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

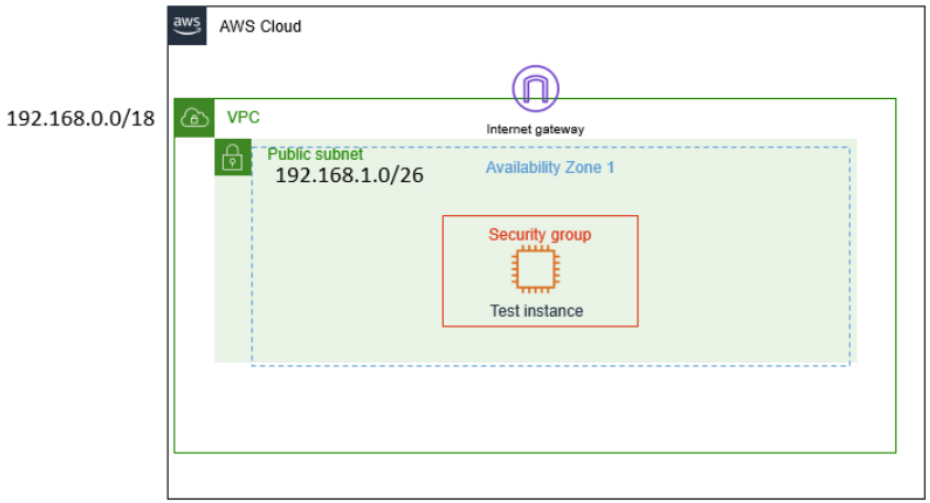
This lab total duration is 60 minutes.

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

264-[NF]-Lab - Networking res

Workbench - Vocareum

labs.vocareum.com/main/main.php?m=editor&asnid=4732158&stepid=4732161&hideNavBar=1

Submit | Details | AWS | Start Lab | End Lab | 00:00:00 | Grades

EN-US

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

▶ **Recall**

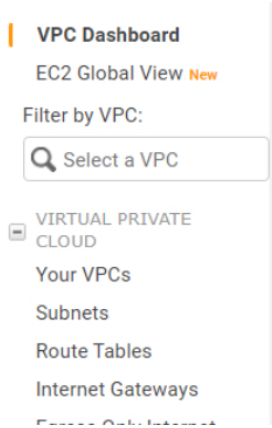
Task 1: Investigate the customer's needs

▼ **Recall**

Recall protocols which can be directly used with AWS's Security Group (SG) and Network Access Control Lists (NACLs). A VPC needs an Internet Gateway (IGW) in order for the VPC to reach the internet, which has the route as 0.0.0.0/0. These routes go on what is called a Route Table, which are associated to subnets so they know where they belong. As mentioned in previous labs, you will follow the order of the navigation console to build this VPC, and a troubleshooting method to build a fully functioning VPC. When building a VPC from scratch, it is easier to work from the top and move down to the bottom since you do not have an instance yet. Think of this as building a sandwich; the VPC is the bun, and the resources are everything in between.

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.



the internet. Keep the VPC CIDR at 192.168.0.0/18 and public subnet CIDR of 192.168.1.0/26.

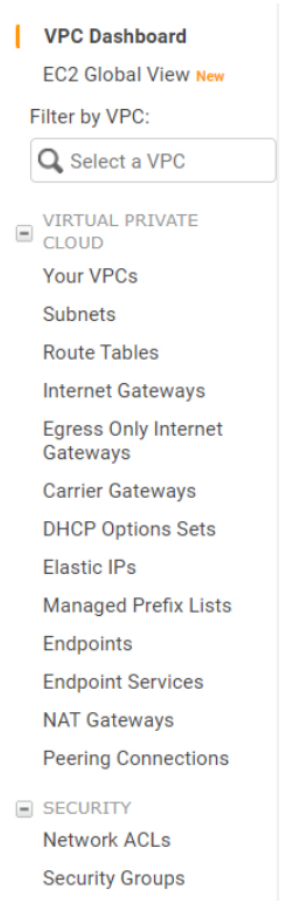


Figure: A great guide to building a VPC is to follow the left hand navigation pane, starting from "Your VPCs" and working your way down.

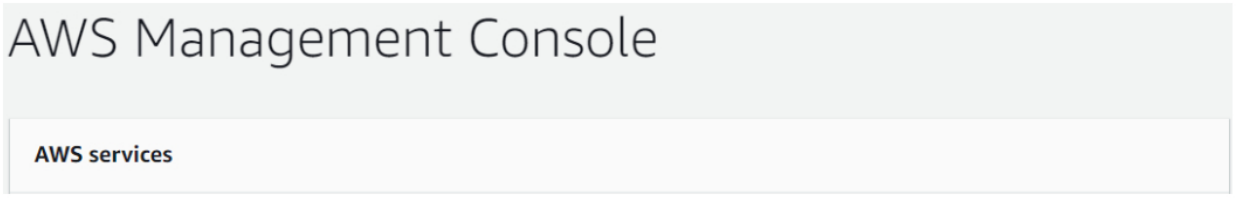
Figure: A great guide to building a VPC is to follow the left hand navigation pane, starting from "Your VPCs" and working your way down.

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

- A **Virtual Private Cloud (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.



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.

