

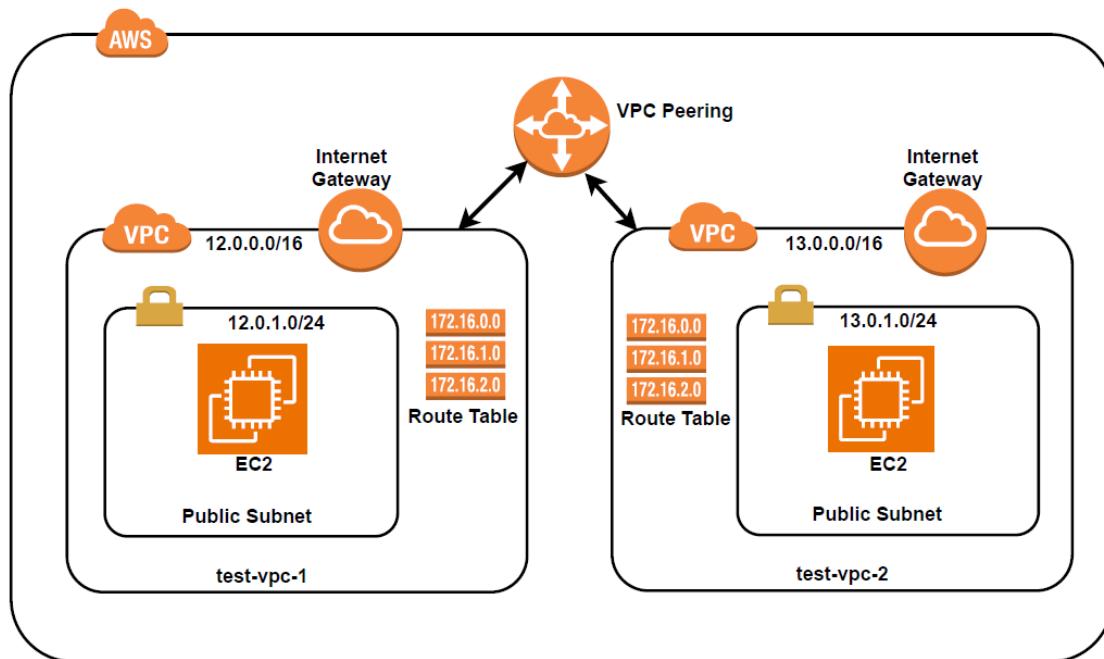
AWS VPC PEERING

- **VPC Peering** is a networking connection between two Virtual Private Clouds (VPCs) that allows you to route traffic between them using private IP addresses.
- Instances in either VPC can communicate with each other as if they are within the same network.
- VPC peering allows for seamless communication across different VPCs, whether they are in the same account, different accounts, or even different regions (inter-region VPC peering).

You can create a VPC peering connection between:

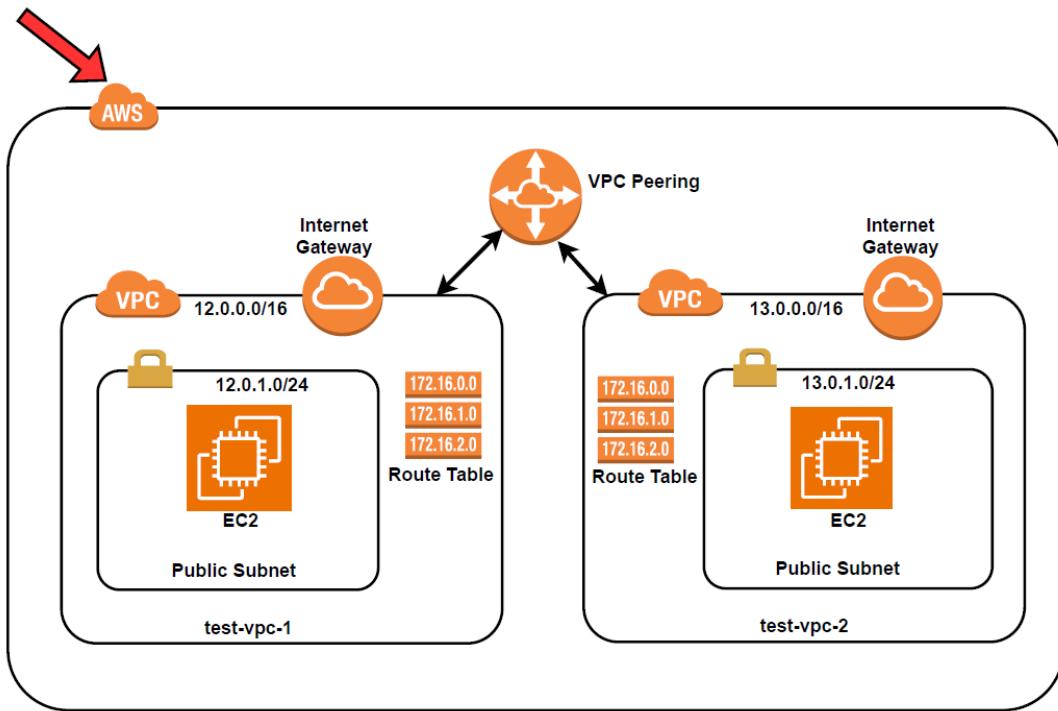
- Your own VPCs
- A VPC in another AWS Account
- A VPC in different AWS Region

In this tutorial, we will walk through the entire process of setting up a VPC peering connection between two VPCs: **test-vpc-1** and **test-vpc-2** that are in the same AWS Account.

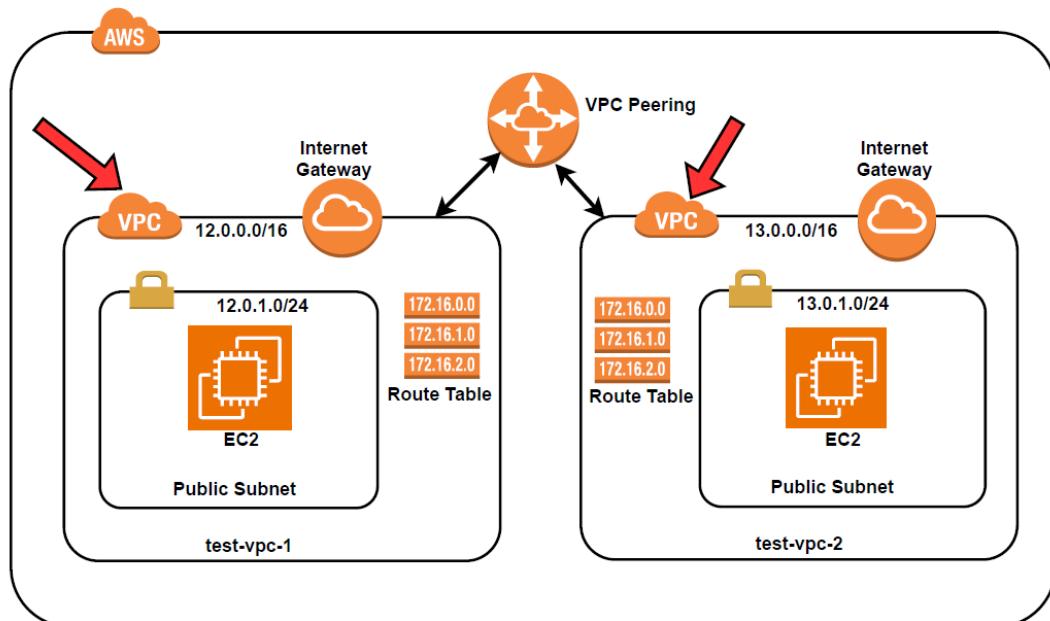


Part 1: Introduction

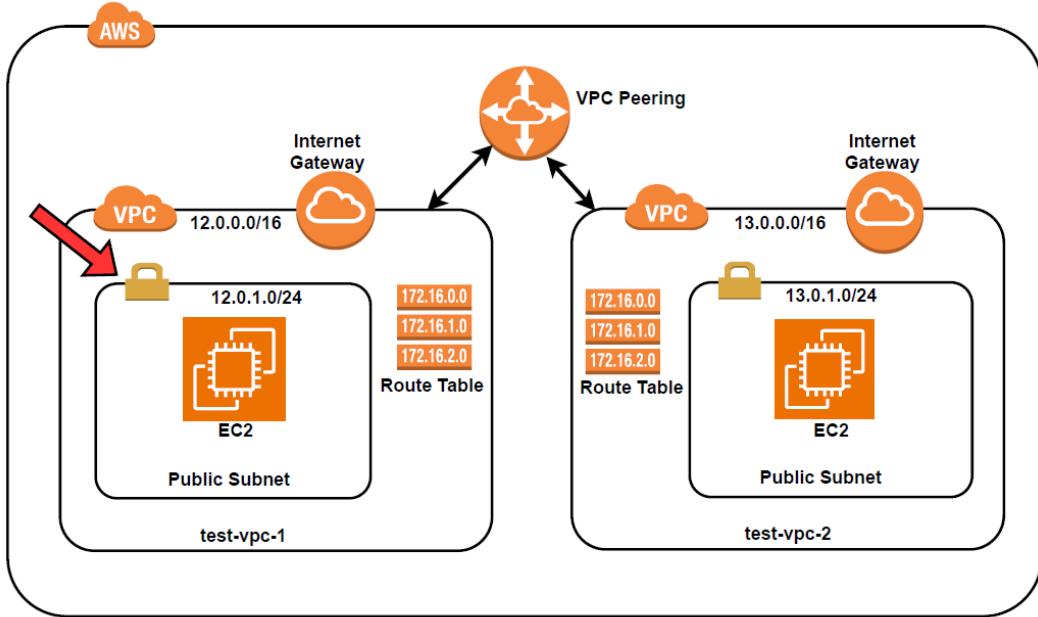
By the end of this session, you will know how to set up your VPC peering between the two VPCs and also you will know the benefits of using VPC peering. Let us start with it.



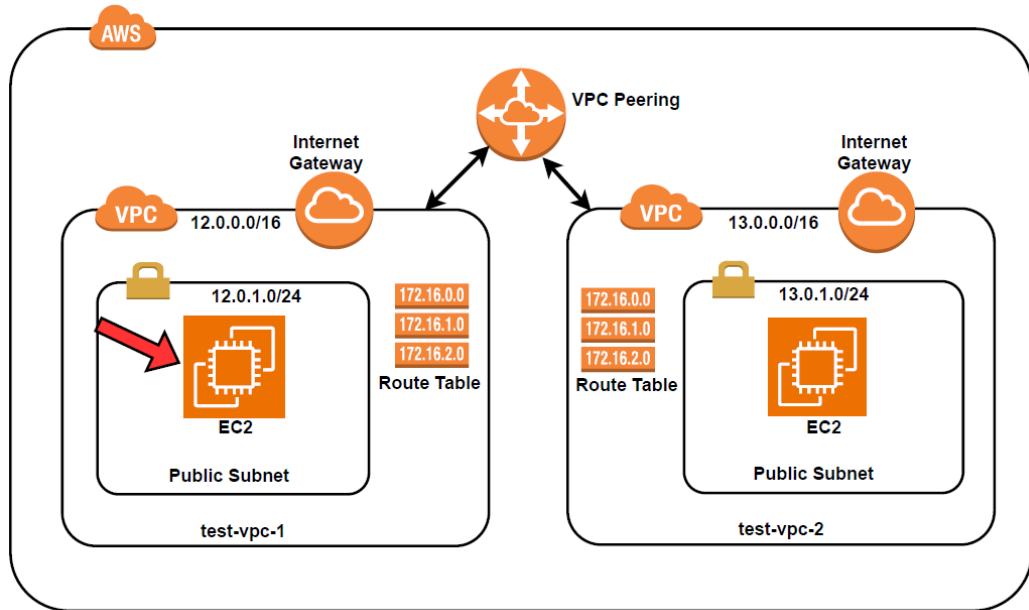
First of all, this is our AWS account on which we are going to set up this whole demo



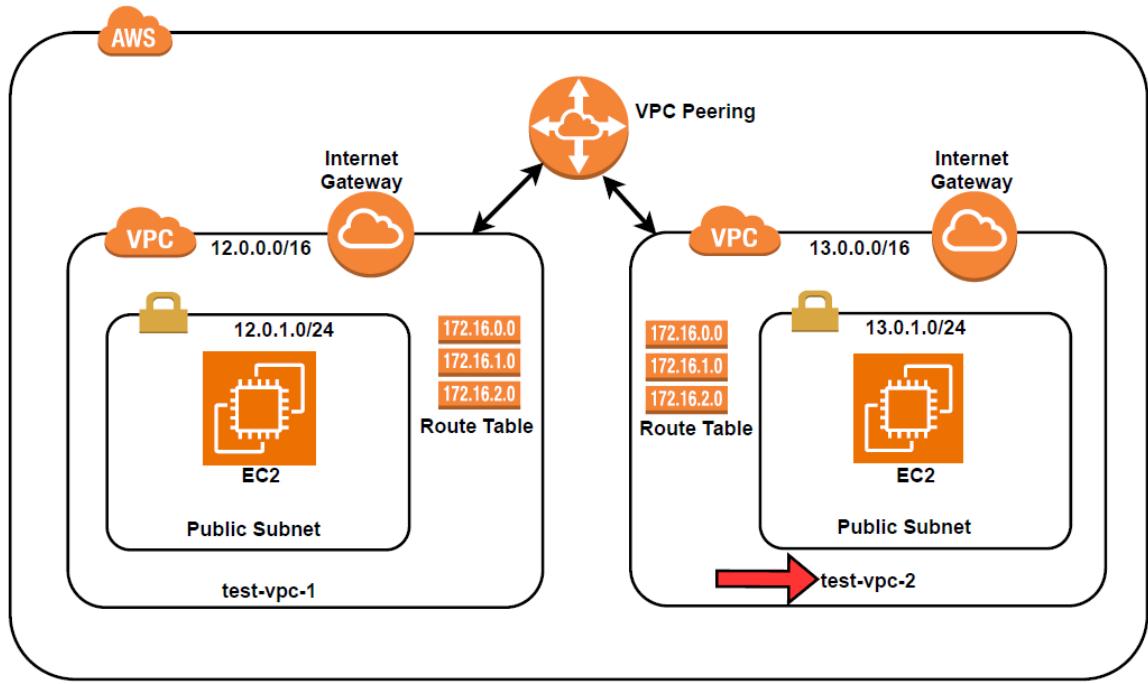
We are going to create two VPCs, on the left-hand side we have **test-vpc-1** and on the right-hand side you will find **test-vpc-2**. These two VPCs are separated from each other.



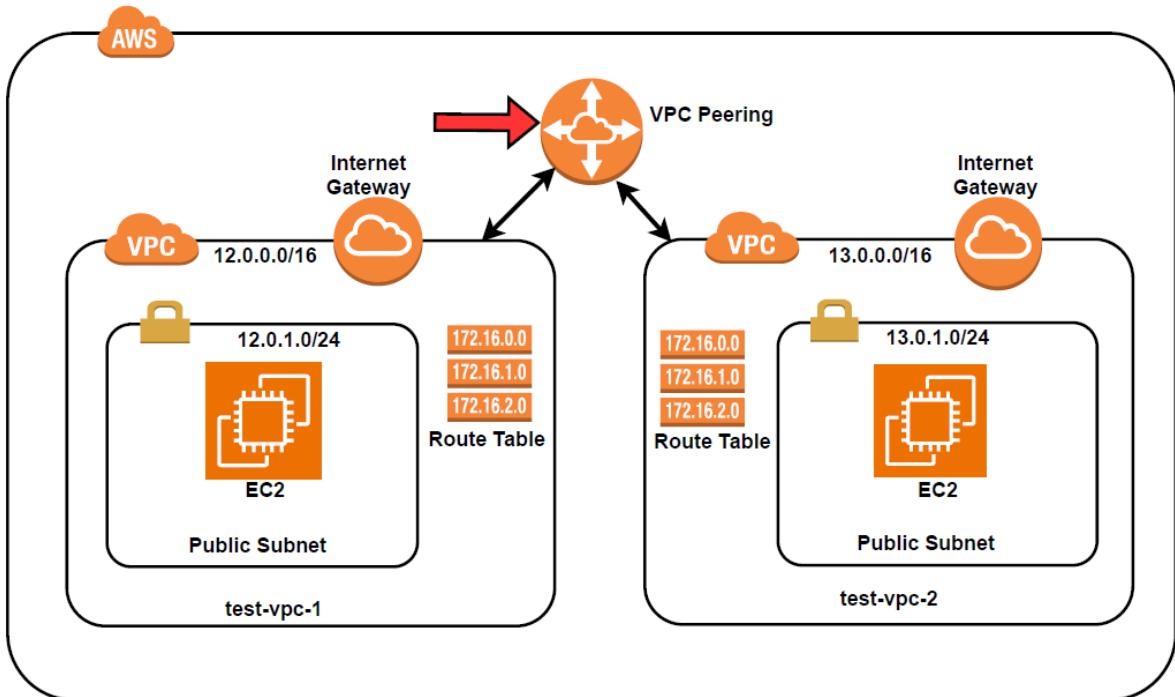
Inside the VPC, we are going to create a **subnet**.



Inside the subnet, we are going to provision **Amazon Linux EC2 instance**.



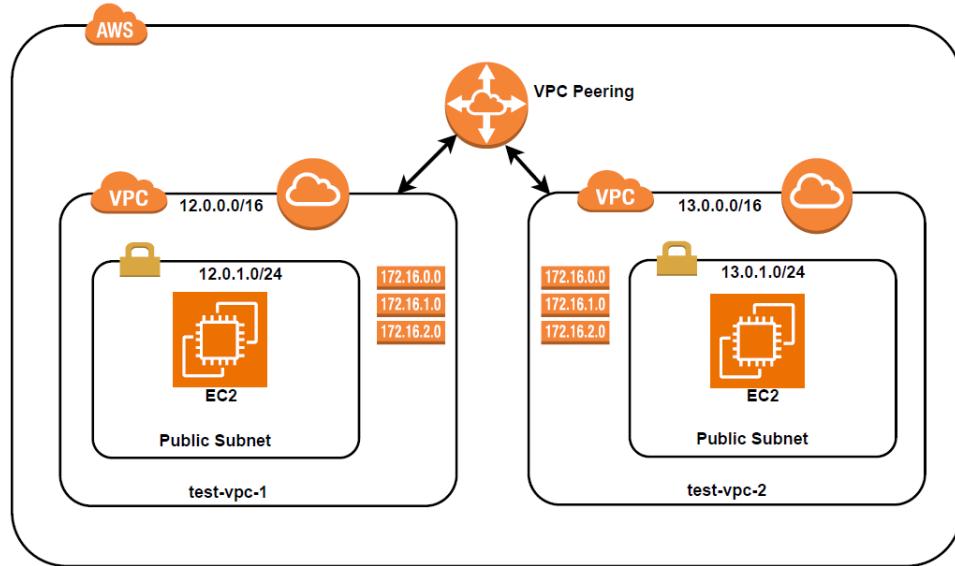
Similarly, on the other VPC (**test-vpc-2**), we are also going to create our subnet and EC2 instance.



After that we are going to set up a **VPC peering**. Once we set up the VPC peering, then the EC2 instance from **test-vpc-1** will be able to communicate with the EC2 instance in **test-vpc-2**.

STEPS

Let us start creating our AWS resources. We will follow these steps:



Step 1: Create the AWS VPC

Name: test-vpc-1 in US East (N. Virginia) us-east-1

CIDR: 12.0.0.0/16

Step 2: Create a public subnet with

CIDR 12.0.1.0/24

Name: test-subnet-vpc-1-1a

Step 3: Create an internet gateway and attach to test-vpc-1

Name: test-igw-vpc-1

Step 4: Create a route table and name it test-rt-vpc-1 and attach the public subnet to this Route Table.

Name: test-rt-vpc-1

Subnet Name: test-subnet-vpc-1-1a

Step 5: Create the AWS VPC

Name: test-vpc-2 in US East (N. Virginia) us-east-1

CIDR: 13.0.0.0/16

Step 6: Create a public subnet with

CIDR 13.0.1.0/24

Name: test-subnet-vpc-2-1a

Step 7: Create an internet gateway and attach to test-vpc-2

Name: test-igw-vpc-2

Step 8: Create a route table and name it test-rt-vpc-2 and attach the public subnet to this RouteTable.

Name: test-rt-vpc-2

Subnet Name: test-subnet-vpc-2-1a

Step 9: Create the VPC peering

Name: peering-connection-between-vpc1-and-vpc2

Step 10: Accept the Peering connection in both VPCs

Step 11: Update the route tables in both VPCs to allow traffic to the other VPC

- In test-rt-vpc-1 route table, add a route for test-vpc-2's CIDR block to the peering-connection-between-vpc1-and-vpc2 connection.
test-vpc-2 CIDR: 13.0.0.0/16
- In test-rt-vpc-2 route table, add a route for test-vpc-1's CIDR block to the peering-connection-between-vpc1-and-vpc2 connection
test-vpc-1 CIDR: 12.0.0.0/16

Step 12: Launch an EC2 instance in both VPC's public subnets

Launch two Ubuntu EC2 Instances

Name First EC2: vpc-1-ec2-instance

Name Second EC2: vpc-2-ec2-instance

SSH connect to the EC2 instances

Step 13: Test the connectivity between the instances using the curl command.

