 'Launch an instance' wizard with the 'Summary' section open. It displays the number of instances (1), software image (Amazon Linux 2023 AMI 2023.9.2...), virtual server type (t2.micro), and firewall (New security group). The 'Launch instance' button is highlighted in orange. The bottom navigation bar includes CloudShell, Feedback, and Console Mobile App.

- Launch an instance with required VPC and click on Launch instance

VPC_TASK1

The screenshot shows the AWS EC2 Instances page for instance `i-05613f9b5c6092165`. The instance is a `t2.micro` type, running in a VPC with ID `vpc-0bc432ffa77faf53e` and subnet `subnet-0a5cbd124a108a546`. It has a private IP of `172.168.4.184` and a public IP of `172.168.4.184`. The instance state is `Running`. A tooltip for the Public IPv4 address shows it has been copied.

The screenshot shows the AWS EC2 Instances page for instance `i-0acfca5c265d29921`. The instance is a `t2.micro` type, running in a VPC with ID `vpc-0bc432ffa77faf53e` and subnet `subnet-0a5cbd124a108a546`. It has a private IP of `172.168.2.189` and a public IP of `52.54.62.127`. The instance state is `Running`. A terminal window titled "MINGW64/c/Users/user/Downloads" is open, showing the command `ssh -i "AmazonLinux2.pem" user@52.54.62.127` being run. The terminal output indicates a key fingerprint warning and a connection attempt to port 22.

- First connect to instance with Public Ip

VPC_TASK1

For private IP instance

```
user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ ssh -i "Neelima-Jenkins.pem" ec2-user@172.168.4.184
ssh: connect to host 172.168.4.184 port 22: Connection timed out

user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ scp -i "Neelima-Jenkins.pem" Neelima-Jenkins.pem ec2-user@<PUBLIC_EC2_PUBLIC_IP>:/home/ec2-user/
bash: PUBLIC_EC2_PUBLIC_IP: No such file or directory

user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ scp -i "Neelima-Jenkins.pem" Neelima-Jenkins.pem ec2-user@52.54.62.127:/home/ec2-user/
Neelima-Jenkins.pem                                         100% 1678      7.5KB/s   00:00

user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ chmod 400 Neelima-Jenkins.pem

user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ |
```

```
[ec2-user@ip-172-168-2-189 ~]$ ssh -i "Neelima-Jenkins.pem" ec2-user@ec2-52-54-62-127.compute-1.amazonaws.com
,      #
~\_\ #####
     Amazon Linux 2023
~~ \####\
~~ \###|
~~  \#/ __ https://aws.amazon.com/linux/amazon-linux-2023
~~  V~' .->
~~  / \
~~ . .-
~~ /_/
~~ /m/'

Last login: Mon Jan  5 18:37:22 2026 from 115.96.62.22
[ec2-user@ip-172-168-2-189 ~]$ hostname -I
172.168.2.189
[ec2-user@ip-172-168-2-189 ~]$ ping -c 3 google.com
PING google.com (142.251.111.138) 56(84) bytes of data.
64 bytes from bk-in-f138.1e100.net (142.251.111.138): icmp_seq=1 ttl=102 time=2.02 ms
64 bytes from bk-in-f138.1e100.net (142.251.111.138): icmp_seq=2 ttl=102 time=1.77 ms
64 bytes from bk-in-f138.1e100.net (142.251.111.138): icmp_seq=3 ttl=102 time=2.31 ms

--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 1.767/2.031/2.312/0.222 ms
[ec2-user@ip-172-168-2-189 ~]$ |
```

- To provide Connectivity for Private subnet we have first copy .pem in Private subnet.
- And then hostname -l provides private ip
- Then check connection by using ping ip.

VPC_TASK1

9. Install Apache Tomcat in private EC2 and deploy a sample app.

The screenshots show the AWS VPC console interface across three pages:

- VPC dashboard > Your VPCs:** Displays the "Your VPCs" table with one entry: "Tomcat_VPC" (vpc-0afcb281693457501). The "State" column shows "Available". Other columns include "Encryption controls" and "DNS hostnames".
- VPC dashboard > Subnets:** Displays the "Subnets (6) Info" table. It lists six subnets: "Public_sub_tom" (subnet-09fb4495eb8f94f0e), "private_sub_tom" (subnet-07e30b9db7fd994e2), and four unnamed subnets (subnet-0fd11907c677b2e8c, subnet-0ca102e70a0c7a70, subnet-0c102e70a0c7a70, subnet-0c102e70a0c7a70). All subnets are in an "Available" state and are associated with the VPC "vpc-0afcb281693457501 | Tomcat_VPC".
- VPC dashboard > Internet gateways:** Displays the "Internet gateways (1/2) Info" table. It lists two internet gateways: "tom_IGW" (igw-0862707541d8d8846) and "igw-0a3d84d8fa117ed61". Both are attached to the VPC "vpc-0afcb281693457501 | Tomcat_VPC".

