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

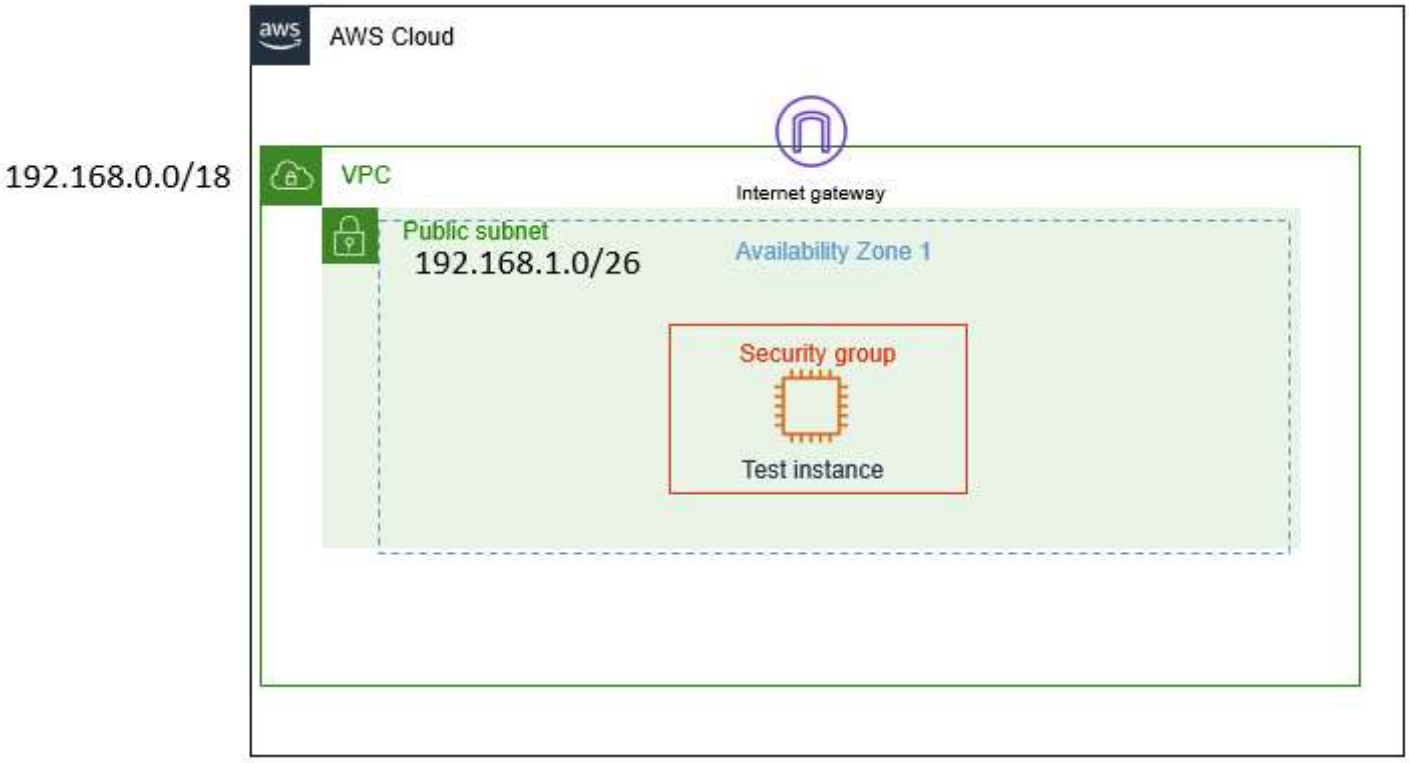
This lab total duration is 60 minutes.

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

▶ **Recall**

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.

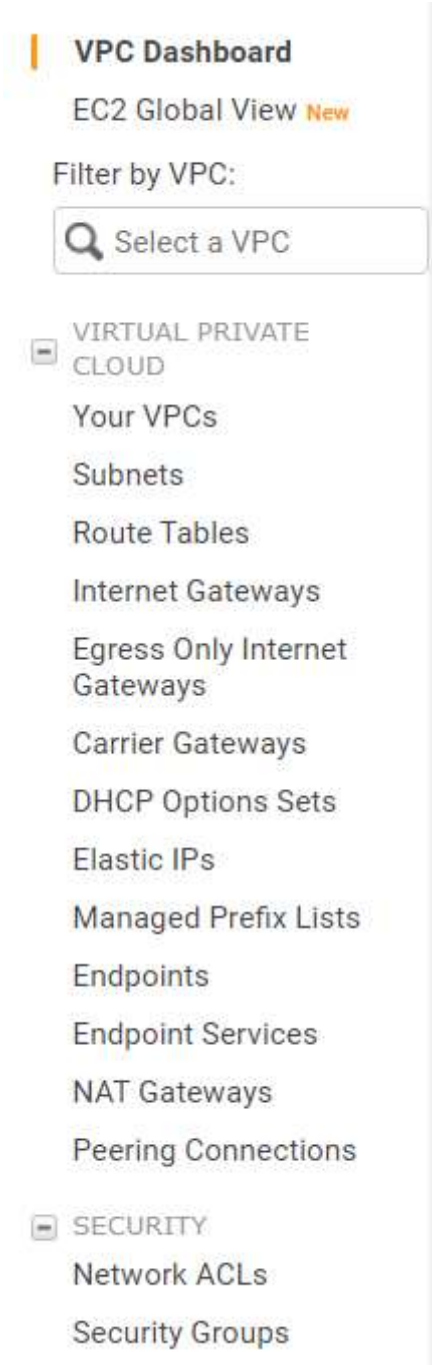


Figure: A great guide to building a VPC is to follow the left hand navigation pane, starting from "Your VPCs" and working your way down.

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.

