

Creating Networking Resources in an Amazon Virtual Private Cloud (VPC)

Objectives

In this lab, you will:

- Summarize the customer scenario
- Create a VPC, Internet Gateway, Route Table, Security Group, Network Access List, and EC2 instance to create a routable network within the VPC
- Familiarize yourself with the console
- Develop a solution to the customers issue found within this lab.

The lab is complete once you can successfully utilize the command ping outside the VPC.

Duration

This lab total duration is 60 minutes.

Scenario

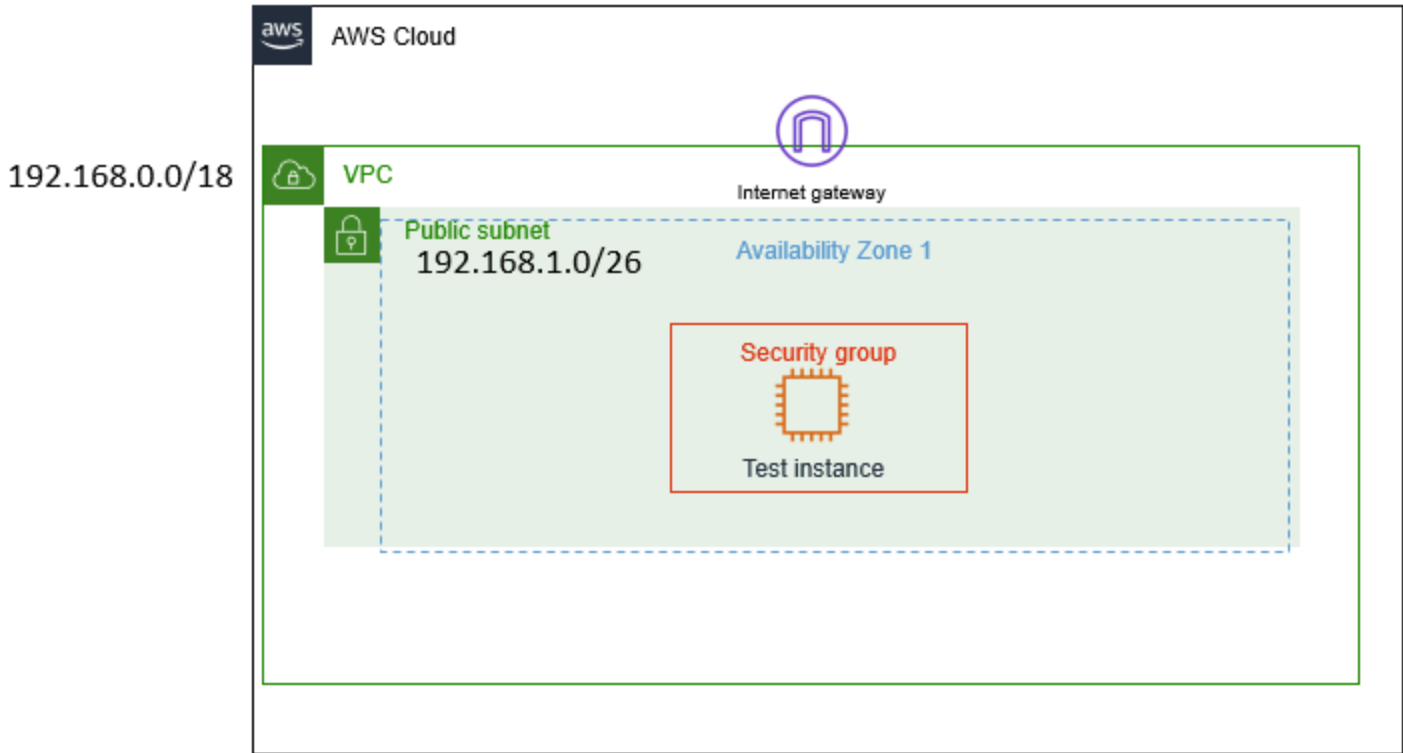
Your role is a Cloud Support Engineer at Amazon Web Services (AWS). During your shift, a customer from a startup company requests assistance regarding a networking issue within their AWS infrastructure. The email and an attachment of their architecture is below.

Email from the customer

Hello Cloud Support!

I previously reached out to you regarding help setting up my VPC. I thought I knew how to attach all the resources to make an internet connection, but I cannot even ping outside the VPC. All I need to do is ping! Can you please help me set up my VPC to where it has network connectivity and can ping? The architecture is below. Thanks!

Brock, startup owner



Customer VPC architecture

AWS service restrictions

In this lab environment, access to AWS services and service actions might be restricted to the ones that are needed to complete the lab instructions. You might encounter errors if you attempt to access other services or perform actions beyond the ones that are described in this lab.

