

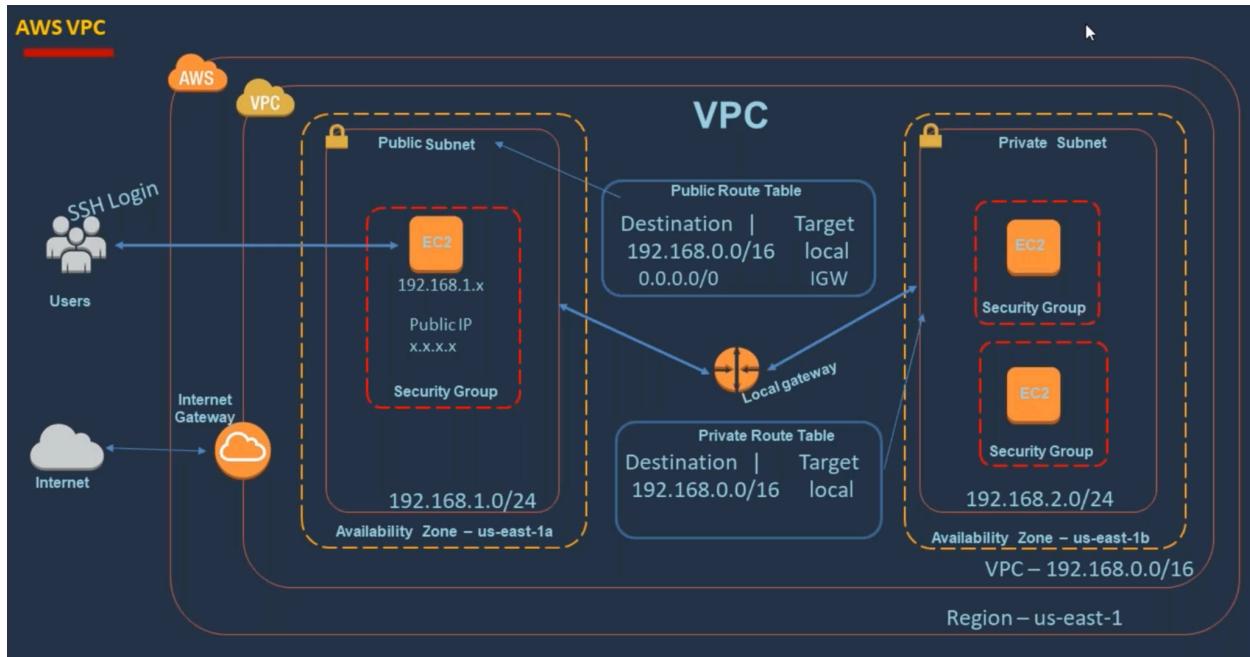
AWS VPC Setup Guide

Introduction

This guide outlines the process of setting up a custom Virtual Private Cloud (VPC) in Amazon Web Services (AWS). It covers creating a VPC, subnets, Internet Gateway (IGW), route tables, and NAT Gateway, as well as launching EC2 instances within the VPC.