VPC_TASK1

VPC dashboard		NAT gateways					
AWS Global View ↗		Find NAT gateways by attribute or tag					
Filter by VPC		Name NAT gateway ID Connectivity... State State message Availability ...					
Virtual private cloud	Actions	Create NAT gateway	< 1 >	☰	☰	☰	☰
Your VPCs							
Subnets							
Route tables							

VPC dashboard		Route tables					
AWS Global View ↗		Find route tables by attribute or tag					
Filter by VPC		Last updated 2 minutes ago					
Virtual private cloud	Actions	Create route table	< 1 >	☰	☰	☰	☰
Your VPCs							
Subnets							
Route tables							

VPC_TASK1

```
ec2-user@ip-172-168-2-157:~  
user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)  
$ ssh -i Oregon.pem Oregon.pem ec2-user@35.93.146.183:/home/ec2-user/  
ssh: Could not resolve hostname oregon.pem: Name or service not known  
  
user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)  
$ scp -i Oregon.pem Oregon.pem ec2-user@35.93.146.183:/home/ec2-user/  
Oregon.pem                                         100% 1678      6.2KB/s   00:00  
  
user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)  
$ ^C  
  
user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)  
$ chmod 400 Oregon.pem  
  
user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)  
$ scp -i Oregon.pem Oregon.pem ec2-user@35.93.146.183:/home/ec2-user/  
Oregon.pem                                         100% 1678      6.3KB/s   00:00  
  
user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)  
$ ssh -i Oregon.pem ec2-user@35.93.146.183  
  
          #  
 ~\_\ ####_      Amazon Linux 2023  
 ~~ \#####\  
 ~~ \###|  
 ~~ \#/ ____ https://aws.amazon.com/linux/amazon-linux-2023  
 ~~ \~' '-'>  
 ~~ .--.  
 ~~ /_/  
 _/m/'  
Last login: Tue Jan  6 20:31:44 2026 from 115.96.62.22  
[ec2-user@ip-172-168-2-157 ~]$  
[ec2-user@ip-172-168-2-157 ~]$ chmod 400 Oregon.pem  
[ec2-user@ip-172-168-2-157 ~]$ ssh -i Oregon.pem ec2-user@172.168.2.157  
          #  
 ~\_\ ####_      Amazon Linux 2023  
 ~~ \#####\  
 ~~ \###|  
 ~~ \#/ ____ https://aws.amazon.com/linux/amazon-linux-2023  
 ~~ \~' '-'>  
 ~~ .--.  
 ~~ /_/  
 _/m/'  
Last login: Tue Jan  6 20:45:06 2026 from 115.96.62.22  
[ec2-user@ip-172-168-2-157 ~]$ |
```

VPC_TASK1

```
[ec2-user@ip-172-168-2-157 ~]$ ls -l /opt
total 0
drwxr-xr-x. 4 root root 33 Dec  3 16:42 aws
[ec2-user@ip-172-168-2-157 ~]$ sudo yum install java-17-amazon-corretto -y
Last metadata expiration check: 0:04:07 ago on Tue Jan  6 20:47:53 2026.
Package java-17-amazon-corretto-1:17.0.17+10-1.amzn2023.1.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-168-2-157 ~]$ java -version
openjdk version "17.0.17" 2025-10-21 LTS
OpenJDK Runtime Environment Corretto-17.0.17.10.1 (build 17.0.17+10-LTS)
OpenJDK 64-Bit Server VM Corretto-17.0.17.10.1 (build 17.0.17+10-LTS, mixed mode, sharing)
[ec2-user@ip-172-168-2-157 ~]$ cd /opt
[ec2-user@ip-172-168-2-157 opt]$ sudo wget https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.50/bin/apache-tomcat-10.1.50.tar.gz
--2026-01-06 20:53:10-- https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.50/bin/apache-tomcat-10.1.50.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 14299418 (14M) [application/x-gzip]
```

```
[ec2-user@ip-172-168-2-157 bin]$ cd /opt/tomcat/webapps
[ec2-user@ip-172-168-2-157 webapps]$ wget https://tomcat.apache.org/tomcat-7.0-doc/appdev/sample/sample.war
--2026-01-06 20:58:58-- https://tomcat.apache.org/tomcat-7.0-doc/appdev/sample/sample.war
Resolving tomcat.apache.org (tomcat.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to tomcat.apache.org (tomcat.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4606 (4.5K)
Saving to: 'sample.war'

sample.war                                              100%[=====] 4606/4606
2026-01-06 20:58:58 (54.0 MB/s) - 'sample.war' saved [4606/4606]

[ec2-user@ip-172-168-2-157 webapps]$ cd /opt/tomcat/bin
[ec2-user@ip-172-168-2-157 bin]$ ./shutdown.sh
Using CATALINA_BASE:  /opt/tomcat
Using CATALINA_HOME:  /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
[ec2-user@ip-172-168-2-157 bin]$ ./startup.sh
Using CATALINA_BASE:  /opt/tomcat
Using CATALINA_HOME:  /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
```

```
[ec2-user@ip-172-168-2-157 bin]$ ls /opt/tomcat/webapps
ROOT  docs  examples  host-manager  manager  sample  sample.war
[ec2-user@ip-172-168-2-157 bin]$ ps -ef | grep tomcat
ec2-user  27087      1  0 21:00 pts/1    00:00:04 /usr/bin/java -Djava.util.logging.config.file=/opt/tomcat/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djdk.tls.ephemeralDHKeySize=2048 -Djava.protocol.handler.pkgs=org.apache.catalina.webresources -Dsun.io.useCanonCaches=false -Dorg.apache.catalina.security.SecurityListener.UMASK=0027 --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.lang.reflect=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED -classpath /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar -Dcatalina.base=/opt/tomcat -Dcatalina.home=/opt/tomcat -Djava.io.tmpdir=/opt/tomcat/temp org.apache.catalina.startup.Bootstrap start
ec2-user  27486  2686  1 21:00 pts/1    00:00:00 grep --color=auto tomcat
[ec2-user@ip-172-168-2-157 bin]$ ./startup.sh
Using CATALINA_BASE:  /opt/tomcat
Using CATALINA_HOME:  /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
```

