Lab 261

Internet Protocols -Public and Private IP addresses

Student: Mane Zakarian
Bootcamp: Forge AWS re/Start UYMON5
Date: 2023

Objectives

In this lab, you will:

- Summarize and investigate the customer scenario
- Analyze the difference between a private and public IP address
- Develop a solution to the customer's issue in this lab
- Summarize and describe your findings (group activity)

Scenario

Your role is a cloud support engineer at Amazon Web Services (AWS). During your shift, a customer from a Fortune 500 company requests assistance regarding a networking issue within their AWS infrastructure. The following is the email and an attachment regarding their architecture:

Ticket from your customer

Hello, Cloud Support!

We currently have one virtual private cloud (VPC) with a CIDR range of 10.0.0.0/16. In this VPC, we have two Amazon Elastic Compute Cloud (Amazon EC2) instances: instance A and instance B. Even though both are in the same subnet and have the same configurations with AWS resources, instance A cannot reach the internet, and instance B can reach the internet. I think it has something to do with the EC2 instances, but I'm not sure. I also had a question about using a public range of IP address such as 12.0.0.0/16 for a VPC that I would like to launch. Would that cause any issues? Attached is our architecture for reference.

Thanks! Jess Cloud Admin

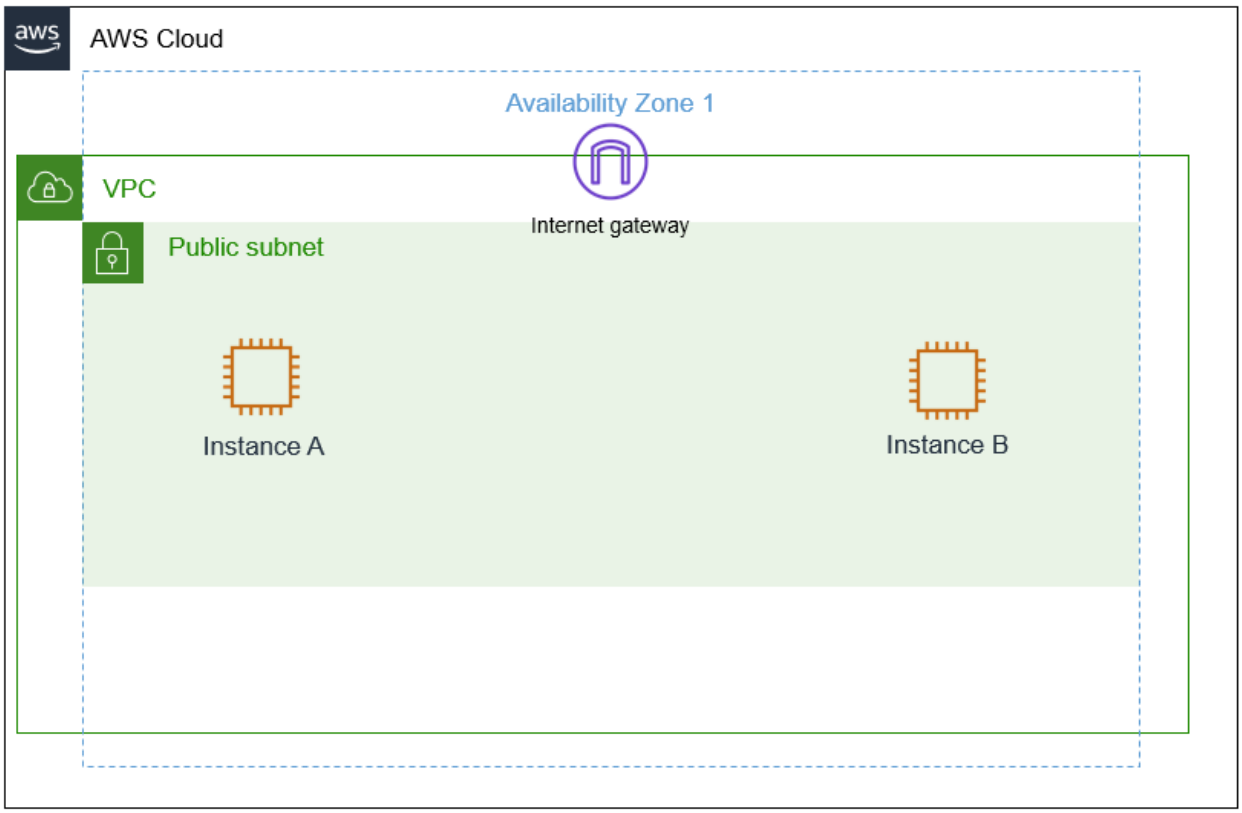


Figure: The customer's architecture, which consists of a VPC, internet gateway, public subnet with instance A, and a public subnet with instance B.

