

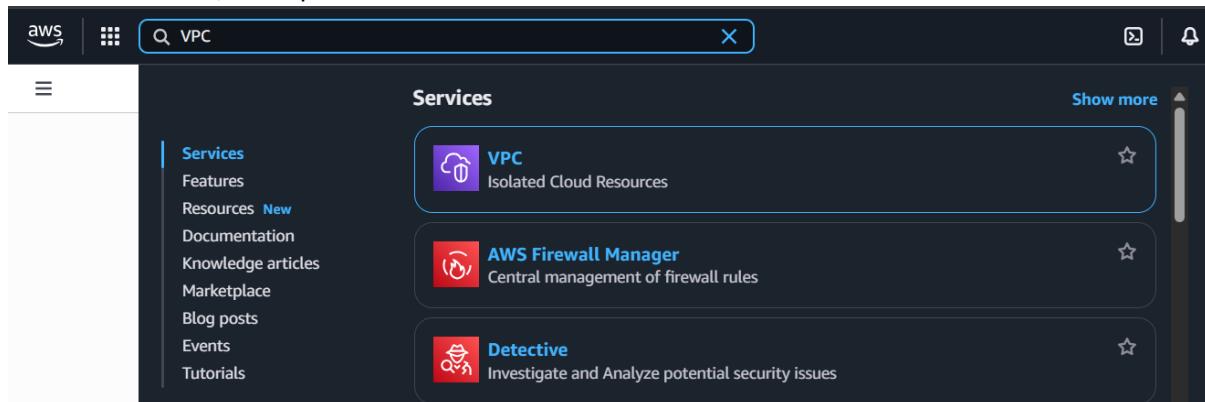
# Day 24

## “CLOUD SECURITY”

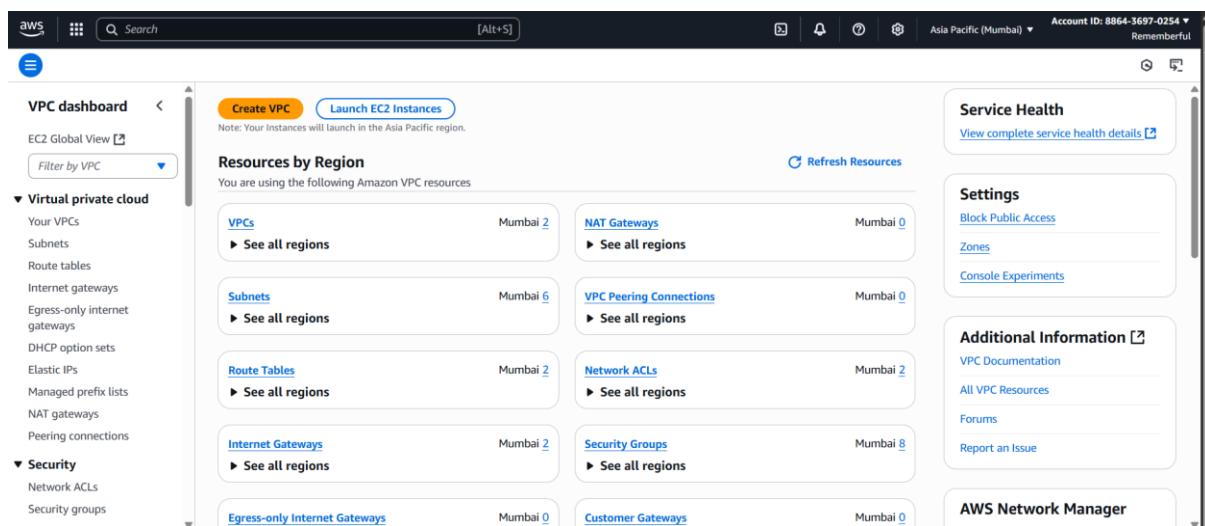
### Creating secure EC2 Instances in AWS Virtual Private Cloud (VPC):

Steps:

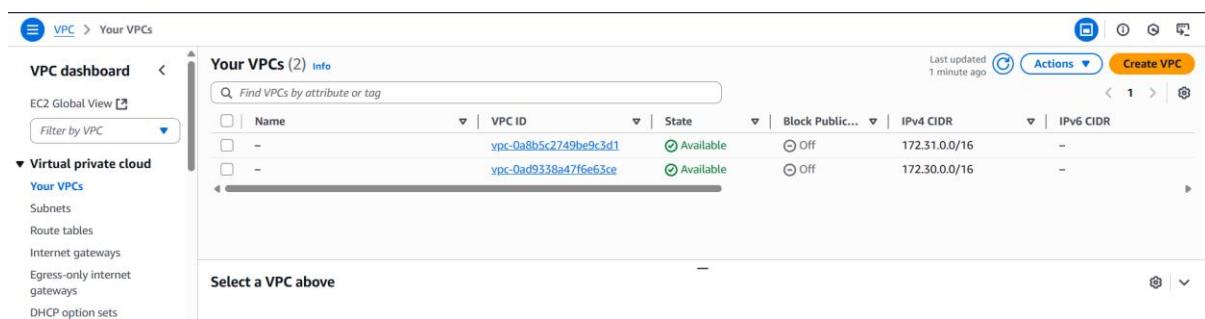
To create the VPC, first open the VPC:



Following window will appear:



Under the “Your VPC” section: click on “Create VPC” button



Following screen will appear:

**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**  
10.0.0.0/24  
CIDR block size must be between /16 and /28.

**IPv6 CIDR block** Info  
 No IPv6 CIDR block  IPAM-allocated IPv6 CIDR block  Amazon-provided IPv6 CIDR block

Fill the name, CIDR, don't change any other settings: then click on the “Create VPC” button.