Accessing the AWS Management Console

1. At the top of these instructions, choose **Start Lab** to launch your lab.

A **Start Lab** panel opens, and it displays the lab status.

Tip: If you need more time to complete the lab, choose the Start Lab button again to restart the timer for the environment.

2. Wait until you see the message *Lab status: ready*, then close the **Start Lab** panel by choosing the **X**.

3. At the top of these instructions, choose **AWS**.

This opens the AWS Management Console in a new browser tab. The system will automatically log you in.

Tip: If a new browser tab does not open, a banner or icon is usually at the top of your browser with a message that your browser is preventing the site from opening pop-up windows. Choose the banner or icon and then choose **Allow pop ups**.

4. Arrange the AWS Management Console tab so that it displays along side these instructions. Ideally, you will be able to see both browser tabs at the same time so that you can follow the lab steps more easily.

Task 1: Investigate the customer's needs

For task 1, you will investigate the customer's request and build a VPC that has network connectivity. You will complete this lab when you can successfully ping from your EC2 instance to the internet showing that the VPC has network connectivity.

In the scenario, Brock, the customer requesting assistance, has requested help in creating resources for his VPC to be routable to the internet. Keep the VPC CIDR at 192.168.0.0/18 and public subnet CIDR of 192.168.1.0/26.

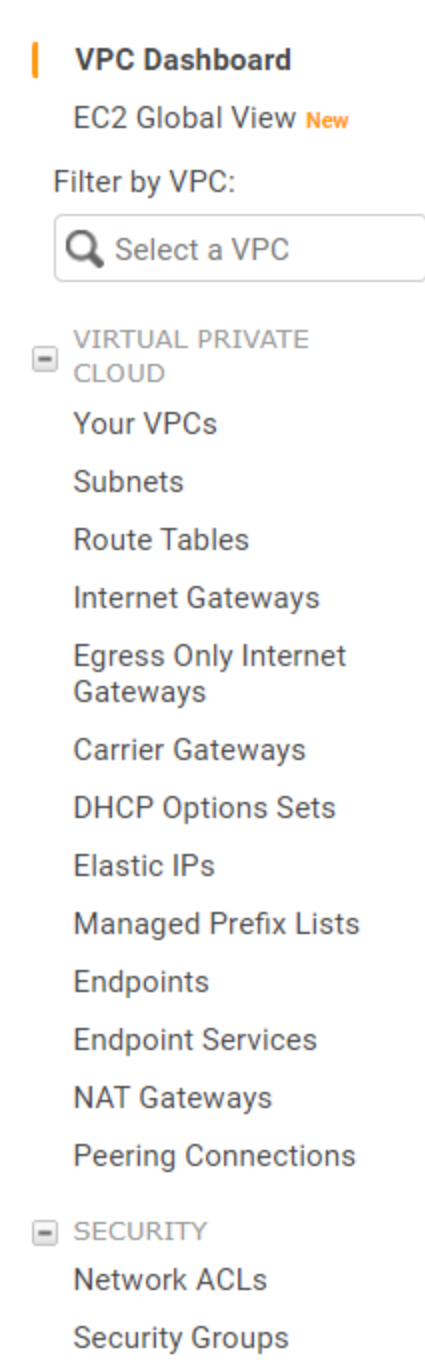


Figure: A great guide to building a VPC is to follow the left hand navigation pane, starting from "Your VPCs" and working your way down.

Before you start, let's review VPC and its components to make it network compatible.

- A **Virtual Private Gateway (VPC)** is like a data center but in the cloud. Its logically isolated from other virtual networks from which you can spin up and launch your AWS resources within minutes.
- **Private Internet Protocol (IP)** addresses are how resources within the VPC communicate with each other. An instance needs a public IP address for it to communicate outside the VPC. The VPC will need networking resources such as an Internet Gateway (IGW) and a route table in order for the instance to reach the internet.
- An **Internet Gateway (IGW)** is what makes it possible for the VPC to have internet connectivity. It has two jobs: perform network address translation (NAT) and be the target to route traffic to the internet for the VPC. An IGW's route on a route table is always 0.0.0.0/0.
- A **subnet** is a range of IP addresses within your VPC.

- A **route table** contains routes for your subnet and directs traffic using the rules defined within the route table. You associate the route table to a subnet. If an IGW was on a route table, the destination would be 0.0.0.0/0 and the target would be IGW.
- **Security groups** and **Network Access Control Lists (NACLs)** work as the firewall within your VPC. Security groups work at the instance level and are stateful, which means they block everything by default. NACLs work at the subnet level and are stateless, which means they do not block everything by default.