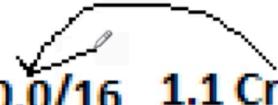
Prerequisites

- AWS account
- Access to AWS Management Console



1 VPC can have only 1 IGW

2 EC2 - public / priv

1. Create VPC - CIDR - 192.168.0.0/16  **1.1 Create IGW**

2. subnet1 - CIDR - 192.168.1.0/24 - us-east-1a

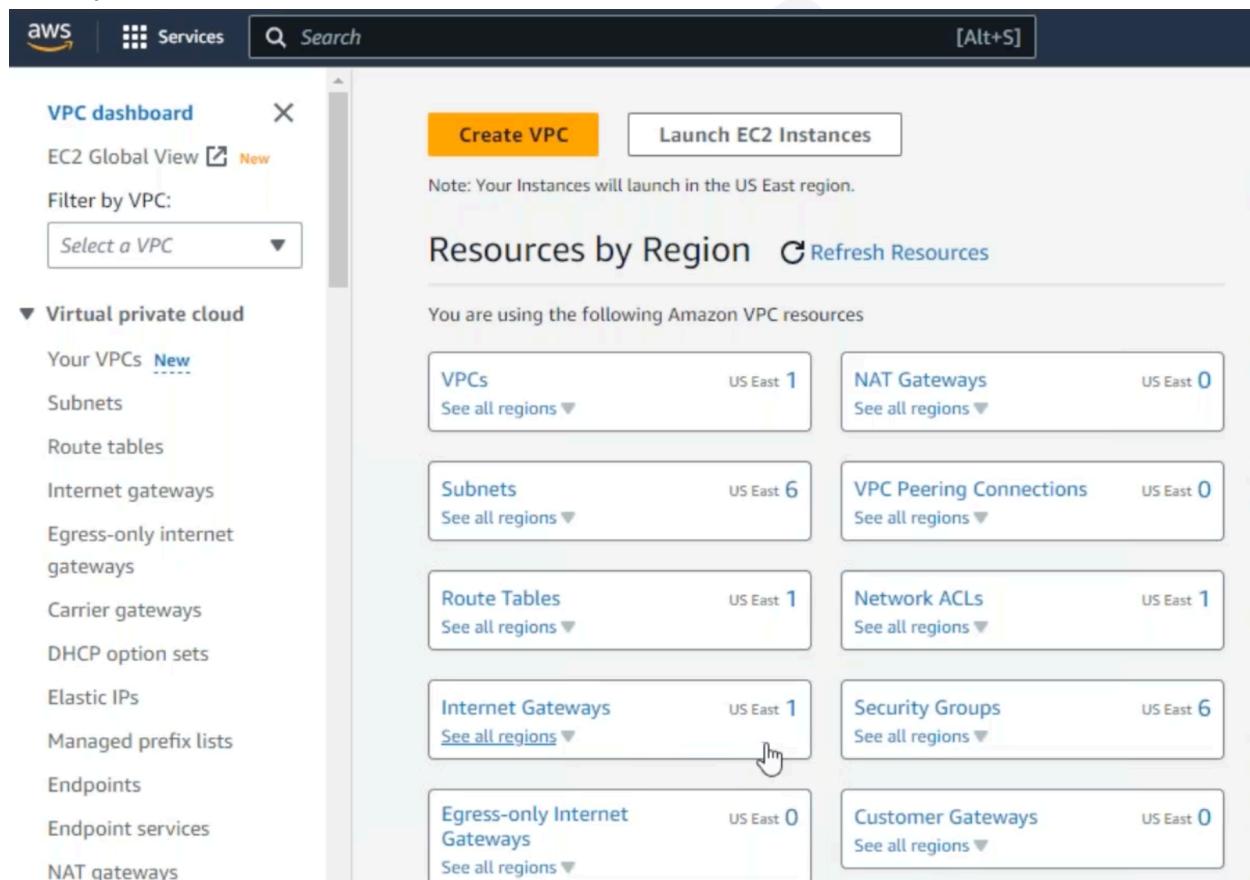
3. subnet2 - CIDR - 192.168.2.0/24 - us-east-1b

4. Private Route Table - local GW

5. Public Route Table - local GW & IGW

6. EC2 instances

Go to AWS Console->VPC. AWS is maintaining default VPC, Subnets, Route table, IGW, Security groups. While deploying EC2, if no custom VPC is specified, this default VPC will be used by AWS.



The screenshot shows the AWS VPC Dashboard. At the top, there are buttons for 'Create VPC' and 'Launch EC2 Instances'. A note says 'Note: Your Instances will launch in the US East region.' Below this, a section titled 'Resources by Region' shows the following counts for the US East region:

Resource Type	Count	Last Modified
VPCs	1	See all regions
NAT Gateways	0	See all regions
Subnets	6	See all regions
VPC Peering Connections	0	See all regions
Route Tables	1	See all regions
Network ACLs	1	See all regions
Internet Gateways	1	See all regions
Security Groups	6	See all regions
Egress-only Internet Gateways	0	See all regions
Customer Gateways	0	See all regions

1. Create a VPC

1. Navigate to AWS Console > VPC
2. Click "Create VPC"
3. Provide a name and CIDR range (e.g., 192.168.0.0/16)
4. Choose Tenancy (Default or Dedicated)

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

IPv4 CIDR block [Info](#)

IPv4 CIDR manual input
 IPAM-allocated IPv4 CIDR block

IPv4 CIDR

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](#)

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.