AWS service restrictions

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

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 environment

Recall that you previously covered public and private IP addresses and CIDRs. As you go through this lab, think about the differences between public and private IP addresses for task 1. For task 2, think about the importance of using private IP ranges rather than public IP ranges.

Note

For this lab, you have already checked the AWS architecture, and everything is routed and attached correctly. This lab does not cover any AWS architecture.

In the scenario, Jess, who is the customer requesting assistance, has two EC2 instances in one VPC. Instance A cannot reach the internet, and instance B can even though they are configured the same within the VPC. Currently, the customer's AWS architecture seems sound because one of their instances works. Jess suspects that the instance configuration may be the issue.

She also has a question about using a public range of IP addresses for the new VPC and has asked if you could provide further insight on her question.

You currently have one VPC with the same CIDR of 10.0.0.0/16 with two instances — instance A and instance B — with the same configurations as the customer. When troubleshooting networking and AWS, you can apply a troubleshooting method where you start from the top and work your way down or start from the bottom and work your way up. You start troubleshooting from the bottom and work your way to the top in layers using an example such as the OSI model. At the very bottom of this architecture is the EC2 instance. Although the cloud architecture does not directly translate to the OSI model, the following is an example of how the OSI and cloud relate.

	OSI Model	AWS infrastructure
Layer 7	Application (how the end user sees it)	Application
Layer 6	Presentation (translator between layers)	Web Servers, application servers
Layer 5	Session (session establishment, security)	EC2 instances
Layer 4	Transport (TCP, flow control)	Security group. NACL
Layer 3	Network (Packets which contain IP addresses)	Route Tables, IGW, Subnets
Layer 2	Data Link (Frames which contain physical MAC addresses)	Route Tables, IGW, Subnets
Layer 1	Physical (cables, physical transmission bits and volts)	Regions, Availability Zones

Table: This is an example of how the AWS infrastructure and its resources have similarities to the OSI model. This information can be beneficial when troubleshooting.

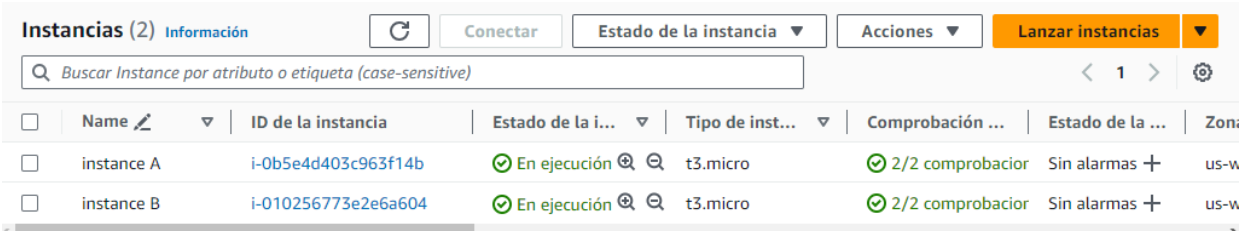
For task 1, you gain an understanding of the customer's environment and replicate their issue.

5. At the upper-right of these instructions, choose **AWS**. The AWS Management Console opens in a new tab.

6. Once you are in the AWS console, type and search for **EC2** in the search bar on the top-left corner. Select EC2 from the list.

   Tip: Alternatively, You can also find EC2 under **Services** - **Compute** in the top left corner

7. You are now in the Amazon EC2 dashboard. In the left navigation menu, choose **Instances**. This option takes you to your current EC2 instances. You should currently see two EC2 instances.



8. Please copy and paste the names and IP addresses of both instances for future reference in a text editor. Select the check box next to **instance A**. At the bottom of the page, choose the **Networking** tab, and note the **Public** and **Private** IPv4 addresses. Once you copy and paste the name and IP addresses, deselect the instance, and then select **instance B** and do the same. Did you notice any differences? Note them if you did.