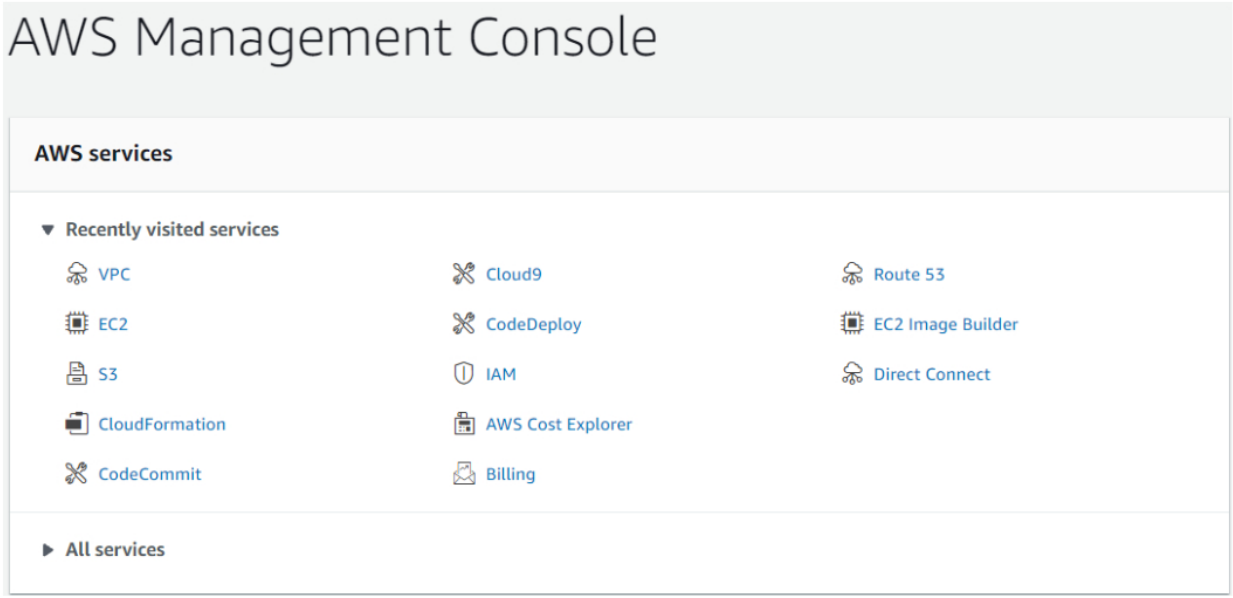
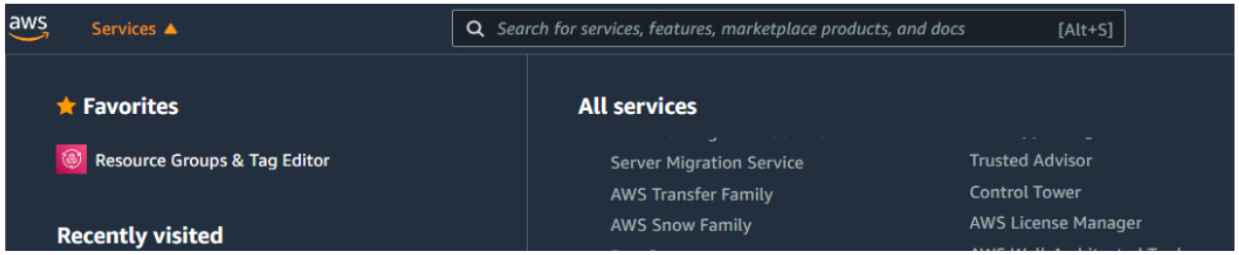