**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-test

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

**IPv4 CIDR**  
10.0.0.0/16  
CIDR block size must be between /16 and /28.

**IPv6 CIDR block** Info  
 No IPv6 CIDR block  IPAM-allocated IPv6 CIDR block  Amazon-provided IPv6 CIDR block

Following confirmation will occurs:

**VPC dashboard**

**vpc-064d8bb93ed66ac4f / My-vpc-test**

**Details** Info

VPC ID	vpc-064d8bb93ed66ac4f	State	Available
DNS resolution	Enabled	Tenancy	default
Main network ACL	acl-0d54d807b0b653bef	Default VPC	No
IPv6 CIDR (Network border group)	-	Network Address Usage metrics	Disabled
		Block Public Access	Off
		DHCP option set	dopt-09593eeb193b19a5
		IPv4 CIDR	10.0.0.0/16
		Route 53 Resolver DNS Firewall rule groups	-
		DNS hostnames	Disabled
		Main route table	rtb-02ce93797b1c0a294
		IPv6 pool	-
		Owner ID	886436970254

**Resource map**

- VPC
- CIDRs
- Flow logs
- Tags
- Integrations

**Resource map** Info

- Subnets (0)
- Route tables (1)
- Network Connections (0)

Now, we need to create an Internet Gateway:

▼ Virtual private cloud

Your VPCs

- Subnets
- Route tables
- Internet gateways
- Egress-only internet gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- NAT gateways
- Peering connections

Following screen will appear:

The screenshot shows the AWS VPC dashboard with the 'Internet gateways' section selected. The left sidebar shows 'Your VPCs' and 'Virtual private cloud' sections. The main area displays a table titled 'Internet gateways (2) Info' with columns: Name, Internet gateway ID, State, VPC ID, and Owner. Two entries are listed:

Name	Internet gateway ID	State	VPC ID	Owner
-	igw-0570d191f381af917	Attached	vpc-0a8b5c2749be9c3d1	886436970254
-	igw-069e78168a6123c4e	Attached	vpc-0ad9338a47f6e63ce	886436970254

A message at the bottom says 'Select an internet gateway above'.

Click on the “Create Internet Gateway” Button: following screen will appear.

The screenshot shows the 'Create internet gateway' wizard. The first step, 'Internet gateway settings', is displayed. It includes fields for 'Name tag' (containing 'my-internet-gateway') and 'Tags - optional' (containing 'Add new tag'). A note states 'An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.' The bottom right has 'Cancel' and 'Create Internet gateway' buttons.

Fill it and rest of the things be default, and then click on “Create Internet Gateway” button:

VPC > Internet gateways > Create internet gateway

### Create internet gateway Info

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

**Internet gateway settings**

Name tag  
Creates a tag with a key of 'Name' and a value that you specify.  
my-gateway-forVPC

**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	Value - optional
Q Name	Q my-gateway-forVPC X Remove

Add new tag  
You can add 49 more tags.

Cancel **Create internet gateway**

Following confirmation will show:

aws Search [Alt+S] Account ID: 8864-3697-0254 ▾ Rememberful

VPC > Internet gateways > igw-018aa371746fb1c40

The following internet gateway was created: igw-018aa371746fb1c40 - my-gateway-forVPC. You can now attach to a VPC to enable the VPC to communicate with the internet. **Attach to a VPC**

**igw-018aa371746fb1c40 / my-gateway-forVPC**

**Details** Info

Internet gateway ID	State	VPC ID	Owner
igw-018aa371746fb1c40	Detached	-	886436970254

**Tags**

Key	Value
Name	my-gateway-forVPC

Actions

At the green confirmation, click on the “Attach to a VPC” button:

The following internet gateway was created: igw-018aa371746fb1c40 - my-gateway-forVPC. You can now attach to a VPC to enable the VPC to communicate with the internet. **Attach to a VPC**

Following window will appear:

aws Search [Alt+S] Account ID: 8864-3697-0254 ▾ Rememberful

VPC > Internet gateways > Attach to VPC (igw-018aa371746fb1c40)

The following internet gateway was created: igw-018aa371746fb1c40 - my-gateway-forVPC. You can now attach to a VPC to enable the VPC to communicate with the internet. **Attach to a VPC**

**Attach to VPC (igw-018aa371746fb1c40) Info**

**VPC**  
Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

**Available VPCs**  
Attach the internet gateway to this VPC.  
Select a VPC

**AWS Command Line Interface command**

Cancel **Attach internet gateway**

Select the required VPC: and then click on “Attach Internet Gateway” button, following confirmation will occur.