Instance A

Instance B

Credentials

Cloud Access

AWS CLI: Show

Cloud Labs

Remaining session time: 01:50:02(111 minutes)
Session started at: 2023-11-03T13:57:21-0700
Session to end at: 2023-11-03T16:21:47-0700

Accumulated lab time: 02:34:00 (154 minutes)

(1) ips -- public:, private:10.0.10.241 (2) ips -- public:34.222.90.164, private:10.0.10.205

SSH key Show Download PEM Download PPK

AWS SSO Download URL

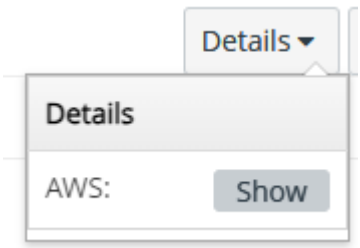
LabRegion	us-west-2
PublicInstanceBIP	34.222.90.164

Task 2: Use SSH to connect to an Amazon Linux EC2 instance

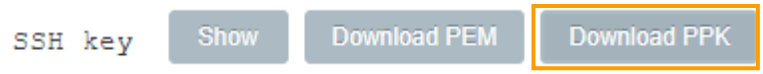
In this task, you will connect to a Amazon Linux EC2 instance. You will use an SSH utility to perform all of these operations.

Windows Users: Using SSH to Connect

1. Select the Details drop-down menu above these instructions you are currently reading, and then select Show. A Credentials window will be presented.



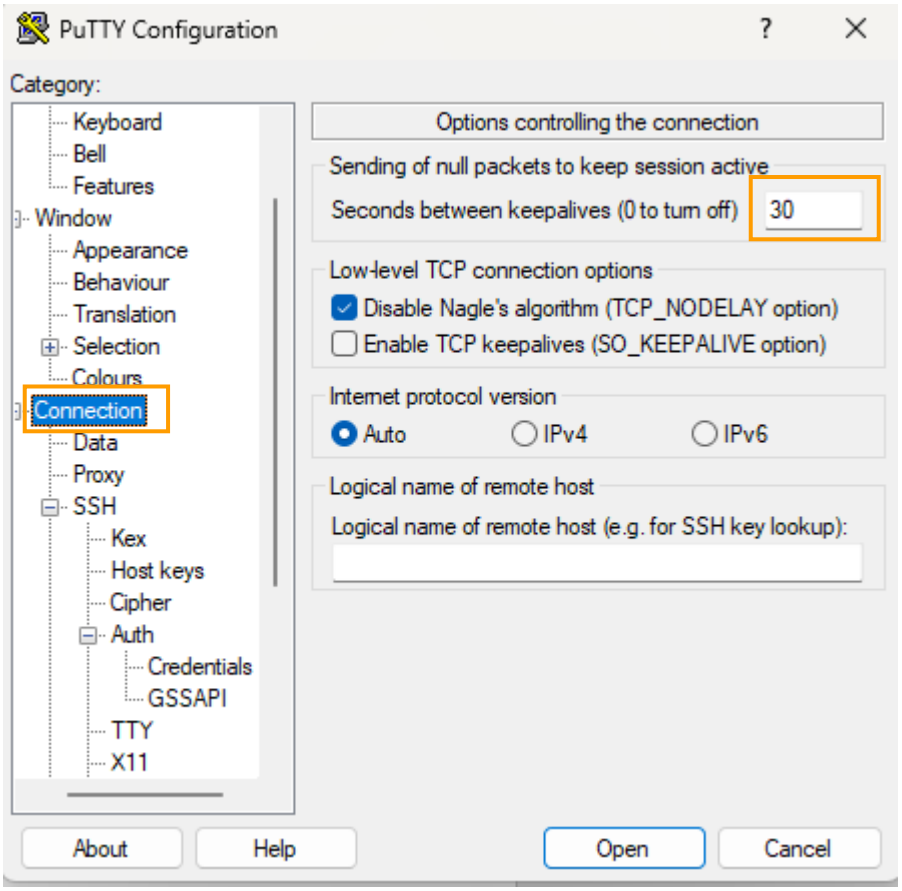
2. Select the **Download PPK** button and save the **labsuser.ppk** file.