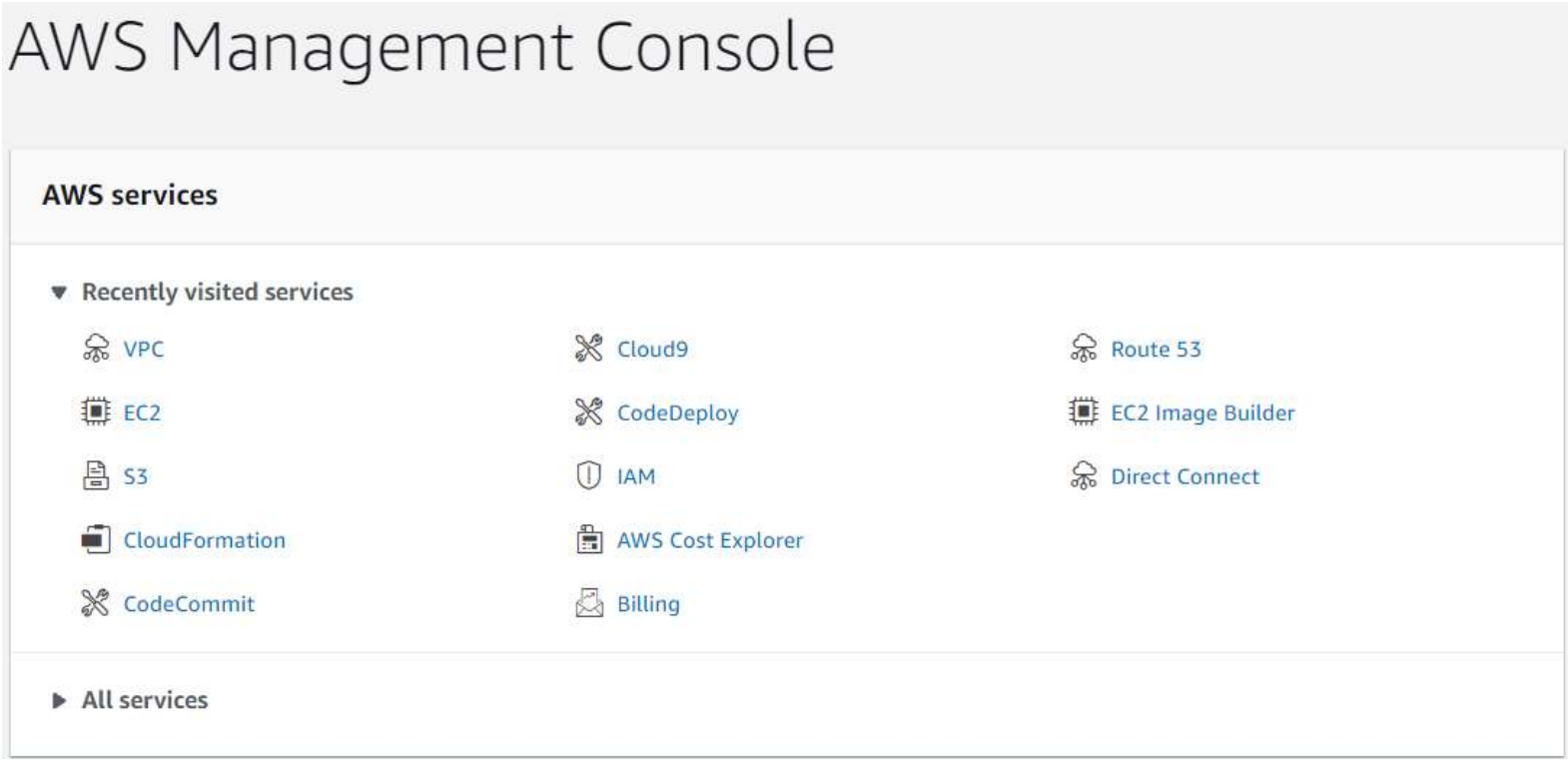


Figure: Recently visited services in the AWS console

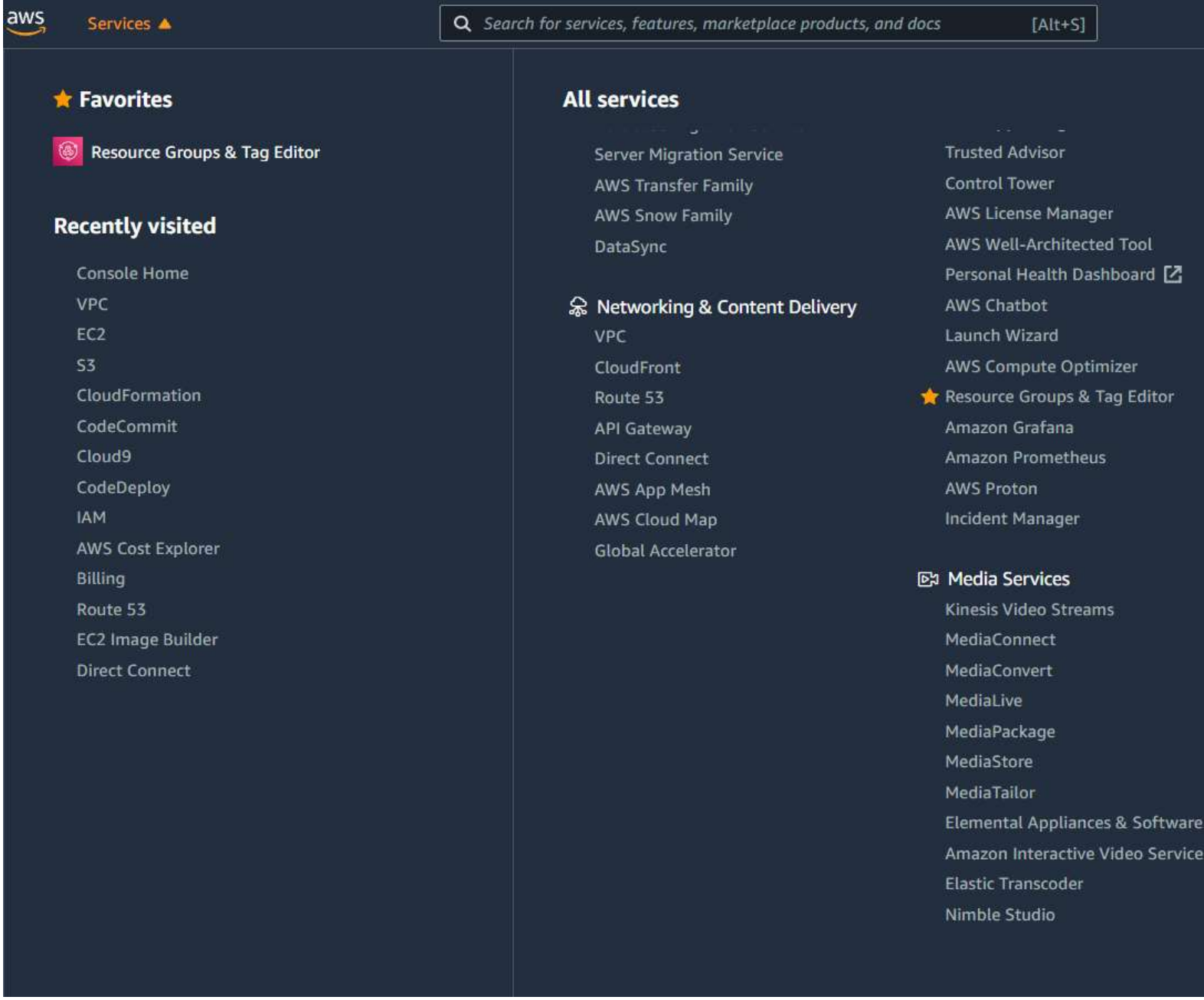


Figure: Services navigation drop down

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.

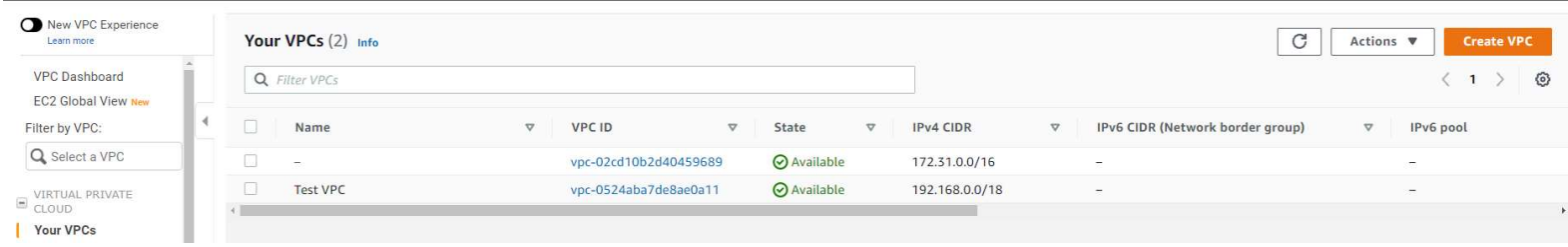


Figure: Navigate to "Your VPCs" and select Create VPC.

8. Name the VPC: `Test VPC`

   IPv4 CIDR block: `192.168.0.0/18`

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

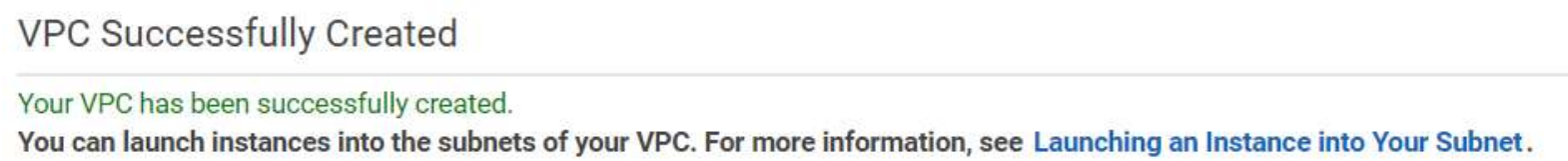


Figure: VPC settings configuration

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.

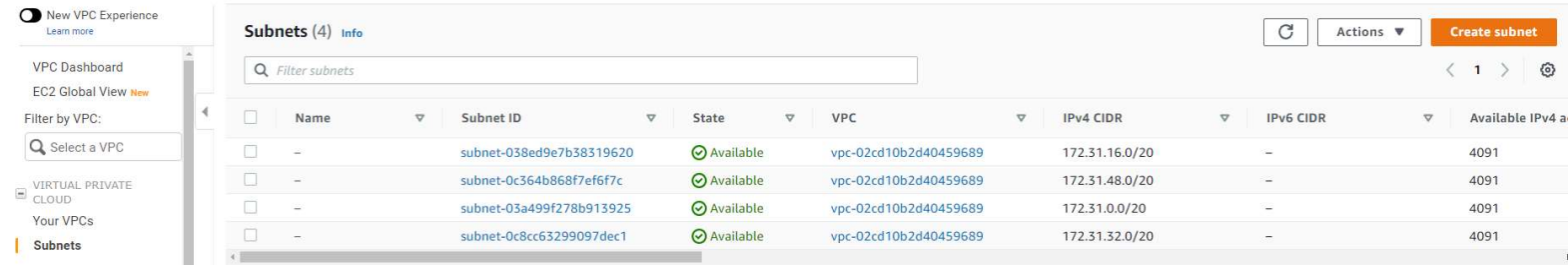


Figure: Select Create subnet

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**.

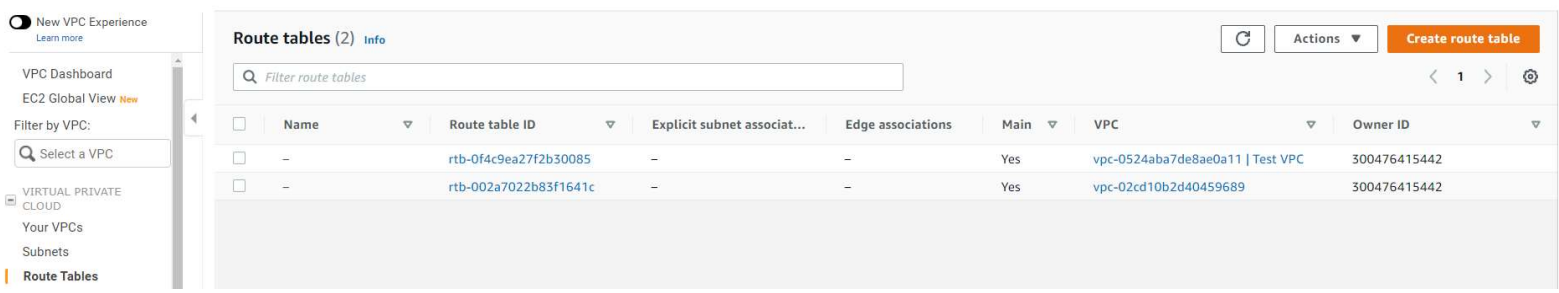


Figure: Select Create route table.