Figure: Recently visited services in the AWS console



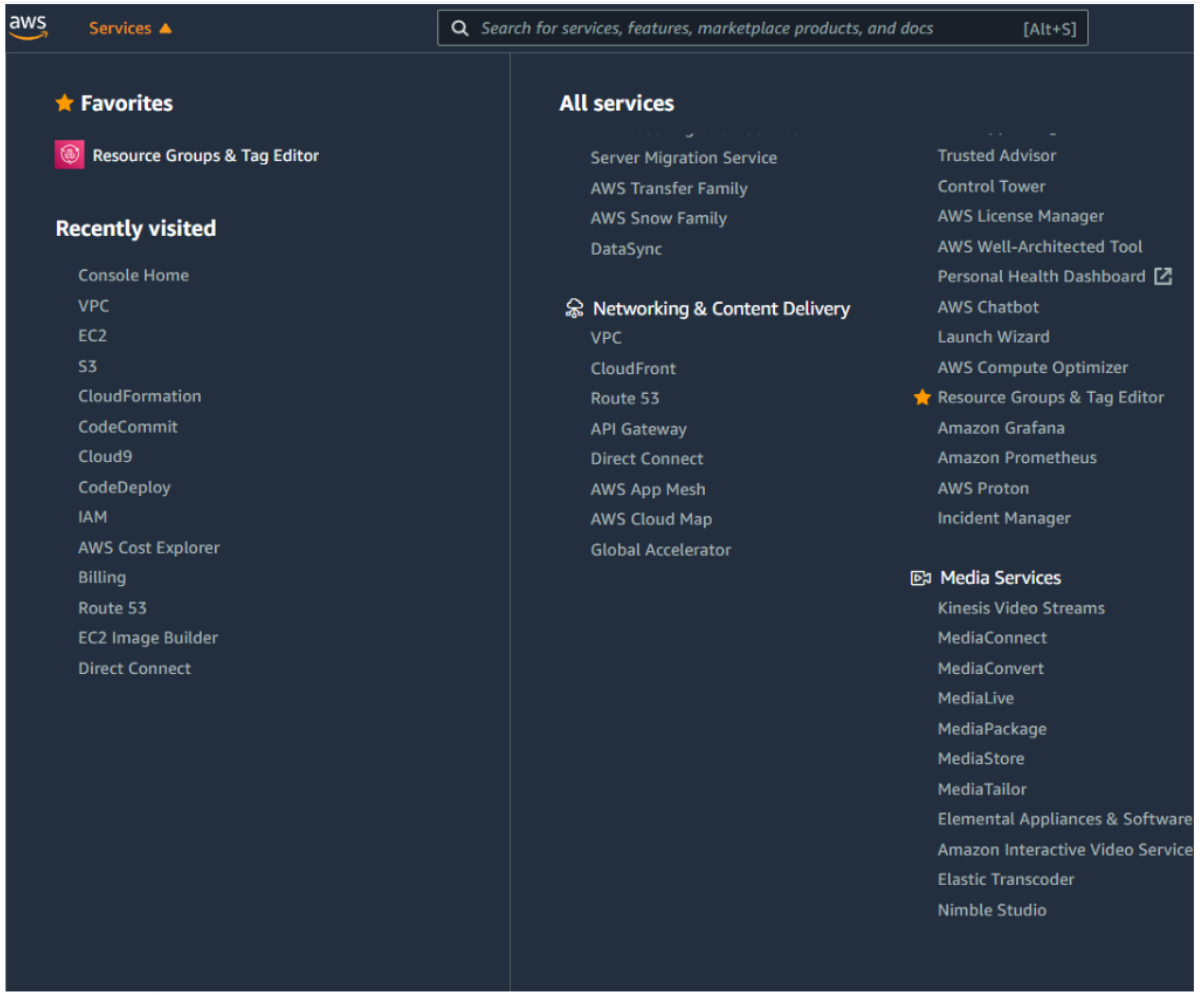


Figure: Services navigation drop down

Creating the VPC

▼ **Recall**

Recall that a VPC is like a data center, but located in the cloud. It's logically isolated from other virtual networks. Now you will build a VPC.

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.

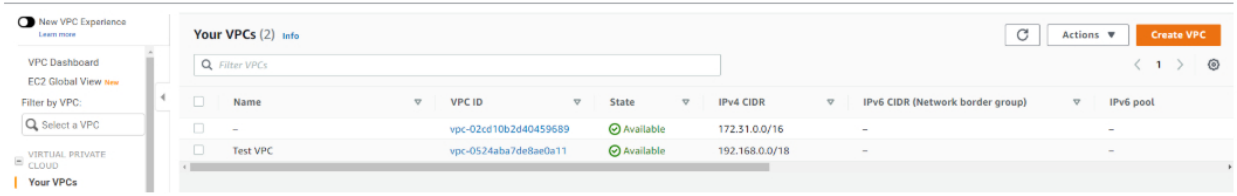
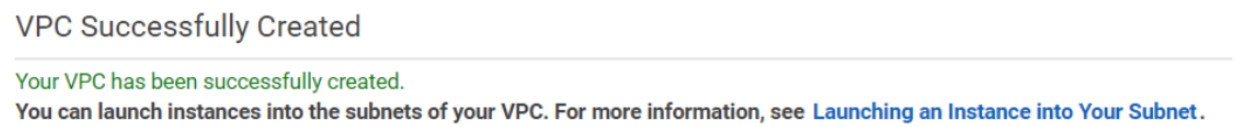


Figure: Navigate to "Your VPCs" and select Create VPC.

8. Name the VPC: `Test VPC`

   IPv4 CIDR block: `192.168.0.0/18`

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



Creating Subnets

▼ **Recall**

Recall that a subnet is a range of IP addresses within your VPC. In your VPC, you can create a public and a private subnet. You can separate subnets according to specific architectural needs. For example, if you have servers that shouldn't be directly accessed by the internet, you would put them in the private subnet. For test servers or instances that require internet connectivity can be placed in the public subnet. Companies also separate subnets according to offices, teams, or floors.

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.

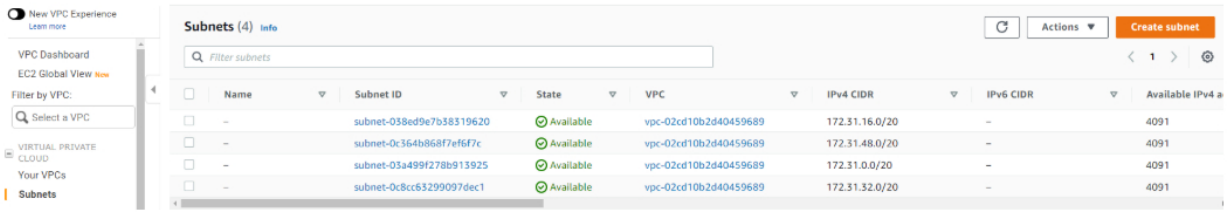


Figure: Select Create subnet

11. Configure like the following picture:

VPC

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	Value - *optional*	
Name	Public subnet	Remove

Figure: Subnet configuration

Create Route Table

▼ Recall

Recall that a route table contains the rules or routes that determine where network traffic within your subnet and VPC will go. It controls the network traffic like a router, and, just like a router, it stores IP addresses within the VPC. You associate a route table to each subnet and put the routes that you need your subnet to be able to reach. For this step, you will create the route table first, and then add the routes as you create AWS resources for the VPC.

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

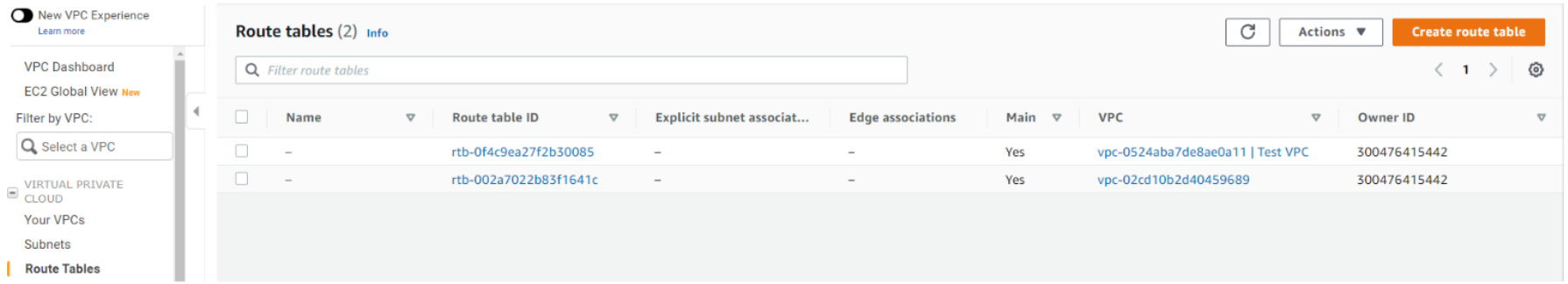
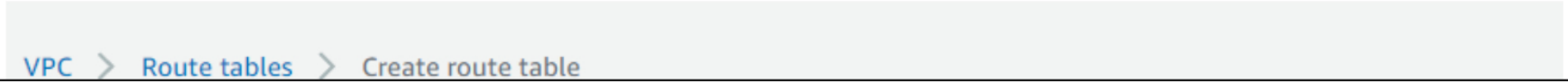


Figure: Select Create route table.