3. Make a note of the **PublicIP** address.

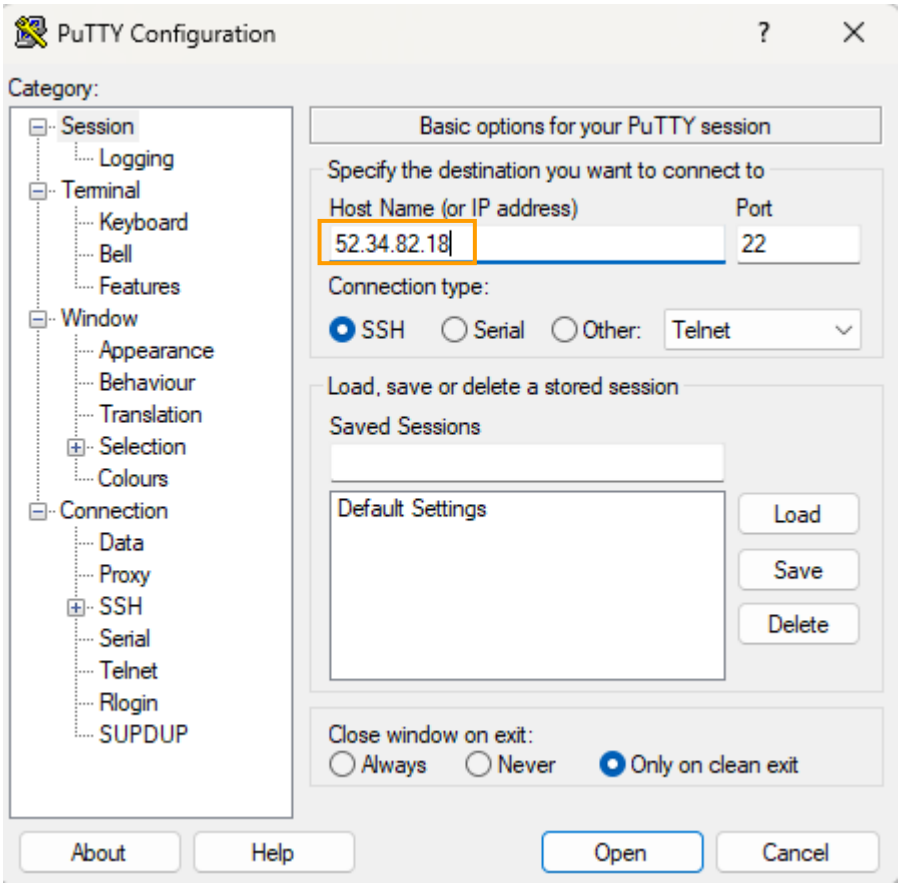
4. Then exit the Details panel by selecting the **X**.
5. Download **PuTTY** to SSH into the Amazon EC2 instance. If you do not have PuTTY installed on your computer.
6. Open **putty.exe**

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

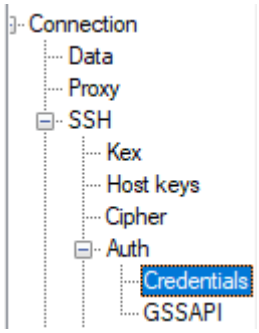


8. Configure your PuTTY session:
    - Select **Session**

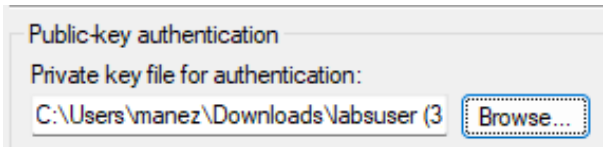
    - **Host Name (or IP address):** Paste the **Public DNS or IPv4 address** of the instance you made a note of earlier. Alternatively, return to the EC2 Console and select **Instances**. Check the box next to the instance you want to connect to and in the *Description* tab copy the **IPv4 Public IP** value



- Back in PuTTY, in the **Connection** list, expand **SSH** and select **Auth** *(don't expand it*



- Select **Browse** and select the lab#.ppk file that you downloaded



- Select **Open** to select it and then select **Open** again.