The screenshot shows the AWS VPC Internet gateways details page. A green banner at the top indicates that the internet gateway has been successfully attached to a VPC. The main section displays the internet gateway's ID (igw-018aa371746fb1c40), state (Attached), VPC ID (vpc-064d8bb93ed66ac4f), and owner (886436970254). Below this, there is a 'Tags' section with one tag named 'my-gateway-forVPC'. The left sidebar shows the 'Virtual private cloud' menu with the 'Internet gateways' option selected.

Make sure that it shows “attached”:

The screenshot shows the AWS VPC Internet gateways list page. It displays three internet gateways: 'igw-0570d191f581af917' (Attached, vpc-0a8b5c2749be9c3d1), 'igw-069e78168a6123c4e' (Attached, vpc-0ad9338a47f6e63ce), and 'igw-018aa371746fb1c40' (Attached, vpc-064d8bb93ed66ac4f). The left sidebar shows the 'Virtual private cloud' menu with the 'Internet gateways' option selected.

Next, we need to create private and public subnet in the VPC, to do so click on the “subnet” option available on the left menu:

The screenshot shows the AWS VPC Virtual private cloud menu. The 'Subnets' option is highlighted. Other options include Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, and Peering connections.

Following screen will appear:

The screenshot shows the AWS VPC Subnets list page. It displays six subnets with their names, subnet IDs, states, VPCs, and IPv4 CIDRs. All subnets are marked as 'Available' and have 'Block Public' set to 'Off'. The left sidebar shows the 'Virtual private cloud' menu with the 'Subnets' option selected.

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
-	subnet-0c0ed0fce109a50da	Available	vpc-0a8b5c2749be9c3d1	Off	172.31.16.0
-	subnet-0fce07b8fd07d80	Available	vpc-0ad9338a47f6e63ce	Off	172.30.0.0/
-	subnet-0cef412d1c11c5360	Available	vpc-0a8b5c2749be9c3d1	Off	172.31.32.0
-	subnet-0997873fcc98020df	Available	vpc-0a8b5c2749be9c3d1	Off	172.31.0.0/
-	subnet-03e51e10085584713	Available	vpc-0ad9338a47f6e63ce	Off	172.30.1.0/

First, we will create a private subnet: click on the “Create subnet” button, following screen will appear.

The screenshot shows the 'Create subnet' wizard. Step 1: VPC selection. It has a 'VPC' section with a dropdown menu labeled 'Select a VPC'. Below it is a 'Subnet settings' section with a note: 'Specify the CIDR blocks and Availability Zone for the subnet.' A message says 'Select a VPC first to create new subnets.' A 'Add new subnet' button is present. At the bottom are 'Cancel' and 'Create subnet' buttons.

Select the VPC ID, and assign the name:

The screenshot shows the 'Create subnet' wizard. Step 2: Subnet configuration. It has a 'VPC' section with a dropdown menu showing 'vpc-064d8bb93ed6ac4f (My-vpc-test)'. Below it is an 'Associated VPC CIDRs' section with an 'IPv4 CIDRs' input field containing '10.0.0.0/16'. Under 'Subnet settings', there's a 'Subnet 1 of 1' section with a 'Subnet name' input field containing 'my-private-subnet-test'. A note below says 'The name can be up to 256 characters long.'

Scroll down and make sure to write the “IPV4 subnet CIDR block”:

The screenshot shows the 'Create subnet' wizard. Step 3: Advanced subnet configuration. It includes sections for 'Subnet 1 of 1': 'Subnet name' (my-private-subnet-test), 'Availability Zone' (No preference), 'IPv4 VPC CIDR block' (10.0.0.0/16), 'IPv4 subnet CIDR block' (10.0.1.0/24), and 'Tags - optional' (Key: Name, Value: my-private-subnet-test). A note at the bottom says 'You can add 49 more tags.'

Leave rest of the settings as it is, and then click on the “create subnet” button: following confirmation will occur.

You have successfully created 1 subnet: subnet-0c694d684907a60b1

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
my-private-subnet-test	subnet-0c694d684907a60b1	Available	vpc-064d8bb93ed66ac4f   My-v...	Off	10.0.1.0/24

So a private subnet is created, now we need to create a public subnet.

For this, again click on the “Create subnet” button:

You have successfully created 1 subnet: subnet-0c694d684907a60b1

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
my-private-subnet-test	subnet-0c694d684907a60b1	Available	vpc-064d8bb93ed66ac4f   My-v...	Off	10.0.1.0/24

Again, do the same process we did above, just make sure to specify the different subnet block.