You have successfully set up VPC peering between two VPCs and tested connectivity between instances in the public subnets of each VPC.

IMPLEMENTATION

Part 2: Creating test-vpc-1

Let us start by creating the VPC, test-vpc-1. Go to AWS Management console and search for "VPC"

The screenshot shows the AWS search interface with the query "VPD" entered in the search bar. The results are categorized into Services, Features, and Resources.

- Services:**
 - VPC: Isolated Cloud Resources
 - AWS Firewall Manager: Central management of firewall rules
 - Detective: Investigate and Analyze potential security issues
- Features:**
 - Dashboard: VPC feature
 - Route 53 VPCs: Route 53 feature
 - VPC Reachability Analyzer: VPC feature
- Resources:** / for a focused search
 - Introducing resource search: Enable to show cross-region resources for your account in search results. Takes less than 5 minutes to set up.
Go to Resource Explorer

At the bottom left, there's a feedback section: "Were these results helpful?" with "Yes" and "No" buttons. At the bottom right, there are links for "CloudShell", "Feedback", and copyright information: "© 2025, Amazon Web Services, Inc. or its affiliates." and "Privacy Terms Cookie preferences".

Click on “VPC” under services

The screenshot shows the VPC dashboard. On the left, there's a sidebar with various VPC-related options like EC2 Global View, Virtual private cloud, Security, PrivateLink and Lattice, and more. An orange arrow points to the "Your VPCs" link under the "Virtual private cloud" section. The main area displays "Resources by Region" with sections for VPCs, Subnets, Route Tables, Internet Gateways, Egress-only Internet Gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, and Route servers. Each section has a "See all regions" button. To the right, there are sections for Service Health, Settings, Additional Information, and AWS Network Manager. At the bottom, there are links for "CloudShell", "Feedback", and copyright information: "© 2025, Amazon Web Services, Inc. or its affiliates." and "Privacy Terms Cookie preferences".

Click on “Your VPCs”

The screenshot shows the AWS VPC dashboard with the following details:

- VPC dashboard** sidebar with sections: EC2 Global View, Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, Route servers), Security (Network ACLs, Security groups), PrivateLink and Lattice (Endpoints, Endpoint services, Service networks), and CloudShell, Feedback.
- Your VPCs (3) Info** table:

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR	DHCP of...
-	vpc-0128e9209eaef1c37	Available	Off	172.31.0.0/16	-	dopt-06
Dev-VPC	vpc-0c565a10e97a2bb88	Available	Off	10.100.0.0/16	-	dopt-06
test-vpc	vpc-0a178e5d2aecc4790	Available	Off	12.0.0.0/16	-	dopt-06
- Select a VPC above** message.
- Bottom navigation: © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

Click on “Create VPC”

The screenshot shows the "Create VPC" wizard with the following steps completed:

- 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. The "VPC only" radio button is selected.
 - Name tag - optional**: my-vpc-01
 - IPv4 CIDR block** (Info):
 - IPv4 CIDR manual input (radio button selected)
 - IPAM-allocated IPv4 CIDR block
 - IPv4 CIDR**: 10.0.0.0/24
 - IPv6 CIDR block** (Info):
 - No IPv6 CIDR block (radio button selected)
 - IPAM-allocated IPv6 CIDR block
 - Amazon-provided IPv6 CIDR block
 - IPv6 CIDR owned by me
 - Tenancy** (Info): Default
- 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.
 - No tags associated with the resource

We will use “VPC only” as shown above. Then we will call our VPC “**test-vpc-1**”

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.
test-vpc-1

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

IPv4 CIDR
10.0.0.0/24

IPv6 CIDR block Info
 No IPv6 CIDR block
 IPAM-allocated IPv6 CIDR block
 Amazon-provided IPv6 CIDR block
 IPv6 CIDR owned by me

Tenancy Info
Default

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
<input type="text" value="Name"/>	<input type="text" value="test-vpc-1"/> X

Add tag Remove tag

You can add 49 more tags

Cancel Preview code Create VPC

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

For the IPv4 CIDR, we will use “**12.0.0.0/16**”

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.
test-vpc-1

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

IPv4 CIDR
12.0.0.0/16

IPv6 CIDR block Info
 No IPv6 CIDR block
 IPAM-allocated IPv6 CIDR block
 Amazon-provided IPv6 CIDR block
 IPv6 CIDR owned by me

Tenancy Info
Default

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
<input type="text" value="Name"/>	<input type="text" value="test-vpc-1"/> X

Add tag Remove tag

You can add 49 more tags

Cancel Preview code Create VPC

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Then click on “**create VPC**”

You successfully created **vpc-0ab72baf7cc45d615 / test-vpc-1**

vpc-0ab72baf7cc45d615 / test-vpc-1

Details Info

VPC ID vpc-0ab72baf7cc45d615	State Available	Block Public Access Off	DNS hostnames Disabled
DNS resolution Enabled	Tenancy default	DHCP option set dopt-067ba7e48365eae16	Main route table rtb-0d57ce1038252a594
Main network ACL acl-038d954a13b72c85	Default VPC No	IPv4 CIDR 12.0.0.0/16	IPv6 pool -
IPv6 CIDR (Network border group) -	Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID 324783324460

Resource map Info

VPC Show details Your AWS virtual network test-vpc-1	Subnets (0) Subnets within this VPC	Route tables (1) Route network traffic to resources rtb-0d57ce1038252a594	Network connections (0) Connections to other networks
---	---	--	---

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

We have created the first VPC called “**test-vpc-1**”, now let us create the second VPC called “**test-vpc-2**”. Click on “**Your VPCs**”

Your VPCs (4) Info

Last updated 3 minutes ago [Actions](#) [Create VPC](#)

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR	DHCP opt...
-	vpc-0128e9209eaef1c37	Available	Off	172.31.0.0/16	-	dopt-06
Dev-VPC	vpc-0c565a10e97a2bb88	Available	Off	10.100.0.0/16	-	dopt-06
test-vpc	vpc-0a178e5d2aecc4790	Available	Off	12.0.0.0/16	-	dopt-06
test-vpc-1	vpc-0ab72baf7cc45d615	Available	Off	12.0.0.0/16	-	dopt-06

Select a VPC above

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Part 3: Creating test-vpc-2

Click on “create VPC”

The screenshot shows the 'Create VPC' settings page. Under 'VPC settings', the 'Resources to create' dropdown is set to 'VPC only'. The 'Name tag - optional' field contains 'my-vpc-01', which is highlighted with an orange arrow. Other fields include 'IPv4 CIDR block' (set to 'IPv4 CIDR manual input' with '10.0.0.0/24'), 'IPv6 CIDR block' (set to 'No IPv6 CIDR block'), and 'Tenancy' (set to 'Default'). A 'Tags' section is present but empty. The bottom navigation bar includes CloudShell, Feedback, and links to 2025 AWS terms.

We will call this second VPC “test-vpc-2”

The screenshot shows the 'Create VPC' settings page again. The 'Name tag - optional' field now contains 'test-vpc-2', highlighted with an orange arrow. The other settings remain the same: 'IPv4 CIDR block' (manual input), 'IPv6 CIDR block' (no block), and 'Tenancy' (Default). The 'Tags' section is shown with one tag: 'Name: test-vpc-2'. The bottom navigation bar includes CloudShell, Feedback, and links to 2025 AWS terms.

For this VPC, we will use the IPv4 CIDR “**13.0.0.0/16**”

The screenshot shows the 'Create VPC' wizard in the AWS VPC service. The configuration includes:

- Name tag - optional:** test-vpc-2
- IPv4 CIDR block:** 13.0.0.0/16
- IPv6 CIDR block:** No IPv6 CIDR block selected.
- Tenancy:** Default
- Tags:** A single tag named 'test' with value 'test-vpc-2' is added.

A red arrow points from the bottom right towards the 'Create VPC' button.

Click on “create VPC”

The screenshot shows the VPC dashboard with the following details for the newly created VPC:

- VPC ID:** [vpc-0bc31d8b7fc84ec75](#)
- State:** Available
- DNS resolution:** Enabled
- Main network ACL:** [acl-0a027f5f53ce79a2d](#)
- IPv6 CIDR:** Network border group
- Network Address Usage metrics:** Disabled
- Block Public Access:** Off
- DHCP option set:** [dopt-067ba7e48365eae16](#)
- IPv4 CIDR:** 13.0.0.0/16
- Route 53 Resolver DNS Firewall rule groups:** –
- DNS hostnames:** Disabled
- Main route table:** [rtb-0d518b476ecf05b28](#)
- IPv6 pool:** –
- Owner ID:** 324783324460

A red arrow points from the left sidebar to the 'Your VPCs' link.

We have created the second VPC. Click on “**Your VPCs**”

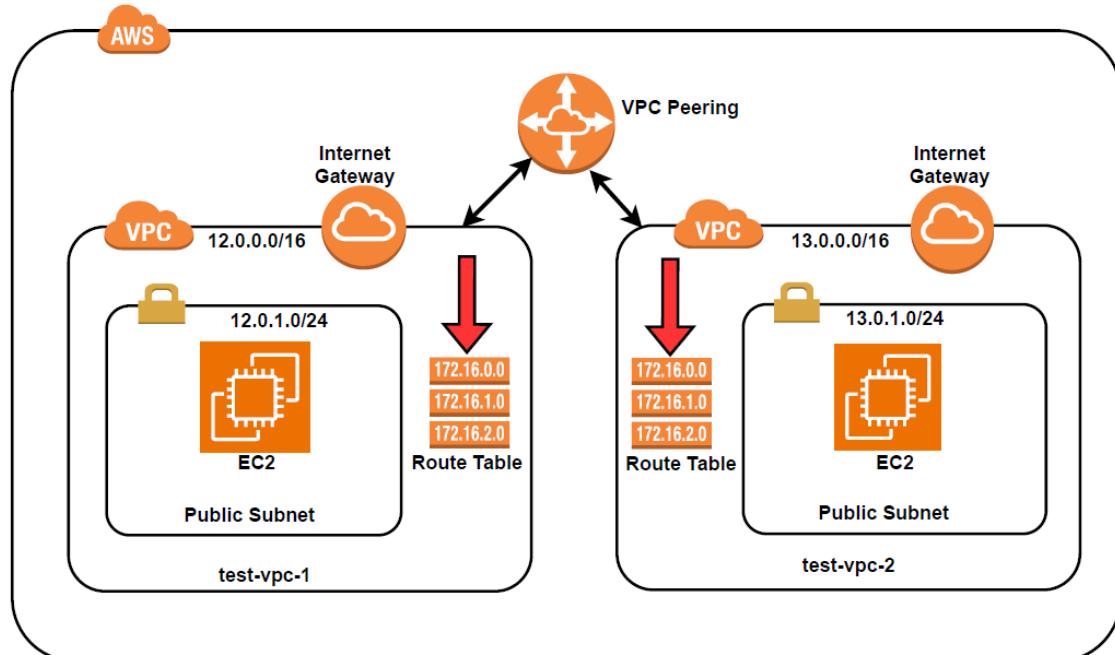
Screenshot of the AWS VPC dashboard showing the list of created VPCs:

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR	DHCP opt...
-	vpc-0128e9209eaf1c37	Available	Off	172.31.0.0/16	-	dopt-06
Dev-VPC	vpc-0565a10e97a2bb88	Available	Off	10.100.0.0/16	-	dopt-06
test-vpc	vpc-0a178e5d2aec4790	Available	Off	12.0.0.0/16	-	dopt-06
test-vpc-1	vpc-0ab72ba7cc45d615	Available	Off	12.0.0.0/16	-	dopt-06
test-vpc-2	vpc-0bc31d8b7fc04ec75	Available	Off	13.0.0.0/16	-	dopt-06

Select a VPC above

You can see our two VPC for this project: **test-vpc-1** and **test-vpc-2**.

Part 4: Create route tables



We are going to create two Route Tables one for **test-vpc-1** and the other for **test-vpc-2**, so that it can route the traffic to our particular subnet.

Let us go back to the VPC dashboard on our AWS Management Console

The screenshot shows the AWS VPC dashboard. In the left sidebar, under 'Virtual private cloud', the 'Route tables' link is highlighted with an orange arrow. The main area displays a table titled 'Your VPCs (5) Info' with columns for Name, VPC ID, State, Block Public..., IPv4 CIDR, IPv6 CIDR, and DHCP options. A success message at the top indicates a new VPC was created. The table includes rows for 'vpc-0128e9209eaef1c37' (Dev-VPN), 'vpc-0c565a10e97a2bb88' (test-VPN), 'vpc-0a178e5d2aecc4790' (test-VPN), 'vpc-0ab72baaf7cc45d615' (test-VPN), and 'vpc-0bc31d8b7fc84ec75' (test-VPN).

Click on “Route Tables”

The screenshot shows the AWS Route tables dashboard. In the left sidebar, the 'Route tables' link is selected. The main area displays a table titled 'Route tables (6) Info' with columns for Name, Route table ID, Explicit subnet associations, Edge associations, Main, VPC, Own..., and a row number column. The table lists six route tables: 'rtb-094350c8a924d963f', 'rtb-091f8c3d6b19ab5ff', 'Dev-VPN-Public-RouteTable' (with 2 subnets), 'test-rt-public' (with 2 subnets), 'rtb-0da6cd7b4e4f2f903', and 'Dev-VPN-Private-RouteTable' (with 6 subnets). The 'Create route table' button is highlighted with an orange arrow.

Click on “create Route Table”

Screenshot of the AWS VPC Route Tables creation page. The 'Name' field is highlighted with a blue border and contains the value 'my-route-table-01'. An orange arrow points from the text above to this field.

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.

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.
No tags associated with the resource.

[Add new tag](#)
You can add 50 more tags.

[Cancel](#) [Create route table](#)

Give the Route Table a name, we will call it “**test-rt-vpc-1**”.

Screenshot of the AWS VPC Route Tables creation page. The 'Name' field is highlighted with a blue border and contains the value 'test-rt-vpc-1'. An orange arrow points from the text above to this field.

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.

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
<input type="text" value="Name"/>	<input type="text" value="test-rt-vpc-1"/> Remove

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

[Cancel](#) [Create route table](#)

Click on the drop down and select our first VPC “**test-vpc-1**”

Screenshot of the AWS VPC Route Tables creation page. The 'VPC' dropdown menu is open, showing the option 'vpc-0ab72baf7cc45d615 (test-vpc-1)'. An orange arrow points from the text above to this dropdown.

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.

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
<input type="text" value="Name"/>	<input type="text" value="test-rt-vpc-1"/> Remove

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

[Cancel](#) [Create route table](#)

Click on “create route table”

The screenshot shows the AWS VPC Route Tables page. A success message at the top states: "Route table rtb-007c3145a4fcbe436 | test-rt-vpc-1 was created successfully." The main section displays the details of the route table "rtb-007c3145a4fcbe436 / test-rt-vpc-1". The "Details" tab is selected, showing the Route table ID (rtb-007c3145a4fcbe436), Main status (No), Owner ID (vpc-0ab72baf7cc45d615 | test-vpc-1), and Explicit subnet associations and Edge associations both listed as "-". Below this, tabs for Subnet associations, Edge associations, Route propagation, and Tags are visible. A table titled "Routes (1)" lists one route: Destination 12.0.0.0/16, Target local, Status Active, Propagated No, and Route Origin Create Route Table. At the bottom right of the table, there is a "Edit routes" button. On the left sidebar, under "Virtual private cloud", the "Route tables" link is highlighted with an orange arrow.

We have created the route table for “test-vpc-1”. Now, let us create a Route Table for “test-vpc-2”.
Click on “Route Tables”

The screenshot shows the AWS VPC Route Tables page with a list of existing route tables. The table has columns for Name, Route table ID, Explicit subnet associations, Edge associations, Main, VPC, and Own... (Owner). The route tables listed are: rtb-094350c8a924d963f, rtb-091f8c3d6b19ab5ff, Dev-VPC-Public-RouteTable (rtb-0869d0039da76f221), test-rt-public (rtb-0fbab8bbc017237268), rtb-0da6cd7b4e4f2f903, Dev-VPC-Private-RouteTable (rtb-02fe1a3bb09f895ad), rtb-0d518b476ecf05b28, test-rt-vpc-1 (rtb-007c3145a4fcbe436), and rtb-0d57ce1038252a594. An orange arrow points to the "Create route table" button at the top right of the table. The left sidebar shows the "Route tables" link under "Virtual private cloud" highlighted with an orange arrow.

Click on “Create Route Table”

AWS VPC > Route tables > Create route table

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.

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.
No tags associated with the resource.

You can add 50 more tags.

[Cancel](#) [Create route table](#)

We will call our Route Table “**test-rt-vpc-2**”

AWS VPC > Route tables > Create route table

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.

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
<input type="text" value="Name"/>	<input type="text" value="test-rt-vpc-2"/> Remove

You can add 49 more tags.

[Cancel](#) [Create route table](#)

Click on the drop down and select our second VPC, “**test-vpc-2**”

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.

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**

Add new tag

You can add 49 more tags.

Create route table

Click on “create route table”

Route table rtb-04a6f429ead18b25c | test-rt-vpc-2 was created successfully.

rtb-04a6f429ead18b25c / test-rt-vpc-2

Details Info

Route table ID rtb-04a6f429ead18b25c	Main <input checked="" type="checkbox"/> No	Explicit subnet associations -	Edge associations -
VPC vpc-0bc31d8b7fc84ec75 test-vpc-2	Owner ID 324783324460		

Routes 1 **Edit routes**

Destination	Target	Status	Propagated	Route Origin
13.0.0.0/16	local	Active	No	Create Route Table

We have created the Route Table for “test-vpc-2”. Click on “Your VPCs”

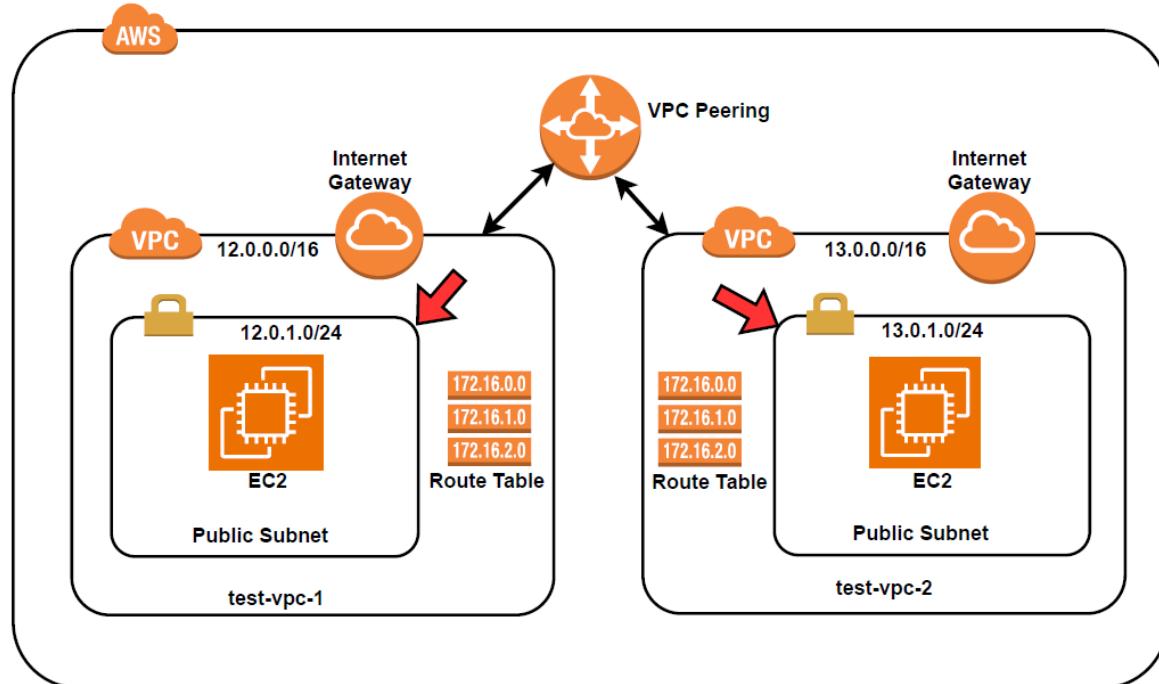
Screenshot of the AWS VPC dashboard showing the list of VPCs. The table displays columns for Name, VPC ID, State, Block Public..., IPv4 CIDR, IPv6 CIDR, and DHCP options.

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR	DHCP options
-	vpc-0128e9209eaef1c37	Available	Off	172.31.0.0/16	-	dopt-06
Dev-VPC	vpc-0c565a10e97a2bb88	Available	Off	10.100.0.0/16	-	dopt-06
test-vpc	vpc-0a178e5d2aecc4790	Available	Off	12.0.0.0/16	-	dopt-06
test-vpc-1	vpc-0ab72ba7cc45d615	Available	Off	12.0.0.0/16	-	dopt-06
test-vpc-2	vpc-0bc31d8b7fc84ec75	Available	Off	13.0.0.0/16	-	dopt-06

Select a VPC above

You can see the two Route Tables we have created.