13. Configure like the following picture:



13. Configure like the following picture:

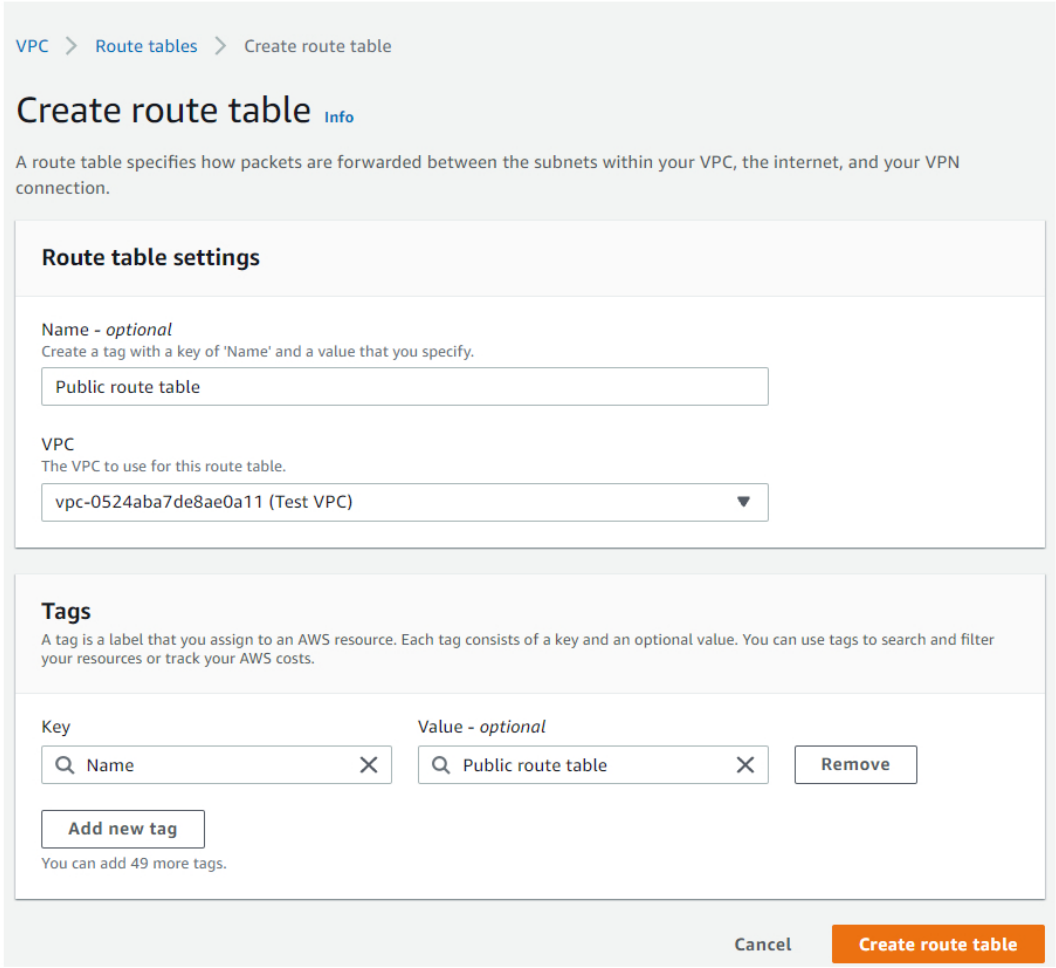


Figure: Route table configuration

Create Internet Gateway and attach Internet Gateway

▼ **In this lab**

Recall that an IGW is what allows the VPC to have internet connectivity and allows communication between resources in your VPC and the internet. The IGW is used as a target in the route table to route internet-routable traffic and to perform network address translation (NAT) for EC2 instances. NAT is a bit beyond the scope of this lab, but it is referenced in the reference section if you'd like to dive deeper.