13. Configure like the following picture:

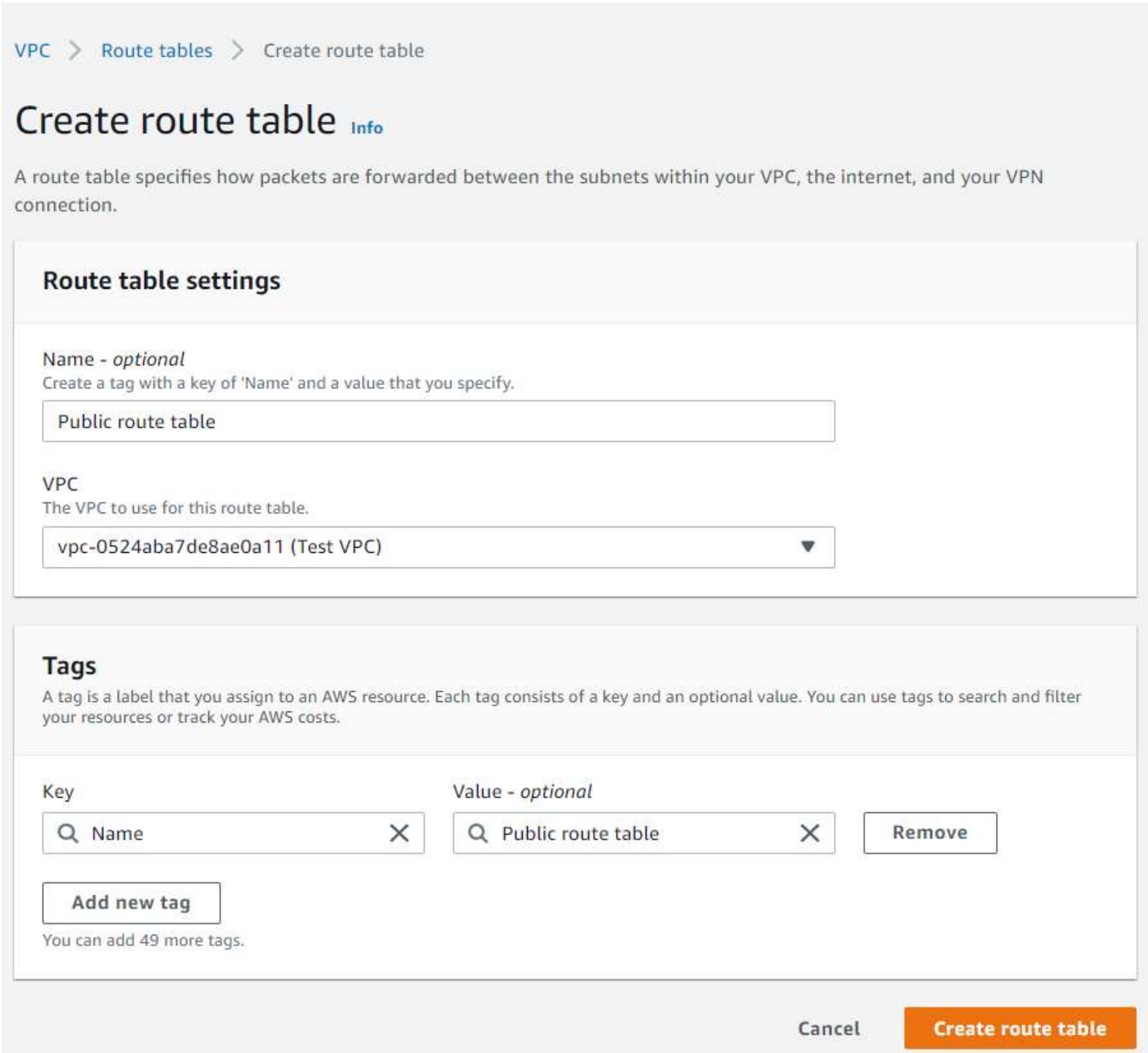


Figure: Route table configuration

Create Internet Gateway and attach Internet Gateway

▶ **In this lab**

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

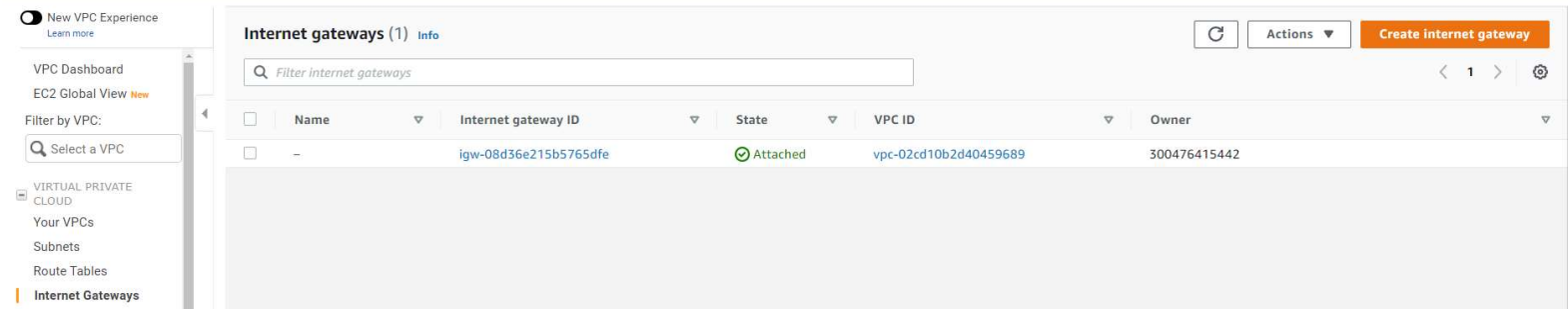


Figure: Select Create internet gateway

15. Configure like the following picture:

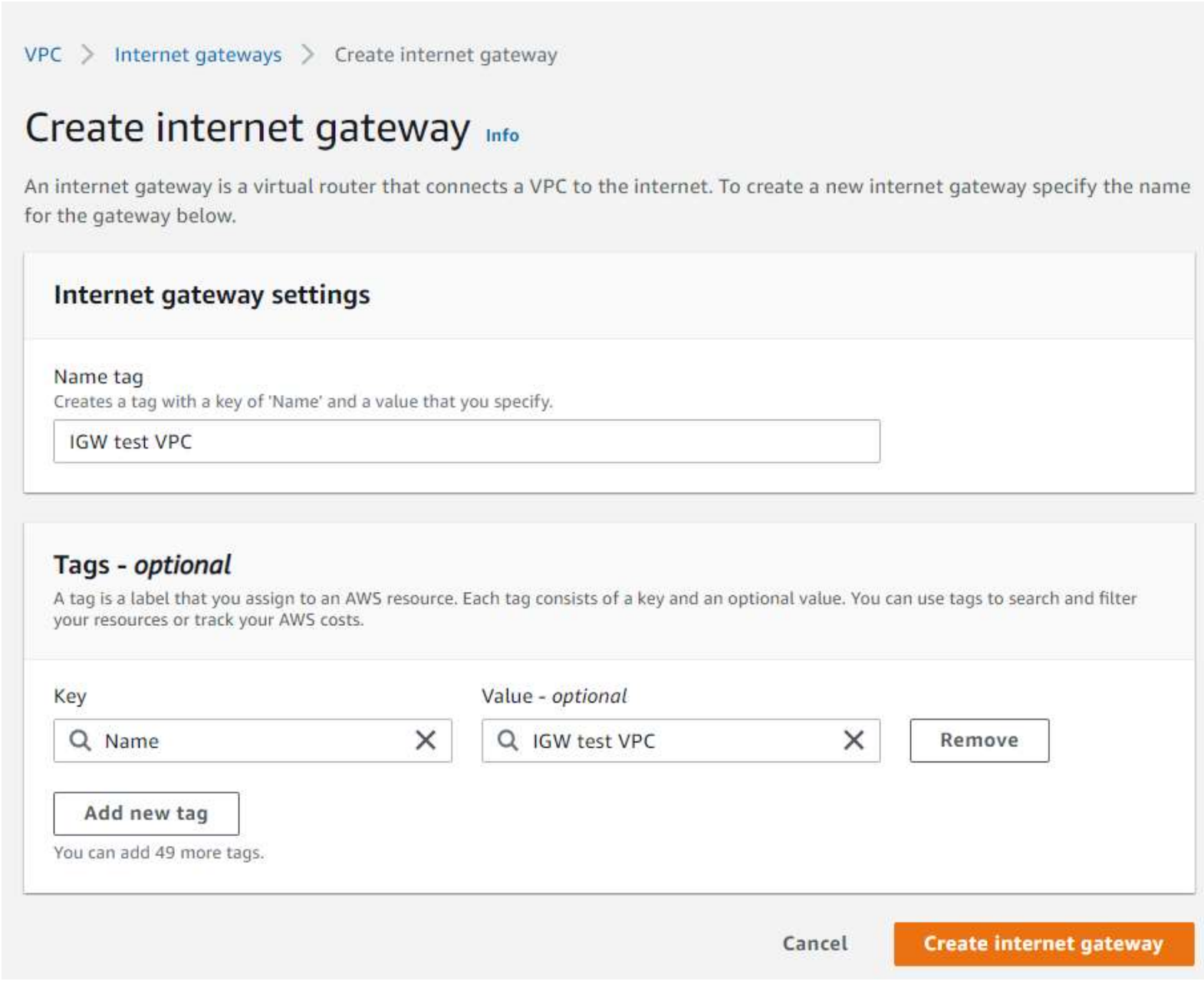


Figure: Internet gateway configuration

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**.

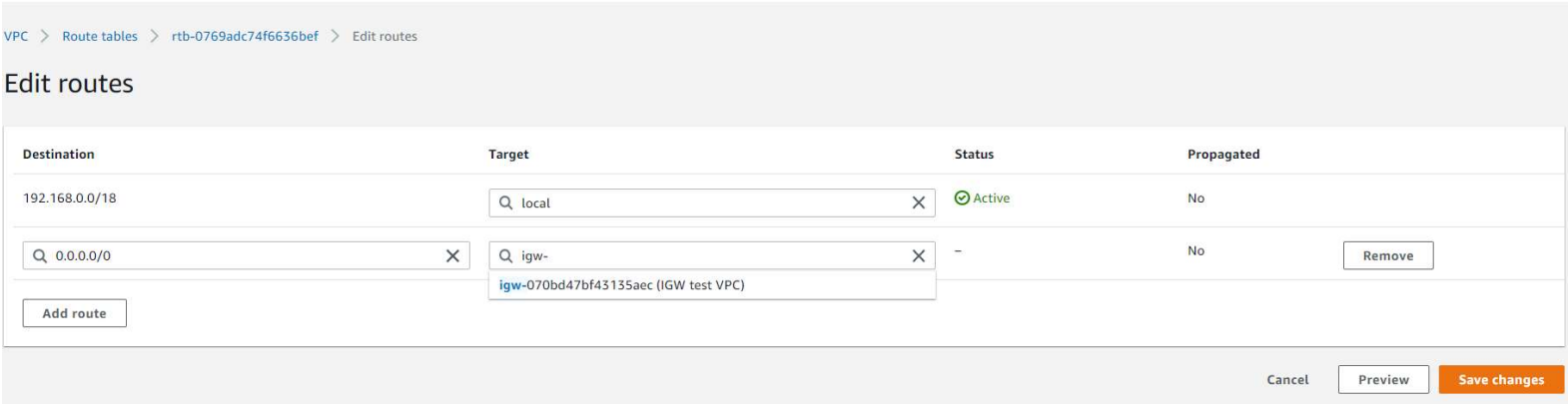


Figure: Adding the IGW in the route table (0.0.0.0/0 as the destination and IGW as the target).

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.