Part 5: Create Subnet



After creating the route tables, the next thing we have to do is to create the subnets. To do that, go to the VPC dashboard on AWS Management Console

aws Search [Alt+S] United States (N. Virginia) Account ID: 3247-8332-4460 sidney

VPC dashboard < Your VPCs

EC2 Global View Filter by VPC

Virtual private cloud

- Your VPCs **Subnets** (arrow)
- Route tables
- Internet gateways
- Egress-only internet gateways
- Carrier gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- NAT gateways
- Peering connections
- Route servers [New](#)

Security

- Network ACLs
- Security groups

PrivateLink and Lattice

- Getting started [Updated](#)
- Endpoints [Updated](#)
- Endpoint services
- Service networks [Updated](#)
- Lattice services

CloudShell Feedback

Your VPCs (5) Info Last updated 36 minutes ago Actions Create VPC

Name	VPC ID	State	Block Public Access	IPv4 CIDR	IPv6 CIDR	DHCP options
-	vpc-0128e9209eaef1c37	Available	Off	172.31.0.0/16	-	dopt-06
Dev-VPC	vpc-0c565a10e97a2bb88	Available	Off	10.100.0.0/16	-	dopt-06
test-vpc	vpc-0a178e5d2aec4790	Available	Off	12.0.0.0/16	-	dopt-06
test-vpc-1	vpc-0ab72ba7ffcc45d615	Available	Off	12.0.0.0/16	-	dopt-06
test-vpc-2	vpc-0bc31d8b7fc84ec75	Available	Off	13.0.0.0/16	-	dopt-06

Select a VPC above

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click on “Subnets”

aws Search [Alt+S] United States (N. Virginia) Account ID: 3247-8332-4460 sidney

VPC dashboard < Subnets

EC2 Global View Filter by VPC

Virtual private cloud

- Your VPCs **Subnets** (arrow)
- Route tables
- Internet gateways
- Egress-only internet gateways
- Carrier gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- NAT gateways
- Peering connections
- Route servers [New](#)

Security

- Network ACLs
- Security groups

PrivateLink and Lattice

- Getting started [Updated](#)
- Endpoints [Updated](#)
- Endpoint services
- Service networks [Updated](#)
- Lattice services

CloudShell Feedback

Subnets (16) Info Last updated 37 minutes ago Actions Create subnet

Name	Subnet ID	State	VPC	Block Public Access	IPv4 CIDR
-	subnet-0920122fe5af20950	Available	vpc-0128e9209eaef1c37	Off	172.31.80.0/20
Dev-private-subnet-1a	subnet-03a340dd0d8b6fd87	Available	vpc-0c565a10e97a2bb88 Dev-VPC	Off	10.100.80.0/21
Dev-app-subnet-1b	subnet-0ef40a6bc568a5a9d	Available	vpc-0c565a10e97a2bb88 Dev-VPC	Off	10.100.160.0/21
test-public-subnet-1-1b	subnet-013af1016c64ee796	Available	vpc-0a178e5d2aec4790 test-vpc	Off	12.0.2.0/24
Dev-public-subnet-1b	subnet-0f7d7a0105ef33410	Available	vpc-0c565a10e97a2bb88 Dev-VPC	Off	10.100.16.0/21
Dev-public-subnet-1a	subnet-02a2a7b7e24f85610	Available	vpc-0c565a10e97a2bb88 Dev-VPC	Off	10.100.8.0/21
Dev-data-subnet-1a	subnet-08c5ed4636f9b19a0	Available	vpc-0c565a10e97a2bb88 Dev-VPC	Off	10.100.168.0/21
-	subnet-0e978029c28591323	Available	vpc-0128e9209eaef1c37	Off	172.31.0.0/20
-	subnet-0c21087ab9e69a290	Available	vpc-0128e9209eaef1c37	Off	172.31.64.0/20
-	subnet-0410d8e91c199a290	Available	vpc-0128e9209eaef1c37	Off	172.31.32.0/20
test-public-subnet-1-1a	subnet-04e5dcf10023b20f3	Available	vpc-0a178e5d2aec4790 test-vpc	Off	12.0.1.0/24
-	subnet-09e0df53f5814abf3	Available	vpc-0128e9209eaef1c37	Off	172.31.16.0/20
-	subnet-0abe0cd1e821b6aa9	Available	vpc-0128e9209eaef1c37	Off	172.31.48.0/20
Dev-private-subnet-1b	subnet-057df62ee87416af9	Available	vpc-0c565a10e97a2bb88 Dev-VPC	Off	10.100.88.0/21
Dev-data-subnet-1b	subnet-06354dec3e56392d0	Available	vpc-0c565a10e97a2bb88 Dev-VPC	Off	10.100.176.0/21
Dev-app-subnet-1a	subnet-0ff18868ba4edf0f0	Available	vpc-0c565a10e97a2bb88 Dev-VPC	Off	10.100.144.0/21

Select a subnet

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click on “create subnet”

aws Search [Alt+S] United States (N. Virginia) Account ID: 3247-8332-4460 sidney

VPC > Subnets > Create subnet

Create subnet Info

VPC

VPC ID
Create subnets in this VPC.

Select a VPC

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

Select a VPC first to create new subnets.

Add new subnet

Cancel Create subnet

Click on the drop down and select the first VPC “**test-vpc-1**”

aws Search [Alt+S] United States (N. Virginia) Account ID: 3247-8332-4460 sidney

VPC > Subnets > Create subnet

Create subnet Info

VPC

VPC ID
Create subnets in this VPC.

vpc-0ab72ba7cc45d615 (test-vpc-1)

Associated VPC CIDRs

IPv4 CIDRs
12.0.0.0/16

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-subnet-01

The name can be up to 256 characters long.

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

IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
12.0.0.0/16

IPv4 subnet CIDR block
12.0.0.0/20

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Then give the subnet a name, I will call it “**test-subnet-vpc-1-1a**”

Screenshot of the AWS VPC Create Subnet page. The 'Availability Zone' dropdown menu is open, showing 'No preference' as the selected option. An orange arrow points to the dropdown menu.

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.

The name can be up to 256 characters long.

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

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

IPv4 subnet CIDR block

 You can add 49 more tags.

Tags - optional

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="test-subnet-vpc-1-1a"/>

[CloudShell](#) [Feedback](#) © 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Then click on the drop down on availability zone, and select “us-east-1a”

Screenshot of the AWS VPC Create Subnet page. The 'Availability Zone' dropdown menu is open, showing 'United States (N. Virginia) / us-east-1a' as the selected option. An orange arrow points to the dropdown menu.

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.

The name can be up to 256 characters long.

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

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

IPv4 subnet CIDR block

 You can add 49 more tags.

Tags - optional

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="test-subnet-vpc-1-1a"/>

[CloudShell](#) [Feedback](#) © 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

On “IPv4 subnet CIDR block”, enter “12.0.1.0/24”

Screenshot of the AWS VPC Subnet creation page:

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.

The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

IPv4 subnet CIDR block
 256 IPs
[Add new subnet](#)

Tags - optional
Key Value - optional
[Add new tag](#) You can add 49 more tags.
[Remove](#)

[Create subnet](#)

Click on “create subnet”

Screenshot of the AWS VPC Subnets list page:

VPC dashboard < **Subnets** Info

You have successfully created 1 subnet: subnet-094aa23195f6eeaba

Subnets (1) Info

[Clear filters](#)

Name	Subnet ID	State	VPC	Block Public Access	IPv4 CIDR
test-subnet-vpc-1-1a	subnet-094aa23195f6eeaba	Available	vpc-0ab72baf7cc45d615 test-vpc-1	Off	12.0.1.0/24

[Actions](#) [Create subnet](#)

Select a subnet

The first subnet for **test-vpc-1** has been created. Let us create the subnet for **test-vpc-2**. Click on “create subnet”

Create subnet Info

VPC

VPC ID
Create subnets in this VPC.

Select a VPC first to create new subnets.

Add new subnet

Cancel **Create subnet**

Select our VPC “**test-vpc-2**”

Create subnet Info

VPC

VPC ID
Create subnets in this VPC.

vpc-0bc31d8b7fc84ec75 (test-vpc-2)

Associated VPC CIDRs

IPv4 CIDRs
13.0.0.0/16

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-subnet-01
The name can be up to 256 characters long.

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

IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
13.0.0.0/16

IPv4 subnet CIDR block
13.0.0.0/20

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Let us give the VPC a name, I will call it “**test-subnet-vpc-2-1a**”

Screenshot of the AWS VPC Create Subnet page. The 'Availability Zone' dropdown menu is open, showing 'No preference' as the selected option. An orange arrow points to the dropdown menu.

Then on “availability zone”, click on the drop down and select “us-east-1a”

Screenshot of the AWS VPC Create Subnet page. The 'Availability Zone' dropdown menu is now set to 'United States (N. Virginia) / us-east-1a'. An orange arrow points to the dropdown menu.

On “IPv4 subnet CIDR block”, enter “13.0.1.0/24”

Screenshot of the AWS VPC Subnet creation interface:

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.
test-subnet-vpc-2-1a
The name can be up to 256 characters long.

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

IPv4 VPC CIDR block **Info**
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
13.0.0.0/16

IPv4 subnet CIDR block
13.0.1.0/24
256 IPs

Tags - optional
Key Value - optional
Name test-subnet-vpc-2-1a
Add new tag You can add 49 more tags.
Remove

Add new subnet

Cancel Create subnet

Click on “Create subnet”

Screenshot of the AWS VPC Subnets list page:

You have successfully created 1 subnet: subnet-09dfeeb15fc0c45ac

Subnets (1) Info

Last updated less than a minute ago Actions Create subnet

Name	Subnet ID	State	VPC	Block Public Access	IPv4 CIDR
test-subnet-vpc-2-1a	subnet-09dfeeb15fc0c45ac	Available	vpc-0bc31d8b7fc84ec75 test-vpc-2	Off	13.0.1.0/24

Select a subnet

The subnet has been created. Click on “Subnets”

Screenshot of the AWS VPC Subnets page showing a list of 18 subnets. The table includes columns for Name, Subnet ID, State, VPC, Block Public Access, and IPv4 CIDR.

Name	Subnet ID	State	VPC	Block Public Access	IPv4 CIDR
Dev-app-subnet-1b	subnet-0ef40a6bc568a5a9d	Available	vpc-0c565a10e97a2bb88 Dev-VPC	Off	10.100.16.0/21
test-public-subnet-1-1b	subnet-013fa1016c64ee796	Available	vpc-0a178e5d2aec4790 test-vpc	Off	12.0.2.0/24
Dev-public-subnet-1b	subnet-0f7d7a0103ef33410	Available	vpc-0c565a10e97a2bb88 Dev-VPC	Off	10.100.16.0/21
Dev-public-subnet-1a	subnet-02a2a7b7e24fb5610	Available	vpc-0c565a10e97a2bb88 Dev-VPC	Off	10.100.8.0/21
Dev-data-subnet-1a	subnet-08c5ed4636f9b19a0	Available	vpc-0c565a10e97a2bb88 Dev-VPC	Off	10.100.168.0/21
-	subnet-0e978029c28591323	Available	vpc-0128e9209eaef1c37	Off	172.31.0.0/20
-	subnet-0c21087ab9e69a290	Available	vpc-0128e9209eaef1c37	Off	172.31.64.0/20
-	subnet-0410d8e91c199d290	Available	vpc-0128e9209eaef1c37	Off	172.31.32.0/20
test-public-subnet-1-1a	subnet-04e5dcf10023b20f3	Available	vpc-0a178e5d2aec4790 test-vpc	Off	12.0.1.0/24
-	subnet-09e0df53f5814abf3	Available	vpc-0128e9209eaef1c37	Off	172.31.16.0/20
-	subnet-0abe0cd1e821b6aa9	Available	vpc-0128e9209eaef1c37	Off	172.31.48.0/20
Dev-private-subnet-1b	subnet-057d62ee87416af9	Available	vpc-0c565a10e97a2bb88 Dev-VPC	Off	10.100.88.0/21
Dev-data-subnet-1b	subnet-06354eec3e56392d0	Available	vpc-0c565a10e97a2bb88 Dev-VPC	Off	10.100.176.0/21
Dev-app-subnet-1a	subnet-0f1f8868ba4ed0f0	Available	vpc-0c565a10e97a2bb88 Dev-VPC	Off	10.100.144.0/21
test-subnet-vpc-1-1a	subnet-094aa23195f6eeaba	Available	vpc-0ab72ba7cc45d615 test-vpc-1	Off	12.0.1.0/24
test-subnet-vpc-2-1a	subnet-09dffeb5fc0c45ac	Available	vpc-0bc31d8b7fc84ec75 test-vpc-2	Off	13.0.1.0/24

You can see the two subnets we have created.

The next thing we have to do is to associate our route tables with our subnets. Go to the Route tables

Screenshot of the AWS VPC Route tables page showing a list of 10 route tables. The table includes columns for Name, Route table ID, Explicit subnet associations, Edge associations, Main, and VPC.

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC
-	rtb-094350c8a924d963f	-	-	Yes	vpc-0128e9209eaef1c37
-	rtb-091f8c3d6b19ab5ff	-	-	Yes	vpc-0c565a10e97a2bb88
Dev-VPC-Public-RouteTable	rtb-0869d0039da76f221	2 subnets	-	No	vpc-0c565a10e97a2bb88 Dev-VPC
test-rt-public	rtb-0fba8bbc017237268	2 subnets	-	No	vpc-0a178e5d2aec4790 test-vpc
-	rtb-0da6cd7b4e4f2f903	-	-	Yes	vpc-0a178e5d2aec4790 test-vpc
Dev-VPC-Private-RouteTable	rtb-02fe1a3bb09f895ad	6 subnets	-	No	vpc-0c565a10e97a2bb88 Dev-VPC
-	rtb-0d518b476ecf05b28	-	-	Yes	vpc-0bc31d8b7fc84ec75 test-vpc
test-rt-vpc-1	rtb-007c3145a4fcbe436	-	-	No	vpc-0ab72ba7cc45d615 test-vpc-1
-	rtb-0d57ce1038252a594	-	-	Yes	vpc-0ab72ba7cc45d615 test-vpc-1
test-rt-vpc-2	rtb-04a6f429ead18b25c	-	-	No	vpc-0bc31d8b7fc84ec75 test-vpc

An orange arrow points to the route table "test-rt-vpc-1".

Click on the first route table “test-rt-vpc-1”

Click on the “Subnet Associations” tab

Click on “Edit Subnet Associations”

AWS VPC > Route tables > rtb-007c3145a4fcbe436 > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1)				
<input type="text"/> Filter subnet associations <input type="checkbox"/> Name Subnet ID IPv4 CIDR IPv6 CIDR Route table ID <input checked="" type="checkbox"/> test-subnet-vpc-1-1a subnet-094aa23195f6eeaba 12.0.1.0/24 - Main (rtb-0d57ce1038252a594)				
Cancel Save associations				

Select the subnet

AWS VPC > Route tables > rtb-007c3145a4fcbe436 > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/1)				
<input type="checkbox"/> Name Subnet ID IPv4 CIDR IPv6 CIDR Route table ID <input checked="" type="checkbox"/> test-subnet-vpc-1-1a subnet-094aa23195f6eeaba 12.0.1.0/24 - Main (rtb-0d57ce1038252a594)				
Selected subnets <input type="text"/> subnet-094aa23195f6eeaba / test-subnet-vpc-1-1a X				
Cancel Save associations				

Click on “Save Associations”

VPC dashboard < Actions ▾

rtb-007c3145a4fcbe436 / test-rt-vpc-1

You have successfully updated subnet associations for rtb-007c3145a4fcbe436 / test-rt-vpc-1.

Details		Explicit subnet associations		Edge associations																						
Route table ID	rtb-007c3145a4fcbe436	Main	No	Owner ID	324783324460																					
VPC	vpc-0ab72ba7cc45d615 test-vpc-1	Explicit subnet associations subnet-094aa23195f6eeaba / test-subnet-vpc-1-1a																								
Routes Subnet associations Edge associations Route propagation Tags																										
Routes (1) <table border="1"> <thead> <tr> <th colspan="5">Routes (1)</th> <th colspan="2" style="text-align: right;">Edit routes</th> </tr> <tr> <th colspan="5"> <input type="text"/> Filter routes Destination Target Status Propagated Route Origin </th> <th colspan="2" style="text-align: right;"> Both 1 All </th> </tr> </thead> <tbody> <tr> <td>12.0.0.0/16</td> <td>local</td> <td>Active</td> <td>No</td> <td></td> <td colspan="2" style="text-align: right;"> Create Route Table </td> </tr> </tbody> </table>						Routes (1)					Edit routes		<input type="text"/> Filter routes Destination Target Status Propagated Route Origin					Both 1 All		12.0.0.0/16	local	Active	No		Create Route Table	
Routes (1)					Edit routes																					
<input type="text"/> Filter routes Destination Target Status Propagated Route Origin					Both 1 All																					
12.0.0.0/16	local	Active	No		Create Route Table																					

Virtual private cloud

- Your VPCs
- Subnets
- Route tables**
- Internet gateways
- Egress-only internet gateways
- Carrier gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- NAT gateways
- Peering connections
- Route servers [New](#)

Security

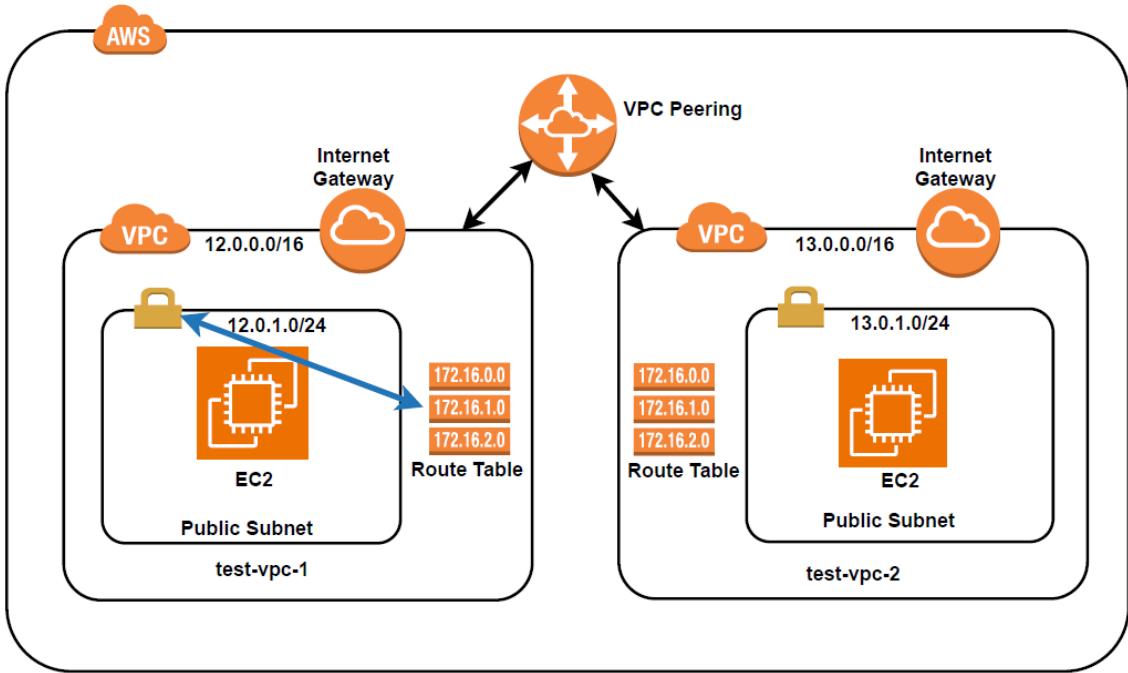
- Network ACLs
- Security groups

PrivateLink and Lattice

- Getting started [Updated](#)
- Endpoints [Updated](#)
- Endpoint services
- Service networks [Updated](#)
- Lattice services

[CloudShell](#) [Feedback](#)

The route table has been associated with the subnet



Our route table has been associated with our subnet in **test-vpc-1**.

Let us associate the second route table to our subnet in test-vpc-2. Click on “**Route Tables**”

The screenshot shows the AWS VPC Route Tables dashboard. The left sidebar navigation includes VPC dashboard, EC2 Global View, Virtual private cloud (with sub-options like Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, Route servers), Security (Network ACLs, Security groups), PrivateLink and Lattice (Endpoints, Endpoint services, Service networks, Lattice services), and CloudShell/Feedback.

The main content area displays a table of Route tables (10) with the following columns: Name, Route table ID, Explicit subnet associ..., Edge associations, Main, VPC, and Own... . An orange arrow points to the "Route table ID" column for the entry "test-rt-vpc-2".

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC	Own...
-	rtb-0d518b476ecf05b28	-	-	Yes	vpc-0bc31d8b7fc84ec75 test-...	324783...
-	rtb-094350c8a924d963f	-	-	Yes	vpc-0128e9209eaef1c37	324783...
-	rtb-091f8c3d6b19ab5ff	-	-	Yes	vpc-0c565a10e97a2bb88 Dev-...	324783...
test-rt-vpc-2	rtb-04a6f429ead18b25c	-	-	No	vpc-0bc31d8b7fc84ec75 test-...	324783...
test-rt-vpc-1	rtb-007c3145a4fcbe436	subnet-094aa23195f6ee...	-	No	vpc-0ab72ba7cc45d615 test-...	324783...
Dev-VPC-Public-RouteTable	rtb-0869d0039da76f221	2 subnets	-	No	vpc-0c565a10e97a2bb88 Dev-...	324783...
test-rt-public	rtb-0ffa8bbc017237268	2 subnets	-	No	vpc-0a178e5d2aecc4790 test-...	324783...
-	rtb-0d57ce1038252a594	-	-	Yes	vpc-0ab72ba7cc45d615 test-...	324783...
-	rtb-0da6cd7b4e4f2f903	-	-	Yes	vpc-0a178e5d2aecc4790 test-...	324783...
Dev-VPC-Private-RouteTable	rtb-02fe1a3bb09f895ad	6 subnets	-	No	vpc-0c565a10e97a2bb88 Dev-...	324783...