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

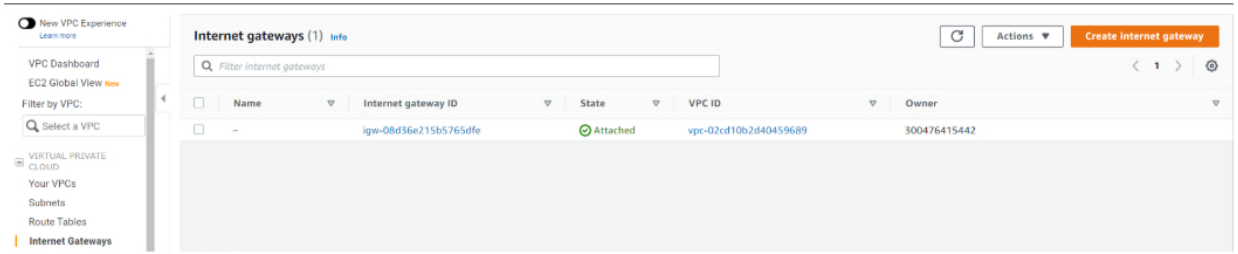


Figure: Select Create internet gateway

15. Configure like the following picture:

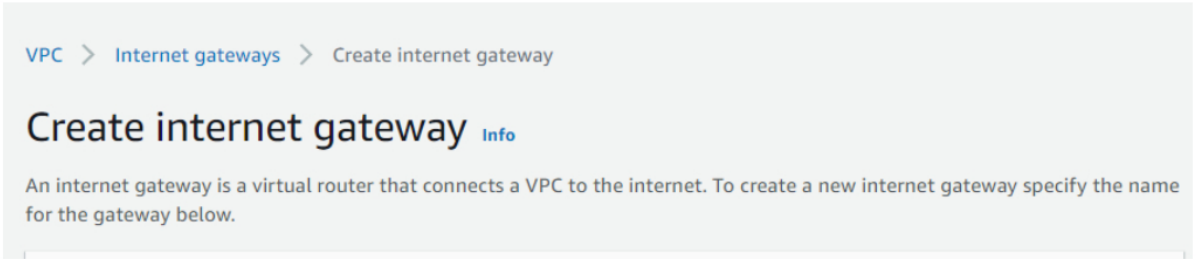


Figure: Select Create internet gateway

15. Configure like the following picture:

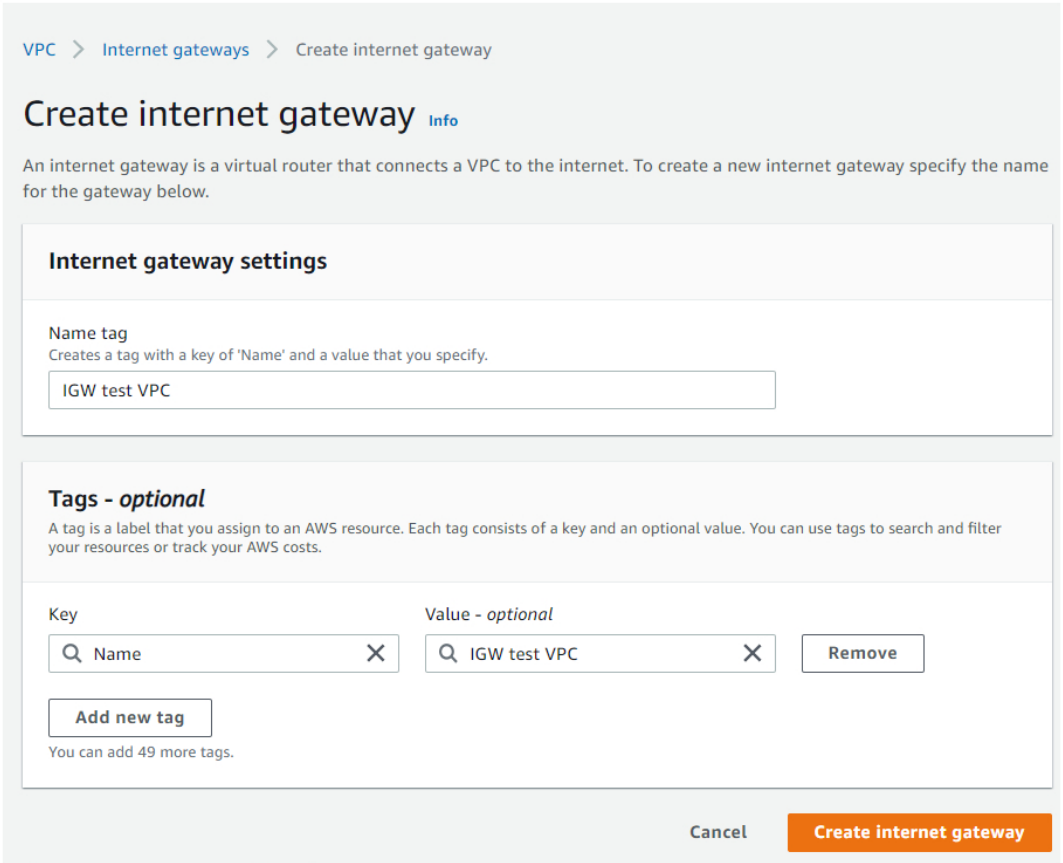


Figure: Internet gateway configuration

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

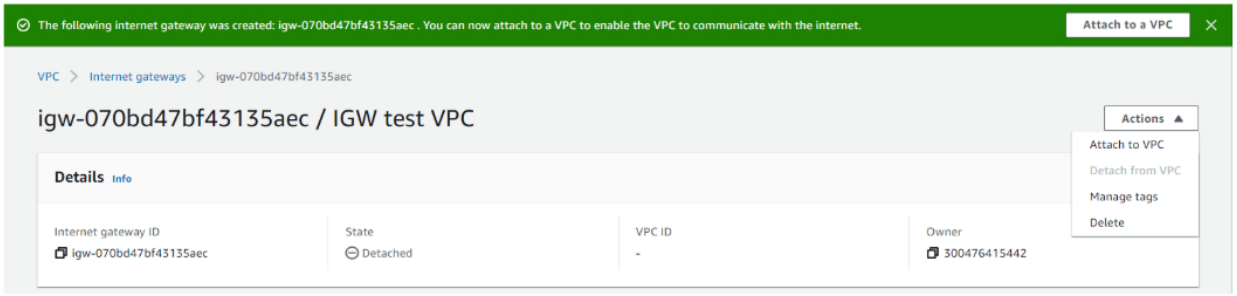


Figure: Attaching the IGW that was just created.

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**.