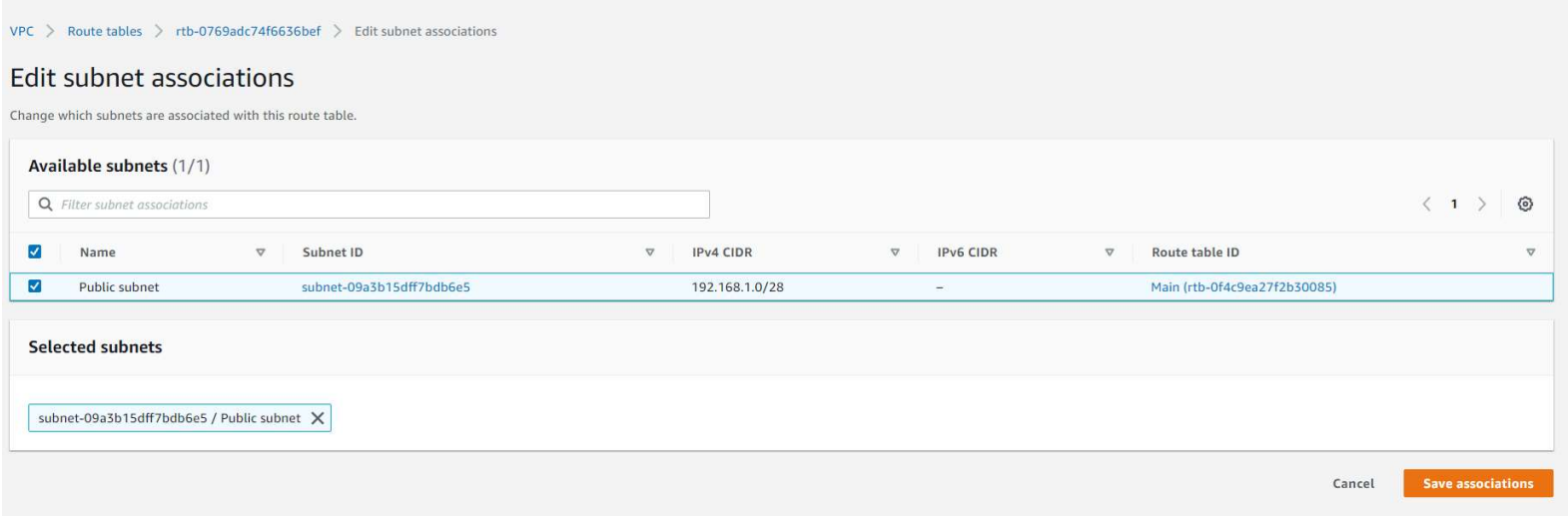


Figure: Associate the Public subnet and select save association.

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 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).

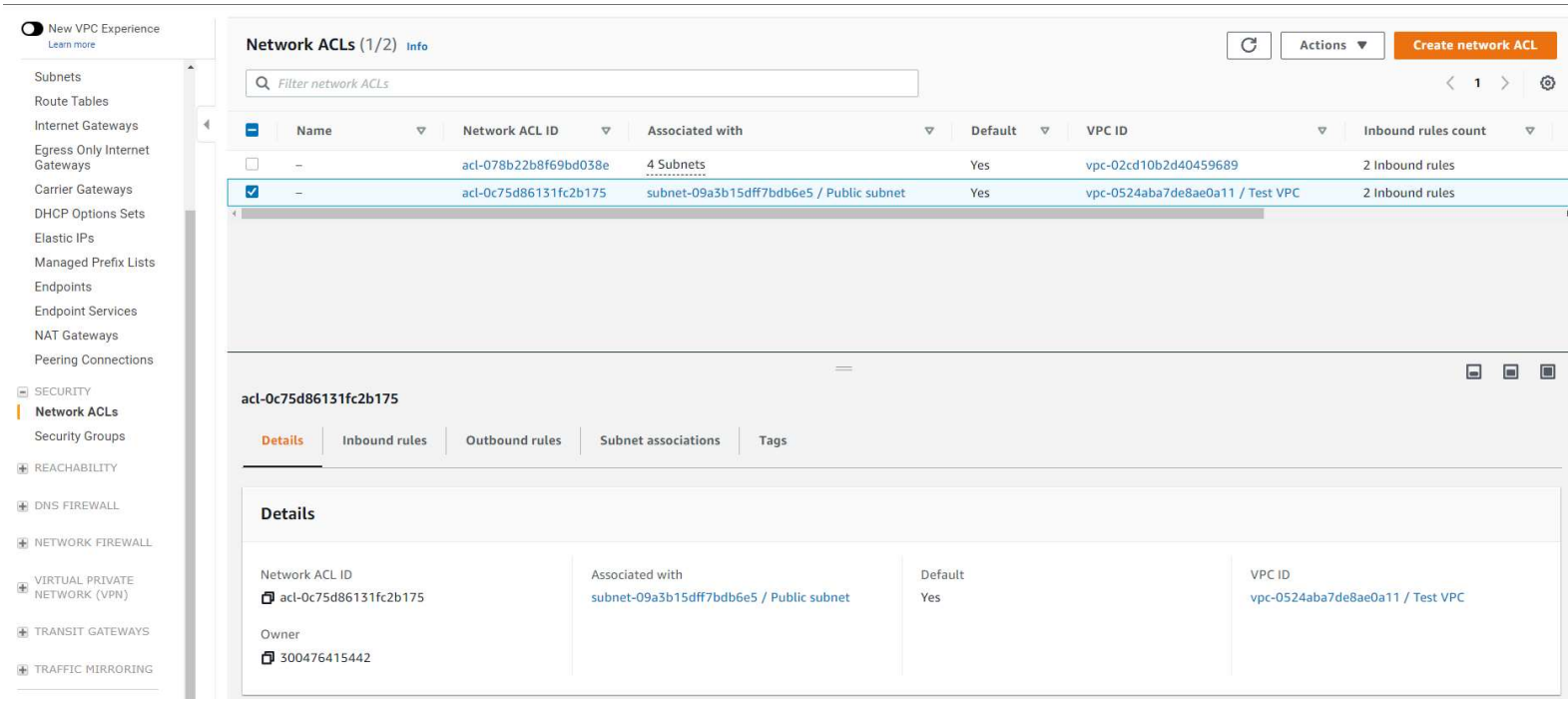


Figure: Select Create network ACL

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.

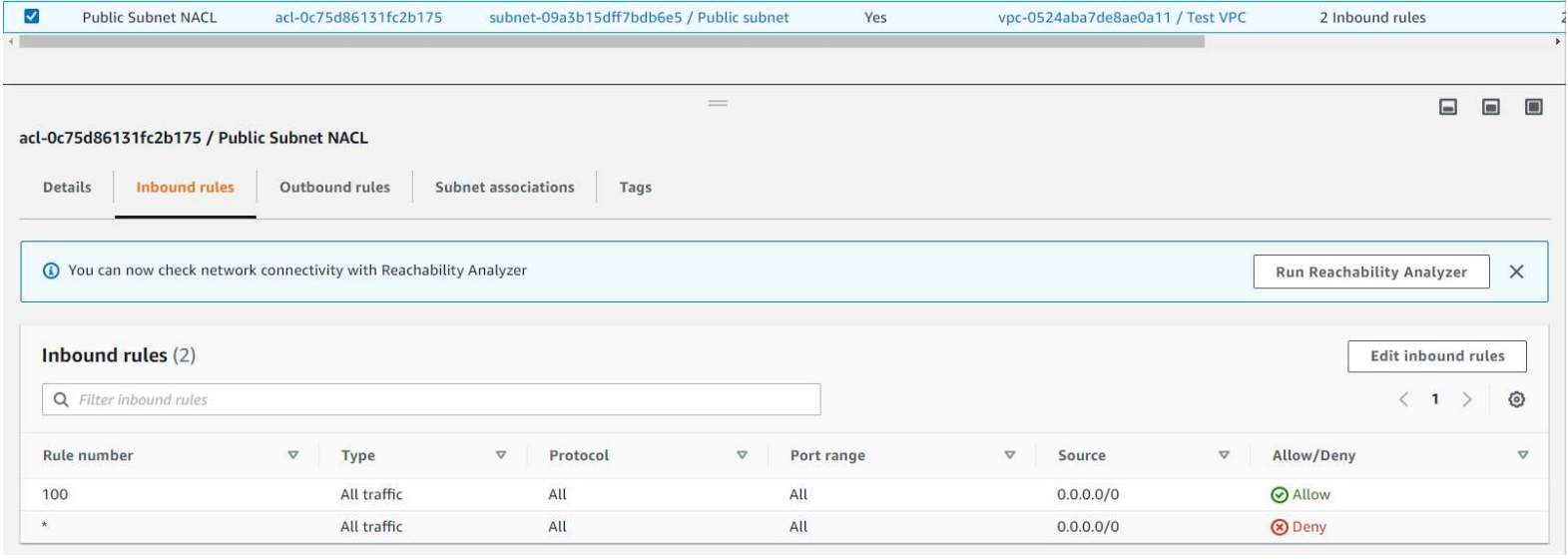


Figure: Default inbound rule configuration for NACL. This will allow all traffic from anywhere and deny anything else that does not match this rule at the subnet level.

Outbound What do you think this rule says?

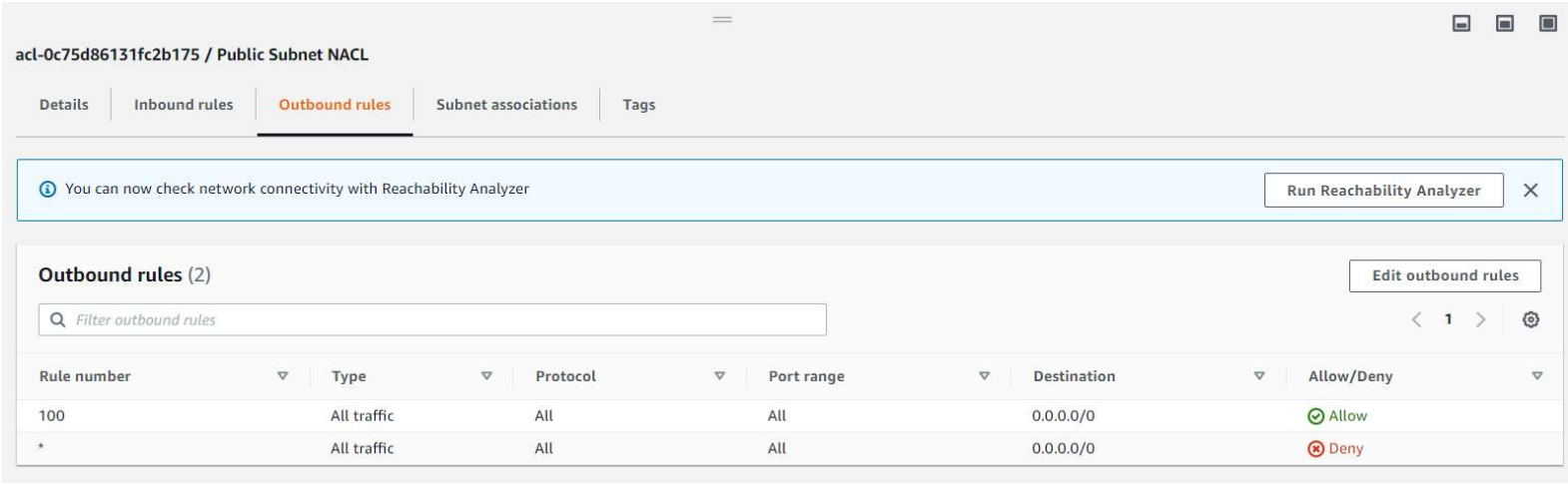


Figure: Default outbound rule configuration for NACL. This will allow all traffic from anywhere and deny anything else that does not match this rule at the subnet level.