Below the table, a section titled "Select a route table" is visible.

Click on the “**Route Table ID**” of the VPC “**test-vpc-2**”, that is “**test-rt-vpc-2**”

VPC dashboard < **rtb-04a6f429ead18b25c / test-rt-vpc-2** Actions ▾

Details Info

Route table ID: rtb-04a6f429ead18b25c
Main: No
Owner ID: vpc-0bc31d8b7fc84ec75 | test-vpc-2

Subnet associations

Edge associations

Routes (1)

Destination	Target	Status	Propagated	Route Origin
13.0.0.0/16	local	Active	No	Create Route Table

Both ▾ Edit routes < 1 > ⚙️

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click on the “Subnet Associations” tab

VPC dashboard < **rtb-04a6f429ead18b25c / test-rt-vpc-2** Actions ▾

Details Info

Route table ID: rtb-04a6f429ead18b25c
Main: No
Owner ID: vpc-0bc31d8b7fc84ec75 | test-vpc-2

Subnet associations

Edge associations

Explicit subnet associations (0)

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

No subnet associations
You do not have any subnet associations.

Edit subnet associations < 1 > ⚙️

Subnets without explicit associations (1)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
test-subnet-vpc-2-1a	subnet-09dffe15fc0c45ac	13.0.1.0/24	-

Edit subnet associations < 1 > ⚙️

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click on “Edit Subnet Associations”

Available subnets (1)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/> test-subnet-vpc-2-1a	subnet-09dffeb15fc0c45ac	13.0.1.0/24	-	Main (rtb-0d518b476ecf05b28)

Cancel **Save associations**

Select the subnet

Available subnets (1/1)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/> test-subnet-vpc-2-1a	subnet-09dffeb15fc0c45ac	13.0.1.0/24	-	Main (rtb-0d518b476ecf05b28)

Selected subnets

subnet-09dffeb15fc0c45ac / test-subnet-vpc-2-1a <input type="button" value="X"/>
--

Cancel **Save associations**

Click on “Save Associations”

You have successfully updated subnet associations for rtb-04a6f429ead18b25c / test-rt-vpc-2.

rtb-04a6f429ead18b25c / test-rt-vpc-2

Details Info

Route table ID <input type="checkbox"/> rtb-04a6f429ead18b25c	Main <input type="checkbox"/> No	Explicit subnet associations subnet-09dffeb15fc0c45ac / test-subnet-vpc-2-1a	Edge associations -
VPC vpc-0bc31d8b7fc84ec75 test-vpc-2	Owner ID <input type="checkbox"/> 324783324460		

Routes (1)

Destination	Target	Status	Propagated	Route Origin
13.0.0.0/16	local	<input checked="" type="checkbox"/> Active	No	Create Route Table

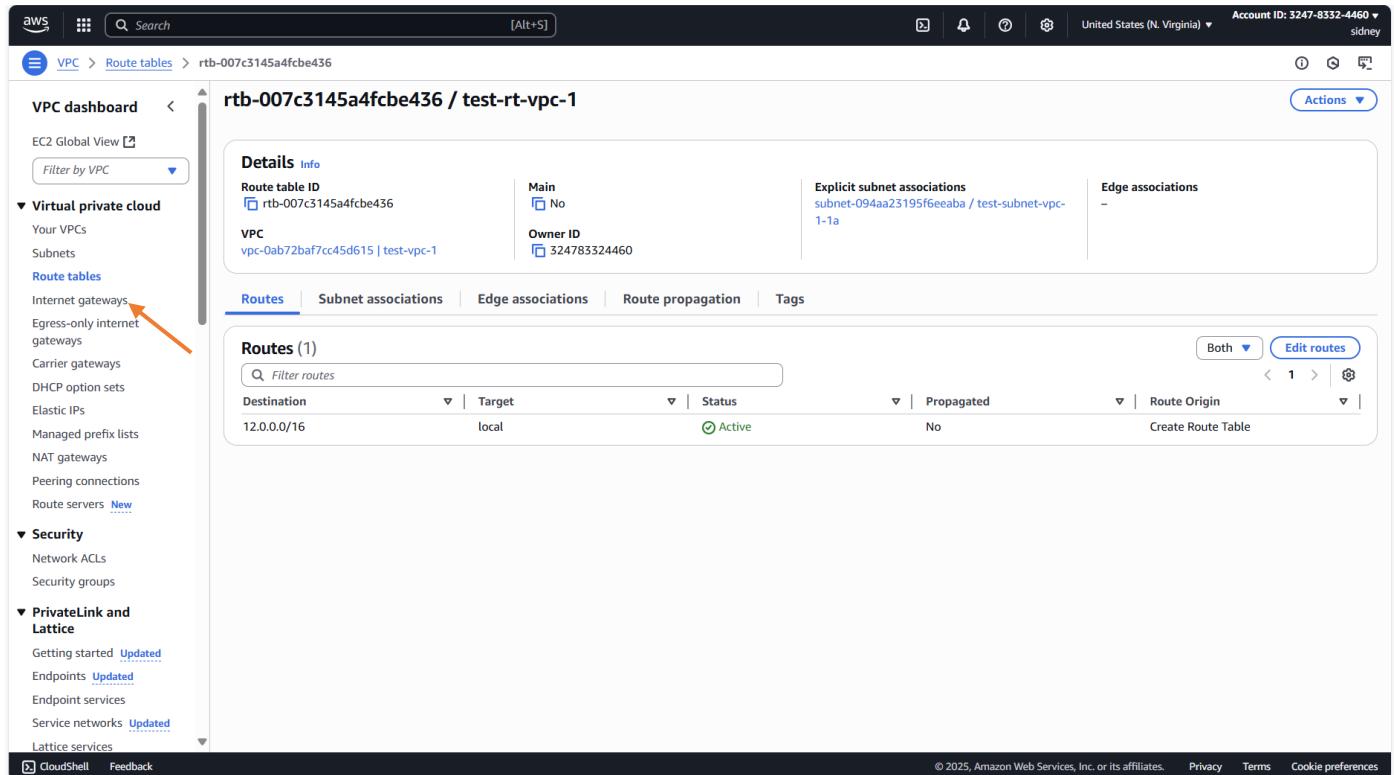
Both

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

We have created two Route tables and associated them with their respective VPCs. But also in the route tables, we need to grant the internet access. To do that we need the **Internet Gateway**.

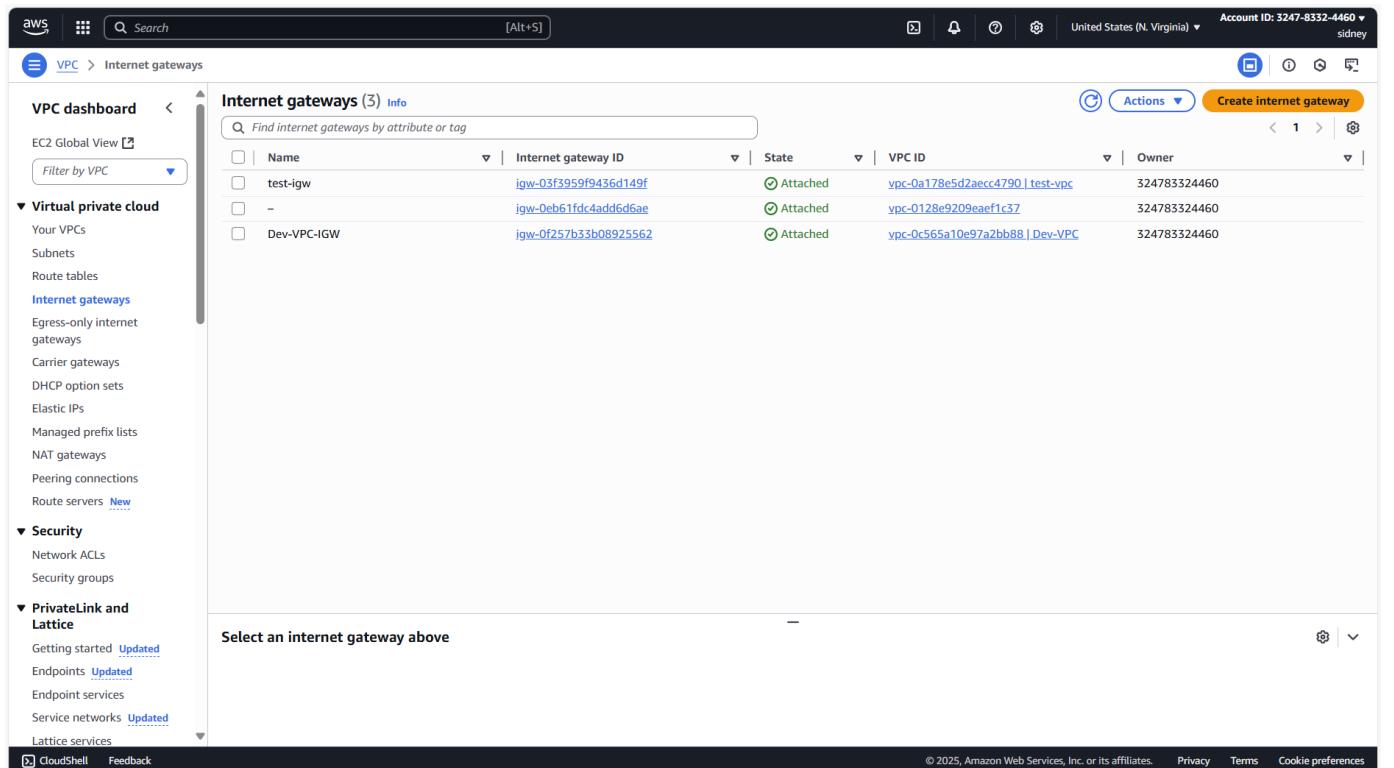
Part 6: Create Internet Gateway

We have to create internet gateway for both VPCs. Let us create the Internet Gateway for the first VPC.



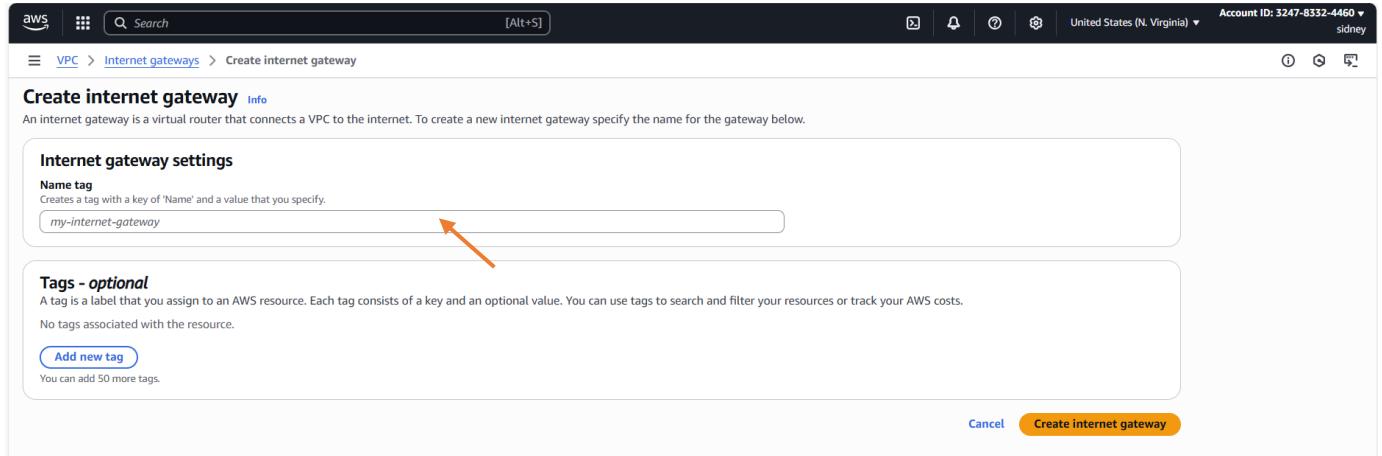
The screenshot shows the AWS VPC Route Tables Details page. The route table ID is rtb-007c3145a4fcbe436, associated with VPC vpc-0ab72ba7cc45d615 | test-vpc-1. The 'Routes' tab is selected, showing one route to destination 12.0.0.0/16 with target local and status Active. The left sidebar has an orange arrow pointing to the 'Internet gateways' link under the 'Route tables' section.

Click on “Internet Gateways”

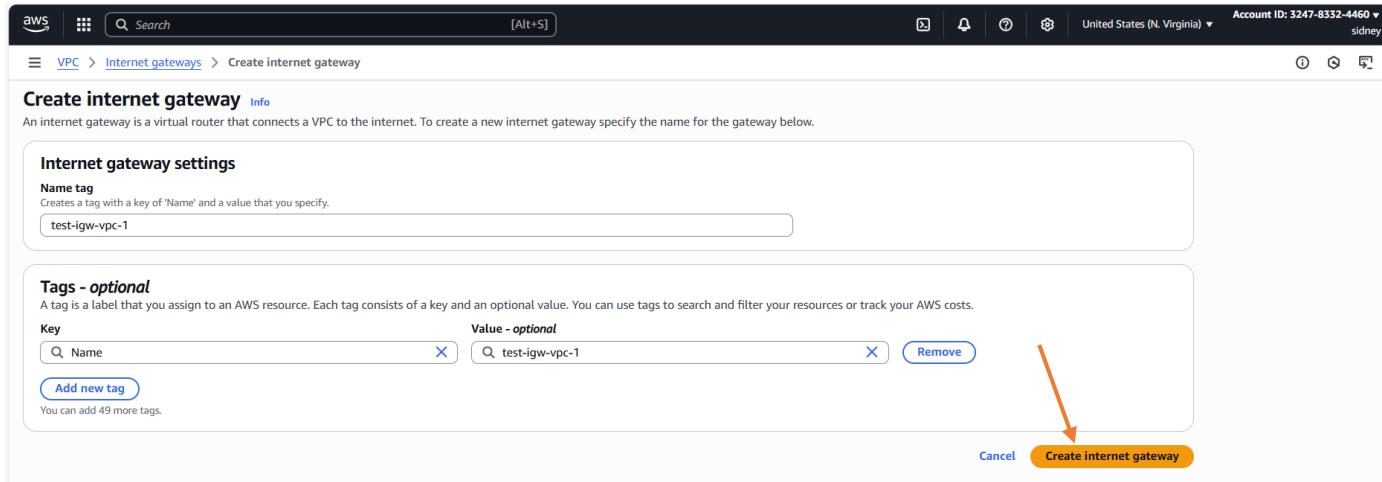


The screenshot shows the AWS VPC Internet Gateways page. Three internet gateways are listed: test-igw (igw-03f3959f9436d149f), - (igw-0eb61fdc4add6d6ae), and Dev-VPC-IGW (igw-0f257b33bb08925562). All are attached to the test-vpc VPC and are owned by account 324783324460. The left sidebar has an orange arrow pointing to the 'Internet gateways' link under the 'Route tables' section.

Click on “Create Internet Gateway”



Give the Internet Gateway a name, I will call it “**test-igw-vpc-1**”



Click on “**create Internet Gateway**”

The screenshot shows the AWS VPC Internet Gateways page. A green banner at the top states: "The following internet gateway was created: igw-0c65e7f9751c6a667 - test-igw-vpc-1. You can now attach to a VPC to enable the VPC to communicate with the internet." Below the banner, the internet gateway details are shown: Internet gateway ID (igw-0c65e7f9751c6a667), State (Detached), VPC ID (-), and Owner (324783324460). The Tags section shows one tag: Name (test-igw-vpc-1). On the left sidebar, the "Internet gateways" section is expanded, showing Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, and Route servers. The "Actions" dropdown menu is visible at the top right.

The internet gateway has been created but it is not yet associated with our VPC. Let us associate the Internet Gateway with our VPC.

This screenshot is identical to the one above, showing the same internet gateway details and the same expanded "Internet gateways" sidebar. The difference is that the "State" field now shows "Attached" instead of "Detached".

Now, we have to attach the Internet Gateway to the VPC. Click on the drop down on “Actions”

VPC dashboard < Actions ▾

EC2 Global View [Alt+S]

Filter by VPC ▼

Virtual private cloud

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

Security

- Network ACLs
- Security groups

PrivateLink and Lattice

- Getting started Updated
- Endpoints Updated
- Endpoint services
- Service networks Updated
- Lattice services

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Select “Attach to VPC”

aws [Alt+S]

☰ VPC > Internet gateways > Attach to VPC (igw-0c65e7f9751c6a667) Actions ▾

United States (N. Virginia) Account ID: 3247-8332-4460 sidney

Attach to VPC (igw-0c65e7f9751c6a667) 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 our VPC “test-vpc-1”

aws [Alt+S]

☰ VPC > Internet gateways > Attach to VPC (igw-0c65e7f9751c6a667) Actions ▾

United States (N. Virginia) Account ID: 3247-8332-4460 sidney

Attach to VPC (igw-0c65e7f9751c6a667) 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.

vpc-0ab72baf7cc45d615

▶ AWS Command Line Interface command

Cancel Attach internet gateway

Click on “Attach Internet Gateway”

Screenshot of the AWS VPC dashboard showing the creation of an Internet Gateway.

Internet gateway igw-0c65e7f9751c6a667 successfully attached to vpc-0ab72baf7cc45d615

igw-0c65e7f9751c6a667 / test-igw-vpc-1

Details **Info**

- Internet gateway ID: igw-0c65e7f9751c6a667
- State: Attached
- VPC ID: vpc-0ab72baf7cc45d615 | test-igw-vpc-1
- Owner: 324783324460

Tags

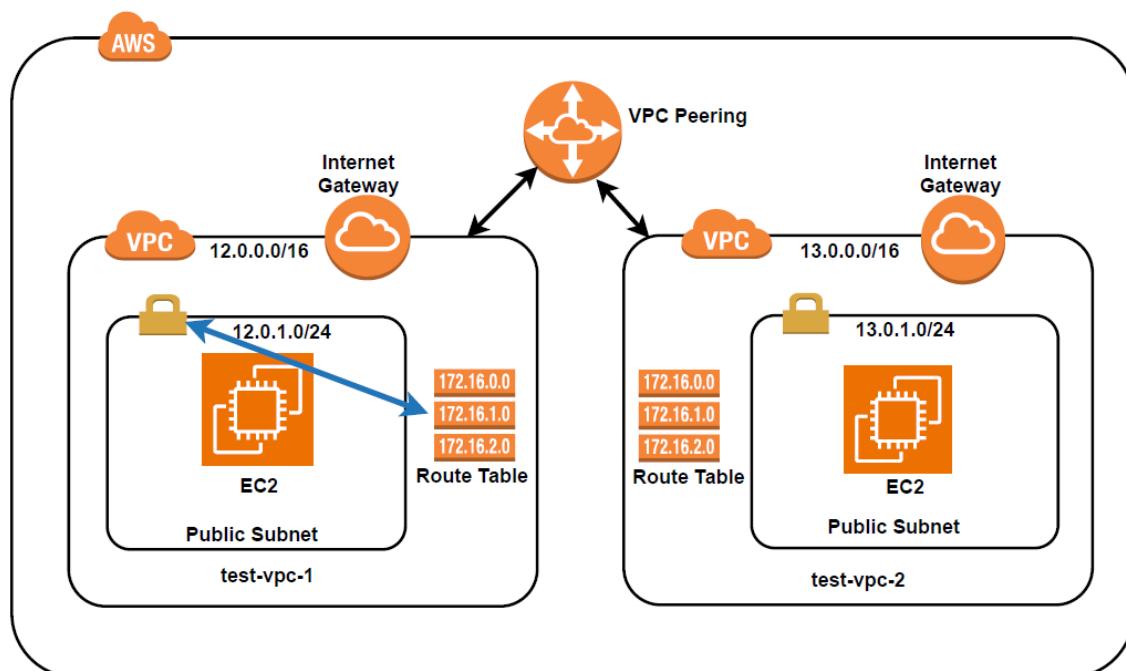
Key	Value
Name	test-igw-vpc-1

Actions

CloudShell **Feedback**

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

We have created the internet gateway for VPC “**test-igw-vpc-1**”.



We have to go to the Route Table and create a route for internet using the internet gateway so that any resources in the subnet can be accessed from the internet. To do this, go back to our route tables

aws Search [Alt+S]

United States (N. Virginia) Account ID: 3247-8332-4460 sidney

VPC > Route tables

Route tables (1/10) Info

Name	Route table ID	Explicit subnet associations	Main	VPC	Own...
-	rtb-094350c8a924d963f	-	Yes	vpc-0128e9209eaef1c37	324783...
-	rtb-091f8c3d6b19ab5ff	-	Yes	vpc-0c565a10e97a2bb88 Dev...	324783...
Dev-VPC-Public-RouteTable	rtb-0869d0039da76f221	2 subnets	No	vpc-0c565a10e97a2bb88 Dev...	324783...
test-rt-public	rtb-0fa8bbc017237268	2 subnets	No	vpc-0a178e5d2aecc4790 test...	324783...
-	rtb-0da6cd7b4e4f2f903	-	Yes	vpc-0c565a10e97a2bb88 Dev...	324783...
Dev-VPC-Private-RouteTable	rtb-02fe1a3bb009f895ad	6 subnets	No	vpc-0c565a10e97a2bb88 Dev...	324783...
-	rtb-0d518b476ecf05b28	-	Yes	vpc-0bc31db7fc84ec75 test...	324783...
test-rt-vpc-1	rtb-007c3145a4fcbe436	subnet-094aa23195f6eeaba / test-subnet-vpc-1-1a	No	vpc-0ab72bafe7cc45d615 test...	324783...
-	rtb-0d57ce1038252a594	-	Yes	vpc-0ab72bafe7cc45d615 test...	324783...
test-rt-vpc-2	rtb-04a6f429ead18b25c	-	No	vpc-0bc31db7fc84ec75 test...	324783...