Steps

5. Select the **AWS** button located in the top right of the Vocareum home environment. This will open the AWS console in a new tab.
6. Once in the AWS console, click **VPC** under **Recently visited services**. If it is not there, navigate to the top left corner, and select **VPC** under **Networking and Content Delivery** in the **Services** navigation pane.

AWS Management Console

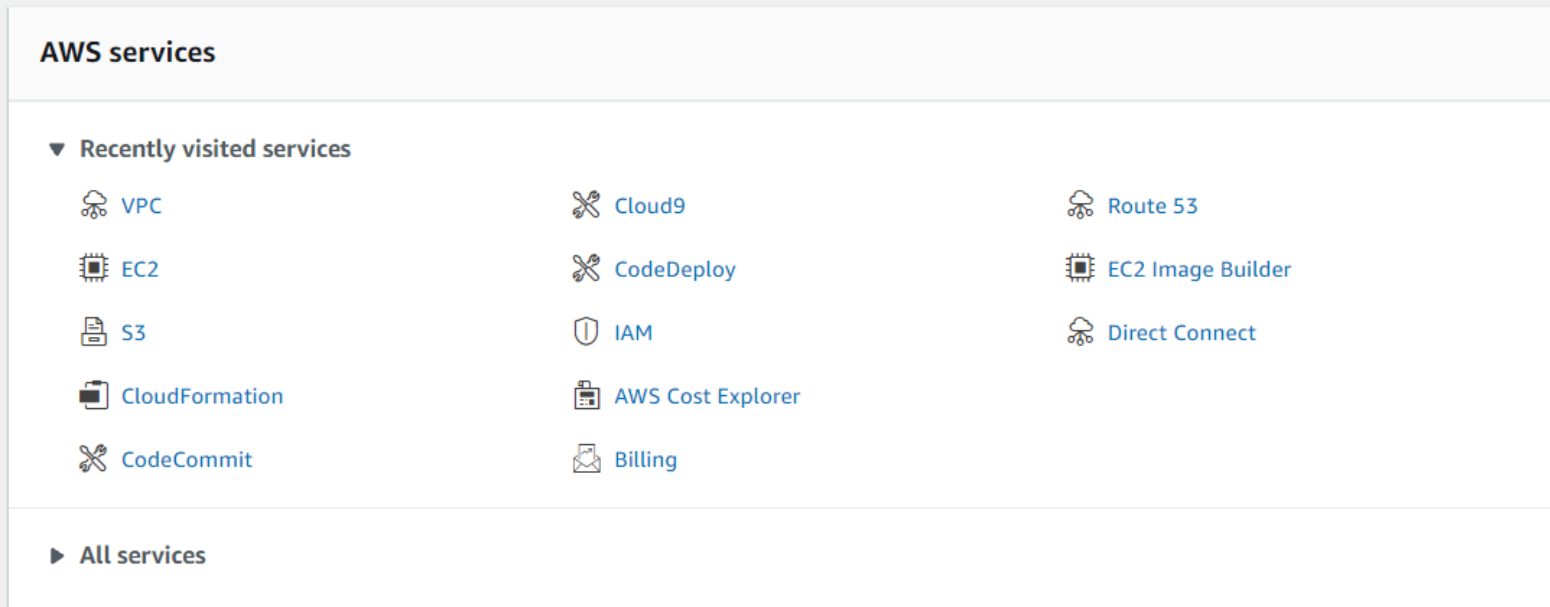


Figure: Recently visited services in the AWS console

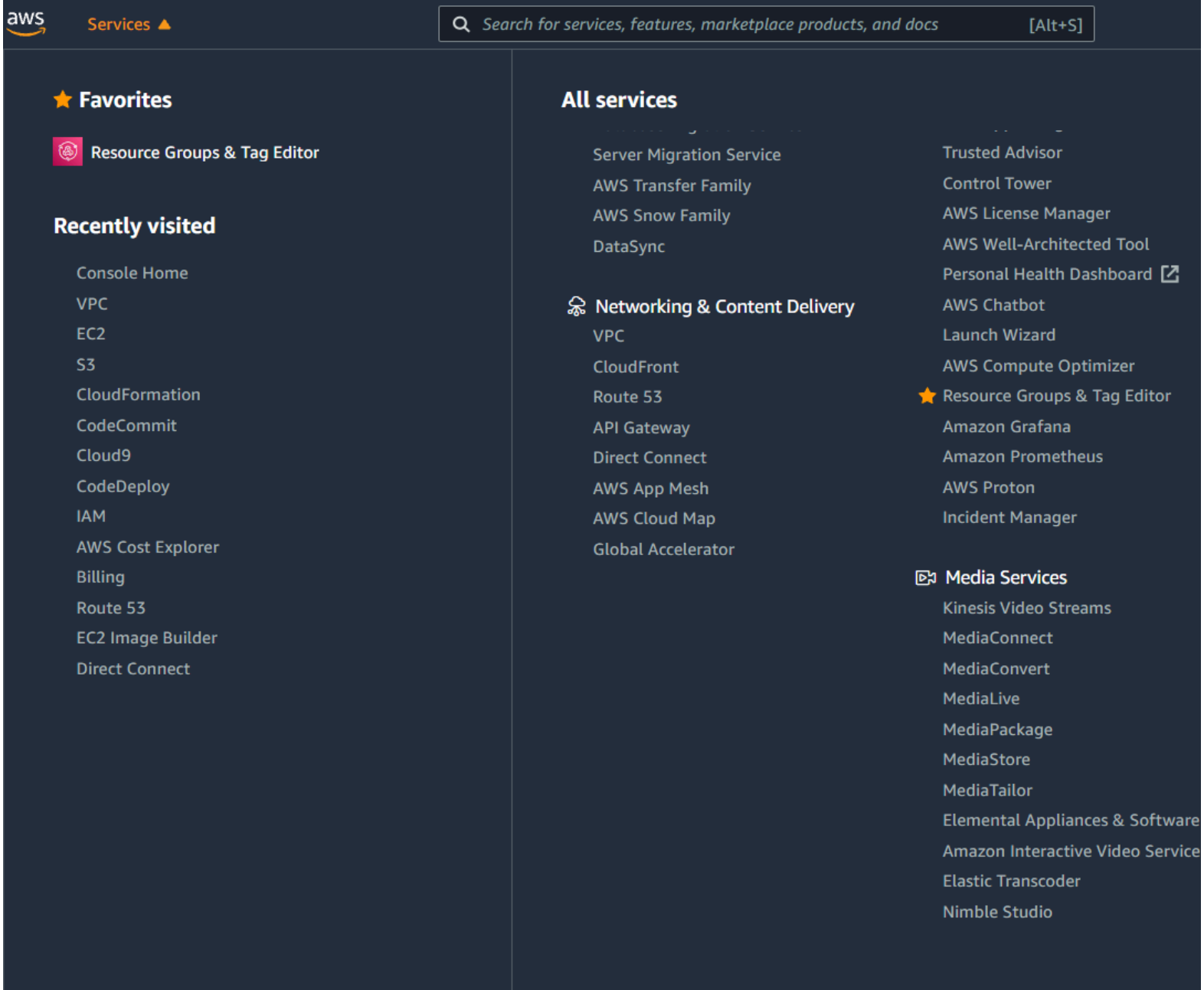


Figure: Services navigation drop down