VPC_TASK1

10. Configure VPC flow logs and store the logs in S3 and CloudWatch.

Step 1: IAM Role/Policy Setup

VPC Flow Logs need permissions to write logs to CloudWatch Logs or S3.

For CloudWatch Logs

Create an IAM role or policy like this:

The screenshot shows the 'Create role' wizard in the AWS IAM console. The current step is 'Choose a use case for the specified service.' under the heading 'Use case'. The 'EC2' option is selected, which allows EC2 instances to call AWS services on behalf of the user. Other options include 'EC2 Role for AWS Systems Manager', 'EC2 Spot Fleet Role', 'EC2 - Spot Fleet Auto Scaling', 'EC2 - Spot Fleet Tagging', 'EC2 - Spot Instances', 'EC2 - Spot Fleet', and 'EC2 - Scheduled Instances'. At the bottom right of the wizard, there are 'Cancel' and 'Next Step' buttons.

The screenshot shows the 'Policies' page in the AWS IAM console. The left sidebar shows the navigation path: 'Identity and Access Management (IAM) > Policies'. The main area displays a green header bar stating 'Policy attached to entity VPC_Flow_Logs.' Below this, the title 'Policies (1449)' is shown with an 'Info' link. A description follows: 'A policy is an object in AWS that defines permissions.' A search bar and a 'Filter by Type' dropdown are present. The main content is a table listing 1449 policies, with columns for 'Policy name', 'Type', 'Used as', and 'Description'. The table includes several AWS managed policies such as 'AccessAnalyzerServiceRole', 'AccountManagementRole', 'AdministratorAccess', and 'AdministratorAccessForAWSAssumeRole'. At the bottom right of the page, there are 'Actions', 'Delete', and 'Create policy' buttons, along with a 'Cancel' button at the top right.

VPC_TASK1

Screenshot 1: AWS CloudWatch Log Management - Create log group

The screenshot shows the 'Create log group' page. On the left, the navigation menu includes 'CloudWatch' (selected), 'Alarms', 'AI Operations', 'GenAI Observability', 'Application Signals', 'Infrastructure Monitoring', 'Logs' (selected), 'Log Management' (New), and 'Log Anomalies'. The main area is titled 'Create log group' and contains 'Log group details' with a note about log classes. The 'Log group name' is set to 'Neelima_CloudWatch', 'Retention setting' is 'Never expire', and 'Log class' is 'Standard'. A 'KMS key ARN - optional' field is also present.

Screenshot 2: AWS VPC - Create flow logs

The screenshot shows the 'Create flow logs' page. Under 'Selected resources', 'Tomcat_VPC' is listed with resource ID 'vpc-0c5edb177a500d469' and state 'Available'. The 'Flow log settings' section includes a 'Name - optional' field with 'my_cloudwatch', a 'Filter' section with 'All' selected, and a 'Maximum aggregation interval' section with '10 minutes' selected. The status bar at the bottom indicates '© 2026, Amazon Web Services, Inc. or its affiliates.' and links for 'CloudShell', 'Feedback', 'Console Mobile App', 'Privacy', 'Terms', and 'Cookie preferences'.