Last updated 34 minutes ago Actions Create route table

Find route tables by attribute or tag

Route tables (1/10) Info

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

Details

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click on the Route Table ID for test-vpc-1

aws Search [Alt+S]

United States (N. Virginia) Account ID: 3247-8332-4460 sidney

VPC > Route tables > rtb-007c3145a4fcbe436 / test-rt-vpc-1

rtb-007c3145a4fcbe436 / test-rt-vpc-1

Details Info

Route table ID rtb-007c3145a4fcbe436	Main <input type="checkbox"/> No	Explicit subnet associations subnet-094aa23195f6eeaba / test-subnet-vpc-1-1a	Edge associations -
VPC vpc-0ab72bafe7cc45d615 test-vpc-1	Owner ID 324783324460		

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

Routes (1)

Destination	Target	Status	Propagated	Route Origin
12.0.0.0/16	local	Active	No	Create Route Table

Both Edit routes

1

Click on “Edit Routes”

aws Search [Alt+S] United States (N. Virginia) Account ID: 3247-8332-4460 sidney

VPC > Route tables > rtb-007c3145a4fcbe436 > Edit routes

Edit routes

Destination	Target	Status	Propagated	Route Origin
12.0.0.0/16	local	Active	No	CreateRouteTable

[Add route](#)

Cancel Preview Save changes

Click on “Add Route”

aws Search [Alt+S] United States (N. Virginia) Account ID: 3247-8332-4460 sidney

VPC > Route tables > rtb-007c3145a4fcbe436 > Edit routes

Edit routes

Destination	Target	Status	Propagated	Route Origin
12.0.0.0/16	local	Active	No	CreateRouteTable

- [Remove](#)

[Add route](#)

Cancel Preview Save changes

Click on the destination

aws Search [Alt+S] United States (N. Virginia) Account ID: 3247-8332-4460 sidney

VPC > Route tables > rtb-007c3145a4fcbe436 > Edit routes

Edit routes

Destination	Target	Status	Propagated	Route Origin
12.0.0.0/16	local	Active	No	CreateRouteTable

- [Remove](#)

0.0.0.0/0
0.0.0.8
0.0.0.16
0.0.0.24
0.0.0.32
::/0
::/16
::/32
::/48
::/64
pl-02cd2c6b (com.amazonaws.us-east-1.dynamodb)
pl-02d12e369a4312e03 (com.amazonaws.global.ipv6.cloudfront.origin-facing)
pl-05c0959a59362110e (com.amazonaws.us-east-1.ipv6.route53-healthchecks)
pl-062e1df8317caab5 (com.amazonaws.us-east-1.route53-healthchecks)
pl-073555187c4e6ccf2 (com.amazonaws.us-east-1.ipv6.vpc-lattice)

Cancel Preview Save changes

Select “0.0.0.0/0”

AWS VPC Route Tables - Edit routes

Destination	Target	Status	Propagated	Route Origin
12.0.0.0/16	local	Active	No	CreateRouteTable
0.0.0.0/0	local		No	CreateRoute

Add route Remove Cancel Preview Save changes

Click on “Target”

AWS VPC Route Tables - Edit routes

Destination	Target	Status	Propagated	Route Origin
12.0.0.0/16	local	Active	No	CreateRouteTable
0.0.0.0/0	local		No	CreateRoute

Add route Remove Internet Gateway Core Network Egress Only Internet Gateway Gateway Load Balancer Endpoint Instance Internet Gateway local NAT Gateway Network Interface Outpost Local Gateway Peering Connection Transit Gateway Virtual Private Gateway Cancel Preview Save changes

Select “Internet Gateway”

AWS VPC Route Tables - Edit routes

Destination	Target	Status	Propagated	Route Origin
12.0.0.0/16	local	Active	No	CreateRouteTable
0.0.0.0/0	Internet Gateway		No	CreateRoute

Add route Remove Q igw- Cancel Preview Save changes

Click on “igw-” and select our internet gateway

Edit routes

Destination	Target	Status	Propagated	Route Origin
12.0.0.0/16	local	Active	No	CreateRouteTable
0.0.0.0/0	Internet Gateway	-	No	CreateRoute

Add route **Remove** **Cancel** **Preview** **Save changes**

Click on “Save Changes”

rtb-007c3145a4fcbe436 / test-rt-vpc-1

Details **Info**

Route table ID: rtb-007c3145a4fcbe436
Main: No
Owner ID: 324783324460
VPC: vpc-0ab72ba7cc45d615 | test-vpc-1
Explicit subnet associations: subnet-094aa23195f6eeaba / test-subnet-vpc-1-1a
Edge associations: -

Routes (2)

Destination	Target	Status	Propagated	Route Origin
0.0.0.0/0	igw-0c65e7f9751c6a667	Active	No	Create Route
12.0.0.0/16	local	Active	No	Create Route Table

Actions

We have created the internet gateway for the first VPC and associate it with the VPC. Now, we have to create the Internet Gateway for the second VPC.

Go to VPC dashboard

Updated routes for rtb-007c3145a4fcbe436 / test-rt-vpc-1 successfully

rtb-007c3145a4fcbe436 / test-rt-vpc-1

Details **Info**

Route table ID: rtb-007c3145a4fcbe436 | Main: No | Owner ID: 324783324460

Explicit subnet associations: subnet-094aa23195f6eeaba / test-subnet-vpc-1-1a | Edge associations: -

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

Routes (2)

Destination	Target	Status	Propagated	Route Origin
0.0.0.0/0	igw-0c65e7f9751c6a667	Active	No	Create Route
12.0.0.0/16	local	Active	No	Create Route Table

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click on “Internet Gateways”

Internet gateways (4) Info

Create internet gateway

Name	Internet gateway ID	State	VPC ID	Owner
test-igw	igw-03f3959f9436d149f	Attached	vpc-0a178e5d2aecc4790 test-vpc	324783324460
-	igw-0eb61fdc4add606ae	Attached	vpc-0128e9209eae1c37	324783324460
Dev-VPC-IGW	igw-0f257b33b08925562	Attached	vpc-0c565a10e97a2bb88 Dev-VPC	324783324460
test-igw-vpc-1	igw-0c65e7f9751c6a667	Attached	vpc-0ab72ba7cc45d615 test-vpc-1	324783324460

Select an internet gateway above

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click on “Create Internet Gateway”

Screenshot of the AWS VPC Internet Gateways creation page. The 'Name tag' field contains 'my-internet-gateway'. An orange arrow points from the text 'We will give the Internet Gateway the name "test-igw-vpc-2"' to this input field.

We will give the Internet Gateway the name “**test-igw-vpc-2**”

Screenshot of the AWS VPC Internet Gateways creation page. The 'Name tag' field contains 'test-igw-vpc-2'. The 'Tags - optional' section shows a single tag 'Name: test-igw-vpc-2'. An orange arrow points from the text 'Click on “Create Internet Gateway”' to the 'Create internet gateway' button.

Click on “**Create Internet Gateway**”

The screenshot shows the AWS VPC dashboard with the 'Internet gateways' section selected. A message at the top indicates that an Internet Gateway was created: 'The following internet gateway was created: igw-016e8c755e87c831f - test-igw-vpc-2. 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-016e8c755e87c831f', State 'Detached', VPC ID '—', and Owner '324783324460'. The 'Tags' section contains a single tag 'Name: test-igw-vpc-2'. On the right, there is a 'Actions' dropdown menu.

Let us now attach the created Internet Gateway to our VPC “**test-vpc-2**”. Click on the drop down on “Action” and select “**Attach to VPC**”.

The screenshot shows the 'Attach to VPC' dialog box. It has a 'VPC' section with a note: 'Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.' Below this is a 'Available VPCs' section with a note: 'Attach the internet gateway to this VPC.' A search bar labeled 'Select a VPC' is present. At the bottom are 'Cancel' and 'Attach internet gateway' buttons. An orange arrow points from the text 'Click on the field under "Available VPCs" and select the VPC "test-vpc-2"' to the 'Select a VPC' search bar.

Click on the field under “Available VPCs” and select the VPC “**test-vpc-2**”

The screenshot shows the 'Attach to VPC' dialog box again. The 'Available VPCs' search bar now contains the text 'vpc-0bc31d8b7fc84ec75'. An orange arrow points from the text 'Click on "Attach Internet Gateway"' to the 'Attach internet gateway' button at the bottom right of the dialog. The 'Attach internet gateway' button is highlighted with a yellow background.

Click on “**Attach Internet Gateway**”

The screenshot shows the AWS VPC Internet Gateways page. A success message at the top states "Internet gateway igw-016e8c755e87c831f successfully attached to vpc-0bc31d8b7fc84ec75". The main card displays details for the internet gateway "igw-016e8c755e87c831f / test-igw-vpc-2". Key information includes:

- Internet gateway ID:** igw-016e8c755e87c831f
- State:** Attached
- VPC ID:** vpc-0bc31d8b7fc84ec75 | test-vpc-2
- Owner:** 324783324460

The "Tags" section shows a single tag: Name = test-igw-vpc-2. The left sidebar lists various VPC-related services like Subnets, Route tables, Internet gateways, and Security groups.

The internet gateway has been attached to our VPC “**test-vpc-2**”. Now, let us go to the Route Table and create a route for internet using the internet gateway so that any resources in the subnet can be accessed from the internet.

To do this, go back to our route tables

The screenshot shows the AWS Route Tables page. The table lists 10 route tables:

Name	Route Table ID	Explicit subnet associations	Main	VPC
-	rtb-094350c8a924d963f	-	-	vpc-0128e9209eaef1c37
-	rtb-091f8c3d6b19ab5ff	-	-	vpc-0c565a10e97a2bb88
Dev-VPC-Public-RouteTable	rtb-0869d0039da76f221	2 subnets	-	vpc-0c565a10e97a2bb88 Dev...
test-rt-public	rtb-0fbabbbc017237268	2 subnets	-	vpc-0a178e5d2aecc4790
-	rtb-0da6cd7b4e4f2f903	-	-	vpc-0a178e5d2aecc4790 test...
Dev-VPC-Private-RouteTable	rtb-02fe1a3bb09f895ad	6 subnets	-	vpc-0c565a10e97a2bb88 Dev...
-	rtb-0d518b476ecf05b28	-	-	vpc-0bc31d8b7fc84ec75 test...
test-rt-vpc-1	rtb-007c3145a4fcbe436	subnet-094aa23195f6ee...	-	vpc-0ab72baaf7cc45d615 test...
-	rtb-0d57ce1038252a594	-	-	vpc-0ab72baaf7cc45d615 test...
test-rt-vpc-2	rtb-04a6f429ead18b25c	-	-	vpc-0bc31d8b7fc84ec75 test...

An orange arrow points to the row for "test-rt-vpc-2". Below the table, a "Select a route table" dropdown is shown.

Click on the “Route Table ID” of the “test-rt-vpc-2”

The screenshot shows the AWS VPC Route Tables page. On the left, there's a navigation sidebar with sections like VPC dashboard, EC2 Global View, Virtual private cloud (with subnets and route tables), Security, PrivateLink and Lattice, and CloudShell/Feedback. The main area displays route table details: Route table ID (rtb-04a6f429ead18b25c), Main (No), VPC (vpc-0bc31d8b7fc84ec75 | test-vpc-2), Owner ID (324783324460). Below this are tabs for Routes, Subnet associations, Edge associations, Route propagation, and Tags. Under the Routes tab, there's a table with one entry: Destination (13.0.0.0/16), Target (local), Status (Active), Propagated (No), and Route Origin (CreateRouteTable). At the bottom right of the routes table, there's an 'Edit routes' button, which is highlighted with an orange arrow.

Click on “Edit Routes”

This screenshot shows the 'Edit routes' page for the route table from the previous step. It has a table with one row: Destination (13.0.0.0/16), Target (local), Status (Active), Propagated (No), and Route Origin (CreateRouteTable). Below the table is an 'Add route' button. At the bottom right are 'Cancel', 'Preview', and 'Save changes' buttons. An orange arrow points to the 'Add route' button.

Click on “Add Route”

This screenshot shows the 'Edit routes' page again. The table now has two rows. The first row is identical to the previous one. The second row has a search icon in the Destination field. At the bottom right are 'Cancel', 'Preview', and 'Save changes' buttons. An orange arrow points to the search icon in the destination field of the second row.

Click on the “destination” and choose anywhere (**0.0.0.0/0**)

Destination: 13.0.0.0/16
Target: local
Status: Active
Propagated: No
Route Origin: CreateRouteTable

Destination: 0.0.0.0/0
Target: (dropdown menu)
Status: (dropdown menu)
Propagated: No
Route Origin: CreateRoute

Add route Remove

Cancel Preview Save changes

Click on the drop down on “Target” and select “Internet Gateway”

Destination: 13.0.0.0/16
Target: local
Status: Active
Propagated: No
Route Origin: CreateRouteTable

Destination: 0.0.0.0/0
Target: Internet Gateway
Status: (dropdown menu)
Propagated: No
Route Origin: CreateRoute

Add route Remove

Cancel Preview Save changes

Click on “igw” and select our Internet Gateway

Destination: 13.0.0.0/16
Target: local
Status: Active
Propagated: No
Route Origin: CreateRouteTable

Destination: 0.0.0.0/0
Target: Internet Gateway
Status: (dropdown menu)
Propagated: No
Route Origin: CreateRoute

Add route Remove

Cancel Preview Save changes

Click on “Save Changes”

The screenshot shows the AWS VPC dashboard with the 'Route tables' section selected. A green success message at the top states: 'Updated routes for rtb-04a6f429ead18b25c / test-rt-vpc-2 successfully'. The main area displays the details for the route table 'rtb-04a6f429ead18b25c / test-rt-vpc-2'. The 'Details' tab is active, showing the Route table ID (rtb-04a6f429ead18b25c), Main status (No), Owner ID (324783324460), and explicit subnet associations (subnet-09d1fe15fc0c45ac). The 'Routes' tab shows two routes: one to 'igw-016e8c755e87c831f' (Status: Active) and another local route (Status: Active). The sidebar on the left includes sections for Virtual private cloud, Security, PrivateLink and Lattice, and CloudShell/Feedback.

We have created the internet gateway for the second VPC and associate it with the VPC

Part 7: Launch EC2 instances in test-vpc-1 and test-vpc-2

We are almost done with our networking part. Let us now provision our EC2 instances.

Go to AWS Management console and search for EC2

The screenshot shows the AWS search results for 'EC2'. The search bar at the top contains 'EC2'. The left sidebar has a 'Services' section with links to EC2, EC2 Image Builder, and EC2 Global View. Below that is a 'Features' section with links to Dashboard, EC2 Instances, and AMIs. A 'Resources' section is present. A modal window titled 'Introducing resource search' is open, explaining its purpose and how to enable it. At the bottom of the sidebar, there's a question 'Were these results helpful?' with 'Yes' and 'No' buttons.

Services

- EC2
- EC2 Image Builder
- EC2 Global View

Features

- Dashboard
- EC2 Instances
- AMIs

Resources / for a focused search

Were these results helpful?

Yes No

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click on “EC2”

The screenshot shows the EC2 landing page. The left sidebar includes sections for EC2 (Dashboard, EC2 Global View, Events), Instances (Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), and Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces). The main content area features the heading 'Amazon Elastic Compute Cloud (EC2)' and the sub-headline 'Create, manage, and monitor virtual servers in the cloud.' It also includes a section on 'Benefits and features' with a bullet point about scalability and control, and a 'Use cases' section. On the right side, there are three callout boxes: 'Launch a virtual server' (with a 'Launch instance' button highlighted by an orange arrow), 'Get started' (with 'Get started walkthroughs' and 'Get started tutorial' buttons), and 'Additional actions' (with 'View running instances' and 'Migrate a server' buttons).

Compute

Amazon Elastic Compute Cloud (EC2)

Create, manage, and monitor virtual servers in the cloud.

Amazon Elastic Compute Cloud (Amazon EC2) offers the broadest and deepest compute platform, with over 600 instance types and a choice of the latest processors, storage, networking, operating systems, and purchase models to help you best match the needs of your workload.

Benefits and features

EC2 offers ultimate scalability and control

Fully resizable compute capacity to support virtually any workload. This service is best if you want:

- Highest level of control of the entire technology stack, allowing full integration with all AWS services
- Wide variety of server size options
- Wide availability of operating systems to choose from including Linux, Windows, and macOS
- Global scalability

Find out more about EC2

Use cases

Launch a virtual server

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance

View dashboard

Get started

Take our walkthroughs to help you launch an instance, learn about EC2 best practices, and set up your account.

Get started walkthroughs

Get started tutorial

Additional actions

View running instances

Migrate a server

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click on “Launch Instance”

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name
e.g. My Web Server

Application and OS Images (Amazon Machine Image) Info

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Recents **Quick Start**

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux Debian

[Browse more AMIs](#) Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 kernel-6.1 AMI
ami-0de716d6197524dd9 (64-bit (x86), uefi-preferred) / ami-0c094e7a3ac492637 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs Free tier eligible

Description
Amazon Linux 2023 (kernel-6.1) is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

[Amazon Linux 2023 AMI 2023.8.20230809.1.../64GiB/HVM/kernel-6.1](#)

[Preview code](#)

Give the EC2 instance a name, I will call it “**vpc-1-ec2-instance**”

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name
vpc-1-ec2-instance

Application and OS Images (Amazon Machine Image) Info

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Recents **Quick Start**

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux Debian

[Browse more AMIs](#) Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 kernel-6.1 AMI
ami-0de716d6197524dd9 (64-bit (x86), uefi-preferred) / ami-0c094e7a3ac492637 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs Free tier eligible

Description
Amazon Linux 2023 (kernel-6.1) is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

[Amazon Linux 2023 AMI 2023.8.20230809.1.../64GiB/HVM/kernel-6.1](#)

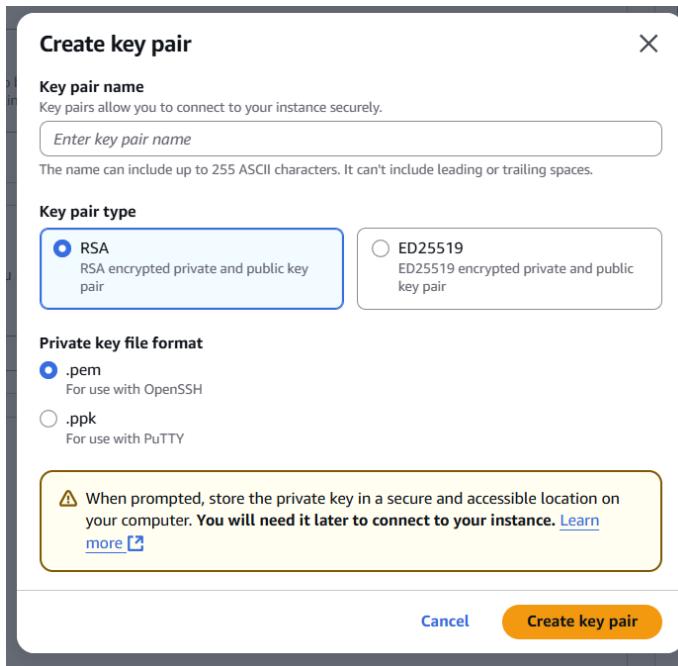
[Preview code](#)

For AMI, we will choose “**Ubuntu**”

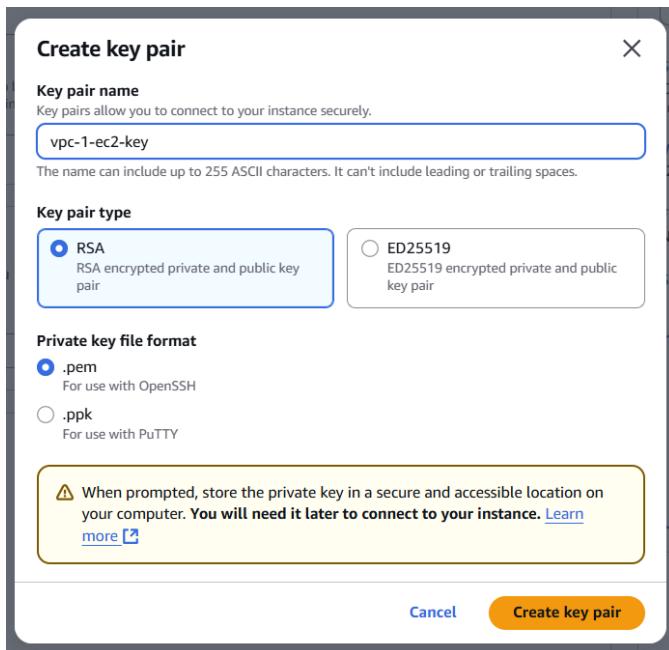
Scroll down to “**Instance Type**” and select “**t2.micro**”