Creating the VPC

► Recall

7. Start at the top of the left navigation pane at **Your VPCs** and work your way down. Select **Your VPCs**, navigate to the top right corner, and select **Create VPC**.

Note

Note, you will be using a top-down theory with the top being the VPC.

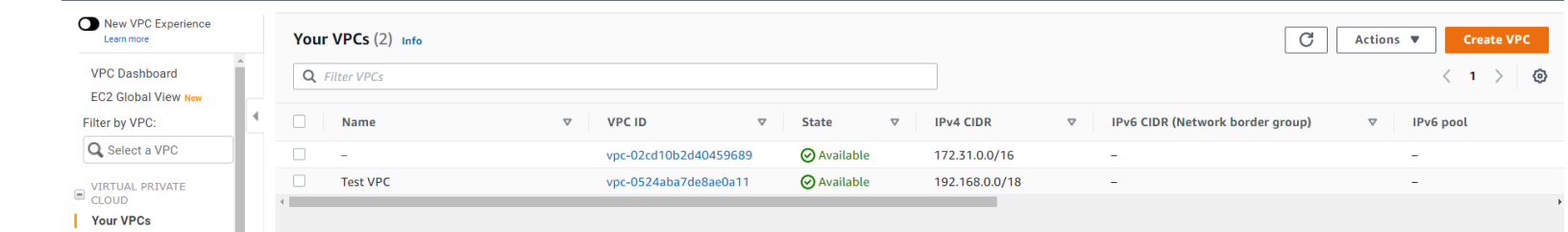


Figure: Navigate to "Your VPCs" and select Create VPC.

8. Name the VPC: `Test VPC`

IPv4 CIDR block: `192.168.0.0/18`

9. Leave everything else as default, and select **Create VPC**.

VPC Successfully Created

Your VPC has been successfully created.