Tenancy values - Default, dedicated. Default is in this VPC Ec2 instances are shared on demand. Dedicated is used for the Dedicated hosts.

Your VPCs (1/2) [Info](#)

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
-	vpc-022ea287266f40d1a	Available	172.31.0.0/16	-
<input checked="" type="checkbox"/> vpc-01	vpc-061cd820962237b5e	Available	192.168.0.0/16	-

vpc-061cd820962237b5e / vpc-01

[Details](#) [Resource map New](#) [CIDRs](#) [Flow logs](#) [Tags](#)

Details

Step1 is completed. Next step is to create IGW. There is default IGW already which is used by default VPC. We need to have our own for VPC-01.

2. Create an Internet Gateway (IGW)

1. From the left menu, select "Internet Gateways"
2. Click "Create Internet Gateway"

Internet gateways (1/1) [Info](#)

<input checked="" type="checkbox"/>	Name	Internet gateway ID	State	VPC ID	Ow
<input checked="" type="checkbox"/>	-	igw-0c54e558cdef7d138	Attached	vpc-022ea287266f40d1a	576

igw-0c54e558cdef7d138

[Details](#) | [Tags](#)

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

Internet gateway settings

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

igw-01

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/> <input type="button" value="X"/>	<input type="text" value="igw-01"/> <input type="button" value="X"/> <input type="button" value="Remove"/>

Add new tag

You can add 49 more tags.

IGW is created and in Detached state.

The following internet gateway was created: igw-07ba7be2781308e2f - igw-01. You can now attach to a VPC to enable the VPC to communicate with the internet.

Internet gateways (2)

Name	Internet gateway ID	State	VPC ID
igw-01	igw-07ba7be2781308e2f	Detached	-
-	igw-0c54e558cdef7d138	Attached	vpc-022ea287266f40d1a

Select an internet gateway above

This IGW needs to be attached to a VPC. Select the IGW, click on Actions-> Attach to VPC

The following internet gateway was created: igw-07ba7be2781308e2f - igw-01. You can now attach to a VPC to enable the VPC to communicate with the internet.

Internet gateways (1/2)

Name	Internet gateway ID	State	VPC ID
igw-01	igw-07ba7be2781308e2f	Detached	-
-	igw-0c54e558cdef7d138	Attached	vpc-022ea287266f40d1a

igw-07ba7be2781308e2f / igw-01

Details Tags

Select VPC from drop down and Attach Internet gateway. Any servers in this VPC can use this IGW.

The screenshot shows the AWS VPC Attach to Internet Gateway interface. At the top, there's a navigation bar with the AWS logo, Services, a search bar, and a [Alt+S] key shortcut. Below the navigation is a breadcrumb trail: VPC > Internet gateways > Attach to VPC (igw-07ba7be2781308e2f). The main title is "Attach to VPC (igw-07ba7be2781308e2f)" with an "Info" link. A "VPC" section header is followed by a sub-instruction: "Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below." A search bar contains the text "vpc-061cd820962237b5e". Below the search bar is a link to "AWS Command Line Interface command". At the bottom right are "Cancel" and "Attach internet gateway" buttons, with the latter being orange and having a cursor icon pointing to it.

Step 1.1 is completed.

Let's create subnets. Click on Subnets from the left menu. There are default subnets listed. We need to create our own subnets. Click on the Create subnet button.

The screenshot shows the AWS Subnets list interface. The left sidebar includes a VPC dashboard, EC2 Global View (New), and a "Virtual private cloud" section with links for Your VPCs (New), Subnets, Route tables, Internet gateways, Egress-only internet gateways, and Carrier gateways. The main area displays a table titled "Subnets (6) Info" with columns: Name, Subnet ID, State, VPC, and IP. The table lists six default subnets under each column. At the top right of the table is a "Create subnet" button, which is highlighted with a cursor icon. The bottom of the screen shows a "Select a subnet" prompt.

Name	Subnet ID	State	VPC	IP
-	subnet-0b3ec85531e99deea	Available	vpc-022ea287266f40d1a	1
-	subnet-0f88bad4a6c2293aa	Available	vpc-022ea287266f40d1a	1
-	subnet-03ef1d6b3d9990ad	Available	vpc-022ea287266f40d1a	1
-	subnet-0748614beb9907472	Available	vpc-022ea287266f40d1a	1
-	subnet-0a88a6157c224b151	Available	vpc-022ea287266f40d1a	1

Select the VPC created in step1. Provide the AZ, CIDR details and create a subnet.