Scroll down to “Key Pair” and click on “create new key pair”

Click on “create new key pair”



We will name the key “**vpc-1-ec2-key**”



Click on “**create key pair**”

▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

▼ **Create new key pair**

Scroll down to “**Network Settings**”

▼ Network settings [Info](#)

Network | [Info](#)
vpc-0128e9209eaef1c37

Subnet | [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP | [Info](#)
Enable
Additional charges apply when outside of free tier allowance

Firewall (security groups) | [Info](#)
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

We'll create a new security group called 'launch-wizard-3' with the following rules:

Allow SSH traffic from Anywhere
0.0.0.0/0
Helps you connect to your instance

Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

⚠️ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. X



Click on “Edit”

▼ Network settings [Info](#)

VPC - required | [Info](#)

vpc-0128e9209eaef1c37
172.31.0.0/16 (default)  

 Create new subnet 

Subnet | [Info](#)

No preference 

Availability Zone | [Info](#)

No preference 

Auto-assign public IP | [Info](#)

Enable 

Additional charges apply when outside of free tier allowance

Firewall (security groups) | [Info](#)

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 

Security group name - required

launch-wizard-3 

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-:/()#@[]+=&{}!\$*

Description - required | [Info](#)

launch-wizard-3 created 2025-08-11T01:31:02.077Z 

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0) 

Type Info	Protocol Info	Port range Info
ssh 	TCP 	22 

Click on the drop down on “VPC” and select our VPC “**test-vpc-1**”

▼ Network settings [Info](#)

VPC - required [Info](#)

vpc-0ab72baf7cc45d615 (test-vpc-1)
12.0.0.0/16

Subnet [Info](#)

subnet-094aa23195f6eeaba test-subnet-vpc-1-1a
VPC: vpc-0ab72baf7cc45d615 Owner: 324783324460 Availability Zone: us-east-1a
Zone type: Availability Zone IP addresses available: 251 CIDR: 12.0.1.0/24

Create new subnet

Auto-assign public IP [Info](#)

Disable

Firewall (security groups) [Info](#)

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

Security group name - required

launch-wizard-3

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-./()#,@[]+=&,{!\$*

Description - required [Info](#)

launch-wizard-3 created 2025-08-11T01:31:02.077Z

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type Info	Protocol Info	Port range Info
ssh	TCP	22
Source type Info	Source Info	Description - optional Info
Anywhere	Add CIDR, prefix list or security group	e.g. SSH for admin desktop

The subnet is already selected. On “auto-sign public IP”, select “Enable”

▼ Network settings [Info](#)

VPC - required [Info](#)

vpc-0ab72baf7cc45d615 (test-vpc-1)
12.0.0.0/16

Subnet [Info](#)

subnet-094aa23195f6eeaba test-subnet-vpc-1-1a
VPC: vpc-0ab72baf7cc45d615 Owner: 324783324460 Availability Zone: us-east-1a
Zone type: Availability Zone IP addresses available: 251 CIDR: 12.0.1.0/24

Create new subnet

Auto-assign public IP [Info](#)

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) [Info](#)

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

Security group name - required

launch-wizard-3

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-./()#,@[]+=&,{!\$*

Description - required [Info](#)

launch-wizard-3 created 2025-08-11T01:31:02.077Z

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type Info	Protocol Info	Port range Info
ssh	TCP	22

Scroll down to “Inbound Security Group Rules”

The screenshot shows the AWS Lambda Firewall (security groups) configuration page. At the top, there are two radio buttons: "Create security group" (selected) and "Select existing security group". Below this, the "Security group name - required" field contains "launch-wizard-3". A note below states: "This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-:/()#@[]+=;&!\$*". The "Description - required" field contains "launch-wizard-3 created 2025-08-11T01:31:02.077Z". The "Inbound Security Group Rules" section lists one rule: "Security group rule 1 (TCP, 22, 0.0.0.0/0)". This rule has fields for Type (ssh), Protocol (TCP), Port range (22), Source type (Anywhere), Source (Add CIDR, prefix list or security group), and Description (e.g. SSH for admin desktop). A warning message at the bottom left says: "⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only." An orange arrow points to the "Add security group rule" button.

Click on “Add security group rule”

The screenshot shows the AWS Lambda Firewall (security groups) configuration page with a new security group rule being added. The "Inbound Security Group Rules" section lists two rules: "Security group rule 1 (TCP, 22, 0.0.0.0/0)" and "Security group rule 2 (TCP, 0)". Rule 1 has the same configuration as in the previous screenshot. Rule 2 has a Type of "Custom TCP", a Protocol of "TCP", a Port range of "0", a Source type of "Custom", and a Source value of "0.0.0.0/0". An orange arrow points to the "Source type" dropdown for Rule 2. A warning message at the bottom left says: "⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only." An orange arrow also points to the "Add security group rule" button.

Click on the drop down on “type” and select “HTTP”

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type | Info ssh

Protocol | Info TCP

Port range | Info 22

Source type | Info Anywhere

Description - optional | Info e.g. SSH for admin desktop

▼ Security group rule 2 (TCP, 80)

Type | Info HTTP

Protocol | Info TCP

Port range | Info 80

Source type | Info Custom

Source | Info Add CIDR, prefix list or security group 0.0.0.0/0 X

Description - optional | Info e.g. SSH for admin desktop

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. X

Add security group rule

► Advanced network configuration

Click on “Source Type” and select “anywhere”

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type | Info ssh

Protocol | Info TCP

Port range | Info 22

Source type | Info Anywhere

Source | Info Add CIDR, prefix list or security group 0.0.0.0/0 X

Description - optional | Info e.g. SSH for admin desktop

▼ Security group rule 2 (TCP, 80, 0.0.0.0/0)

Type | Info HTTP

Protocol | Info TCP

Port range | Info 80

Source type | Info Anywhere

Source | Info Add CIDR, prefix list or security group 0.0.0.0/0 X

Description - optional | Info e.g. SSH for admin desktop

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. X

Add security group rule

► Advanced network configuration

Scroll down to “Advanced Details”

The screenshot shows the 'Configure storage' section of the AWS Lambda setup. It displays a volume configuration: 1x 8 GiB gp3, labeled as a Root volume, 3000 IOPS, Not encrypted. A note indicates that free-tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Below this is an 'Add new volume' button. A message states that the selected AMI contains instance store volumes, but the instance does not allow any instance store volumes. A note also says to click refresh to view backup information. The 'File systems' section shows 0x File systems with an 'Edit' button. An orange arrow points from the text "Click on ‘Advanced Details’ and scroll down to ‘User Data’" to the 'Advanced details' link.

Click on “Advanced Details” and scroll down to “User Data”

The screenshot shows the 'Advanced Details' section of the AWS Lambda setup. It includes fields for V2 only (token required), Metadata response hop limit (set to 2), Allow tags in metadata (set to Select), and User data - optional (with a note about base64 encoding). On the right, the 'Summary' section shows 1 instance. It lists the Software Image (AMI) as Canonical, Ubuntu, 24.04, amd64, and the Virtual server type (instance type) as t2.micro. It also shows Firewall (security group) as New security group and Storage (volumes) as 1 volume(s) - 8 GiB. A note about the free tier is displayed. At the bottom are 'Cancel', 'Launch instance', and 'Preview code' buttons.

Paste the code below in the “User Data”

```
#!/bin/bash
yes | sudo apt update
yes | sudo apt install apache2
echo "<h1>Server Details</h1><p><strong>Hostname:</strong> $(hostname)</p><p><strong>IP Address:</strong> $(hostname -I | cut -d" " -f1)</p>"> /var/www/html/index.html
sudo systemctl restart apache2
```

⚠️ For V2 requests, you must include a session token in all instance metadata requests.
Applications or agents that use V1 for instance metadata access will break.

Metadata response hop limit | [Info](#)
2

Allow tags in metadata | [Info](#)
Select

User data - optional | [Info](#)
Upload a file with your user data or enter it in the field.
[Choose file](#)

```
#!/bin/bash
yes | sudo apt update
yes | sudo apt install apache2
echo "<h1>Server Details</h1><p><strong>Hostname:</strong> $(hostname)</p><strong>IP Address:</strong> $(hostname -i | cut -d" " -f1)</p>" >/var/www/html/index.html
sudo systemctl restart apache2
```

User data has already been base64 encoded

Summary

Number of instances | [Info](#)
1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6... [read more](#)
ami-020cba7c55df1f1615

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#) [Launch instance](#) [Preview code](#)

Click on “Launch Instance”

The screenshot shows the AWS EC2 "Launch an instance" page. At the top, there's a green success message: "Success: Successfully initiated launch of instance (i-06ac98d061867840c)". Below this, there's a "Launch log" link. The main area is titled "Next Steps" with a sub-instruction: "What would you like to do next with this instance, for example "create alarm" or "create backup"". There are six cards with the following options:

- Create billing and free tier usage alerts**: To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds. Includes a "Create billing alerts" button.
- Connect to your instance**: Once your instance is running, log into it from your local computer. Includes a "Connect to instance" button.
- Connect an RDS database**: Configure the connection between an EC2 instance and a database to allow traffic flow between them. Includes a "Connect an RDS database" button.
- Create EBS snapshot policy**: Create a policy that automates the creation, retention, and deletion of EBS snapshots. Includes a "Create EBS snapshot policy" button.
- Manage detailed monitoring**: Enable or disable detailed monitoring for the instance. If you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period. Includes a "Manage detailed monitoring" button.
- Create Load Balancer**: Create a application, network gateway or classic Elastic Load Balancer. Includes a "Create Load Balancer" button.
- Create AWS budget**: AWS Budgets allows you to create budgets, forecast spend, and take action on your costs and usage from a single location. Includes a "Create AWS budget" button.
- Manage CloudWatch alarms**: Create or update Amazon CloudWatch alarms for the instance. Includes a "Manage CloudWatch alarms" button.

At the bottom of the page, the URL is https://us-east-1.console.aws.amazon.com/billing/home?region=us-east-1#/budgets/cre... and the footer includes links for Privacy, Terms, and Cookie preferences.

Click on “Instances”

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with sections like EC2, Instances, Images, Elastic Block Store, Network & Security, and more. The main area displays a table titled 'Instances (1) Info'. The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. One row is shown: 'vpc-1-ec2-instance' with Instance ID 'i-06ac98d061867840c', State 'Running', Type 't2.micro', Status 'Initializing', and Availability Zone 'us-east-1a'. Below the table, a section titled 'Select an instance' is visible.

Wait for the instance to pass the “2/2 check”

This screenshot is identical to the one above, showing the AWS EC2 Instances page. The instance 'vpc-1-ec2-instance' is still listed in the 'Running' state. However, the 'Status check' column now shows '2/2 checks passed' instead of 'Initializing'. The rest of the interface and data are the same.

The “2/2 check” has passed. Let us check if the Apache has been installed successfully.

Select the EC2 instance

The screenshot shows the AWS EC2 Instances page. On the left, a sidebar navigation includes EC2, Dashboard, EC2 Global View, Events, Instances (with sub-options like Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, and Network Interfaces. The main content area displays 'Instances (1/1) Info' for 'vpc-1-ec2-instance'. The instance details include:

- Public IPv4 address:** 54.80.96.177
- Instance state:** Running
- Instance type:** t2.micro
- VPC ID:** vpc-0ab72baf7cc45d615 (test-vpc-1)

Copy the “Public IPv4 address” and test on your browser

The screenshot shows a Microsoft Edge browser window with the URL 54.80.96.177. The page title is "Server Details". The content of the page is:

Hostname: ip-12-0-1-230
IP Address: 12.0.1.230

You can see that the Apache has been installed successfully.

For VPC peering, we are going to do the SSH and then we are going to check a few more things later.

Now, we have to create another EC2 instance in **test-vpc-2**.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like Dashboard, EC2 Global View, Events, Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), and CloudShell/Feedback. The main area displays a table titled 'Instances (1/1) Info' with one row for 'vpc-1-ec2-instance'. The instance details include: Instance ID (i-06ac98d061867840c), Instance state (Running), Instance type (t2.micro), Status check (2/2 checks passed), and Availability Zone (us-east-1a). At the top right of the table, there are buttons for Connect, Instance state, Actions, and Launch instances (which is highlighted with an orange arrow). Below the table, there's a detailed view for 'i-06ac98d061867840c (vpc-1-ec2-instance)' with tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. The 'Details' tab shows information such as Public IPv4 address (54.80.96.177), Instance state (Running), Private IP DNS name (ip-12-0-1-230.ec2.internal), Instance type (t2.micro), and VPC ID (vpc-0ab72baf7cc45d615 (test-vpc-1)).

Click on “Launch Instance”

The screenshot shows the 'Launch an instance' wizard. It starts with the 'Name and tags' step, where 'My Web Server' is entered in the 'Name' field. The next step is 'Application and OS Images (Amazon Machine Image)', which has the 'Quick Start' tab selected, showing recent AMIs: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. Below this is the 'Amazon Machine Image (AMI)' section, which shows 'Amazon Linux 2023 kernel-6.1 AMI' selected. The final step is 'Summary', which shows a summary of the launch configuration: 1 instance, Software Image (Amazon Linux 2023 AMI 2023.8.2...), Virtual server type (t2.micro), Firewall (New security group), and Storage (1 volume(s) - 8 GiB). A note about the free tier is displayed in a callout box: 'Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs. 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.' At the bottom right, there are 'Cancel', 'Launch instance', and 'Preview code' buttons.

We will call the EC2 instance “**vpc-2-ec2-instance**”

Sidney

Search [Alt+S]

United States (N. Virginia) Account ID: 3247-8332-4460 sidney

[EC2](#) > [Instances](#) > Launch an instance

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name Add additional tags

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Search our full catalog including 1000s of application and OS images

Recent [Quick Start](#)

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux Debian

aws Mac ubuntu® Microsoft Red Hat SUSE debian

Amazon Machine Image (AMI)

Amazon Linux 2023 kernel-6.1 AMI ami-0de716d6197524d9 (64-bit (x86), uefi-preferred) / ami-0c094e7a3ac492637 (64-bit (Arm), uefi) Free tier eligible

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 (kernel-6.1) is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.0.20250609.1 x86_64 HVM kernel-6.1

[CloudShell](#) [Feedback](#)

Summary

Number of instances [Info](#)

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.8.2... [read more](#)

ami-0de716d6197524d9

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.

[Cancel](#) [Launch instance](#) [Preview code](#)

Then follow the step as we did in the other EC2 instance.

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Search our full catalog including 1000s of application and OS images

Recent [Quick Start](#)

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux Debian

aws Mac ubuntu® Microsoft Red Hat SUSE debian

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type ami-020cba7c55df1f615 (64-bit (x86)) / ami-07041441b708acbd6 (64-bit (Arm)) Free tier eligible

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Ubuntu Server 24.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture	AMI ID	Publish Date	Username	Verified provider
64-bit (x86)	ami-020cba7c55df1f615	2025-06-10	ubuntu	Verified provider

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t2.micro	Free tier eligible
Family: t2	1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing:	0.0162 USD per Hour
On-Demand Ubuntu Pro base pricing:	0.0134 USD per Hour
On-Demand SUSE base pricing:	0.0116 USD per Hour
On-Demand RHEL base pricing:	0.026 USD per Hour
On-Demand Linux base pricing:	0.0116 USD per Hour

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

[Create new key pair](#)

Scroll down to “Network Settings”

▼ Network settings [Info](#)

[Edit](#)

Network | [Info](#)

vpc-0128e9209eaef1c37

Subnet | [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP | [Info](#)

Enable

[Additional charges apply](#) when outside of [free tier allowance](#)

Firewall (security groups) | [Info](#)

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

We'll create a new security group called 'launch-wizard-4' with the following rules:

<input checked="" type="checkbox"/> Allow SSH traffic from Helps you connect to your instance	Anywhere 0.0.0.0/0
<input type="checkbox"/> Allow HTTPS traffic from the internet To set up an endpoint, for example when creating a web server	
<input type="checkbox"/> Allow HTTP traffic from the internet To set up an endpoint, for example when creating a web server	

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Click on “Edit”

▼ Network settings [Info](#)

VPC - required | [Info](#)

vpc-0bc31d8b7fc84ec75 (test-vpc-2)
13.0.0.0/16

Subnet | [Info](#)

subnet-09dffeb15fc0c45ac test-subnet-vpc-2-1a
VPC: vpc-0bc31d8b7fc84ec75 Owner: 324783324460 Availability Zone: us-east-1a
Zone type: Availability Zone IP addresses available: 251 CIDR: 13.0.1.0/24

Create new subnet [Create new subnet](#)

Auto-assign public IP | [Info](#)

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) | [Info](#)

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

Security group name - required

launch-wizard-4

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-:/()#@+=;&{}\$*

Description - required | [Info](#)

launch-wizard-4 created 2025-08-11T02:53:21.450Z

Scroll down

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type | [Info](#) **Protocol** | [Info](#) **Port range** | [Info](#)

ssh TCP 22

Source type | [Info](#) **Source** | [Info](#) **Description - optional** | [Info](#)

Anywhere e.g. SSH for admin desktop

0.0.0.0/0 [X](#)

Remove

▼ Security group rule 2 (TCP, 80, 0.0.0.0/0)

Type | [Info](#) **Protocol** | [Info](#) **Port range** | [Info](#)

HTTP TCP 80

Source type | [Info](#) **Source** | [Info](#) **Description - optional** | [Info](#)

Anywhere e.g. SSH for admin desktop

0.0.0.0/0 [X](#)

Remove

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. [X](#)

Add security group rule

► Advanced network configuration

Scroll down

Configure storage [Info](#) [Advanced](#)

1x GiB Root volume, 3000 IOPS, Not encrypted

[Free tier eligible customers can get up to 30 GB of EBS General Purpose \(SSD\) or Magnetic storage](#) [X](#)

[Add new volume](#)

The selected AMI contains instance store volumes, however the instance does not allow any instance store volumes. None of the instance store volumes from the AMI will be accessible from the instance

[Click refresh to view backup information](#) [Edit](#)

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems [Edit](#)

Scroll down to “Advanced Details” and click on it

[For V2 requests, you must include a session token in all instance metadata requests.
Applications or agents that use V1 for instance metadata access will break.](#)

Metadata response hop limit [Info](#)

Allow tags in metadata [Info](#)

User data - optional [Info](#)
Upload a file with your user data or enter it in the field.
 [Choose file](#)

User data has already been base64 encoded

Summary
Number of instances [Info](#)

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6... [read more](#)
ami-020cba7c55df1f615

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

[Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage \(or t3.micro where t2.micro isn't available\) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.](#) [X](#)

[Cancel](#) [Launch instance](#) [Preview code](#)

Paste the code below in the “user data”

```
#!/bin/bash
yes | sudo apt update
yes | sudo apt install apache2
echo "<h1>Server Details</h1><p><strong>Hostname:</strong>
$(hostname)<p/><p><strong>IP Address:</strong> $(hostname -I | cut -d" " -f1)</p>">
/var/www/html/index.html
sudo systemctl restart apache2
```

⚠️ For V2 requests, you must include a session token in all instance metadata requests.
Applications or agents that use V1 for instance metadata access will break.

Metadata response hop limit | [Info](#)
2

Allow tags in metadata | [Info](#)
Select

User data - optional | [Info](#)
Upload a file with your user data or enter it in the field.
[Choose file](#)

```
#!/bin/bash
yes | sudo apt update
yes | sudo apt install apache2
echo "<h1>Server Details</h1><p><strong>Hostname:</strong> ${hostname}</p><p><strong>IP Address:</strong> ${hostname} -l | cut -d" -f1</p>> /var/www/html/index.html
sudo systemctl restart apache2
```

User data has already been base64 encoded

▼ Summary

Number of instances | [Info](#)
1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64...[read more](#)
ami-020cba7c55df1f615

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

ⓘ Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#) [Launch instance](#) [Preview code](#)

Click on “Launch Instance”

The screenshot shows the AWS EC2 Instances "Launch an instance" page. At the top, there is a green success message: "Successfully initiated launch of instance (i-04f02bd873e6850d9)". Below this, there is a "Next Steps" section with several options:

- Create billing and free tier usage alerts**: To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds. Includes a "Create billing alerts" button.
- Connect to your instance**: Once your instance is running, log into it from your local computer. Includes a "Connect to instance" button.
- Connect an RDS database**: Configure the connection between an EC2 instance and a database to allow traffic flow between them. Includes a "Connect an RDS database" button.
- Create EBS snapshot policy**: Create a policy that automates the creation, retention, and deletion of EBS snapshots. Includes a "Create EBS snapshot policy" button.
- Manage detailed monitoring**: Enable or disable detailed monitoring for the instance. Includes a "Manage detailed monitoring" button.
- Create Load Balancer**: Create a application, network gateway or classic Elastic Load Balancer. Includes a "Create Load Balancer" button.
- Create AWS budget**: AWS Budgets allows you to create budgets, forecast spend, and take action on your costs and usage from a single location. Includes a "Create AWS budget" button.
- Manage CloudWatch alarms**: Create or update Amazon CloudWatch alarms for the instance. Includes a "Manage CloudWatch alarms" button.