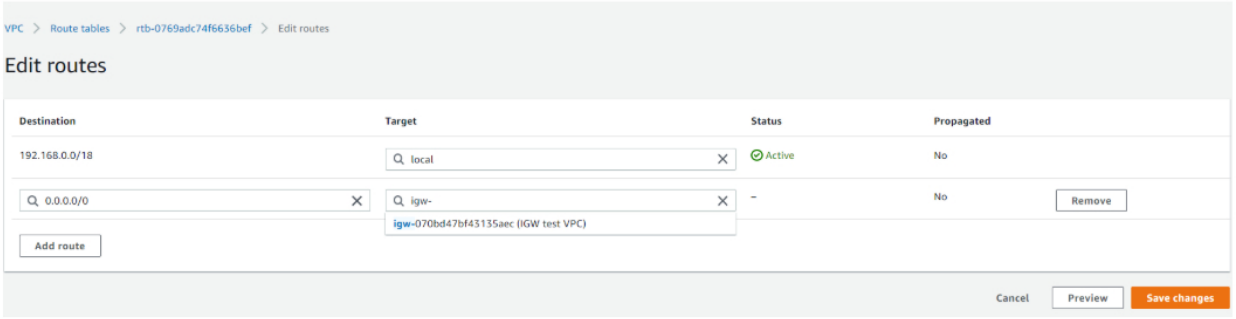


Figure: Adding the IGW in the route table (0.0.0.0/0 as the destination and IGW as the target).

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.

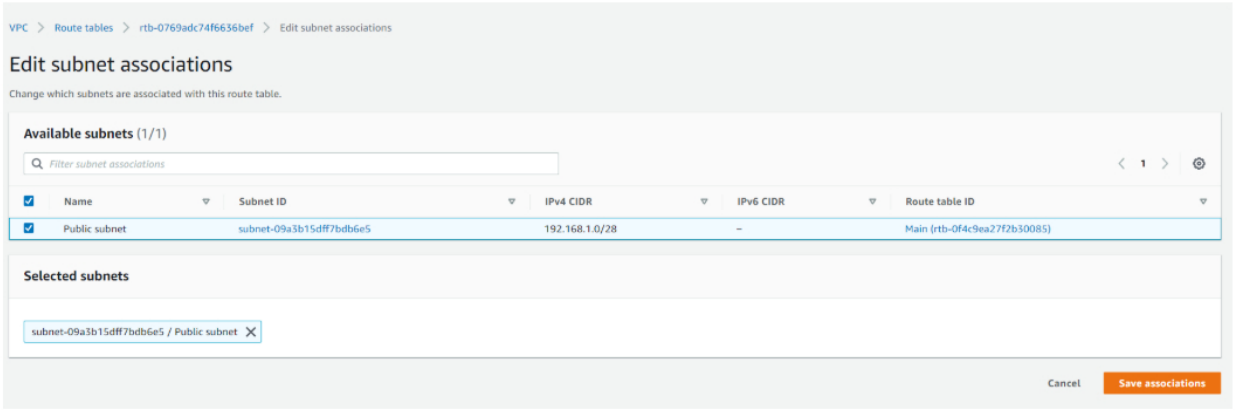


Figure: Associate the Public subnet and select save association.

19. Select **Save assocation**.

> 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

Figure: Associate the Public subnet and select save association.

19. Select **Save assocation**.

> 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**

Recall that an NACL is a layer of security that acts like a firewall at the subnet level. The rules to set up a NACL are similar to security groups in the way that they control traffic. The following rules apply: NACLs must be associated to a subnet, NACLs are stateless, and they have the following parts:
- Rule number: The lowest number rule gets evaluated first. As soon as a rule matches traffic, its applied; for example: 10 or 100. Rule 10 would get evaluated first.
- Type of traffic; for example: HTTP or SSH
- Protocol: You can specify all or certain types here
- Port range: All or specific ones
- Destination: Only applies to outbound rules
- Allow or Deny specified traffic.

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).

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).

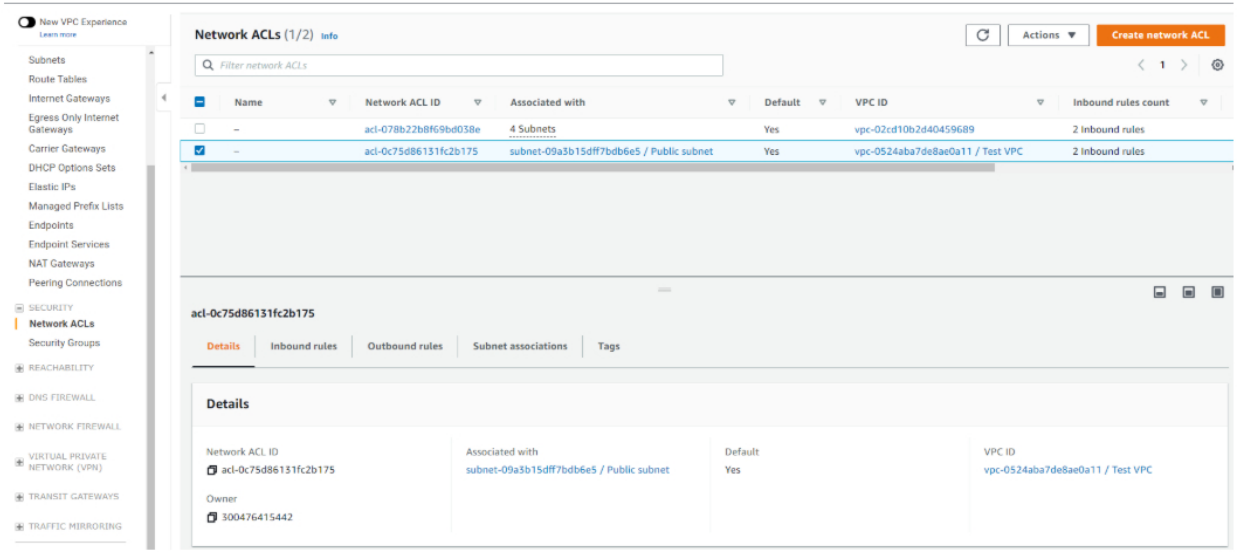


Figure: Select Create network ACL

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`

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.