Creating a Security Group

▶ **Recall**

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.

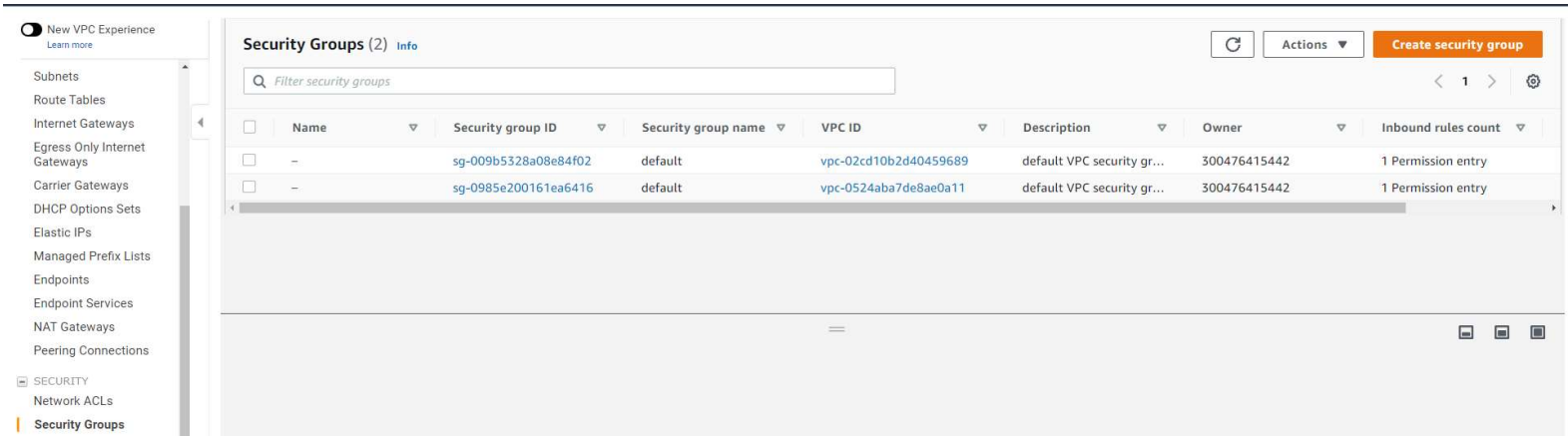


Figure: Select Create security group

Configure like the following image of the Basic details page:

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

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.

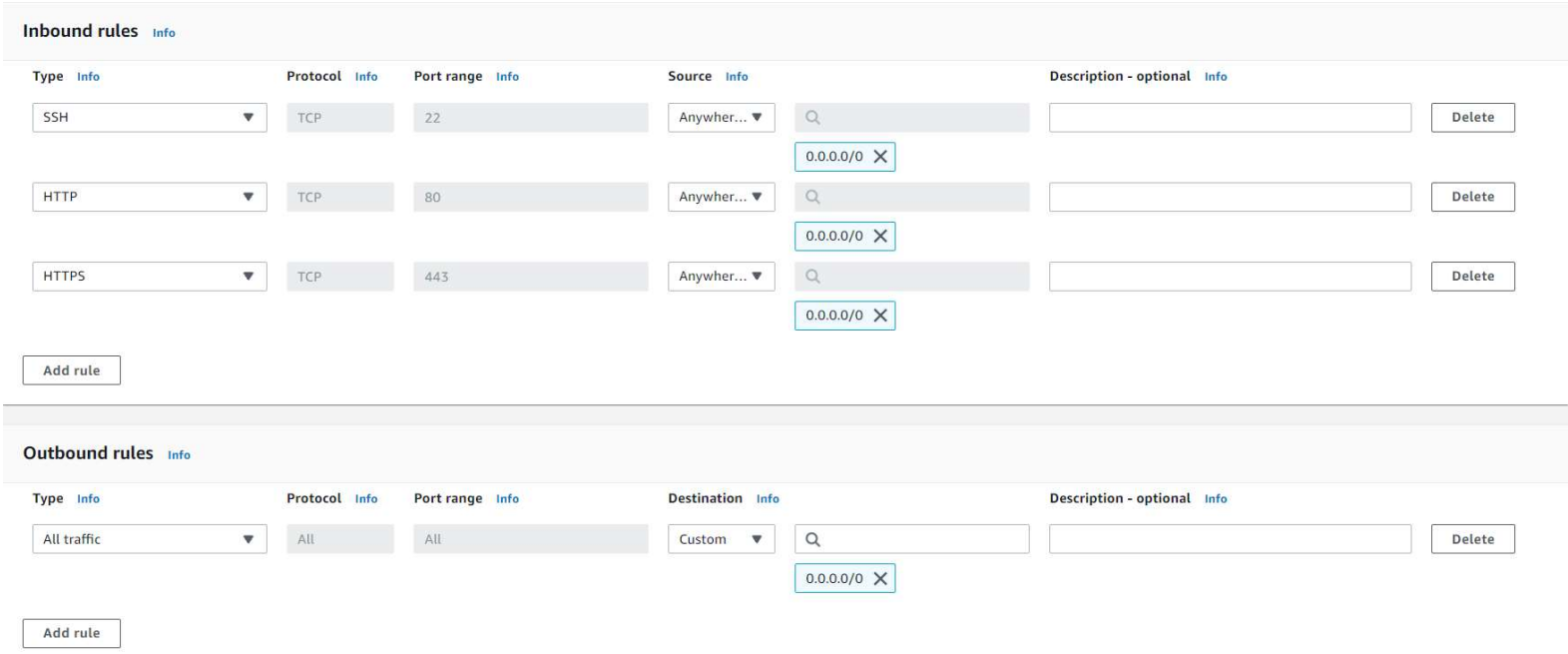


Figure: Configuration details for inbound and outbound rules for the security group

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.