Create subnet Info

VPC

VPC ID
Create subnets in this VPC.

Select a VPC

Select a VPC	
<input type="text"/>	<input type="button"/>
vpc-022ea287266f40d1a 172.31.0.0/16	(default)
vpc-061cd820962237b5e (vpc-01) 192.168.0.0/16	

Select a VPC first to create new subnets.

aws | Services [Alt+S]

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 CIDR block Info

▼ Tags - optional

Key	Value - optional	Remove
<input type="text" value="Name"/> <input type="button"/>	<input type="text" value="subnet-01"/> <input type="button"/>	<input type="button" value="Remove"/>

You can add 49 more tags.

As per our plan, let's create one more subnet.

aws | Services | Search [Alt+S]

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 CIDR block [Info](#)

Tags - optional

Key	Value - optional	Remove
<input type="text" value="Name"/> X	<input type="text" value="subnet-02"/> X	Remove

Add new tag
You can add 49 more tags.
Remove

Add new subnet

Cancel **Create subnet**

Step2 and Step3 are done.

Let us create a route table. Route table is mandatory for every subnet. If a custom route table is not created, AWS will provide a default route table.

Select route table from left menu and click on Create route table button.

aws | Services | Search [Alt+S] | N. Virginia

VPC dashboard | EC2 Global View [New](#)

Filter by VPC: [Select a VPC](#)

Virtual private cloud
Your VPCs [New](#)
Subnets
Route tables

Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs

Route tables (2) [Info](#)

Name	Route table ID	Explicit subnet associati...	Edge associations	Main
-	rtb-03b73104de6324bcf	-	-	Yes
-	rtb-0d5d3fb6876219ce2	-	-	Yes

Create route table

Select a route table

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 - <i>optional</i>
<input type="text" value="Name"/>	<input type="text" value="priv-route-table-01"/>

Add new tag
You can add 49 more tags.

Create route table

For the route table created, AWS is adding the default route. Its routing the traffic to destination via local gateway.

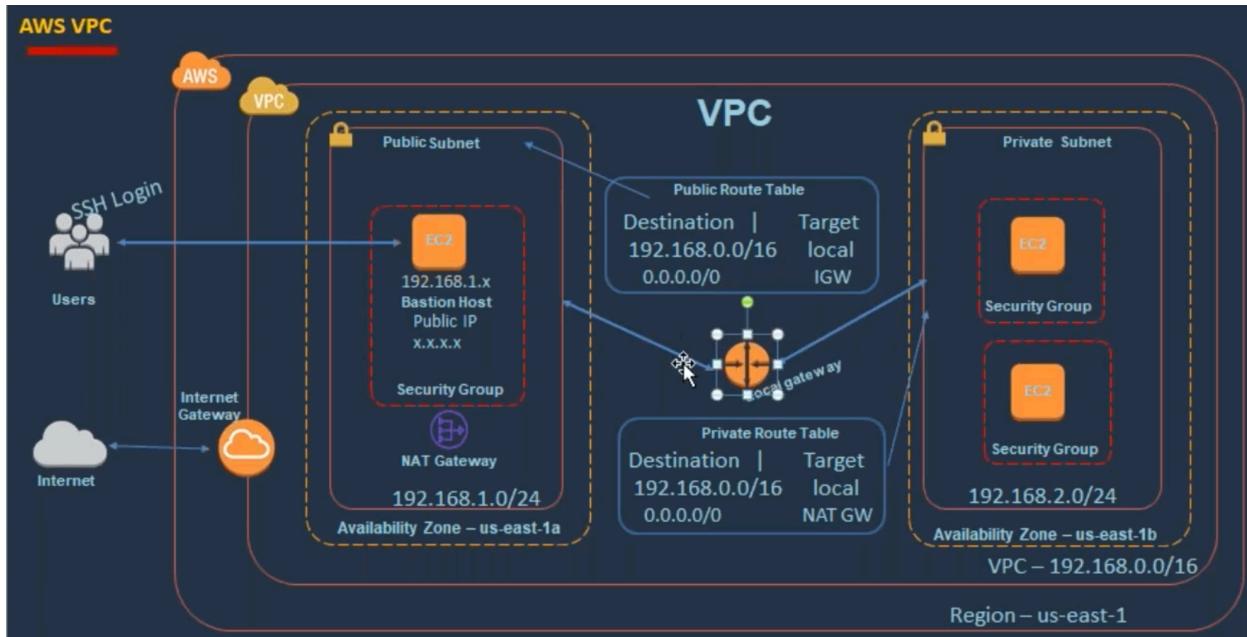
The screenshot shows the AWS VPC Route Table details page. At the top, a green banner displays the message: "Route table rtb-0b6d60b545f5e998b | priv-route-table-01 was created successfully." Below the banner, the route table information is listed:

Route table ID	rtb-0b6d60b545f5e998b	Main	No	Explicit subnet associations	-	Edge associations	-
VPC	vpc-061cd820962237b5e vpc-01	Owner ID	576341600583				

The "Routes" tab is selected, showing one route entry:

Destination	Target	Status	Propagated
192.168.0.0/16	local	Active	No

From our diagram, you can see the local gateway between the subnets. Creating local GW, adding routes everything is taken care of by AWS internally.



We need to associate our custom route table with a subnet. When it is created, the route table is not associated with any subnets. Select the route table, click on Subnet associations->Edit subnet associations.

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
No subnet associations You do not have any subnet associations.			

As per our example, we want subnet-2 as a private subnet. Select the subnet. Current association is with the Main route table which is the default route table created by AWS. we want to change that to our custom route table. Save associations. One subnet can have 1 route table associated. Main route table will be replaced by custom route table after association.

VPC > Route tables > rtb-0b6d60b545f5e998b > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/2)					
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID	
<input checked="" type="checkbox"/> subnet-02	subnet-03c6853d7e5453309	192.168.2.0/24	-	Main (rtb-0d5d3fb6876219ce2)	
<input type="checkbox"/> subnet-01	subnet-03bae9f5b858120fb	192.168.1.0/24	-	Main (rtb-0d5d3fb6876219ce2)	

Selected subnets

- subnet-03c6853d7e5453309 / subnet-02

All the EC2 servers in this subnet, follow these routes to route the traffic.

AWS Services Search [Alt+S] N. Virginia

VPC dashboard

EC2 Global View

Filter by VPC: Select a VPC

Virtual private cloud Your VPCs New Subnets

Route tables Internet gateways Egress-only internet gateways Carrier gateways DHCP option sets Elastic IPs Managed prefix lists Endpoints

You have successfully updated subnet associations for rtb-0b6d60b545f5e998b / priv-route-table-01.

Route table ID: rtb-0b6d60b545f5e998b	Main: No	Explicit subnet associations: subnet-03c6853d7e5453309 / subnet-02	Edge associations: -
VPC: vpc-061cd820963257b5e vpt-01	Owner ID: 576341600585		

Routes Subnet associations Edge associations Route propagation Tags

Routes (1)

Destination	Target	Status	Propagated
192.168.0.0/16	local	Active	No

Step 4 is completed.

Let us create one more route table to make it public. Add IGW route to make it public route table.

VPC > Route tables > rtb-06ce614bb5446e08d > Edit routes

Edit routes

Destination	Target	Status	Propagated
192.168.0.0/16	local	Active	No
0.0.0.0/0	igw-07ba7be2781308e2f	-	No

Associate subnet to use public route table.

VPC > Route tables > rtb-06ce614bb5446e08d > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/2)					
	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input type="checkbox"/>	subnet-02	subnet-03c6853d7e5453309	192.168.2.0/24	-	rtb-0b6d60b545f5e998b / priv-route-table-01
<input checked="" type="checkbox"/>	subnet-01	subnet-03bae9f5b858120fb	192.168.1.0/24	-	Main (rtb-0d5d3fb6876219ce2)

Selected subnets

subnet-03bae9f5b858120fb / subnet-01 X

Cancel **Save associations**

Step5 is completed.

Subnet-01 is a public subnet and Subnet-02 is a private subnet.