Screenshot 3: AWS VPC - Your VPCs

The screenshot shows the 'Your VPCs' page. It displays a success message: 'Successfully created flow log for the following resource: vpc-0c5edb177a500d469'. Below this, a card for 'fl-0461dfffe79bb70ee / my_cloudwatch' shows detailed configuration: Flow Log ID 'fl-0461dfffe79bb70ee', Destination Type 'cloud-watch-logs', Destination Name 'Neelima_CloudWatch', Traffic Type 'All', Max Aggregation Interval '10 minutes', File Format 'Hive Compatible Partitions', IAM Role 'arn:aws:iam::814588432081:role/logs_VPC', Log Format 'Default', and Cross Account IAM Role 'Default'. The status bar at the bottom indicates '© 2026, Amazon Web Services, Inc. or its affiliates.' and links for 'CloudShell', 'Feedback', 'Console Mobile App', 'Privacy', 'Terms', and 'Cookie preferences'.

VPC_TASK1

The screenshot shows the AWS CloudWatch Log Management interface. The left sidebar has a 'Logs' section with 'Log Management' selected. The main area is titled 'Log events' and displays a table of log entries. The columns are 'Timestamp', 'Message', and 'Log stream name'. The messages show network traffic between various AWS services like EC2 instances and VPC endpoints.

Timestamp	Message	Log stream name
2026-01-06T18:47:50.000Z	2 814588432081 - 185.248.13.83 3.229.86.22 52747 443 -	nat-12bb241b09e3be9ed-all
2026-01-06T18:47:50.000Z	2 814588432081 - 87.121.22.4 3.229.86.22 28820 443 6 -	nat-12bb241b09e3be9ed-all
2026-01-06T18:47:50.000Z	2 814588432081 - 185.248.15.13 3.229.86.22 21119 443 -	nat-12bb241b09e3be9ed-all
2026-01-06T18:47:50.000Z	2 814588432081 - 95.141.248.155 3.229.86.22 48150 443 -	nat-12bb241b09e3be9ed-all
2026-01-06T18:47:50.000Z	2 814588432081 eni-0b92c5cb2575d570d 147.185.133.86 1..	eni-0b92c5cb2575d570d-all
2026-01-06T18:47:50.000Z	2 814588432081 eni-0b92c5cb2575d570d 79.124.62.122 19..	eni-0b92c5cb2575d570d-all
2026-01-06T18:47:50.000Z	2 814588432081 eni-0b92c5cb2575d570d 91.230.168.219 1..	eni-0b92c5cb2575d570d-all

For S3

The screenshot shows the 'Create flow log' configuration page. It includes sections for 'Selected resources', 'Flow log settings' (with a 'Name' field set to 'Log_s3'), and a 'Filter' section where 'All' is selected. The 'Maximum annotation interval' is set to 100ms.

The screenshot shows the 'Create bucket' configuration page. It includes sections for 'General configuration' (AWS Region: US East (N. Virginia) us-east-1, Bucket type: General purpose selected), 'Bucket name' (Logs_s3), and 'Copy settings from existing bucket' (Choose bucket). The bucket name must be 3 to 63 characters and unique.

VPC_TASK1

The screenshots illustrate the process of setting up VPC flow logs:

- S3 Bucket Creation:** A screenshot of the Amazon S3 Buckets page. It shows a success message: "Successfully created bucket 's3neelima'". Below it, a table lists three buckets: "s3neelima" (created Jan 7, 2026), "venkat-516" (created Oct 17, 2025), and "venkey-s3-516" (created Oct 11, 2025). Buttons for "Copy ARN", "Empty", "Delete", and "Create bucket" are visible.
- VPC Flow Log Configuration:** A screenshot of the VPC > Your VPCs > Create flow logs page. It shows the "S3 bucket ARN" field set to "arn:aws:s3:::s3neelima". A note says "Please note, a resource-based policy will be created for you and attached to the target bucket." A "Create S3 bucket" button is present.
- VPC Flow Log Details:** A screenshot of the VPC dashboard > fl-Oaa7fe3d30f3fb5b8 / s3logs page. It shows the flow log details:

Flow Log ID	Destination Type	Traffic Type	File Format
fl-Oaa7fe3d30f3fb5b8	s3	All	Plain text
Name	Destination Name	Max Aggregation Interval	Hive Compatible Partitions
s3logs	s3neelima	1 minute	Not enabled
State	IAM Role	Log Format	Partition Logs
Active	-	Default	Daily
Creation Time	Cross Account IAM Role		
Wednesday, January 7, 2026 at 00:39:29 GMT+5:30	-		