**Create subnet**

**VPC**  
VPC ID  
Create subnets in this VPC.  
vpc-064d8bb93ed66ac4f (My-vpc-test)

**Associated VPC CIDRs**  
IPv4 CIDRs  
10.0.0.0/16

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

**Subnet 1 of 1**

**Subnet name**  
Create a tag with a key of 'Name' and a value that you specify.  
my-public-subnet-

**Availability Zone**  
Choose the zone in which your subnet will reside, or let Amazon choose one for you.  
No preference

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

**IPv4 subnet CIDR block**  
10.0.2.0/24  
256 IPs

**Tags - optional**

Click on the “create subnet” button, following confirmation will occur.

You have successfully created 1 subnet: subnet-0e9def89cdd30fad7

Subnets (8) Info

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
-	subnet-0c694d684907a60b1	Available	vpc-064d8bb93ed6ac4f   My-v...	Off	10.0.1.0/24
my-private-subnet-test	subnet-0e9def89cdd30fad7	Available	vpc-064d8bb93ed6ac4f   My-v...	Off	10.0.2.0/24
my-public-subnet-					

Select a subnet

Thus, we have two subnets for private and public access.

Next, we will create the route tables to route the network traffic: where to find this route table?

Virtual private cloud

- Your VPCs
- Subnets**
- Route tables
- Internet gateways
- Egress-only internet gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- NAT gateways
- Peering connections

Following screen will appear:

Route tables (2) Info

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
-	rtb-005f530f9cecd4504	-	-	Yes	vpc-0a8b5c2749be9c3d1
-	rtb-07f6f57f45968188e	-	-	Yes	vpc-0ad9338a47f6e65ce

Select a route table

To create a route table for the public subnet, click on the “Create route table” button: following screen will appear.

**Create route table** Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

**Route table settings**

**Name - optional**  
Create a tag with a key of 'Name' and a value that you specify.

**VPC**  
The VPC to use for this route table.  
Select a VPC

**Tags**  
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	Value - optional
Q Name	X Q - Remove

Add new tag  
You can add 49 more tags.

**Create route table**

Fill the details and then click on the orange button:

**Create route table** Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

**Route table settings**

**Name - optional**  
Create a tag with a key of 'Name' and a value that you specify.

**VPC**  
The VPC to use for this route table.  
vpc-064d8bb93ed66ac4f (My-vpc-test)

**Tags**  
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	Value - optional
Q Name	X Q my-routetable-public-test Remove

Add new tag  
You can add 49 more tags.

**Create route table**

Following confirmation will occur:

**VPC dashboard**

Route table rtb-0fa976ce70f7edf63 | my-routetable-public-test was created successfully.

**rtb-0fa976ce70f7edf63 / my-routetable-public-test**

**Details** Info

**Virtual private cloud**

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
-	rtb-0fa976ce70f7edf63	-	-	No	vpc-064d8bb93ed66ac4f
my-routetable-public-test	rtb-0fa976ce70f7edf63	-	-	No	vpc-064d8bb93ed66ac4f
my-routetable-private-test	rtb-02ce93797b1c0a294	-	-	Yes	vpc-064d8bb93ed66ac4f

Go back to the section of “Route Tables”, make sure to edit the name of default route table created or the VPC to identify the private route table:

**Virtual private cloud**

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
-	rtb-0fa976ce70f7edf63	-	-	No	vpc-064d8bb93ed66ac4f
my-routetable-public-test	rtb-0fa976ce70f7edf63	-	-	No	vpc-064d8bb93ed66ac4f
my-routetable-private-test	rtb-02ce93797b1c0a294	-	-	Yes	vpc-064d8bb93ed66ac4f

Now, we will configure the public route table as public and add the public subnet and internet gateway.

Click on the public route table: and click on the “routes” section.

The screenshot shows the AWS VPC Route Tables page. In the left sidebar, under 'Route tables', 'my-routetable-public-test' is selected. The main area displays a table of route tables with one row selected: 'my-routetable-public-test' (rtb-0fa976ce70f7edf63). Below this, the details for 'rtb-0fa976ce70f7edf63 / my-routetable-public-test' are shown, with the 'Routes' tab selected. The 'Routes' table has one entry: Destination 10.0.0.0/16, Target local, Status Active, Propagated No, and Route Origin Create Route Table.

Click on the “Edit Routes” button: following screen will appear

The screenshot shows the 'Edit routes' dialog box for the public route table. It contains a table with one route entry: Destination 10.0.0.0/16, Target local, Status Active, Propagated No, and Route Origin CreateRouteTable. There is an 'Add route' button at the bottom left and 'Save changes' button at the bottom right.

Click on “add route” button to add the IP: write 0.0.0.0/0 for allowing traffic from any source, and then select the Internet gateway option from the target section.

The screenshot shows the 'Edit routes' dialog box after adding a new route. The table now includes two routes: one to 10.0.0.0/16 (Target local) and another to 0.0.0.0/0 (Target Internet Gateway, ID igw-018aa371746fb1c40). The 'Add route' and 'Save changes' buttons are visible at the bottom.