If the server needs to be accessible from the outside world, it should have public IP and the associated route table should have an entry with IGW. We can enable an option to automatically assign public IP for all the instances created under subnet-01. Select subnet->Actions->Edit subnet settings.

VPC dashboard X

EC2 Global View New

Filter by VPC:

Select a VPC

Virtual private cloud

Your VPCs New

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

VPC > Subnets > subnet-03bae9f5b858120fb

subnet-03bae9f5b858120fb / subnet-01

Details		Actions	
Subnet ID	subnet-03bae9f5b858120fb	Subnet ARN	arn:aws:ec2:us-east-1:576341600583:subnet/subnet-03bae9f5b858120fb
Available IPv4 addresses	251	State	Available
Network border group	us-east-1	Availability Zone	us-east-1a
Default subnet	No	Route table	rtb-06ce614bb5446e08d public-route-table-01
Customer-owned IPv4 pool		Auto-assign IPv6 address	No

Create flow log

Edit subnet settings

Edit IPv6 CIDRs

Edit network ACL association

Edit route table association

Edit CIDR reservations

Share subnet

Manage tags

Delete

Auto-assign customer-owned IPv4 address

No

Enable auto-assign IP4 addresses.

VPC > Subnets > subnet-03bae9f5b858120fb > Edit subnet settings

Edit subnet settings Info

Subnet

Subnet ID	Name
<input type="checkbox"/> subnet-03bae9f5b858120fb	<input type="checkbox"/> subnet-01

Auto-assign IP settings Info

Enable the auto-assign IP settings to automatically request a public IPv4 or IPv6 address for a new network interface in this subnet.

- Enable auto-assign public IPv4 address Info
- Enable auto-assign customer-owned IPv4 address Info
Option disabled because no customer owned pools found.

We can start creating EC2 servers in this custom VPC. Create a security group and use it while creating EC2 instances.

Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
NAT gateways
Peering connections

Security

- Network ACLs
- Security groups**

DNS firewall

- Rule groups
- Domain lists

Security Groups (7) Info

Actions ▾ Export security groups to CSV ▾

<input type="checkbox"/>	Name	Security group ID	Security group name	VPC ID	Description
<input type="checkbox"/>	-	sg-08b2c91c766173e8d	launch-wizard-1	vpc-022ea287266f40d1a	launch-wizard-1 create...
<input type="checkbox"/>	-	sg-01ae2875f07d211c5	default	vpc-061cd820962237b5e	default VPC security gr...
<input type="checkbox"/>	-	sg-035909b83f8879044	launch-wizard-2	vpc-022ea287266f40d1a	launch-wizard-2 create...
<input type="checkbox"/>	-	sg-01be904bc725e9d36	sredpt10-sg	vpc-022ea287266f40d1a	Allow Port 22
<input type="checkbox"/>	-	sg-04efd8d86aa3dcf43	default	vpc-022ea287266f40d1a	default VPC security gr...

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name Info

Name cannot be edited after creation.

Description Info

VPC Info

Inbound rules Info

Outbound rules [Info](#)

Type Info	Protocol Info	Port range Info	Destination Info	Description - optional Info
All traffic	All	All	Custom ▾ <input type="text" value="0.0.0.0"/> <input type="button" value="X"/>	<input type="text"/> Delete

[Add rule](#)

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

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

[Cancel](#) [Create security group](#)

Create Ec2 server. Click Launch instance. Select name, AMI etc. In the Network settings, click Edit.

▼ Network settings [Info](#)

VPC - required [Info](#)

vpc-061cd820962237b5e (vpc-01)
192.168.0.0/16

Subnet [Info](#)

subnet-03c6853d7e5453309 subnet-02
VPC: vpc-061cd820962237b5e Owner: 576341600583
Availability Zone: us-east-1b IP addresses available: 251 CIDR: 192.168.2.0/24