At the bottom of the page, there are links for "CloudShell", "Feedback", and copyright information: "© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

Click on “Instances”

Instances (1/1) [Info](#)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input checked="" type="checkbox"/> vpc-1-ec2-instance	i-06ac98d061867840c	Running View details Logs	t2.micro	2/2 checks passed View alarms +	us-east-1a	

i-06ac98d061867840c (vpc-1-ec2-instance)

- [Details](#) [Status and alarms](#) [Monitoring](#) [Security](#) [Networking](#) [Storage](#) [Tags](#)

Instance summary [Info](#)

Instance ID	i-06ac98d061867840c	Public IPv4 address	54.80.96.177 open address
IPv6 address	-	Instance state	Running
Hostname type	IP name: ip-12-0-1-230.ec2.internal	Private IP DNS name (IPv4 only)	ip-12-0-1-230.ec2.internal
Answer private resource DNS name	-	Instance type	t2.micro
Auto-assigned IP address	54.80.96.177 [Public IP]	VPC ID	vpc-0ab72ba7cc45d615 (test-vpc-1)
		Elastic IP addresses	-
		AWS Compute Optimizer finding	Opt-in to AWS Compute Optimizer for recommendations.

Click on “Refresh”

Instances (2) [Info](#)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input checked="" type="checkbox"/> vpc-1-ec2-instance	i-06ac98d061867840c	Running View details Logs	t2.micro	2/2 checks passed View alarms +	us-east-1a	
<input type="checkbox"/> vpc-2-ec2-instance	i-04f02bd873e6850d9	Running View details Logs	t2.micro	Initializing View alarms +	us-east-1a	

Select an instance

You can see that the second instance has been created. Let us wait for it to pass the “**2/2 check**”

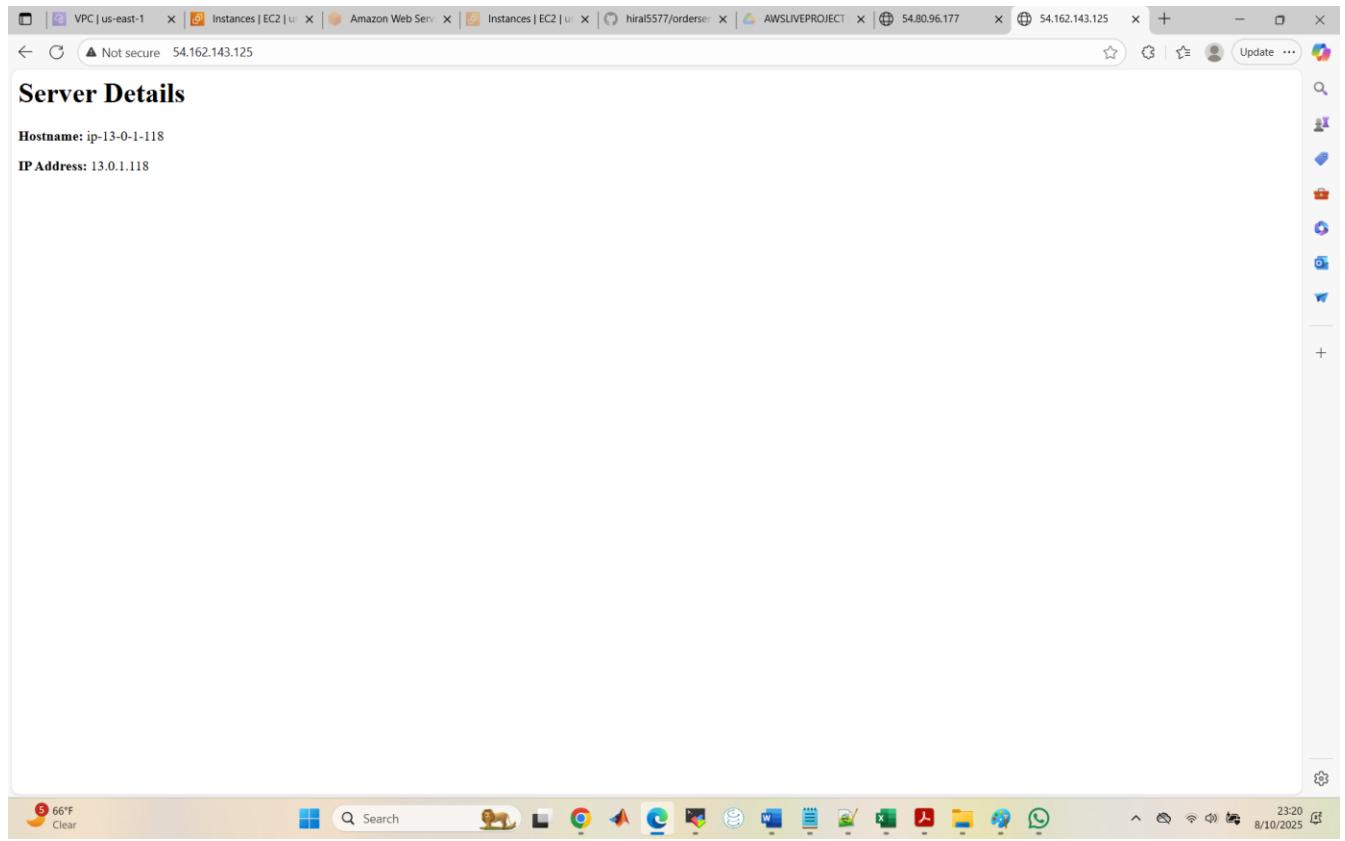
The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with sections like EC2, Dashboard, EC2 Global View, Events, Instances (with sub-options like Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images, Elastic Block Store, and Network & Security. The main area displays a table of instances. The first instance, 'vpc-1-ec2-instance', has an ID of i-06ac98d061867840c, is running, is a t2.micro type, has 2/2 checks passed, and is in the us-east-1a zone. The second instance, 'vpc-2-ec2-instance', has an ID of i-04f02bd873e6850d9, is also running, is a t2.micro type, has 2/2 checks passed, and is in the us-east-1a zone. Below the table, a modal window titled 'Select an instance' is open, showing the same two instances.

It has passed the “**2/2 check**”. Let us check if the Apache has been installed successfully.

Select the EC2 instance

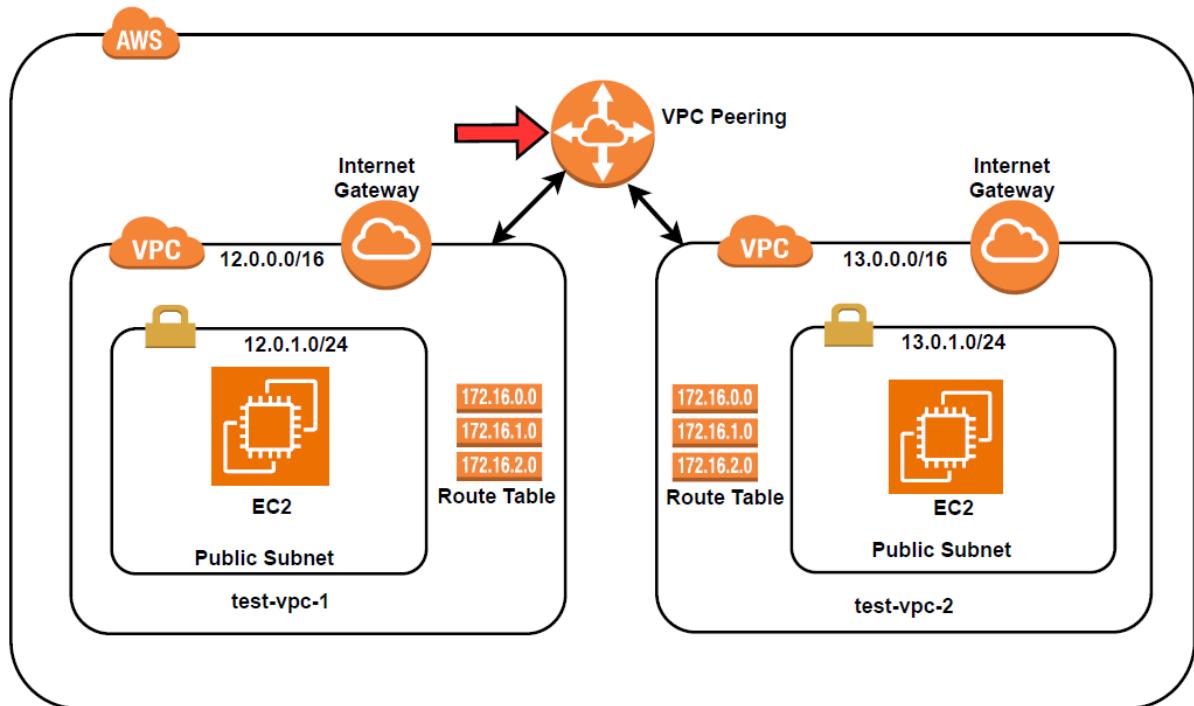
This screenshot shows the same EC2 Instances page as above, but with the 'vpc-2-ec2-instance' instance selected. The instance details pane on the right is expanded, showing various attributes. A red arrow points to the 'Public IPv4 address' field, which contains the value 54.162.143.125. Other visible fields include Instance ID (i-04f02bd873e6850d9), Instance state (Running), Private IP DNS name (ip-13-0-1-118.ec2.internal), Instance type (t2.micro), VPC ID (vpc-0bc31d8b7fc84ec75), Subnet ID (subnet-09dfeb15fc0c45ac), and Instance ARN (arn:aws:ec2:us-east-1:324783324460:instance/i-04f02bd873e6).

Copy the “**Public IPv4 address**” and test on your browser



You can see that the Apache has been installed successfully.

Part 8: Setting up VPC Peering



Now, we are going to create a VPC peering between the two VPCs.

Go to VPC dashboard on AWS Management Console

The screenshot shows the AWS VPC dashboard. On the left sidebar, under the 'Virtual private cloud' section, 'Peering connections' is highlighted with an orange arrow. The main content area displays various VPC resources with counts for the N. Virginia region:

- VPCs: 5
- NAT Gateways: 0
- Subnets: 18
- Route Tables: 10
- Internet Gateways: 5
- Egress-only Internet Gateways: 0
- DHCP option sets: 1
- Customer Gateways: 0
- Managed prefix lists: 0
- Network ACLs: 5
- Security Groups: 13
- Endpoints: 0
- Virtual Private Gateways: 0
- Site-to-Site VPN Connections: 0

On the right side, there are sections for Service Health, Settings, Additional Information, and AWS Network Manager.

Click on “Peering Connections”

The screenshot shows the 'Peering connections' page. The left sidebar has 'Peering connections' selected. At the top right, there is a yellow button labeled 'Create peering connection' with an orange arrow pointing to it. Below the button, the text 'Select a peering connection above' is displayed.

Click on “create peering connection”

Create peering connection

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them privately. [Info](#)

Peering connection settings

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

Select a local VPC to peer with

VPC ID (Requester)
[Select a VPC](#)

Select another VPC to peer with

Account
 My account
 Another account

Region
 This Region (us-east-1)
 Another Region

VPC ID (Acceptor)
[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.
No tags associated with the resource.

[Add new tag](#)
You can add 50 more tags.

For the name, we will call it “**peering-connection-between-vpc1-and-vpc2**”

Create peering connection

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them privately. [Info](#)

Peering connection settings

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

Select a local VPC to peer with

VPC ID (Requester)
[Select a VPC](#)

Select another VPC to peer with

Account
 My account
 Another account

Region
 This Region (us-east-1)
 Another Region

VPC ID (Acceptor)
[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
<input type="text" value="Name"/>	<input type="text" value="peering-connection-between-vpc1-and-vpc2"/>

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

Click on the drop down and select “**test-vpc-1**”

Screenshot of the AWS VPC Peering Connections creation page. The 'VPC ID (Requester)' dropdown is set to 'vpc-0ab72ba7cc45d615 (test-vpc-1)'. The 'VPC CIDRs for vpc-0ab72ba7cc45d615 (test-vpc-1)' table shows one entry: CIDR 12.0.0.0/16, Status Associated, Status reason -. Below this, the 'VPC ID (Acceptor)' dropdown is shown with the placeholder '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

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click on the drop down and select “**test-vpc-2**”

Screenshot of the AWS VPC Peering Connections creation page. The 'VPC ID (Requester)' dropdown is set to 'vpc-0ab72ba7cc45d615 (test-vpc-1)'. The 'VPC CIDRs for vpc-0ab72ba7cc45d615 (test-vpc-1)' table shows one entry: CIDR 12.0.0.0/16, Status Associated, Status reason -. Below this, the 'VPC ID (Acceptor)' dropdown is shown with the placeholder 'Select a VPC'. An orange arrow points to the dropdown menu.

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

You can add 49 more tags.

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click on “**create peering connection**”

A screenshot of the AWS VPC Peering Connections page. At the top, a green banner says "A VPC peering connection pcx-02ff2a4f0ad5f18ea / peering-connection-between-vpc1-and-vpc2 has been requested." Below it, a blue box says "Pending acceptance" with a note about accepting or rejecting the request by Sunday, August 17, 2025. The main area shows "Details" for the peering connection, including Requester owner ID (324783324460), Peering connection ID (pcx-02ff2a4f0ad5f18ea), Status (Pending Acceptance), and Expiration time (Sunday, August 17, 2025 at 23:34:12 EDT). It also lists Acceptor owner ID (324783324460), Requester VPC (vpc-0ab72baf7cc45d615 / test-vpc-1), Requester CIDRs (12.0.0.0/16), Requester Region (N. Virginia (us-east-1)), and Acceptor VPC (vpc-0bc31d8b7fc84ec75 / test-vpc-2). The "Actions" dropdown menu is highlighted with an orange arrow.

Click on the drop down on “Action”

A screenshot of the same AWS VPC Peering Connections page, but now the "Actions" dropdown menu is open. The "Accept request" option is highlighted with an orange arrow. The menu also includes "Reject request", "Edit DNS settings", "Manage tags", and "Delete peering connection". The rest of the page content remains the same, showing the pending acceptance details and DNS settings.

Select “Accept Request”

A screenshot of the AWS VPC Peering connections page. On the left, there's a sidebar with various VPC-related options like 'Virtual private cloud', 'Security', and 'PrivateLink and Lattice'. The main area shows a peering connection named 'pcx-02ff2a4f0ad5f18ea'. A green banner at the top says 'A VPC peering connection ppx-02ff2a4f0ad5f18ea / peering-connection-between-vpc1-and-vpc2 has been requested.' Below it, a 'Pending acceptance' message indicates you can accept or reject the request by Sunday, August 17, 2025. A modal dialog box titled 'Accept VPC peering connection request' is overlaid, containing fields for Requester and Acceptor VPC details, CIDRs, and owner IDs. The 'Accept request' button is highlighted with an orange arrow.

Click on “Accept Request”

A screenshot of the AWS VPC Peering connections page after accepting the request. The green banner at the top now says 'Your VPC peering connection (pcx-02ff2a4f0ad5f18ea | peering-connection-between-vpc1-and-vpc2) has been established.' It also says 'To send and receive traffic across this VPC peering connection, you must add a route to the peered VPC in one or more of your VPC route tables.' The peering connection details show 'Status: Active' and 'Expiration time: -'. The 'Accepter owner ID' and 'Requester VPC' fields are populated with their respective values. The 'Accepter CIDRs' and 'Requester Region' fields are also present. The 'Accepter VPC' and 'Accepter Region' fields are listed below the requester's information.

You can see our peering connection has been established

There is one more thing we need to set up before this connection works and we are able to communicate between the two EC2 instances.

The next thing we need to set is the Route Table. First go to our two VPCs and open them on separate tabs.

VPC Details - test-vpc-1

VPC ID: vpc-0ab72baf7cc45d615	State: Available	Block Public Access: Off	DNS hostnames: Disabled
DNS resolution: Enabled	Tenancy: default	DHCP option set: dopt-067ba7e48365ea16	Main route table: rtb-0d57ce1038252a594
Main network ACL: acl-03a8d954a13b72c85	Default VPC: No	IPv4 CIDR: 12.0.0.0/16	IPv6 pool: -
IPv6 CIDR (Network border group): -	Network Address Usage metrics: Disabled	Route 53 Resolver DNS Firewall rule groups: -	Owner ID: 324783324460

Resource map

- VPC: [test-vpc-1](#)
- Subnets (1): [us-east-1a](#) ([test-subnet-vpc-1-1a](#))
- Route tables (2): [test-rt-vpc-1](#) ([rtb-0d57ce1038252a594](#))
- Network connections (1): [test-igw-vpc-1](#)

VPC Details - test-vpc-2

VPC ID: vpc-0bc31d8b7fc84ec75	State: Available	Block Public Access: Off	DNS hostnames: Disabled
DNS resolution: Enabled	Tenancy: default	DHCP option set: dopt-067ba7e48365ea16	Main route table: rtb-0d518b476ecf05b28
Main network ACL: acl-0a27f5f33ce79a2d	Default VPC: No	IPv4 CIDR: 13.0.0.0/16	IPv6 pool: -
IPv6 CIDR (Network border group): -	Network Address Usage metrics: Disabled	Route 53 Resolver DNS Firewall rule groups: -	Owner ID: 324783324460

Resource map

- VPC: [test-vpc-2](#)
- Subnets (1): [us-east-1a](#) ([test-subnet-vpc-2-1a](#))
- Route tables (2): [test-rt-vpc-2](#) ([rtb-0d518b476ecf05b28](#))
- Network connections (1): [test-igw-vpc-2](#)

Go to the Route Tables and modify the route tables.

The screenshot shows the AWS VPC Route Tables page. The left sidebar includes sections for VPC dashboard, EC2 Global View, Virtual private cloud (Your VPCs, Subnets, Route tables), Security (Network ACLs, Security groups), PrivateLink and Lattice (Endpoints, Endpoint services, Service networks, Lattice services). The main content area displays a table titled 'Route tables (1/10)'. The table has columns for Name, Route table ID, Explicit subnet associations, Edge associations, Main, VPC, and Own... (Actions, Create route table). One row for 'test-rt-vpc-2' is selected, highlighted with a blue border. At the bottom of the page, there are tabs for Details, Routes, Subnet associations, Edge associations, Route propagation, and Tags. The 'Details' tab is active.

Open the route table of test-vpc-1.

The screenshot shows the details of the route table 'rtb-007c3145a4fcbe436 / test-rt-vpc-1'. The left sidebar is identical to the previous screenshot. The main content area shows the 'Details' tab for the route table. It includes fields for Route table ID (rtb-007c3145a4fcbe436), Main (No), VPC (vpc-0ab72baf7cc45d615 | test-vpc-1), Owner ID (324783324460), and Explicit subnet associations (subnet-094aa23195f6eeaba / test-subnet-vpc-1-1a). Below this, there is a 'Routes' section with a table showing two routes: '0.0.0.0/0' with target 'igw-0c65e7f9751c6a667' and status 'Active', and '12.0.0.0/16' with target 'local' and status 'Active'. The 'Edit routes' button is highlighted with a red arrow. At the bottom of the page, there are tabs for Details, Routes, Subnet associations, Edge associations, Route propagation, and Tags. The 'Routes' tab is active.

Click on “Edit Route”

Destination Target Status Propagated Route Origin

12.0.0.0/16	local	Active	No	CreateRouteTable
0.0.0.0/0	Internet Gateway	Active	No	CreateRoute

Add route

Cancel Preview Save changes

Click on “Add Route”

Destination Target Status Propagated Route Origin

12.0.0.0/16	local	Active	No	CreateRouteTable
0.0.0.0/0	Internet Gateway	Active	No	CreateRoute
	-	-	No	CreateRoute

Add route

Cancel Preview Save changes

Copy the IPv4 CIDR of test-vpc-2 and paste on the “destination”

Destination Target Status Propagated Route Origin

12.0.0.0/16	local	Active	No	CreateRouteTable
0.0.0.0/0	Internet Gateway	Active	No	CreateRoute
13.0.0.0/16	-	-	No	CreateRoute

Add route

Cancel Preview Save changes

Click on the drop down on “Target”

Select “Peering Connection”

Click on “pcx-” and select our connection

Click on “Save Changes”

Updated routes for rtb-007c3145a4fcbe436 / test-rt-vpc-1 successfully

Details Info

Route table ID: rtb-007c3145a4fcbe436

Main: No

VPC: vpc-0ab72ba7cc45d615 | test-vpc-1

Owner ID: 324783324460

Explicit subnet associations: subnet-094aa23195f6eeaba / test-subnet-vpc-1-1a

Edge associations: -

Routes (3)

Destination	Target	Status	Propagated	Route Origin
0.0.0.0/0	igw-0c5e7f9751c6a667	Active	No	Create Route
12.0.0.0/16	local	Active	No	Create Route Table
13.0.0.0/16	pcv-02ff2a4f0ad5f18ea	Active	No	Create Route

Now we do same for the other VPC. Click on “Edit Route” on the Route table of “test-vpc-2”

Edit routes

Destination	Target	Status	Propagated	Route Origin
13.0.0.0/16	local	Active	No	CreateRouteTable
0.0.0.0/0	Internet Gateway	Active	No	CreateRoute

Add route

Cancel Preview Save changes

Click on “Add Route”

Edit routes

Destination	Target	Status	Propagated	Route Origin
13.0.0.0/16	local	Active	No	CreateRouteTable
0.0.0.0/0	Internet Gateway	Active	No	CreateRoute

Add route

Cancel Preview Save changes

Add the IPv4 CIDR of “test-vpc-1” here

Screenshot of the AWS VPC Route Tables 'Edit routes' page. The page shows a list of routes:

Destination	Target	Status	Propagated	Route Origin
13.0.0.0/16	local	Active	No	CreateRouteTable
0.0.0.0/0	Internet Gateway	Active	No	CreateRoute
12.0.0.0/16	(dropdown menu)	-	No	CreateRoute

Buttons at the bottom: Cancel, Preview, Save changes.