Click on “save changes” button: following confirmation will occur.

The screenshot shows the VPC Route Tables page again. The public route table now has two routes: one to 10.0.0.0/16 (Target local) and one to 0.0.0.0/0 (Target Internet Gateway, ID igw-018aa371746fb1c40). A green banner at the top indicates that the routes were updated successfully. The 'Routes' tab is selected, showing the updated route table.

Now, again select the public route table and then go to the “Subnet association” section:

The screenshot shows the AWS VPC Route Tables page. A green banner at the top indicates "Updated routes for rtb-0fa976ce70f7edf63 / my-routetable-public-test successfully". The left sidebar shows "Virtual private cloud" and "Route tables" selected. The main table lists two route tables: "my-routetable-public-test" (selected) and "my-routetable-private-test". The "Subnet associations" tab is selected under the "rtb-0fa976ce70f7edf63 / my-routetable-public-test" section, which currently has 0 explicit subnet associations.

Click on the “Edit subnet association” button: following window will appear

The screenshot shows the "Edit subnet associations" dialog box. It displays a table of "Available subnets (2)" with columns: Name, Subnet ID, IPv4 CIDR, IPv6 CIDR, and Route table ID. Two subnets are listed: "my-private-subnet-test" and "my-public-subnet-". Below the table, a "Selected subnets" section shows "subnet-0e9def89cdd30fad7 / my-public-subnet-". Buttons for "Cancel" and "Save associations" are at the bottom right.

Select the subnet for public association:

The screenshot shows the "Edit subnet associations" dialog box again. A specific subnet, "subnet-0e9def89cdd30fad7 / my-public-subnet-", is selected in the "Available subnets" table. The "Selected subnets" section now contains this subnet. Buttons for "Cancel" and "Save associations" are visible.

Click on the “Save association” button: following confirmation will occur.

The screenshot shows a green confirmation banner: "You have successfully updated subnet associations for rtb-0fa976ce70f7edf63 / my-routetable-public-test." The "Route tables (1 / 1)" section is shown below, with the same route table and subnet association information as the previous screens.

Clearly the association is added successfully:

You have successfully updated subnet associations for rtb-0fa976ce70f7edf63 / my-routetable-public-test.

**Route tables (1/4) Info**

Name	Route table ID	Explicit subnet assoc...	Edge associations	Main	VPC
-	rtb-005f530f9cec4504	-	-	Yes	vpc-0a8b5c2749be9c3d1
-	rtb-07f6f57f45968188e	-	-	Yes	vpc-0ad9338a47f6e63ce
<b>my-routetable-public-test</b>	<b>rtb-0fa976ce70f7edf63</b>	<b>subnet-0e9def89cdd30fa...</b>	<b>-</b>	<b>No</b>	<b>vpc-064d8bb93ed66ac4f1</b>

**rtb-0fa976ce70f7edf63 / my-routetable-public-test**

Details | Routes | **Subnet associations** | Edge associations | Route propagation | Tags

**Explicit subnet associations (1)**

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
my-public-subnet-	subnet-0e9def89cdd30fad7	10.0.2.0/24	-

Now, we will add the subnet for the private route table:

You have successfully updated subnet associations for rtb-0fa976ce70f7edf63 / my-routetable-public-test.

**Route tables (1/4) Info**

Name	Route table ID	Explicit subnet assoc...	Edge associations	Main	VPC
-	rtb-07f6f57f45968188e	-	-	Yes	vpc-0ad9338a47f6e63ce
my-routetable-public-test	rtb-0fa976ce70f7edf63	subnet-0e9def89cdd30fa...	-	No	vpc-064d8bb93ed66ac4f1
<b>my-routetable-private-test</b>	<b>rtb-02ce93797b1c0a294</b>	<b>-</b>	<b>-</b>	<b>Yes</b>	<b>vpc-064d8bb93ed66ac4f1</b>

**rtb-02ce93797b1c0a294 / my-routetable-private-test**

Details | Routes | **Subnet associations** | Edge associations | Route propagation | Tags

**Explicit subnet associations (0)**

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
------	-----------	-----------	-----------

No subnet associations  
You do not have any subnet associations.

Then,

**Edit subnet associations**

Change which subnets are associated with this route table.

**Available subnets (1/2)**

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<b>my-private-subnet-test</b>	<b>subnet-0c694d684907a60b1</b>	<b>10.0.1.0/24</b>	<b>-</b>	<b>Main (rtb-02ce93797b1c0a294 / my-ro...)</b>
my-public-subnet-	subnet-0e9def89cdd30fad7	10.0.2.0/24	-	rtb-0fa976ce70f7edf63 / my-routetab...

**Selected subnets**

subnet-0c694d684907a60b1 / my-private-subnet-test <b>X</b>
--

**Cancel** **Save associations**

Following confirmation will occur:

The screenshot shows the AWS VPC Route Tables page. A green success message at the top states: "You have successfully updated subnet associations for rtb-02ce93797b1c0a294 / my-routetable-private-test." Below this, the "Route tables (1/4) Info" section shows a table with three rows. The first two rows are empty, and the third row is for "my-routetable-public-test" with route table ID "rtb-0fa976ce70f7edf63". It has one explicit subnet association: "subnet-0e9def89cccd30fa..." with IPv4 CIDR "10.0.1.0/24". The "Actions" button is highlighted.