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

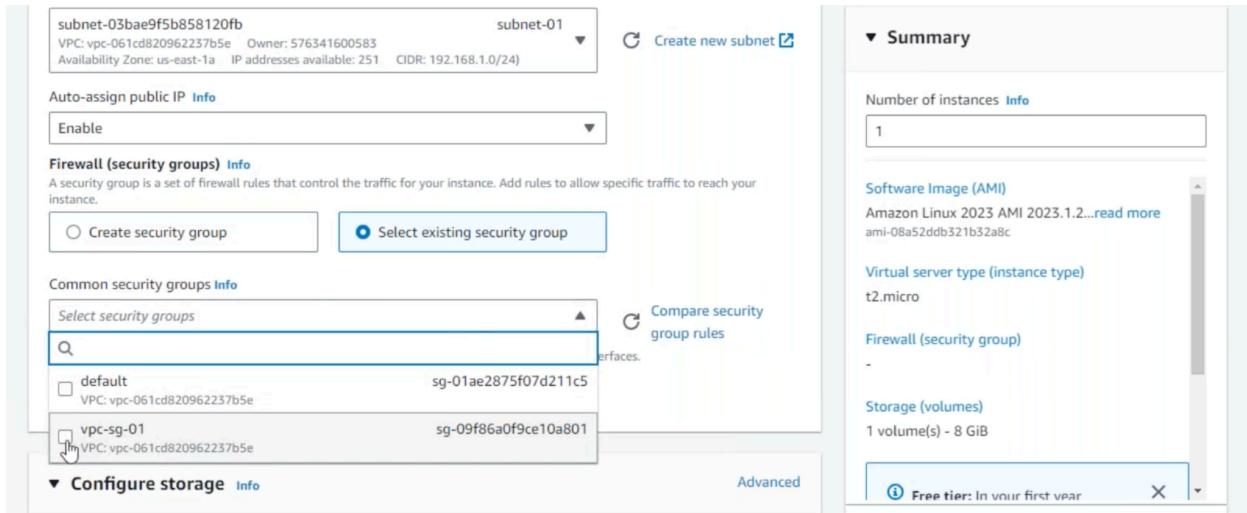
Common security groups [Info](#)

Select security groups

vpc-sg-01 sg-09f86a0f9ce10a801 X
VPC: vpc-061cd820962237b5e

[Compare security group rules](#)

Create a public EC2 instance



SSH to the public EC2 server. From there you can SSH to private EC2. As the local GW takes care of routing.

```
[ec2-user@ip-192-168-1-203 ~]$ 
[ec2-user@ip-192-168-1-203 ~]$ 
[ec2-user@ip-192-168-1-203 ~]$ ssh ec2-user@192.168.2.39
The authenticity of host '192.168.2.39 (192.168.2.39)' can't be established.
ED25519 key fingerprint is SHA256:s6g+/tra42zhAG/b4uP15x87uAEq39bqwtBbNqs28SM.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.2.39' (ED25519) to the list of known hosts.
[ec2-user@192.168.2.39: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ec2-user@ip-192-168-1-203 ~]$ vi sre.pem
[ec2-user@ip-192-168-1-203 ~]$ chmod 600 sre.pem
[ec2-user@ip-192-168-1-203 ~]$ ssh -i sre.pem ec2-user@192.168.2.39
'#
~\##_
~~\##_#_ Amazon Linux 2023
~~\##_#_#
~~\##_|#
~~\#/__ https://aws.amazon.com/linux/amazon-linux-2023
~~\V~'__->
~~\_./
~~\_./_/ /_
~~\_./_/ /_
~~\_./_/ /_
[ec2-user@ip-192-168-2-39 ~]$ 
```

To host your application in this private server, we need to install required packages. For which we need internet access from a private server. Private servers should have outbound service to install required software. People from the internet should not be able to connect to private servers. We use NAT gateway for this use case. NAT gateway should be deployed in a public subnet. Which then can use the IGW to connect to the internet. NAT gateway can connect to private servers using local GW. Private server will send the internet request to NAT gateway, and it uses IGW to connect to the internet and reply back with the required packages to the private server. In the private route table, we need to add a route saying that, if the destination is not within VPC, use NAT gateway

EX: Destination 0.0.0.0/0 Target NAT GW

From VPC section left menu, select NAT gateways->Create NAT gateway

NAT gateways Info

Filter NAT gateways

Name	NAT gateway ID	Connectivity	State	State message	Primary
Select a NAT gateway					

Create NAT gateway

Select public subnet. NAT gateway requires public IP to connect to the internet. Select allocate elastic IP.

Name: nat-01
The name can be up to 256 characters long.