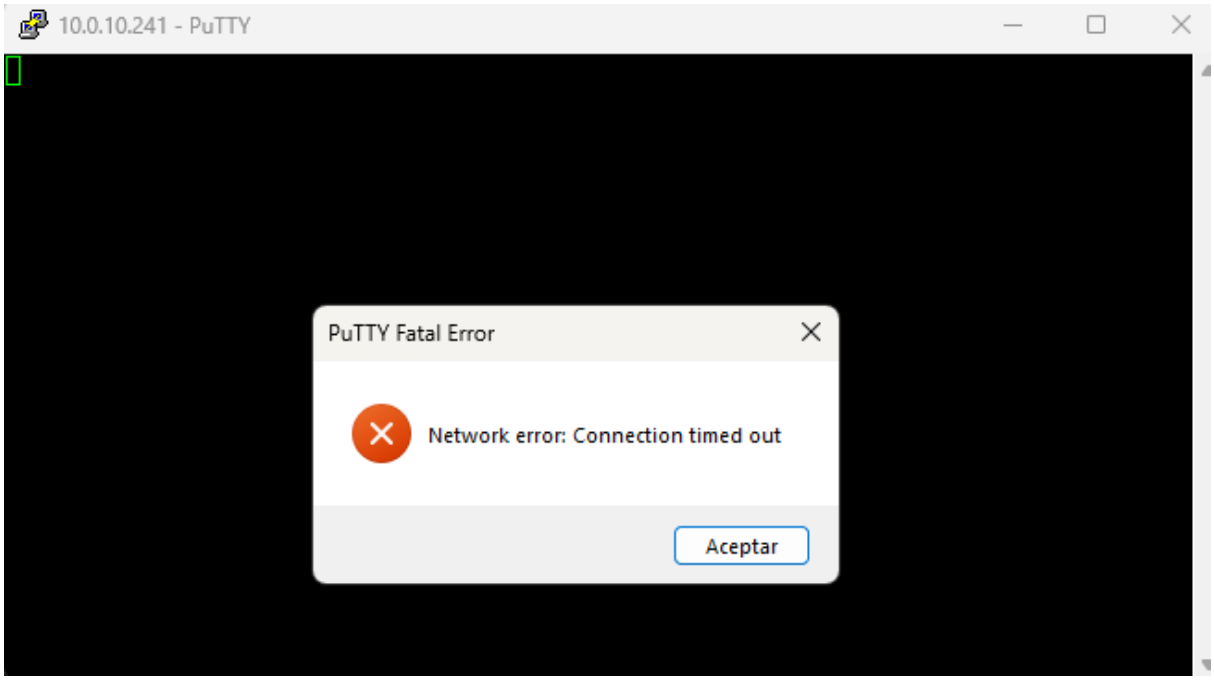
9. Select **Yes**, to trust and connect to the host.

10. When prompted **login as**, enter: `ec2-user` This will connect you to the EC2 instance.

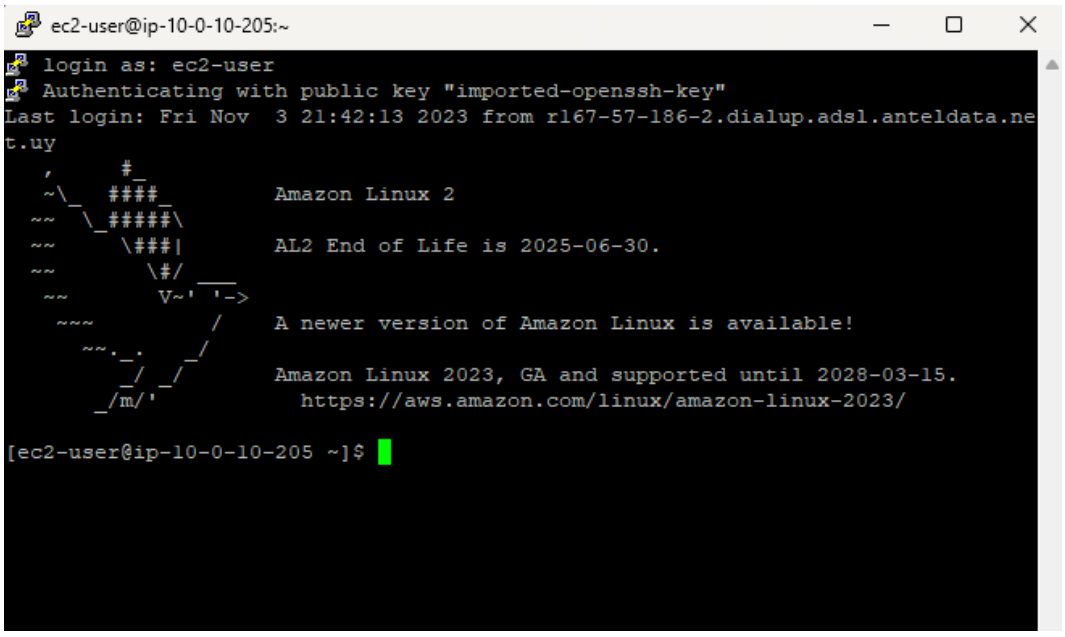
Question - Were you able to use the SSH to connect to both instances? Why or why not?

Answer: If you were not able to connect to instance A, it was due to this instance being assigned only a private IP address. Private IP addresses cannot be accessed from outside the VPC. This is why you are only able to connect to instance B. Instance B has a public IP address assigned to it allowing access from outside the VPC, which allows you to use the SSH utility to connect to the instance.
The customer asked for your insight regarding using a public CIDR for a new VPC that she would like to launch. Refer to module 4 and gather some evidence and summarize a short explanation of your findings to explain to the customer why or why not you recommend this approach.