Now, we will configure the traffic from the internet, using the “Security Groups” section: following window will occur.

The screenshot shows the AWS Security Groups page. The left sidebar shows sections for Security, PrivateLink and Lattice, and Network ACLs. The main table lists nine security groups. The first group, "sg-0f0763db40672805d", is selected. Its details are shown in a modal dialog titled "Select a security group".

Name	Security group ID	Security group name	VPC ID	Description
-	sg-072839513ecc16cc2	launch-wizard-3	vpc-0a8b5c2749be9c3d1	launch-wizard-
-	sg-0296133aa4f909f27c	default	vpc-0ad9338a47f6e63ce	default VPC sec
-	sg-0cc95d2c54db4d76c	launch-wizard-1	vpc-0a8b5c2749be9c3d1	launch-wizard-
-	sg-0ff8a73b04ef15617	launch-wizard-6	vpc-0a8b5c2749be9c3d1	launch-wizard-
-	sg-0179ae47375b5b765	default	vpc-0a8b5c2749be9c3d1	default VPC sec

We can actually see the default security group created for our VPC:

The screenshot shows the AWS Security Groups page. The left sidebar shows sections for Security, PrivateLink and Lattice, and Network ACLs. The main table lists nine security groups. The second group, "sg-0f0763db40672805d", is selected. Its details are shown in a modal dialog titled "Select a security group".

Name	Security group ID	Security group name	VPC ID	Description
-	sg-0179ae47375b5b765	default	vpc-0a8b5c2749be9c3d1	default VPC sec
<input checked="" type="checkbox"/>	sg-0f0763db40672805d	default	vpc-064d8bb93ed66ac4f	default VPC sec
-	sg-014d7223c88ae31c3	launch-wizard-5	vpc-0a8b5c2749be9c3d1	launch-wizard-
-	aditya-sg	launch-wizard-4	vpc-0a8b5c2749be9c3d1	launch-wizard-
-	sg-0633273b46a8fef61	launch-wizard-2	vpc-0a8b5c2749be9c3d1	launch-wizard-

Edit its name:

The screenshot shows the AWS Security Groups page. The left sidebar shows sections for Security, PrivateLink and Lattice, and Network ACLs. The main table lists nine security groups. The second group, "sg-0f0763db40672805d", now has a red circle with a question mark icon next to it, indicating it's being edited. Its details are shown in a modal dialog titled "Select a security group".

Name	Security group ID	Security group name	VPC ID	Description
-	sg-0179ae47375b5b765	default	vpc-0a8b5c2749be9c3d1	default VPC sec
<input checked="" type="checkbox"/>	sg-0f0763db40672805d	my-sg-test	vpc-064d8bb93ed66ac4f	default VPC sec
-	sg-014d7223c88ae31c3	launch-wizard-5	vpc-0a8b5c2749be9c3d1	launch-wizard-
-	aditya-sg	launch-wizard-4	vpc-0a8b5c2749be9c3d1	launch-wizard-
-	sg-0633273b46a8fef61	launch-wizard-2	vpc-0a8b5c2749be9c3d1	launch-wizard-

Now, we will configure the inbound traffic for that security group:

Name	Security group ID	Security group name	VPC ID	Description
sg-0f0763db40672805d	sg-0f0763db40672805d	default	vpc-0a8b5c2749be9c3d1	default VPC sec
my-sg-test	sg-0f0763db40672805d	default	vpc-0a8b5c2749be9c3d1	default VPC sec
sg-014d7223c88aae31c3	sg-014d7223c88aae31c3	launch-wizard-5	vpc-0a8b5c2749be9c3d1	launch-wizard-
aditya-sg	sg-0633273b46a8fef61	launch-wizard-4	vpc-0a8b5c2749be9c3d1	launch-wizard-

Click on the “Edit Inbound rules” button: following screen will appear.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-040c8663d036b77d0	All traffic	All	All	Custom sg-0f0763db40672805d	

Click on “add rule” button, and select the type as “All traffic”, and in source “My IP” to make sure only your IP can access.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-040c8663d036b77d0	All traffic	All	All	Custom sg-0f0763db40672805d	
-	All traffic	All	All	My IP	

Then click on the “save rules” button: following confirmation will occur.

ⓘ Inbound security group rules successfully modified on security group (sg-0f0763db40672805d | default)

So, inbound rules is set for the Security groups.