You can launch instances into the subnets of your VPC. For more information, see [Launching an Instance into Your Subnet](#).

Figure: VPC settings configuration

Creating Subnets

► Recall

10. Now that the VPC is complete, look at the left navigation pane and select **Subnets**. In the top right corner, select **Create subnet**.

Note

Please note: Although almost anything can be created in any order, it is easier to have an approach. Having a flow or an approach will assist you in troubleshooting issues and ensure that you do not forget a resource.

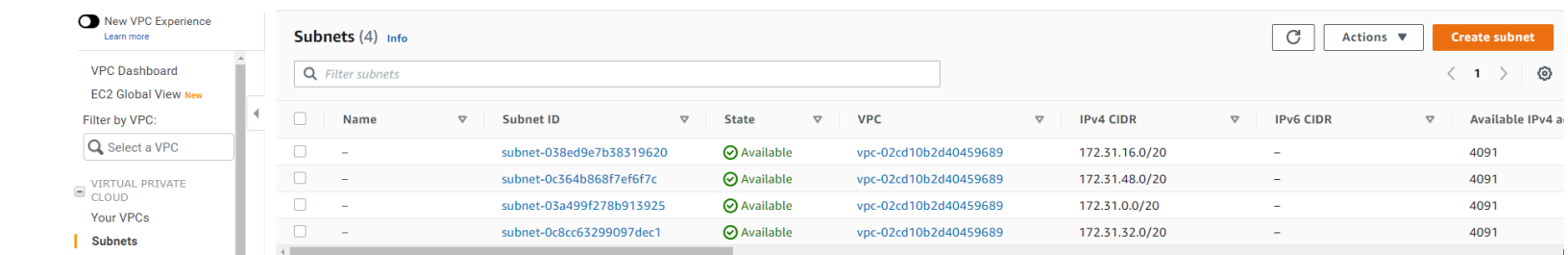


Figure: Select Create subnet

11. Configure like the following picture:

VPC

VPC ID

Create subnets in this VPC.

vpc-0524aba7de8ae0a11 (Test VPC) ▼

Associated VPC CIDRs

IPv4 CIDRs

192.168.0.0/18

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.

Public subnet

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 CIDR block [Info](#)

🔍 192.168.1.0/28 ✕

▼ Tags - optional

Key

🔍 Name ✕

Value - optional

🔍 Public subnet ✕

Remove

Figure: Subnet configuration

Create Route Table

► Recall

12. Navigate to the left navigation pane, and select **Route Tables**. In the top right corner select **Create route table**.

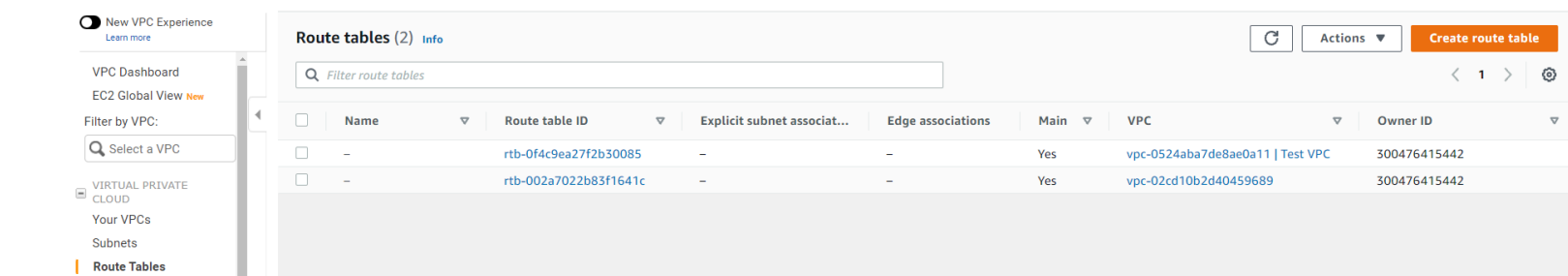


Figure: Select Create route table.

13. Configure like the following picture:

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.

Public route table

VPC
The VPC to use for this route table.

vpc-0524aba7de8ae0a11 (Test VPC)

Tags

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

Key **Value - optional**

Q Name X Q Public route table X Remove

Add new tag

You can add 49 more tags.

Cancel Create route table

Figure: Route table configuration

Create Internet Gateway and attach Internet Gateway

14. From the left navigation pane, select **Internet Gateways**. Create an Internet Gateway (IGW) by selecting **Create internet gateway** at the top right corner.