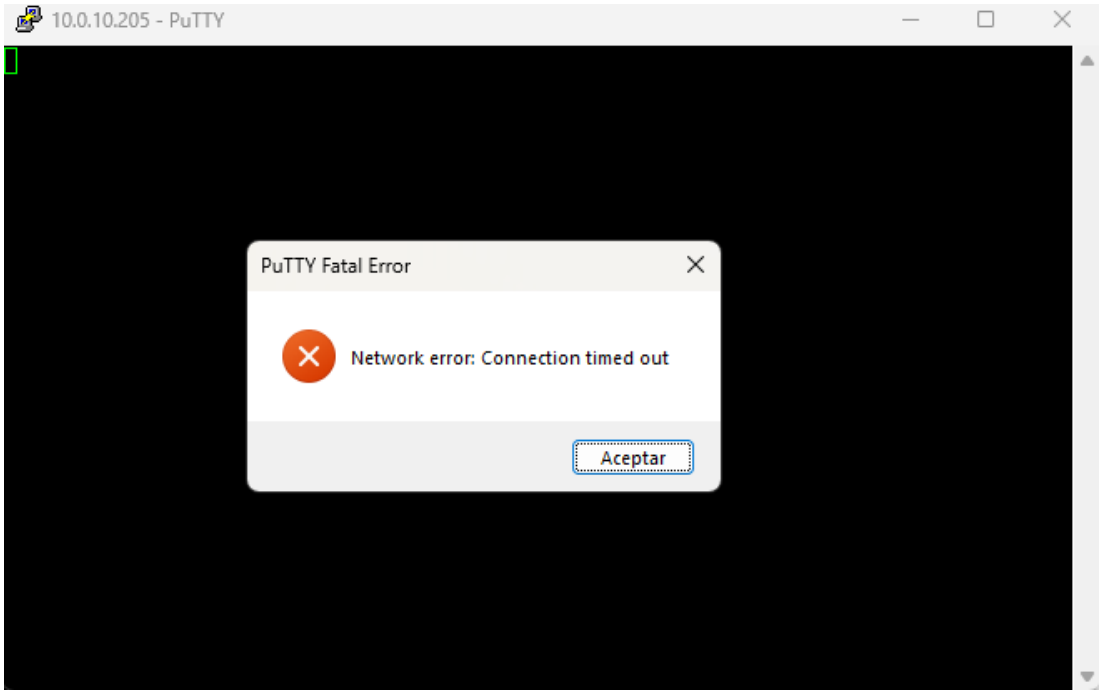
We noted that accessing the private IP from the instance A, we are not able to access. See figure:



For the instance B, on the other hand, we were able to access the public IP. See Figure:



For the instance B, we could not access the private IP. See Figure:



Note: For any of these two instances we will not be able to access the Private IP addresses since these are not visible or accessible from the public internet. They are part of a reserved address space, and the routers on the internet are not configured to route traffic to or from private IP addresses.

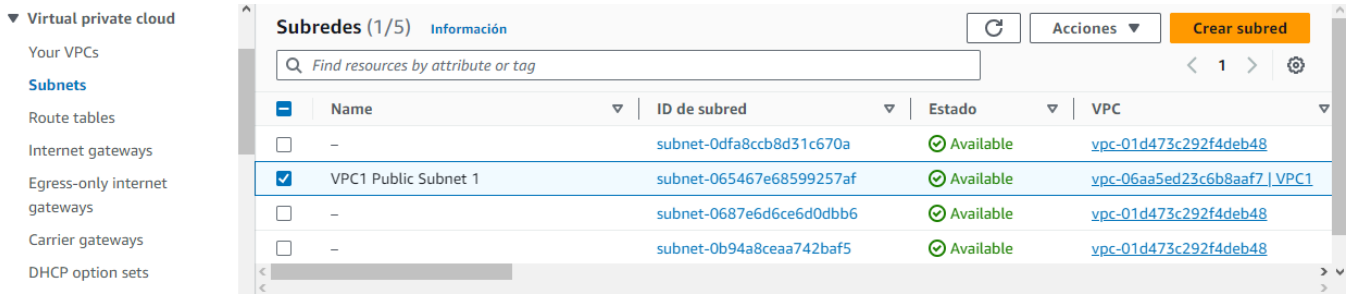
The user tried to create a second instance (Instance A), but this one has only a private IP. So what we can do next, is to stablish the next instances automatically to a public IP.