Now, open the ACL:

Name	Network ACL ID	Associated with	Default	VPC ID	Inbound
-	acl-048215e5962e62d86	3 Subnets	Yes	vpc-0a8b5c2749be9c3d1	2 Inbx
-	acl-0c0c16beec66025d0	3 Subnets	Yes	vpc-0ad9338a47f6e63ce	2 Inbx
<b>-</b>	<b>acl-0d54d807b0b653bef</b>	<b>2 Subnets</b>	<b>Yes</b>	<b>vpc-064d8bb93ed66ac4f / My-vpc-test</b>	<b>2 Inbx</b>

Identify the default Network ACL:

Name	Network ACL ID	Associated with	Default	VPC ID	Inbound
-	acl-048215e5962e62d86	3 Subnets	Yes	vpc-0a8b5c2749be9c3d1	2 Inbx
-	acl-0c0c16beec66025d0	3 Subnets	Yes	vpc-0ad9338a47f6e63ce	2 Inbx
<b>-</b>	<b>acl-0d54d807b0b653bef</b>	<b>2 Subnets</b>	<b>Yes</b>	<b>vpc-064d8bb93ed66ac4f / My-vpc-test</b>	<b>2 Inbx</b>

Change its name:

Name	Network ACL ID	Associated with	Default	VPC ID	Inbound
-	acl-048215e5962e62d86	3 Subnets	Yes	vpc-0a8b5c2749be9c3d1	2 Inbx
-	acl-0c0c16beec66025d0	3 Subnets	Yes	vpc-0ad9338a47f6e63ce	2 Inbx
<b>my-acl-test</b>	<b>acl-0d54d807b0b653bef</b>	<b>2 Subnets</b>	<b>Yes</b>	<b>vpc-064d8bb93ed66ac4f / My-vpc-test</b>	<b>2 Inbx</b>

Next, click on the “Inbound rule” section: clearly all the traffic is allowed, but we need only traffic from our IP.

Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	<b>Allow</b>
*	All traffic	All	All	0.0.0.0/0	<b>Deny</b>

Click on the “Edit Inbound rules” button: following screen will appear.

The screenshot shows the 'Edit inbound rules' interface. It displays two rules:

Rule number	Type Info	Protocol Info	Port range Info	Source Info	Allow/Deny Info
100	All traffic	All	All	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

Buttons at the bottom include 'Add new rule', 'Sort by rule number', 'Cancel', 'Preview changes', and 'Save changes'.

Specify your IP, in the Source:

The screenshot shows the 'Edit inbound rules' interface with the source IP updated to 160.238.92.241/32. The rest of the configuration is identical to the previous screenshot.

Click on the Save changes button: following confirmation will occur.

The screenshot shows a confirmation message: "You have successfully updated inbound rules for acl-0d54d807b0b653bef / my-acl-test". Below the message are buttons for 'Actions' and 'Create network ACL'.

So, till now, ACL is also configured properly.

Now, we will move towards the EC2 Instance creation:

The screenshot shows the AWS Services page with a search bar containing 'EC2'. The sidebar shows 'VPC' is selected under 'Services'. The main area shows various services and features:

- Services**: EC2 (Virtual Servers in the Cloud), EC2 Image Builder (A managed service to automate build, customize and deploy OS images), EC2 Global View (Provides a global dashboard and search functionality).
- Features**: EC2 Global View (Provides a global dashboard and search functionality that lets you ...).

Following screen will appear:

The screenshot shows the AWS EC2 Dashboard for the Asia Pacific (Mumbai) Region. The left sidebar includes links for Dashboard, Instances, Images, Elastic Block Store, Network & Security, and Load Balancing. The main area displays 'Resources' with counts for Instances (running), Auto Scaling Groups, Capacity Reservations, Dedicated Hosts, Elastic IPs, Instances, Key pairs, Load balancers, Placement groups, Security groups, Snapshots, and Volumes. Below this is a 'Launch instance' section with a 'Launch instance' button and a note about launching in the Asia Pacific (Mumbai) Region. To the right is a 'Service health' section showing the status of AWS Health Dashboard and a 'Zones' section. On the far right, there's a 'EC2 Free Tier' summary with sections for 'Offer usage (monthly)' and 'View all AWS Free Tier offers'.

Click on launch instance button: following screen appears.

The screenshot shows the 'Launch an instance' wizard. Step 1: Set instance details. It asks for a 'Name and tags' (e.g., 'My Web Server') and lists 'Add additional tags'. Step 2: Application and OS Images (Amazon Machine Image). It shows a search bar ('Search our full catalog including 1000s of application and OS images') and a grid of quick start AMIs: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. A note says 'Including AMIs from AWS, Marketplace and the Community'. On the right, a 'Summary' section shows 1 instance, the selected 'Software Image (AMI)' (Amazon Linux 2023 AMI 2023.8.2...), 'Virtual server type (instance type)' (t2.micro), 'Firewall (security group)' (New security group), and 'Storage (volumes)' (1 volume(s) - 8 GiB). A note about 'Free tier' is visible. Buttons for 'Cancel', 'Launch instance', and 'Preview code' are at the bottom.