22. Navigate to Services at the top left, and select **EC2**. From the EC2 dashboard, select **Instances**.

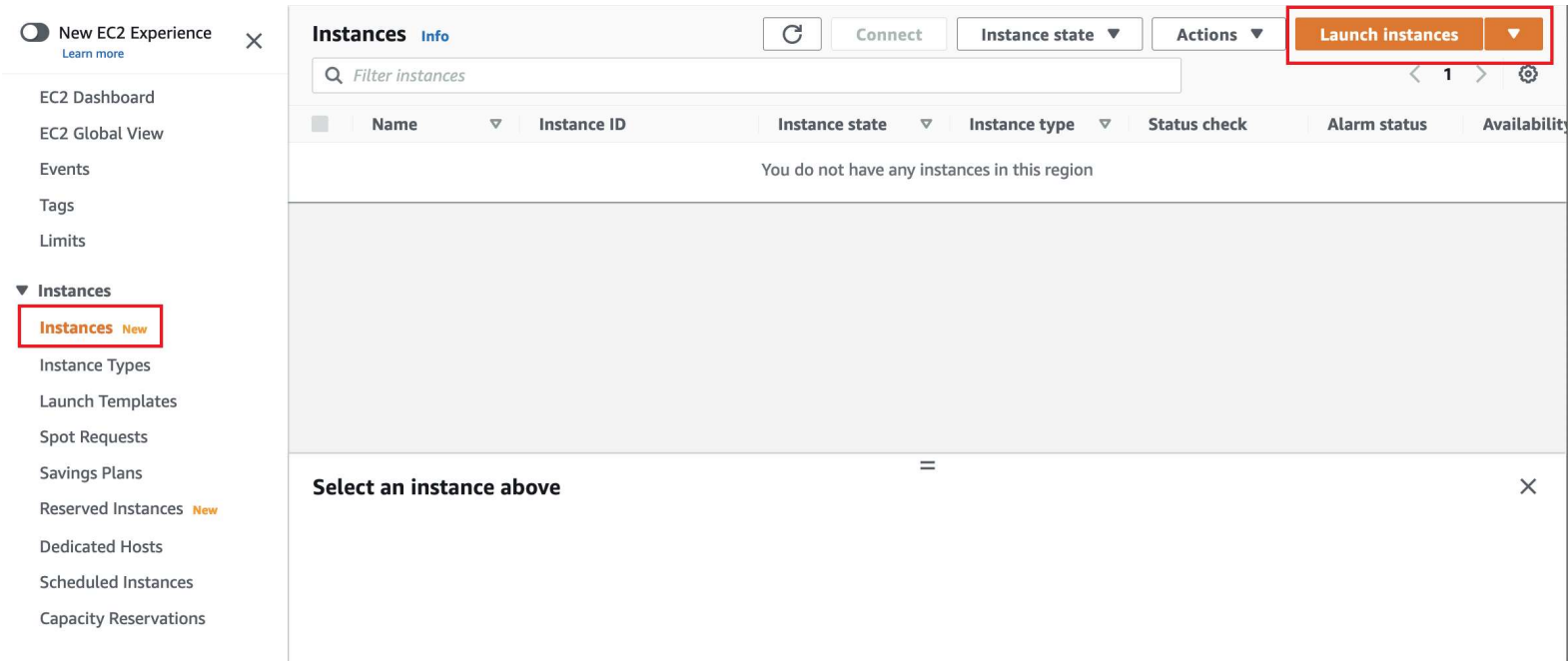


Figure: Launch an EC2 instance by selecting Launch instance

23. Select **Launch instances** from the top right corner. Then complete the following steps:

24. For Step 1, keep the defaults.

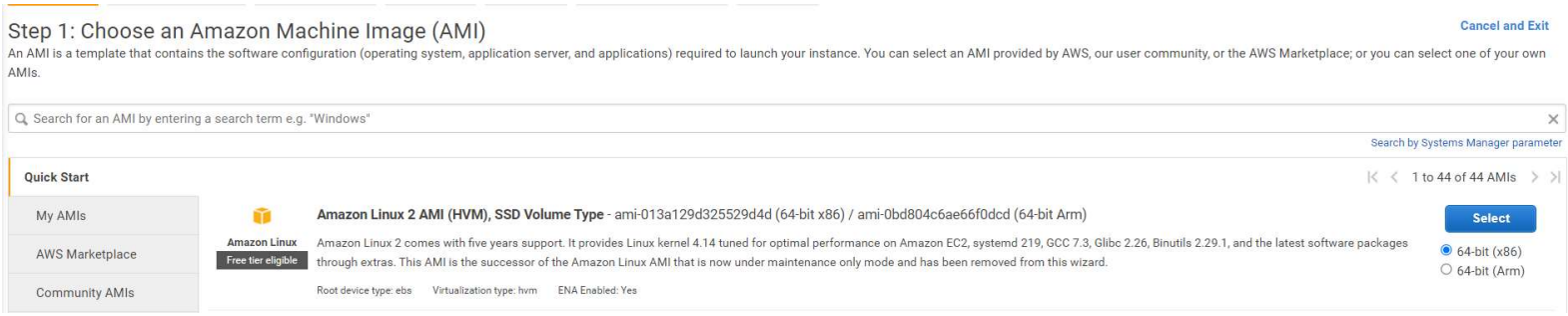


Figure: Choose the AMI: Amazon Linux 2 AMI in step 1

25. For step 2, keep the defaults.

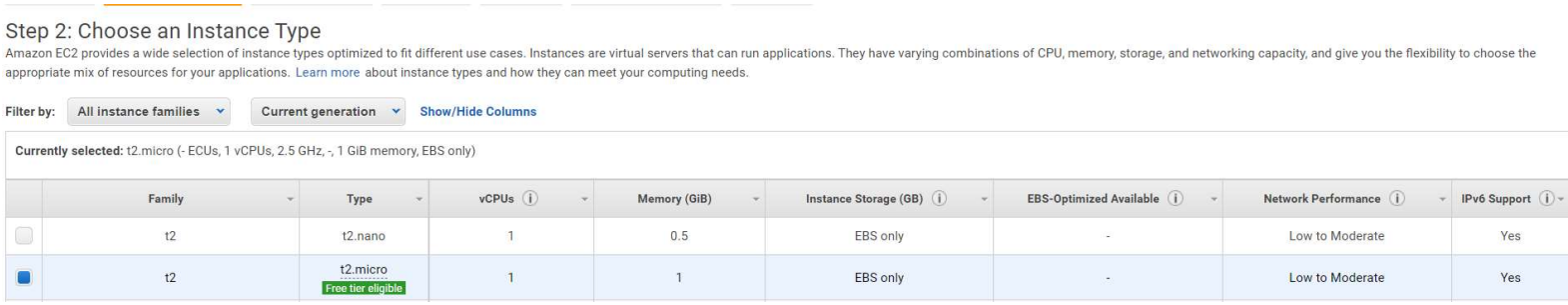


Figure: Choose the instance type, t3.micro, in step 2.

26. For step 3, configure the Instance as shown below:

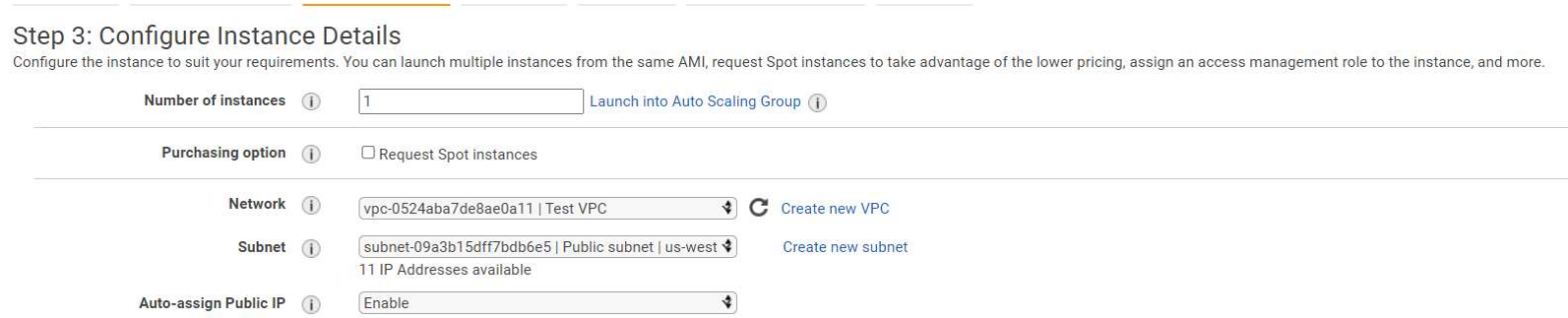


Figure: Step 3 is to configure the instance details. An important note here is to enable the "Auto-assign Public IP" if you want a public IP for an instance without having to assign an EIP.

27. For step 4, keep the defaults.

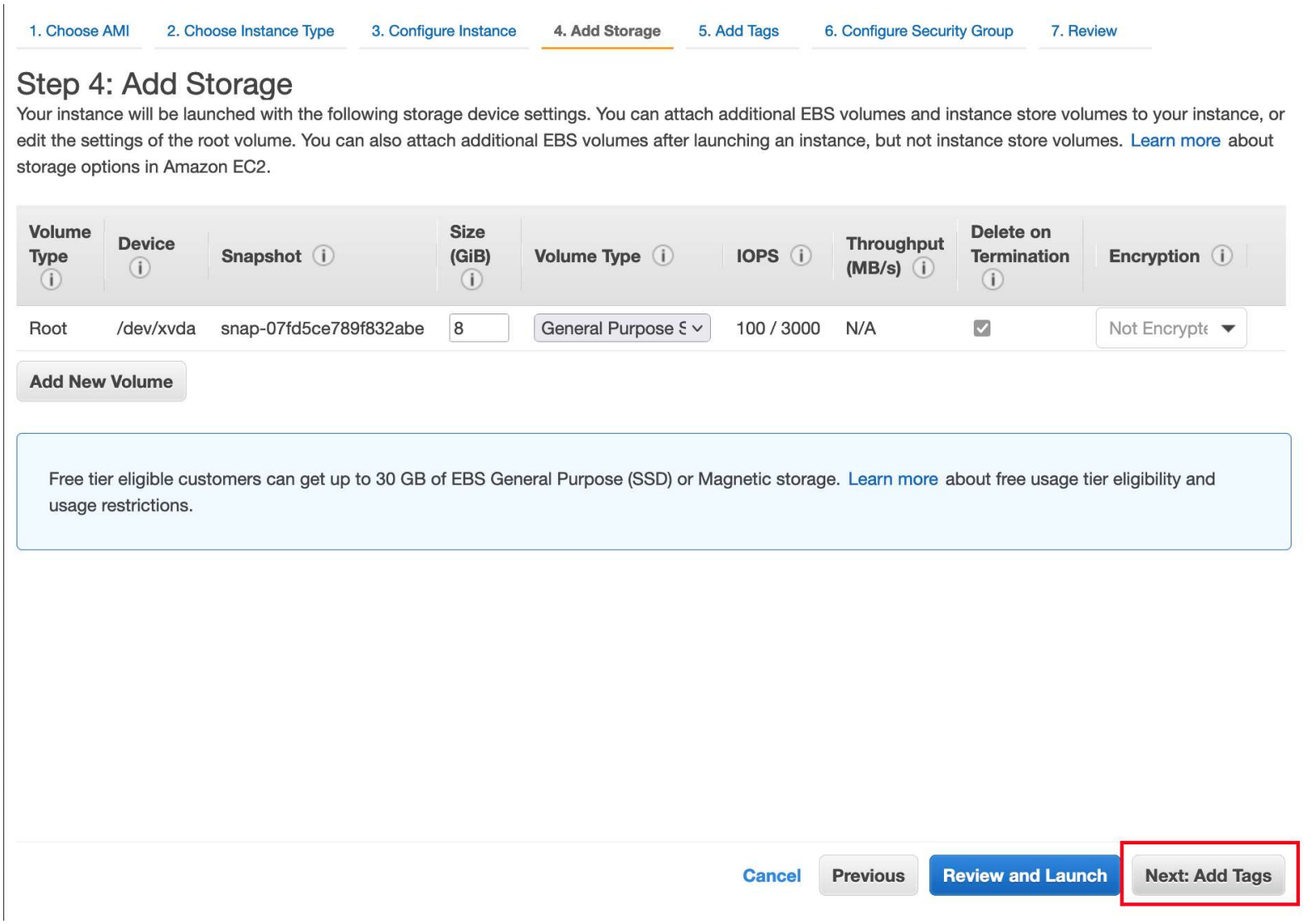


Figure: Step 4 has the option to add additional or change the storage. This will be kept at the default storage.