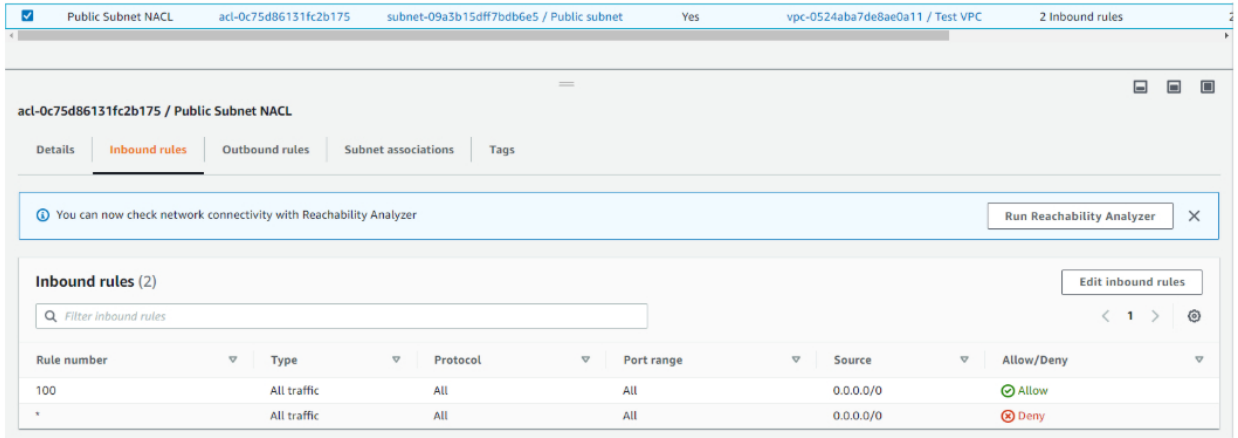


Figure: Default inbound rule configuration for NACL. This will allow all traffic from anywhere and deny anything else that does not

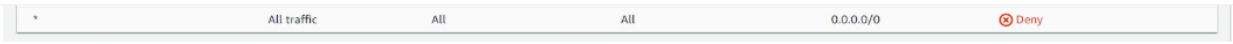


Figure: Default inbound rule configuration for NACL. This will allow all traffic from anywhere and deny anything else that does not match this rule at the subnet level.

Outbound What do you think this rule says?

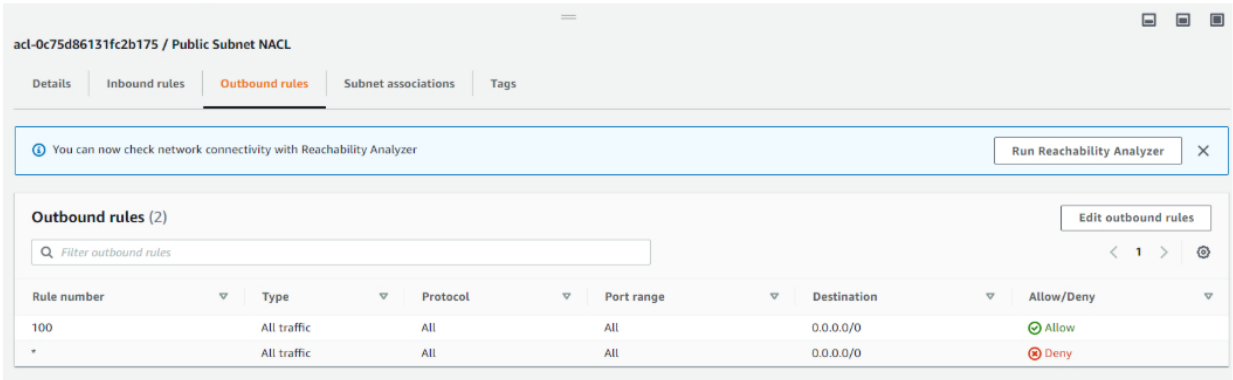


Figure: Default outbound rule configuration for NACL. This will allow all traffic from anywhere and deny anything else that does not match this rule at the subnet level.

Creating a Security Group

▼ **Recall**

Recall that a security group is a virtual firewall at the instance level that controls inbound and outbound traffic. Just like a NACL, security groups control traffic; however, security groups cannot deny traffic. Security groups are stateful; you must allow traffic through the security group as it blocks everything by default, and it must be associated to an instance. A security group has the following parts for both inbound and outbound rules:

- Inbound Source: It can be an IP or a specific resource
- Outbound Destination: Can by an IP such as anywhere (0.0.0.0/0)
- Protocol: Example UDP or TCP

- Outbound Destination: Can by an IP such as anywhere (0.0.0.0/0)
- Protocol: Example UDP or TCP
- Port range: All or specific range
- Description: You can input a description

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

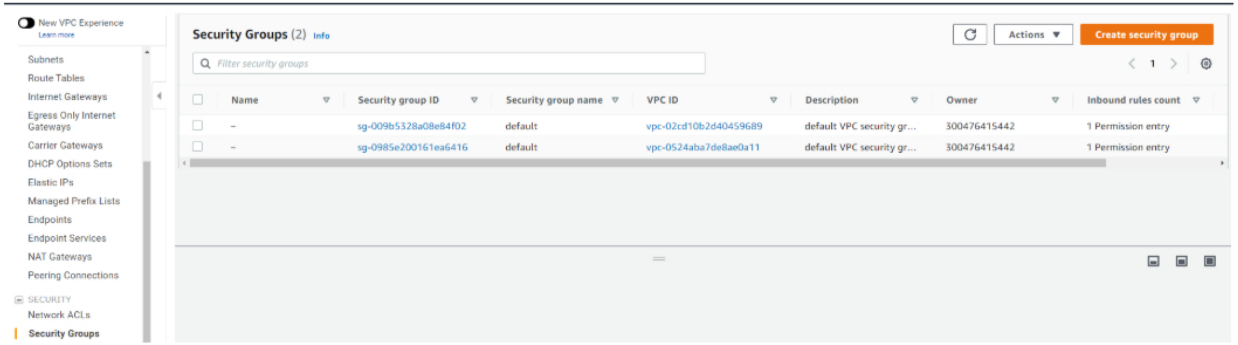


Figure: Select Create security group

Configure like the following image of the Basic details page:

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

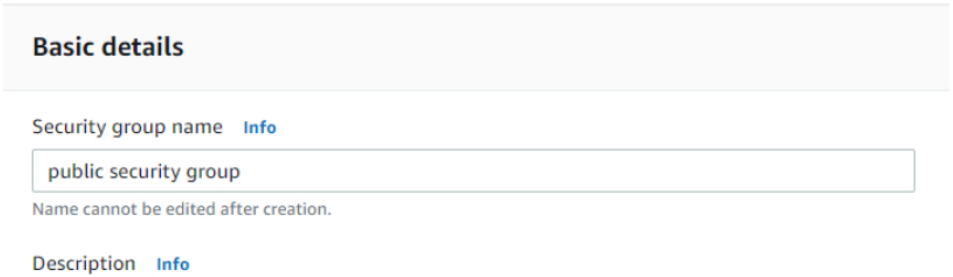


Figure: Configure the Basic details page

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.

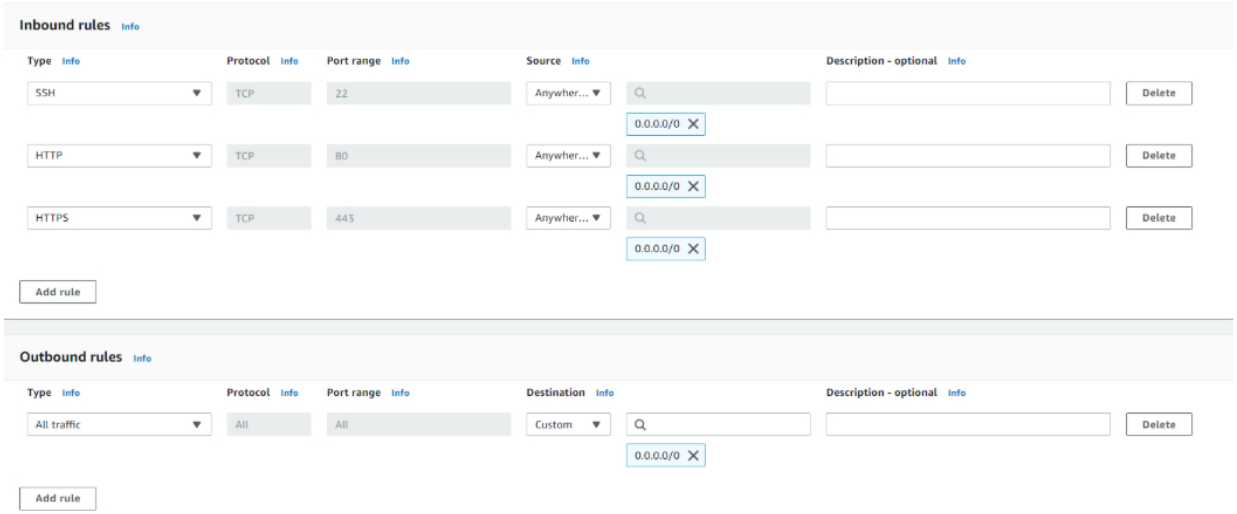


Figure: Configuration details for inbound and outbound rules for the security group

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.

31. On the AWS Management Console, in the **Search** bar, enter and choose `EC2` to go to the **EC2 Management Console**.
32. In the left navigation pane, choose **Instances**.
33. 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**.

34. 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**.

35. Choose **Launch instance**.
36. 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

35. Choose **Launch instance**.
36. 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**

In this task, you will connect to a Amazon Linux EC2 instance. You will use an SSH utility to perform all of these operations.

The following instructions vary slightly depending on whether you are using Windows or Mac/Linux.

Windows Users: Using SSH to Connect

These instructions are specifically for Windows users. If you are using macOS or Linux, skip to the next section.

37. Select the Details drop-down menu above these instructions you are currently reading, and then select Show. A Credentials window will be presented.
38. Select the **Download PPK** button and save the **labsuser.ppk** file.
    Typically your browser will save it to the Downloads directory.
39. Make a note of the **PublicIP** address.
40. Then exit the Details panel by selecting the **X**.
41. Download **PuTTY** to SSH into the Amazon EC2 instance. If you do not have PuTTY installed on your computer, download it here.

Task 3: Use ping to test internet connectivity

53. Run the following command to test internet connectivity:

```
ping google.com
```

After a few seconds, exit ping by holding **CTRL+C** on Windows or **CMD+C** on Mac to exit. You should get the following result:

Successful ping:

```
[ec2-user@ip-192-168-1-8 ~]$ ping google.com
PING google.com (142.250.217.110) 56(84) bytes of data.
64 bytes from sea09s30-in-f14.1e100.net (142.250.217.110): icmp_seq=1 ttl=93 time=6.02 ms
64 bytes from sea09s30-in-f14.1e100.net (142.250.217.110): icmp_seq=2 ttl=93 time=5.96 ms
64 bytes from sea09s30-in-f14.1e100.net (142.250.217.110): icmp_seq=3 ttl=93 time=6.23 ms
64 bytes from sea09s30-in-f14.1e100.net (142.250.217.110): icmp_seq=4 ttl=93 time=6.01 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 5.969/6.060/6.230/0.126 ms
[ec2-user@ip-192-168-1-8 ~]$
```

Run ping to test connectivity. The above results are saying you have replies from google.com and have 0% packet loss.

If you are getting replies back, that means that you have connectivity.

Lab Complete 🎓

🏁 Congratulations! You have completed the lab.

54. Choose ■ **End Lab** at the top of this page, and then select **Yes** to confirm that you want to end the lab.

    A panel indicates that *You may close this message box now. Lab resources are terminating...*

55. Choose the **X** in the upper-right corner to close the **End Lab** panel.