Scroll down, keep other things as it is, just make the following changes in the “network settings” section:

**Network settings**

**VPC - required**

vpc-064d8bb93ed66ac4f (My-vpc-test)  
10.0.0.0/16

**Subnet**

subnet-0e9def89cd30fad7  
my-public-subnet-  
VPC: vpc-064d8bb93ed66ac4f Owner: 886436970254 Availability Zone: ap-south-1a (aps1-az1)  
Zone type: Availability Zone IP addresses available: 251 CIDR: 10.0.2.0/24

**Create new subnet**

**Auto-assign public IP**

Enable

**Firewall (security groups)**

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

**Create security group** **Select existing security group**

**Common security groups**

Select security groups

default sg-0f0763db40672805d X  
VPC: vpc-064d8bb93ed66ac4f

**Compare security group rules**

**Advanced network configuration**

Click on “launch instance” button: following confirmation will occur.



We may recheck the status of the instance here:

**EC2**

- Dashboard
- EC2 Global View
- Events
- Instances**
- Images
- Elastic Block Store
- Volumes
- Snapshots
- Lifecycle Manager
- Network & Security
- Security Groups
- Elastic IPs

**Instances (1) Info**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public
test-instance	i-00fd9e3113c564ee6	Running	t2.micro	Initializing		ap-south-1a	-	13.232

**Select an instance**

Click on the instance and note the Public IP address:

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links for EC2, Dashboard, EC2 Global View, Events, Instances, Images, Elastic Block Store, Network & Security, Load Balancing, and Trust Stores. The main area displays a table titled 'Instances (1/1) Info'. The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public. One row is selected, showing 'test-instance' with Instance ID 'i-00fd9e3113c564ee6', State 'Running', Type 't2.micro', Status 'Initializing', and Availability Zone 'ap-south-1a'. The Public IPv4 DNS is listed as '13.232'. Below the table, a detailed view for 'test-instance' is shown with tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. Under the Details tab, the 'Instance summary' section shows the Instance ID 'i-00fd9e3113c564ee6', Public IPv4 address '13.232.62.218', Instance state 'Running', and Hostname type.

Now, we will create another instance for the private subnet: click on “launch instance” button again.

For this new instance, keep things as default, just change the “network settings” section as:

The screenshot shows the 'Network settings' configuration page. It includes sections for VPC (set to 'vpc-064d8bb93ed66ac4f (My-vpc-test)'), Subnet (set to 'subnet-0c694d684907a60b1'), Auto-assign public IP (set to 'Enable'), Firewall (with 'Select existing security group' selected), and Common security groups (listing 'default sg-0f0763db40672805d'). There are also buttons for 'Create new subnet' and 'Compare security group rules'.

Then click on the “Launch instance” button: clearly “my-test-private” is created.

The screenshot shows the AWS EC2 Instances page again. The sidebar and table structure are identical to the previous screenshot. The table now shows two rows: 'test-instance' (selected) with Instance ID 'i-00fd9e3113c564ee6' and 'my-test-private' with Instance ID 'i-0fe2e2a4cb03be20'. Both instances are in the 'Running' state, t2.micro type, and ap-south-1a availability zone. The Public IPv4 DNS for 'test-instance' is '13.232.62.218' and for 'my-test-private' is '13.232.205.187'.

We can also see the details of the private instance :

Instances (1/2) info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
my-test-private	i-0fce2e2a4cb03be20	Running	t2.micro	Initializing	<a href="#">View alarms +</a>	ap-south-1a	-	13.232.205.187	-
test-instance	i-00f9e983113c564ee6	Running	t2.micro	2/2 checks passed	<a href="#">View alarms +</a>	ap-south-1a	-	13.232.62.218	-

**i-0fce2e2a4cb03be20 (my-test-private)**

**Details** Status and alarms Monitoring Security Networking Storage Tags

**Instance summary**

Instance ID	i-0fce2e2a4cb03be20	Public IPv4 address	13.232.205.187 <a href="#">open address</a>
IPv6 address	-	Private IP DNS name (IPv4 only)	ip-10-0-1-198.ap-south-1.compute.internal
Hostname type	IP name: ip-10-0-1-198.ap-south-1.compute.internal	Instance type	t2.micro
Answer private resource DNS name		Elastic IP addresses	

Now, we will check if we are able to achieve our goal or not, for which we will ping the Public IP of the non-private instance: clearly it is working.

```
C:\Users\Aditya>ping 13.232.62.218

Pinging 13.232.62.218 with 32 bytes of data:
Reply from 13.232.62.218: bytes=32 time=44ms TTL=120
Reply from 13.232.62.218: bytes=32 time=44ms TTL=120
Reply from 13.232.62.218: bytes=32 time=44ms TTL=120
Reply from 13.232.62.218: bytes=32 time=43ms TTL=120

Ping statistics for 13.232.62.218:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 43ms, Maximum = 44ms, Average = 43ms

C:\Users\Aditya>
```

Similarly, pinging private instance will not work:

```
C:\Users\Aditya>ping 13.232.205.187

Pinging 13.232.205.187 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 13.232.205.187:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

--The End--