28. For step 5, keep the defaults.

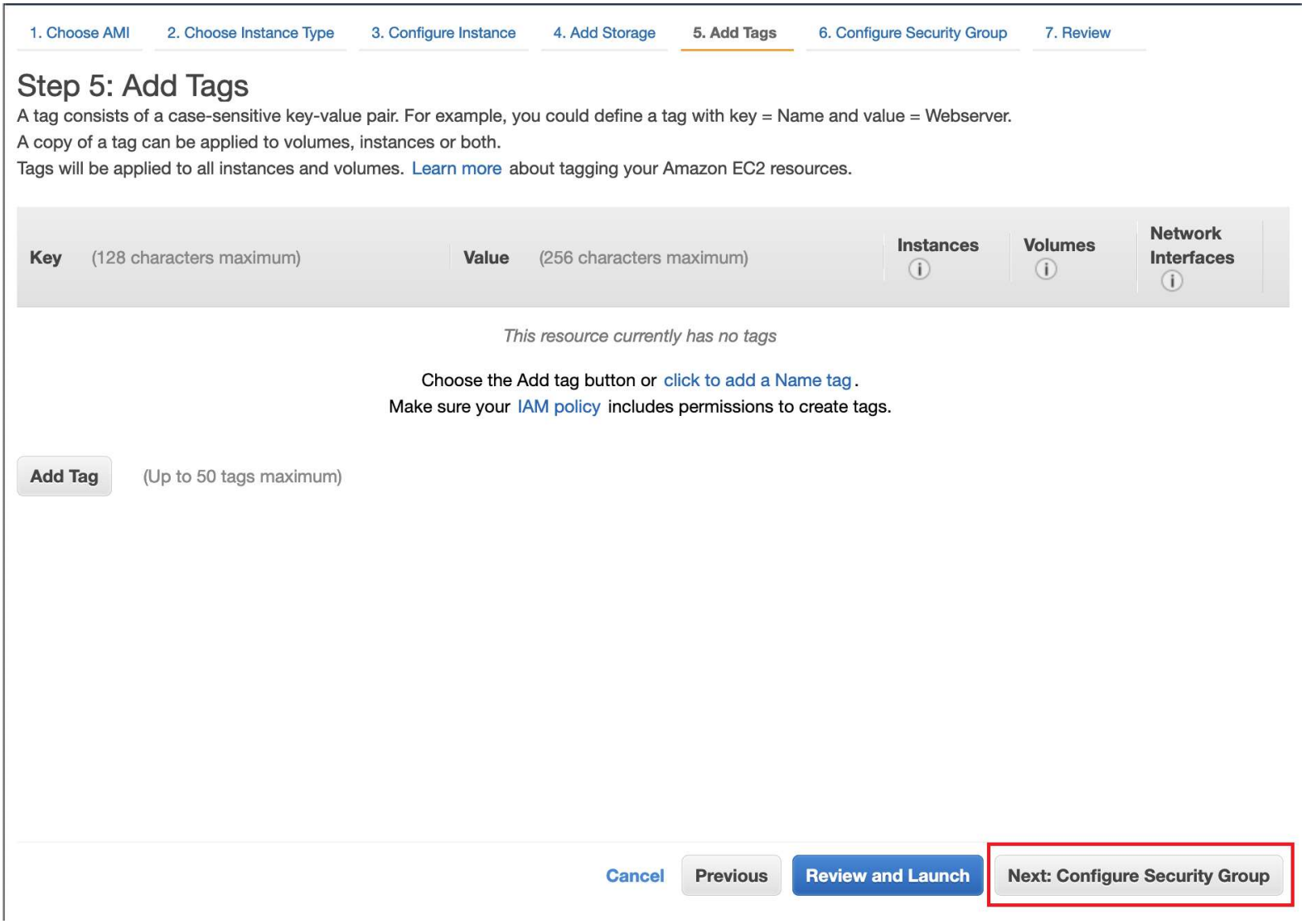


Figure: Step 5 has the option to add tags, which gives the instance a name and value. However, since we are launching just one instance, we will leave this blank. This is a great option when you have multiple resources and need to assign them specific values and names.

29. For step 6, configure the security group as shown below:

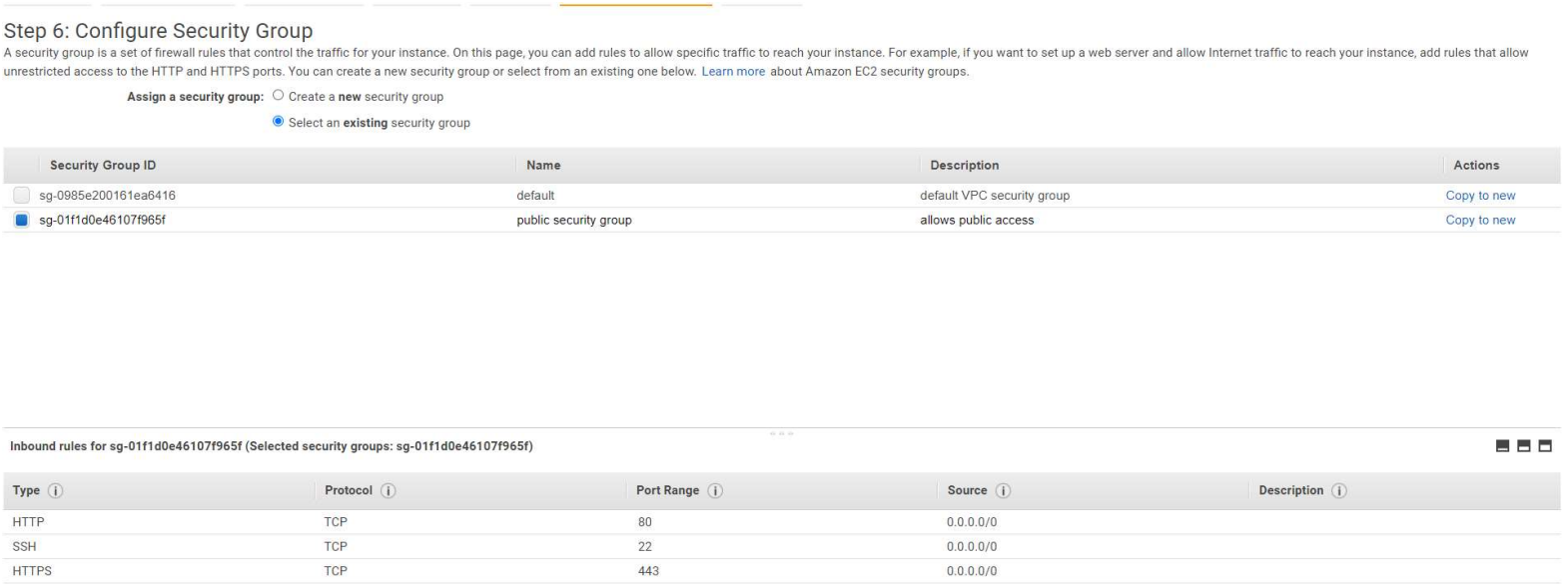


Figure: In step 6, you can always create a new security group or select an existing one. Select the option "Select an existing security group" and select the security group you created, "public security group".

30. For step 7, review your settings. It should look like the following:

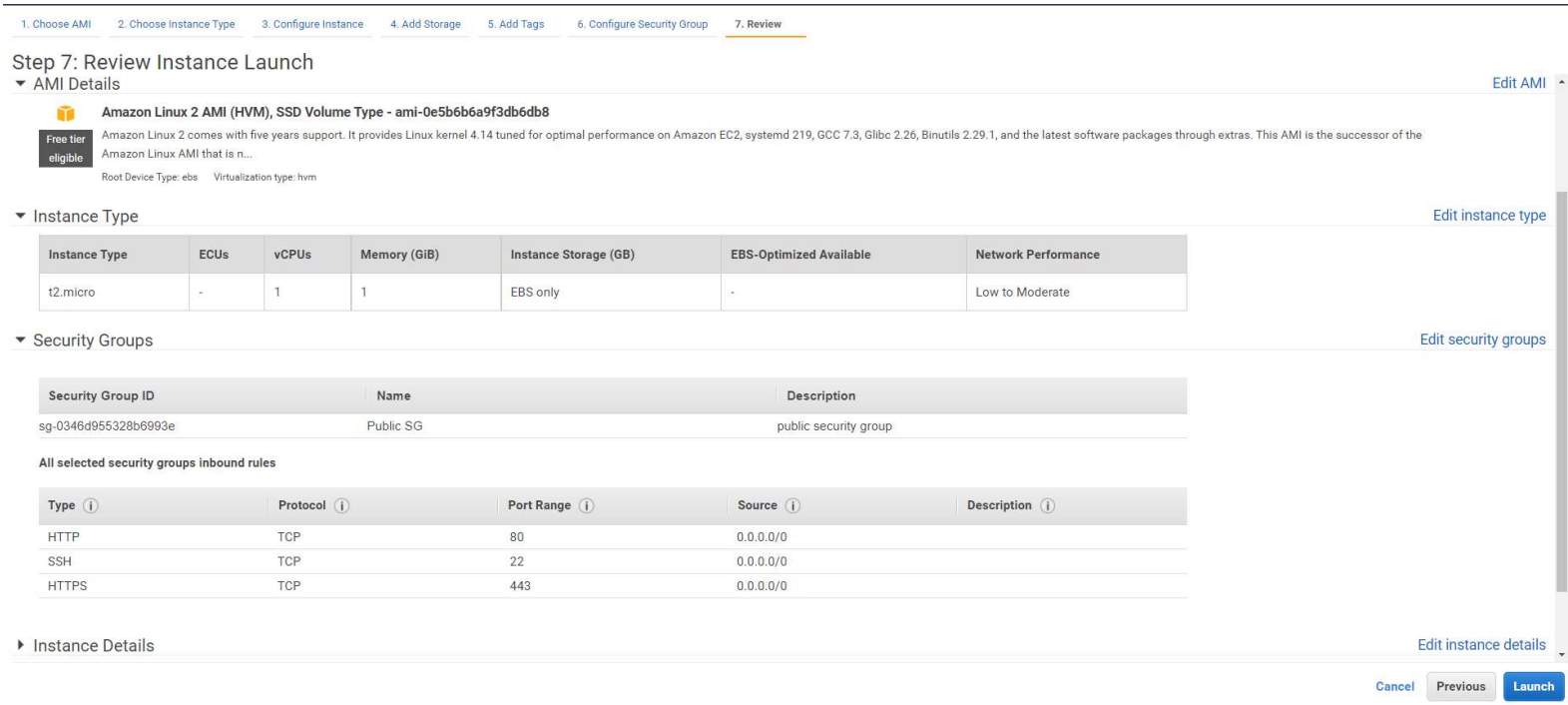


Figure: With all the steps followed correctly, the review screen should look like the example.