To stablish the next instances automatically to a public IP, we follow the next steps:

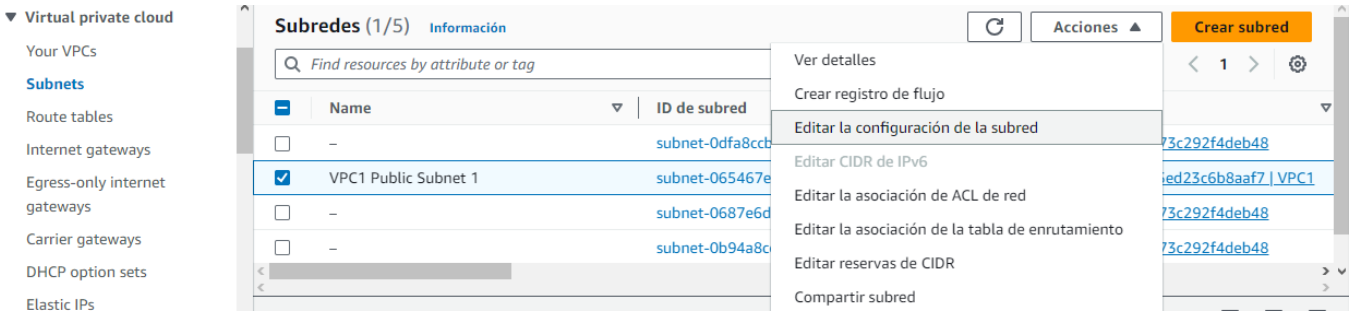
1. In the AWS console, type and search for **VPC** in the search bar on the top-left corner. Select VPC from the list.

Figure: The search bar can be used to find the Amazon VPC service. Once you find the service, select it.

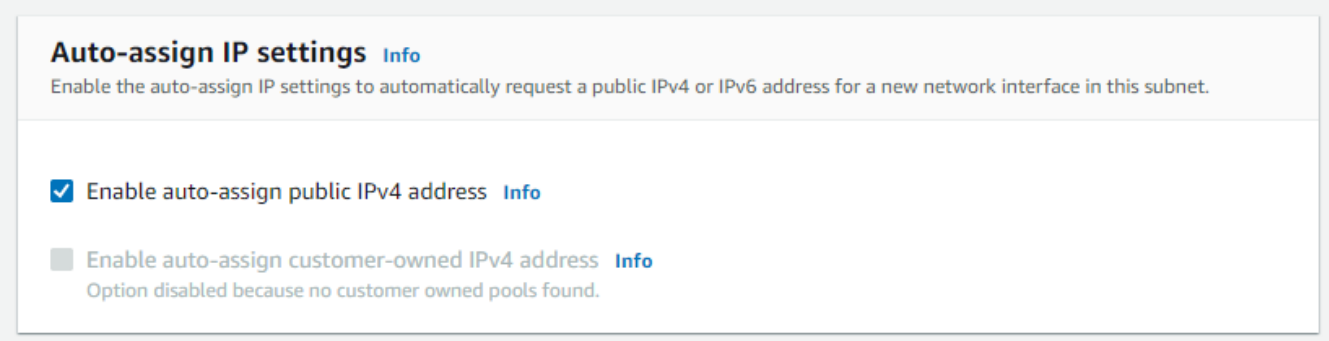
2. You are now in the Amazon VPC dashboard. In the left navigation menu, choose Subnets. This option takes you to your current VPC subnets. You should currently see the available subnets.