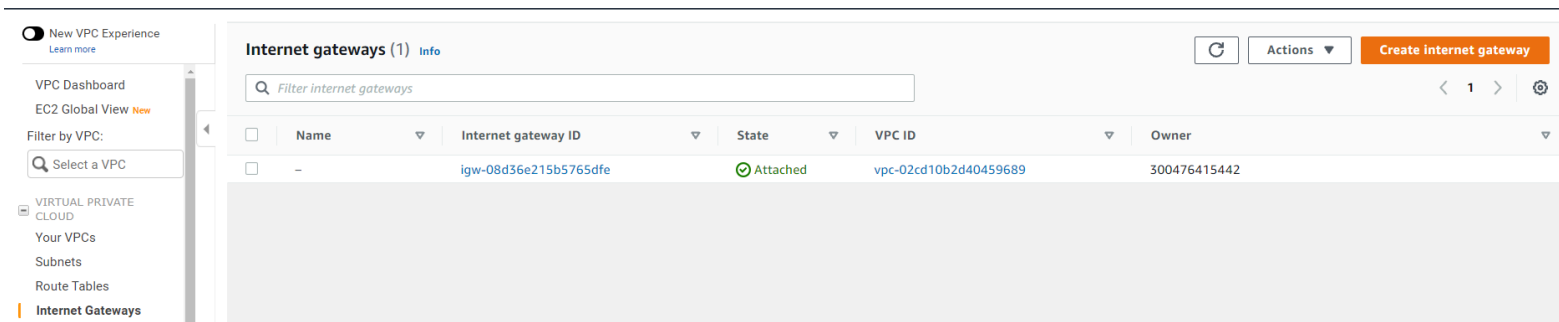


Figure: Select Create internet gateway

15. Configure like the following picture:

VPC > Internet gateways > Create internet gateway

Create internet gateway [Info](#)

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

Internet gateway settings

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

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="text" value="IGW test VPC"/>	<input type="button" value="Remove"/>

You can add 49 more tags.

Figure: Internet gateway configuration

16. Once created, attach the **Internet Gateway** to the VPC by selecting **Actions** at the top right corner and clicking **Attach to VPC**.

VPC > Internet gateways > igw-070bd47bf43135aec

igw-070bd47bf43135aec / IGW test VPC

Details Info

Internet gateway ID igw-070bd47bf43135aec	State ⊖ Detached	VPC ID -	Owner 300476415442
--	---------------------	-------------	-----------------------

Actions ▲

- Attach to VPC
- Detach from VPC
- Manage tags
- Delete

Figure: Attaching the IGW that was just created.

Now your IGW is attached! You now need to add its route to the route table and associate the subnet you created to the route table.

Add route to route table and associate subnet to route table

17. Navigate to the **Route Table** section on the left navigation pane. Select **Public Route Table**, and the scroll to the bottom and select the **Routes** tab. Select the Edit routes button located in the routes box.

On the Edit routes page, the first IP address is the local route and cannot be changed.

Select **Add route**.

- In the **Destination** section, type **0.0.0.0/0** in the search box. This is the route to the IGW. You are telling the route table that any traffic that needs internet connection will use 0.0.0.0/0 to reach the IGW so that it can reach the internet.
- Click in the **Target** section and select **Internet Gateway** since you are targeting any traffic that needs to go to the internet to the IGW. Once you select the IGW, you will see your **TEST VPC IGW** appear. Select that IGW, navigate to the bottom right, and select **Save changes**.

VPC > Route tables > rtb-0769adc74f6636bef > Edit routes

Edit routes

Destination	Target	Status	Propagated
192.168.0.0/18	Q local X	✓ Active	No
Q 0.0.0.0/0 X	Q igw- X igw-070bd47bf43135aec (IGW test VPC)	-	No
Add route			

Cancel Preview Save changes

Figure: Adding the IGW in the route table (0.0.0.0/0 as the destination and IGW as the target).

Now your traffic has a route to the internet via the IGW.

18. From the Public route table dashboard, select the **Subnet associations** tab. Select the **Edit subnet associations** button.

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/1)

<input checked="" type="checkbox"/>	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/>	Public subnet	subnet-09a3b15dff7bdb6e5	192.168.1.0/28	–	Main (rtb-0f4c9ea27f2b30085)

Selected subnets

subnet-09a3b15dff7bdb6e5 / Public subnet
 ✕

Cancel
Save associations

Figure: Associate the Public subnet and select save association.

19. Select **Save association**.

Note: Every route table needs to be associated to a subnet. You are now associating this route table to this subnet. As you probably noticed, the naming convention is kept the same (public route table, public subnet, etc) in order to associate the same resources together. Keep this in mind when your network and resources grow. You can have multiples of the same resources and it can get confusing to which belongs where.