31. Choose the following key:

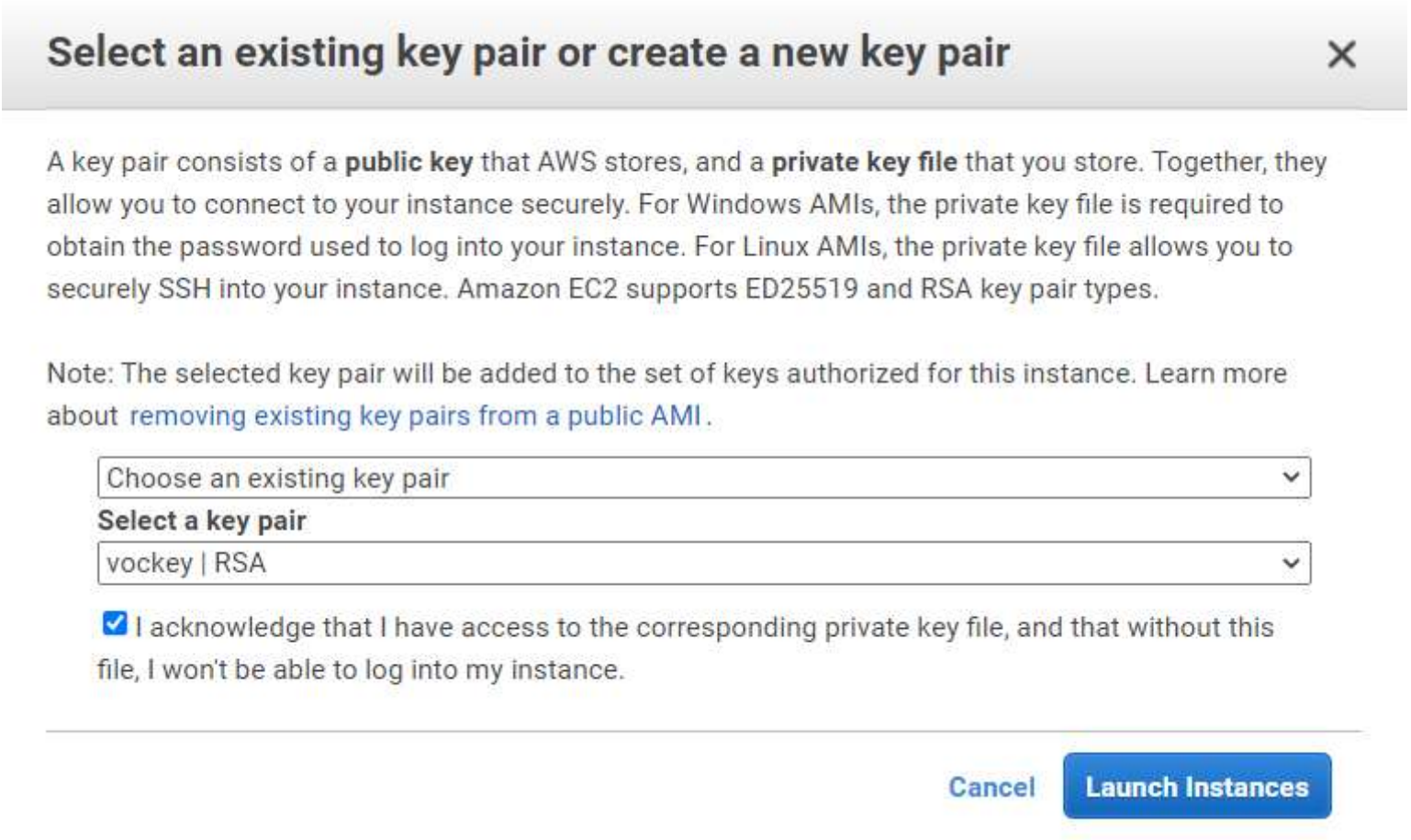


Figure: You will choose an existing key pair: vockey | RSA. You must acknowledge that you have the corresponding private key file that can be downloaded in the details tab in the Vocreaum lab page.

While waiting for the instance to enter the ready state, open the Details tab located at the top of this lab window and download the **PPK** file, if on Windows, or the **pem** file, if on a Mac.

SSH into the EC2 instance by using PuTTY. You can follow these steps here:

Use SSH to connect to an Amazon Linux EC2 instance

▶ **Ways to connect Amazon Linux EC2**

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.

32. Select the Details drop-down menu above these instructions you are currently reading, and then select Show. A Credentials window will be presented.
33. Select the **Download PPK** button and save the **labsuser.ppk** file.
    Typically your browser will save it to the Downloads directory.
34. Make a note of **PublicInstanceBIP**, the IPV4 server's address you have to connect to.
35. Then exit the Details panel by selecting the **X**.
36. Download **PuTTY** to SSH into the Amazon EC2 instance. If you do not have PuTTY installed on your computer, download it here.
37. Open **putty.exe**.

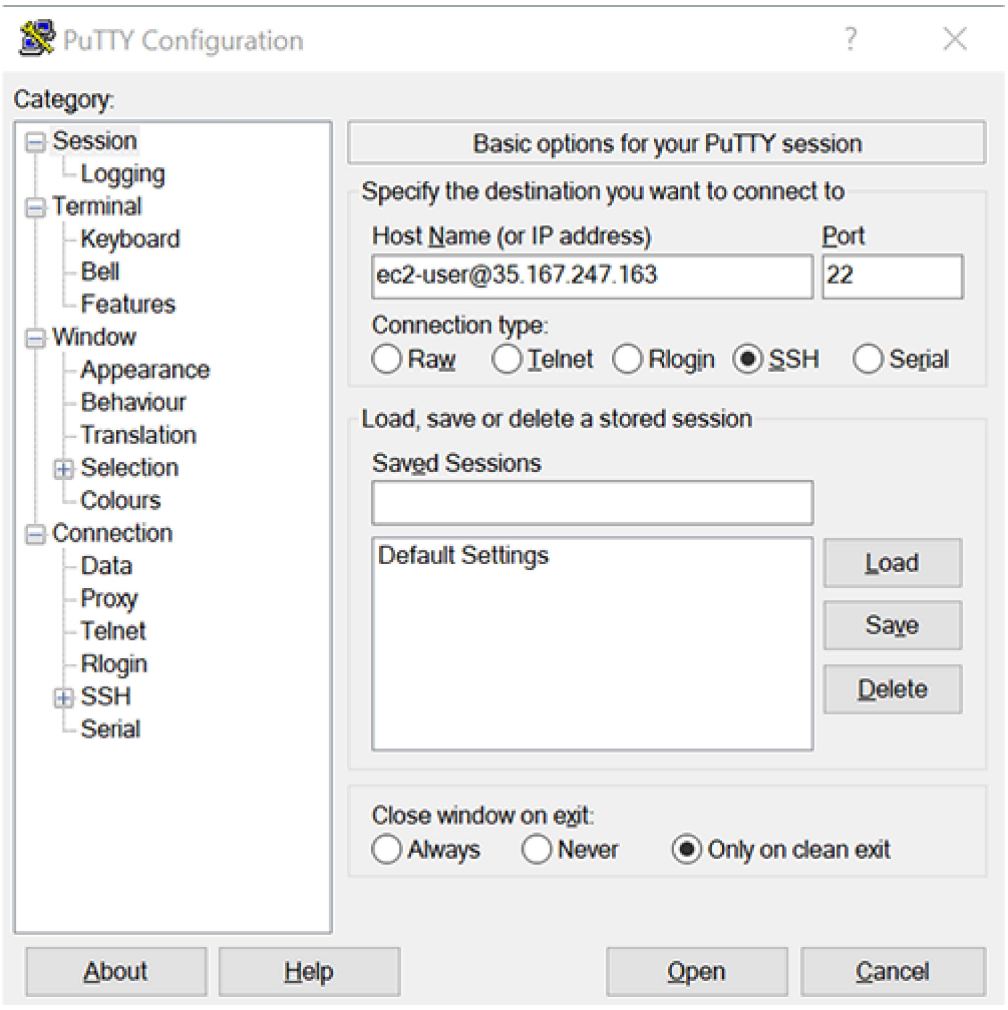


Figure: SSH using Putty for windows.

38. Configure PuTTY timeout to keep the PuTTY session open for a longer period of time.:
    - Select **Connection**
    - Set **Seconds between keepalives** to `30`
39. Configure your PuTTY session:

- Select **Session**
- **Host Name (or IP address):** Paste the server's address of the instance you made a note of earlier.
- Back in PuTTY, in the **Connection** list, expand ⊞ **SSH**
- Select **Auth** *(don't expand it)*
- Select **Browse**
- Browse to and select the **labuser.ppk** file that you downloaded
- Select **Open** to select it
- Select **Open** again.

40. Select **Yes**, to trust and connect to the host.
41. When prompted **login as**, enter: `ec2-user`
    This will connect you to the EC2 instance.
42. Windows Users: Select here to skip ahead to the next task.

macOS and Linux Users

These instructions are specifically for Mac/Linux users. If you are a Windows user, skip ahead to the next task.

43. Select the Details drop-down menu above these instructions you are currently reading, and then select Show. A Credentials window will be presented.
44. Select the **Download PEM** button and save the **labsuser.pem** file.
45. Make a note of **PublicIP**, the IPV4 server's address you have to connect to.
46. Then exit the Details panel by selecting the **X**.
47. Open a terminal window, and change directory `cd` to the directory where the *labsuser.pem* file was downloaded. For example, if the *labuser.pem* file was saved to your Downloads directory, run this command:

```
cd ~/Downloads
```

48. Change the permissions on the key to be read-only, by running this command:

```
chmod 400 labsuser.pem
```

49. Run the below command *(replace **<public-ip>** with the server's address you copied earlier)*:

```
ssh -i labsuser.pem ec2-user@<public-ip>
```

Figure: SSH using a terminal for Mac.

50. Type `yes` when prompted to allow the first connection to this remote SSH server.
    Because you are using a key pair for authentication, you will not be prompted for a password.

Task 3: Use ping to test internet connectivity

51. 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.

52. 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...*

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

Recap

▶ **In this lab**

Additional Resources

What is Amazon VPC?
IP Addressing in your VPC
Route tables for your VPC
Internet Gateways
Network ACLs
Security Groups

For more information about AWS Training and Certification, see AWS Training and Certification Opens in new window
Your feedback is welcome and appreciated.

If you would like to share any suggestions or corrections, please provide the details in our AWS Training and Certification Contact Form Opens in new window