Click on the drop down on “Target” and select “Peering Connection”

Screenshot of the AWS VPC Route Tables 'Edit routes' page. The 'Target' dropdown for the route with Destination 12.0.0.0/16 has been changed to 'Peering Connection'. The dropdown menu shows 'pcx-' as an option.

Destination	Target	Status	Propagated	Route Origin
13.0.0.0/16	local	Active	No	CreateRouteTable
0.0.0.0/0	Internet Gateway	Active	No	CreateRoute
12.0.0.0/16	Peering Connection	-	No	CreateRoute

Buttons at the bottom: Cancel, Preview, Save changes.

Click on “pcx-” and select our peering connection

Screenshot of the AWS VPC Route Tables 'Edit routes' page. The 'Target' dropdown for the route with Destination 12.0.0.0/16 has been changed to 'pcx-02ff2a4f0ad5f18ea'. The dropdown menu shows 'pcx-' as an option.

Destination	Target	Status	Propagated	Route Origin
13.0.0.0/16	local	Active	No	CreateRouteTable
0.0.0.0/0	Internet Gateway	Active	No	CreateRoute
12.0.0.0/16	Peering Connection	-	No	CreateRoute

Buttons at the bottom: Cancel, Preview, Save changes.

Click on “Save Changes”

The screenshot shows the AWS VPC Route Tables page. At the top, there is a green success message: "Updated routes for rtb-04a6f429ead18b25c / test-rt-vpc-2 successfully". The main title is "rtb-04a6f429ead18b25c / test-rt-vpc-2". On the left, the navigation menu includes sections for VPC dashboard, Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, Route servers), Security (Network ACLs, Security groups), PrivateLink and Lattice (Getting started, Endpoints, Endpoint services, Service networks, Lattice services), and CloudShell/Feedback.

Details

Route table ID rtb-04a6f429ead18b25c	Main No	Explicit subnet associations subnet-09dfeeb15fc0c45ac / test-subnet-vpc-2-1a
VPC vpc-0bc31d8b7fc84ec75 test-vpc-2	Owner ID 324783324460	Edge associations -

Routes (3)

Destination	Target	Status	Propagated	Route Origin
0.0.0.0/0	igw-016e8c755e87c831f	Active	No	Create Route
12.0.0.0/16	pxx-02ff2a4f0ad5f18ea	Active	No	Create Route
13.0.0.0/16	local	Active	No	Create Route Table

Both [Edit routes](#)

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

You can see our Route has been modified.

Part 9: Testing the Connection

The next thing we will do is to SSH into the EC2 instance and try to call each other from each EC2 instances. In testing the communication, we will SSH connect to the EC2 instance in test-vpc-1. Then run the command

`curl <Private IP of EC2 instance in test-vpc-2>`. And vice versa.

Go to EC2 dashboard and select the two EC2 instances on separate tabs

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various navigation options like Dashboard, EC2 Global View, Events, Instances, Images, Elastic Block Store, and Network & Security. The main area displays two instances: "vpc-1-ec2-instance" and "vpc-2-ec2-instance". Both instances are listed as "Running" with "t2.micro" instance type. The "Connect" button is highlighted with an orange arrow. Below the instances, there's a detailed view for "vpc-1-ec2-instance" showing security details, inbound rules, and outbound rules.

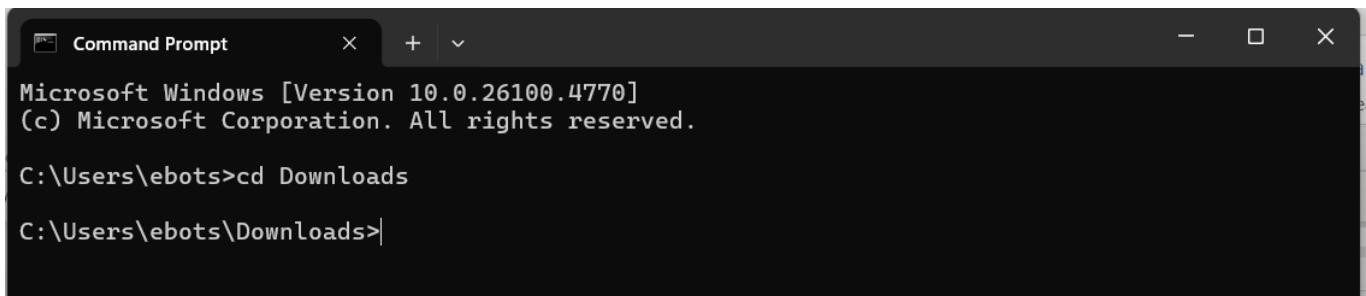
This screenshot shows the same AWS EC2 Instances page as the previous one, but with the focus on the second instance, "vpc-2-ec2-instance". The "Connect" button is again highlighted with an orange arrow. The detailed view for "vpc-2-ec2-instance" is shown below, displaying its instance ID, public and private IP addresses, and various configuration details.

Click on “Connect” in the first EC2 instance

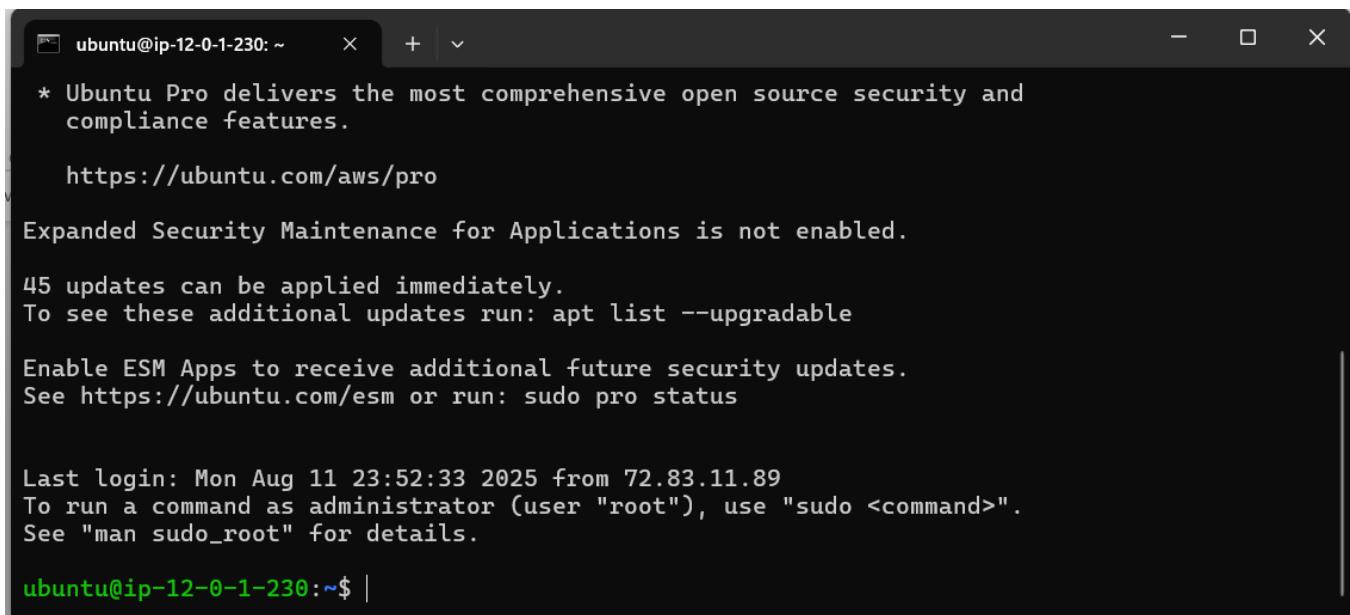
The screenshot shows the AWS EC2 Connect interface. At the top, there's a navigation bar with 'EC2' and 'Instances'. Below it, a 'Connect' section has tabs for 'Info', 'EC2 Instance Connect', 'Session Manager', 'SSH client' (which is selected), and 'EC2 serial console'. Under 'SSH client', there's an 'Instance ID' section with 'i-06ac98d061867840c (vpc-1-ec2-instance)'. Below that are four numbered steps: 1. Open an SSH client, 2. Locate your private key file, 3. Run this command, if necessary, to ensure your key is not publicly viewable, and 4. Connect to your instance using its Public IP. An 'Example' section shows the command: 'ssh -i "vpc-1-ec2-key.pem" ubuntu@98.83.158.230'. A note below says: 'Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.' A 'Cancel' button is at the bottom right.

Copy the command: **ssh -i "vpc-1-ec2-key.pem" ubuntu@98.83.158.230**

Then open command prompt and navigate to where the key is stored, that is our “**Downloads**” folder



Then run the command: **ssh -i "vpc-1-ec2-key.pem" ubuntu@98.83.158.230**



We are now connected to our EC2 instance in the “**test-vpc-1**”. Now let us copy the “**Private IPv4 address**”

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with sections like EC2, Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, and AMI Catalog. Below that are Elastic Block Store, Network & Security, and other options. The main content area shows 'Instances (1/2) Info' with two items: 'vpc-1-ec2-instance' (running, t2.micro, 3.92.128.206) and 'vpc-2-ec2-instance' (running, t2.micro, 3.92.128.206). A detailed view for 'vpc-2-ec2-instance' is shown, with tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. Under the Details tab, the 'Instance summary' section includes fields like Instance ID (i-04f02bd873e6850d9), Instance state (Running), Instance type (t2.micro), VPC ID (vpc-0bc31d8b7fc84ec75), and Subnet ID (subnet-09dfeeb15fc0cc45ac). The Public IPv4 address is 3.92.128.206, and the Private IPv4 address is 13.0.1.118. An orange arrow points from the Public IPv4 address to the Private IPv4 address.

Open Command prompt and navigate to the “Downloads” folder where our keys are saved.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\ebots> cd Downloads
PS C:\Users\ebots\Downloads> |
```

You should get a response by running the command:

```
curl 13.0.1.118
```

```

ubuntu@ip-12-0-1-230: ~      +  v
https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

45 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Mon Aug 11 23:52:33 2025 from 72.83.11.89
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-12-0-1-230:~$ curl 13.0.1.118
<h1>Server Details</h1><p><strong>Hostname:</strong> ip-13-0-1-118</p><p><strong>IP Addresses:</strong> 13.0.1.118</p>
ubuntu@ip-12-0-1-230:~$ |

```

As you can see, we are getting a response from the EC2 instance in “**test-vpc-2**”. We are in IP address “**12-0-1-230**” which is my VPC “**test-vpc-1**” IP range and we are accessing the IP address “**13-0-11-118**”.

Now, let us try the other way. We will SSH connect to the EC2 instance in “**test-vpc-2**” and run the command:

```
curl <Private IPv4 of the EC2 instance in test-vpc-1>
```

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), and CloudShell. The main area displays two instances:

Name	Instance ID	Instance state	Instance type	Status check	Availability Zone
vpc-1-ec2-instance	i-06ac98d061867840c	Running	t2.micro	2/2 checks passed	us-east-1a
vpc-2-ec2-instance	i-04f02bd873e6850d9	Running	t2.micro	2/2 checks passed	us-east-1a

Below the instances, the details for the selected instance ('vpc-2-ec2-instance') are shown. A tooltip indicates that the private IPv4 address '13.0.1.118' has been copied. The details include:

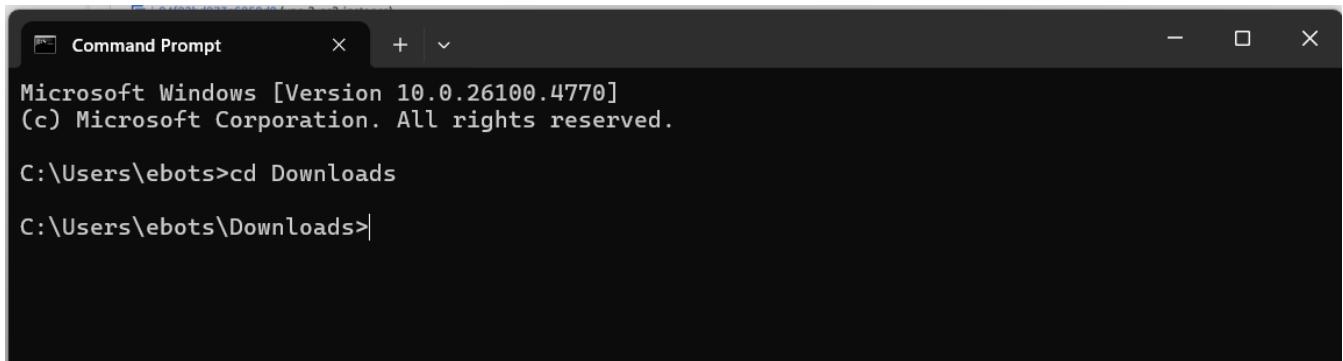
- Public IPv4 address: 3.92.128.206
- Instance state: Running
- Private IP DNS name (IPv4 only): ip-13-0-1-118.ec2.internal
- Instance type: t2.micro
- VPC ID: vpc-0bc51d8b7fc84ec75 (test-vpc-2)
- Subnet ID: subnet-09dffeb15fc045ac (test-subnet-vpc-2-1a)
- Instance ARN: arn:aws:ec2:us-east-1:324783324460:instance/i-04f02bd873e6
- Auto Scaling Group name: -
- Managed: false

Click on “**Connect**”

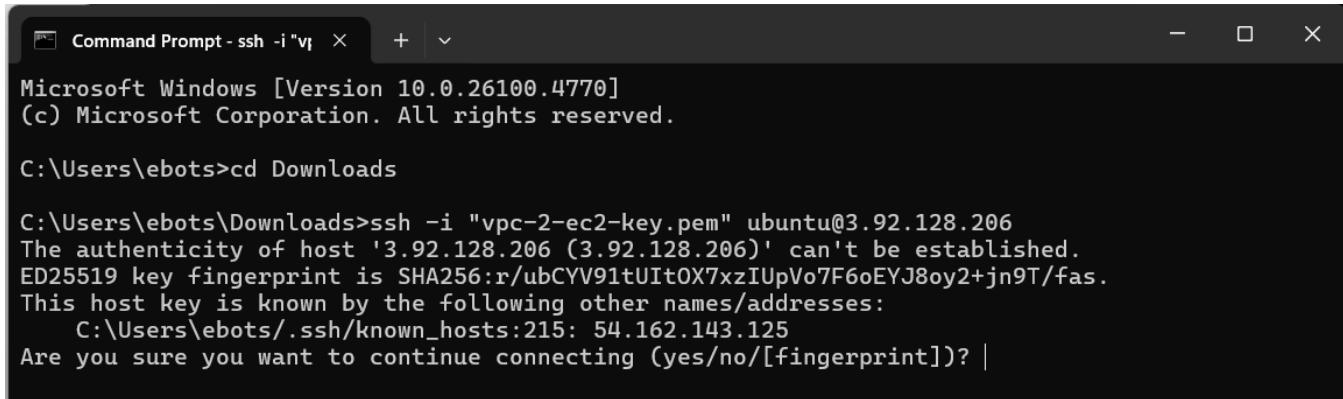
The screenshot shows the AWS EC2 Connect page. At the top, there's a navigation bar with the AWS logo, a search bar, and account information (Account ID: 3247-8332-4460, sidney). Below the navigation is a breadcrumb trail: EC2 > Instances > i-04f02bd873e6850d9 > Connect to instance. The main content area has a title "Connect" with a "Info" link. It says "Connect to an instance using the browser-based client." Below this are tabs: EC2 Instance Connect, Session Manager, **SSH client**, and EC2 serial console. Under the "SSH client" tab, there's a section for "Instance ID" with a dropdown menu showing "i-04f02bd873e6850d9 (vpc-2-ec2-instance)". A numbered list of steps follows: 1. Open an SSH client. 2. Locate your private key file. The key used to launch this instance is vpc-2-ec2-key.pem. 3. Run this command, if necessary, to ensure your key is not publicly viewable. `chmod 400 "vpc-2-ec2-key.pem"`. 4. Connect to your instance using its Public IP: `3.92.128.206`. An "Example:" section shows the command: `ssh -i "vpc-2-ec2-key.pem" ubuntu@3.92.128.206`. A note at the bottom says: "Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username." A "Cancel" button is at the bottom right.

Copy the command: `ssh -i "vpc-2-ec2-key.pem" ubuntu@3.92.128.206`

Open command prompt and navigate to the “Downloads” folder



Run the command: `ssh -i "vpc-2-ec2-key.pem" ubuntu@3.92.128.206`



Type “yes” and press **ENTER**

```

ubuntu@ip-13-0-1-118: ~      + | - | X
System load: 0.08          Processes: 110
Usage of /: 34.7% of 6.71GB  Users logged in: 0
Memory usage: 22%          IPv4 address for enX0: 13.0.1.118
Swap usage: 0%

```

Expanded Security Maintenance for Applications is not enabled.

45 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See <https://ubuntu.com/esm> or run: sudo pro status

Last login: Mon Aug 11 04:37:00 2025 from 72.83.11.89
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-13-0-1-118:~\$ |

Then copy the “Private IPv4 address” of the EC2 instance in “test-vpc-1”

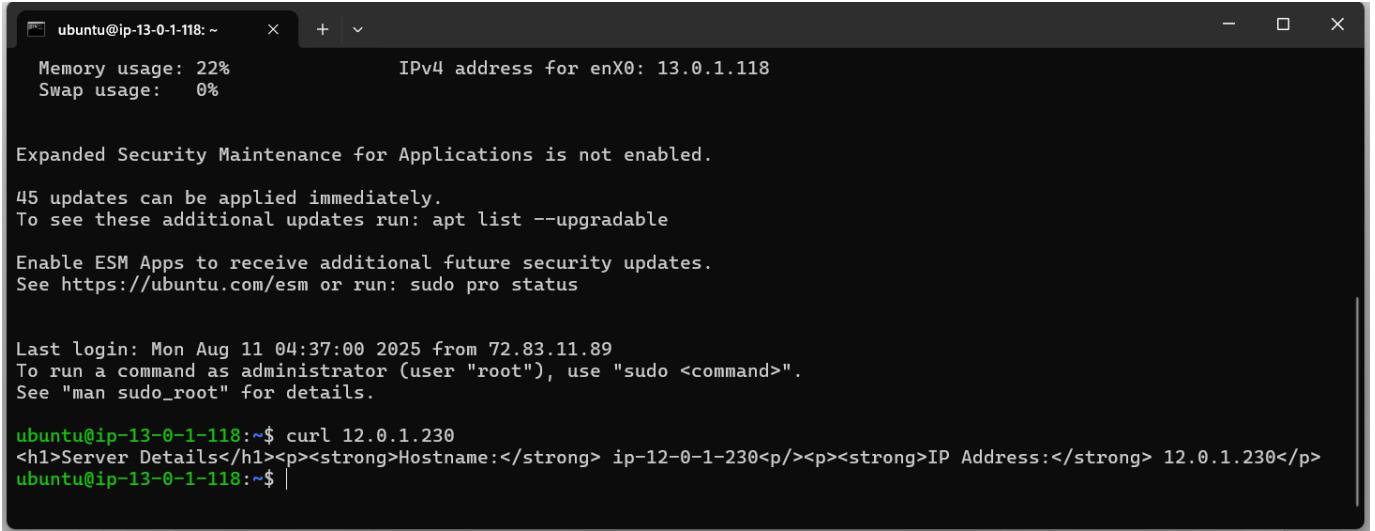
The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various navigation options like Dashboard, EC2 Global View, Events, Instances, Images, Elastic Block Store, Network & Security, and more. The main area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Availability Zone
vpc-1-ec2-instance	i-06ac98d061867840c	Running	t2.micro	2/2 checks passed	View alarms + us-east-1a
vpc-2-ec2-instance	i-04f02bd873e6850d9	Running	t2.micro	2/2 checks passed	View alarms + us-east-1a

Below the table, there's a detailed view for the first instance (i-06ac98d061867840c). The 'Details' tab is selected. In the 'Networking' section, under 'Public IPv4 address', it shows 98.83.158.230. To the right of this, under 'Private IP addresses', it shows 12.0.1.230. An orange arrow points to this private IP address.

Then run the command:

```
curl 12.0.1.230
```



ubuntu@ip-13-0-1-118:~ + - X

Memory usage: 22% Swap usage: 0% IPv4 address for enX0: 13.0.1.118

Expanded Security Maintenance for Applications is not enabled.

45 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See <https://ubuntu.com/esm> or run: sudo pro status

Last login: Mon Aug 11 04:37:00 2025 from 72.83.11.89
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-13-0-1-118:~\$ curl 12.0.1.230
<h1>Server Details</h1><p>Hostname: ip-12-0-1-230</p><p>IP Address: 12.0.1.230</p>
ubuntu@ip-13-0-1-118:~\$ |

You can see that we are getting a response. You can see I am in IP address “**13-0-1-118**” and I can access the IP address “**12-0-1-230**”.

Part 10: Conclusion

We are able to communicate within the VPCs. So, we are able to access the EC2 instance running in another VPC.