Creating a Network ACL

► Recall

20. From the left navigation pane, select **Network ACLs**. Navigate to the top right corner and select **Create network ACL** to create a Network Access Control Lists (NACLs).

New VPC Experience

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

Carrier Gateways

DHCP Options Sets

Elastic IPs

Managed Prefix Lists

Endpoints

Endpoint Services

NAT Gateways

Peering Connections

SECURITY

Network ACLs

Security Groups

REACHABILITY

DNS FIREWALL

NETWORK FIREWALL

VIRTUAL PRIVATE NETWORK (VPN)

TRANSIT GATEWAYS

TRAFFIC MIRRORING

Network ACLs (1/2) Info

Filter network ACLs

<input type="checkbox"/>	Name	Network ACL ID	Associated with	Default	VPC ID	Inbound rules count
<input type="checkbox"/>	–	acl-078b22b8f69bd038e	4 Subnets	Yes	vpc-02cd10b2d40459689	2 Inbound rules
<input checked="" type="checkbox"/>	–	acl-0c75d86131fc2b175	subnet-09a3b15dff7bdb6e5 / Public subnet	Yes	vpc-0524aba7de8ae0a11 / Test VPC	2 Inbound rules

acl-0c75d86131fc2b175

Details

Network ACL ID

acl-0c75d86131fc2b175

Associated with

subnet-09a3b15dff7bdb6e5 / Public subnet

Default

Yes

VPC ID

vpc-0524aba7de8ae0a11 / Test VPC

Owner

300476415442

Figure: Select Create network ACL

21. On the **Create network ACL**, configure the following:

- **Name:** `Public Subnet NACL`
- **VPC:** Choose `Test VPC` from dropdown
- Choose **Create network ACL**

22. On the **Network ACLs** option, from the list of ACLs select **Public Subnet ACL**
23. From the tabs below, select **Inbound rules** and then choose **Edit inbound rules**
24. On the **Edit inbound rules**, choose **Add new rule** and configure:
 - Rule number: Enter **100**
 - Type: Choose **All traffic** from dropdown
25. Choose **Save changes**
26. Back on the **Network ACLs** option, ensure that **Public Subnet ACL** is selected
27. Choose **Outbound rules** and then choose **Edit outbound rules*
28. On the **Edit outbound rules**, choose **Add new rule** and configure:
 - Rule number: Enter **100**
 - Type: Choose **All traffic** from dropdown
29. Choose **Save changes**

Inbound After creating the NACL, it will should look like the following. This indicates there is only one rule number, which is 100, that states that all traffic, all protocols, all port ranges, from any source (0.0.0.0/0) are allowed to enter (inbound) the subnet. The asterisk * indicates that anything else that does not match this rule is denied.

<input checked="" type="checkbox"/>	Public Subnet NACL	acl-0c75d86131fc2b175	subnet-09a3b15dff7bdb6e5 / Public subnet	Yes	vpc-0524aba7de8ae0a11 / Test VPC	2 Inbound rules
-------------------------------------	--------------------	-----------------------	--	-----	----------------------------------	-----------------

acl-0c75d86131fc2b175 / Public Subnet NACL

Details

Inbound rules

Outbound rules

Subnet associations

Tags

You can now check network connectivity with Reachability Analyzer

Run Reachability Analyzer

×

Inbound rules (2)

Edit inbound rules

Filter inbound rules

< 1 > ⚙

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

Figure: Default inbound rule configuration for NACL. This will allow all traffic from anywhere and deny anything else that does not match this rule at the subnet level.

Outbound What do you think this rule says?

acl-0c75d86131fc2b175 / Public Subnet NACL

Details

Inbound rules

Outbound rules

Subnet associations

Tags

You can now check network connectivity with Reachability Analyzer

Run Reachability Analyzer

Outbound rules (2)

Edit outbound rules

Filter outbound rules

<

1

>

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

Figure: Default outbound rule configuration for NACL. This will allow all traffic from anywhere and deny anything else that does not match this rule at the subnet level.

Creating a Security Group

► Recall

21. From the left navigation pane, select **Security Groups**. Navigate to the top right corner and select **Create security group** to create a security group.