Subnet: subnet-03bae9f5b858120fb (subnet-01)

Connectivity type: Public

Elastic IP allocation ID: Select an Elastic IP

Allocate Elastic IP

Additional settings

Tags: Key: Name, Value: nat-01

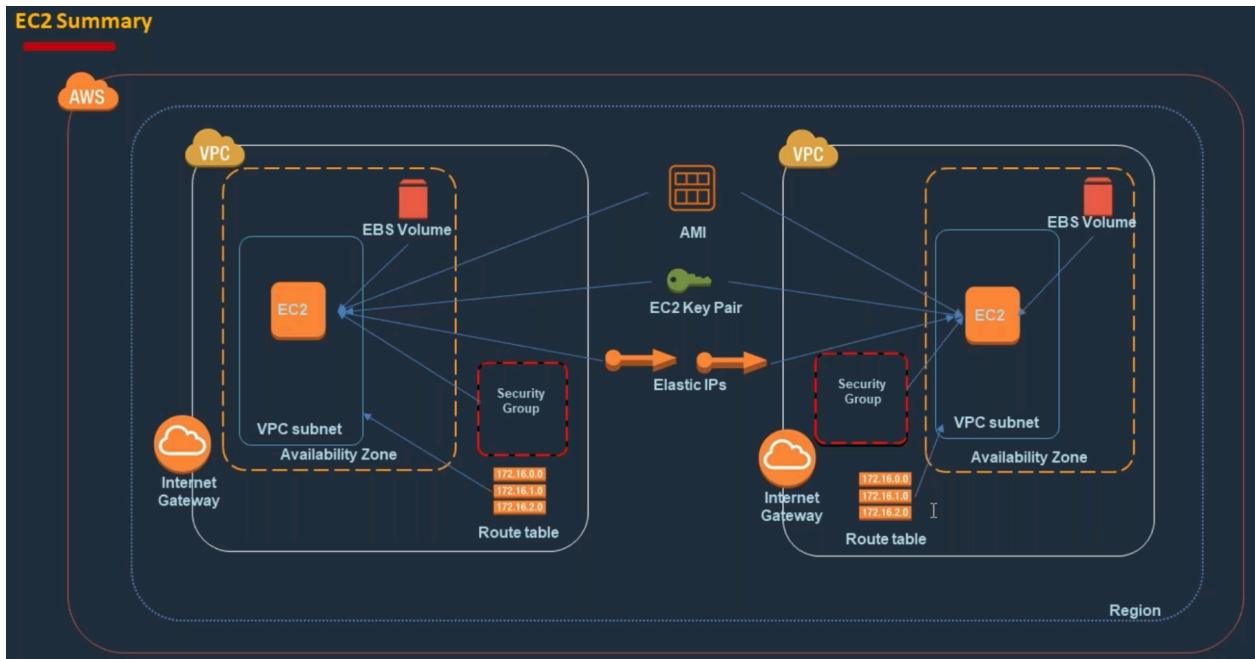
Add an entry in the private subnet route table for NAT gateway

The screenshot shows the 'Edit routes' page for a specific route table. A new route is being added with the destination '0.0.0.0/0' and target 'nat-0c1330d86a22d9455'. The status is 'Active' and propagation is set to 'No'. There are buttons for 'Add route', 'Cancel', 'Preview', and 'Save changes'.

NAT gateway is created with both public IP and private IP

The screenshot shows the 'NAT gateways (1/1)' page. It lists one NAT gateway named 'nat-0c1330d86a22d9455 / nat-01'. The details show it is 'Available' with a primary public IP of '54.157.202.181' and a primary private IP of '192.168.1.41'. The VPC associated with it is 'vpc-061cd820962237b'. The left sidebar shows navigation links for Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, NAT gateways, Peering connections, Security, DNS firewall, and Network Firewall.

Try to install packages on a private server. It should work now.



Summary

- VPCs are region-specific and can contain multiple subnets
- Subnets are associated with specific Availability Zones
- EC2 instances are deployed into specific subnets
- EBS volumes are zone-specific and must be in the same zone as their EC2 instance
- AMIs are region-specific
- Key pairs and Elastic IPs are region-level resources
- Security Groups are VPC-specific
- Route tables are VPC-specific
- One VPC can have only one IGW

Remember to consider costs associated with running AWS resources, especially for components like NAT Gateways which incur hourly charges.