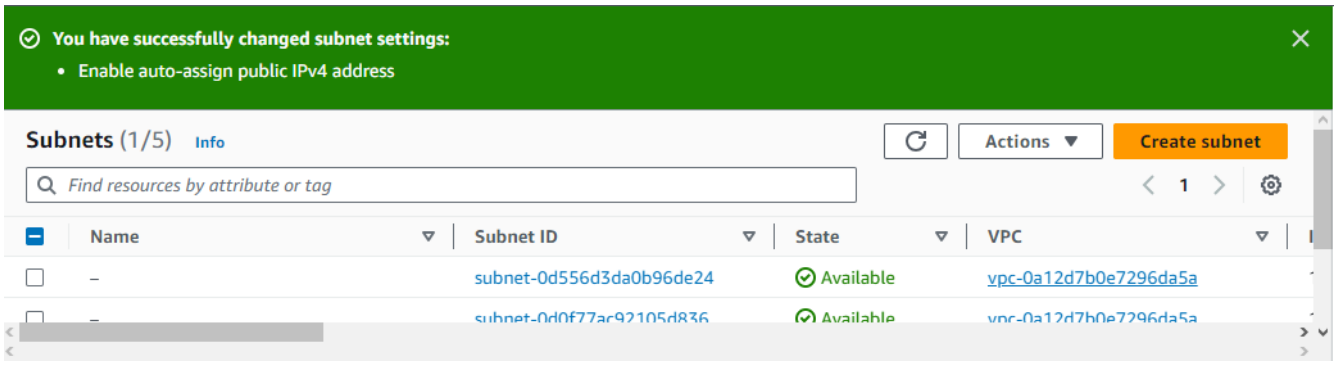


3. We select the subnet “VPC1 Public Subnet 1”, and we click on Actions and edit the subnet configuration.



4. We enable Auto-Assign public IPv4 address and save changes:





5. Now, on Instances dashboard, we launch a new instance to verify that it will automatically enable a pubic IPv4 address:

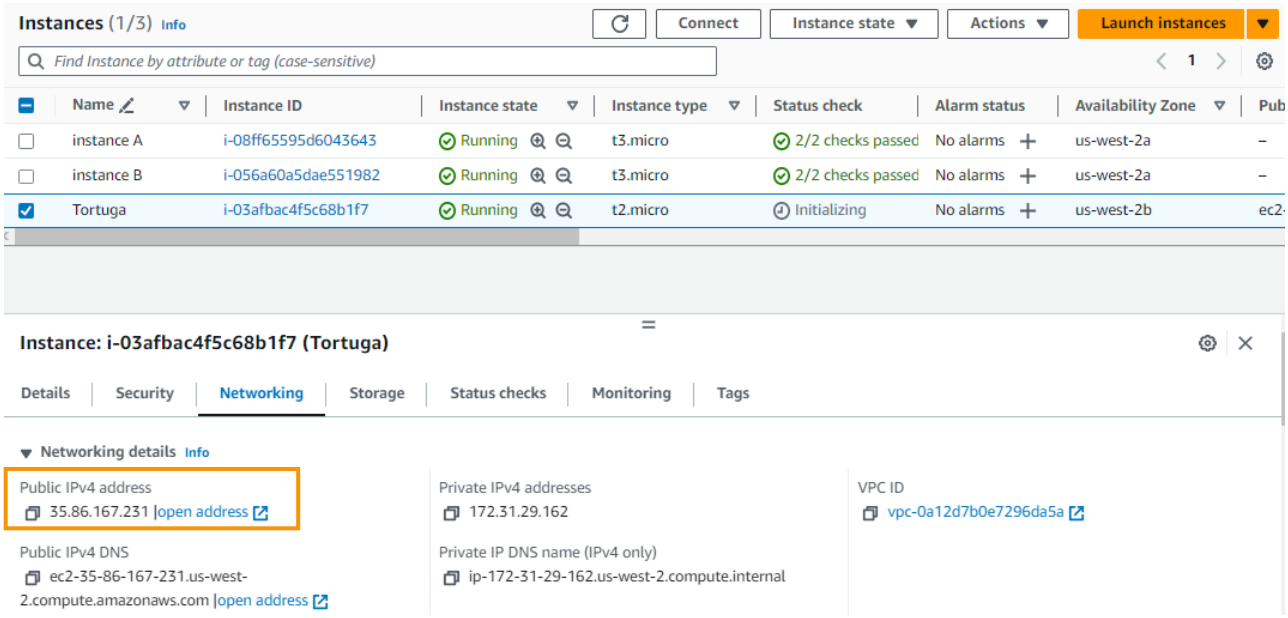


Figure: On the network section, we can see that IPv4 address has been automatically enabled.

Task 3: Send the Response to the customer (group activity)

In groups of two, submit your findings.

Person 2 will act as Jess the customer, and Person 1 will act as the cloud support engineer. Person 1 will talk over their findings with person 2.

Note

This task should take only 5-10 minutes. If a group activity is not possible due to COVID, please have one student walk through their findings to the class.

Lab Complete

Congratulations! You have completed the lab.

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

A message *Ended AWS Lab Successfully* is briefly displayed, indicating that the lab has ended.

Recap:

In this lab you have investigated the customer's environment and applied troubleshooting techniques that allowed you to resolve the customers' issue. Within the scenario, you discovered that the customer's EC2 instance (instance A) needed a public IP address to connect to the internet. This was tested by using an SSH utility to connect to the instance. Private IP addresses are used within the VPC and cannot establish a connection to the internet. As module 4 noted, you discovered that using a public range of IP addresses for a VPC can result in complications from having replies back from other unrelated resources.