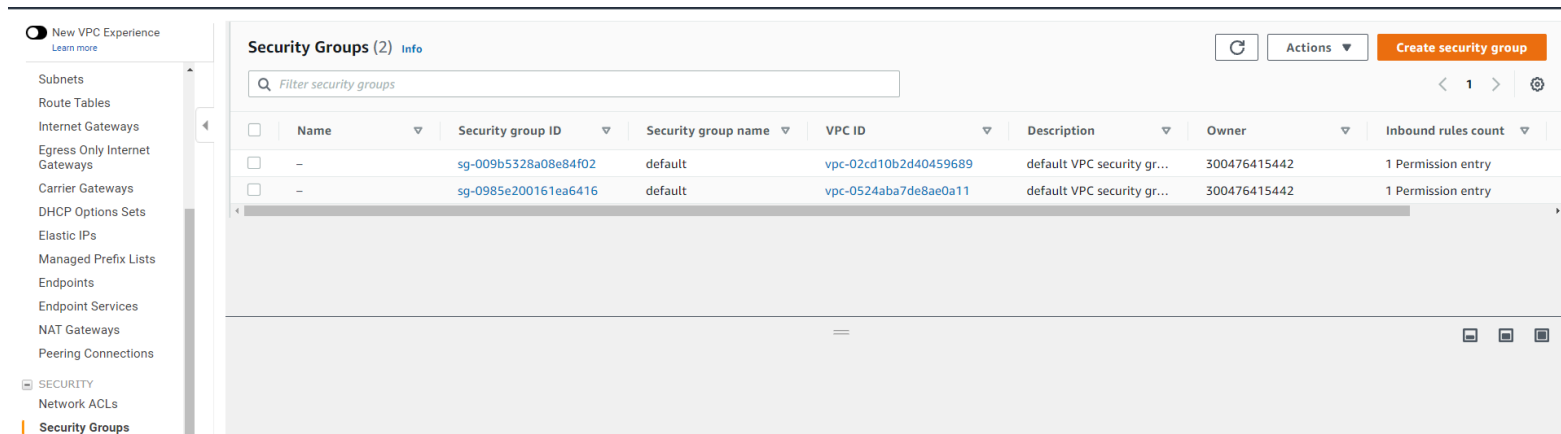


Figure: Select Create security group

Configure like the following image of the Basic details page:

Note: In the VPC portion, remove the current VPC, and select **Test VPC**.

Basic details

Security group name

public security group

Name cannot be edited after creation.

Description

allows public access

VPC

Q vpc-0524aba7de8ae0a11

X

Figure: Configure the Basic details page

The completed security group is shown below. This indicates that for **Inbound rules** you are allowing SSH, HTTP, and HTTPS types of traffic, each of which has its own protocols and port range. The source from which this traffic reaches your instance can be originating from anywhere. For **Outbound rules**, you are allowing all traffic from outside your instance.

Inbound rules

Info

Type	Info	Protocol	Info	Port range	Info	Source	Info	Description - optional	Info
SSH		TCP		22		Anywher...			Delete
							0.0.0.0/0		
HTTP		TCP		80		Anywher...			Delete
							0.0.0.0/0		
HTTPS		TCP		443		Anywher...			Delete
							0.0.0.0/0		

Add rule

Outbound rules

Info

Type	Info	Protocol	Info	Port range	Info	Destination	Info	Description - optional	Info
All traffic		All		All		Custom			Delete
							0.0.0.0/0		

Add rule

Figure: Configuration details for inbound and outbound rules for the security group

You now have a functional VPC. The next task is to launch an EC2 instance to ensure that everything works.

Task 2: Launch EC2 instance and SSH into instance

In task 2, you will launch an EC2 instance within your Public subnet and test connectivity by running the command **ping**. This will validate that your infrastructure is correct, such as security groups and network ACLs, to ensure that they are not blocking any traffic from your instance to the internet and vice versa. This will validate that you have a route to the IGW via the route table and that the IGW is attached.

22. On the AWS Management Console, in the **Search** bar, enter and choose **EC2** to go to the **EC2 Management Console**.

23. In the left navigation pane, choose **Instances**.

24. Choose **Launch instances** and configure the following options:

- In the **Name and tags** section, leave the Name blank.
- In the **Application and OS Images (Amazon Machine Image)** section, configure the following options:
 - Quick Start:** Choose **Amazon Linux**.
 - Amazon Machine Image (AMI):** Choose **Amazon Linux 2023 AMI**.
- In the **Instance type** section, choose **t3.micro**.
- In the **Key pair (login)** section, choose **vockey**.

41. In the **Network settings** section, choose Edit and configure the following options:

- VPC - required:** Choose **Test VPC**.
- Subnet:** Choose **Public Subnet**.
- Auto-assign public IP:** Choose **Enable**.
- Firewall (security groups):** Choose **Select existing security group**.
 - Choose **public security group**.

42. Choose **Launch instance**.

43. To display the launched instance, choose **View all instances**.

□ The EC2 instance named **Bastion Server** is initially in a *Pending* state. The **Instance state** then changes to □ *Running* to indicate that the instance has finished booting.

Use SSH to connect to an Amazon Linux EC2 instance

► Ways to connect Amazon Linux EC2