

Name: Pelumi Johnson

Date: 12/2/2025

Course: CMIT 265 | Fundamentals of Networking

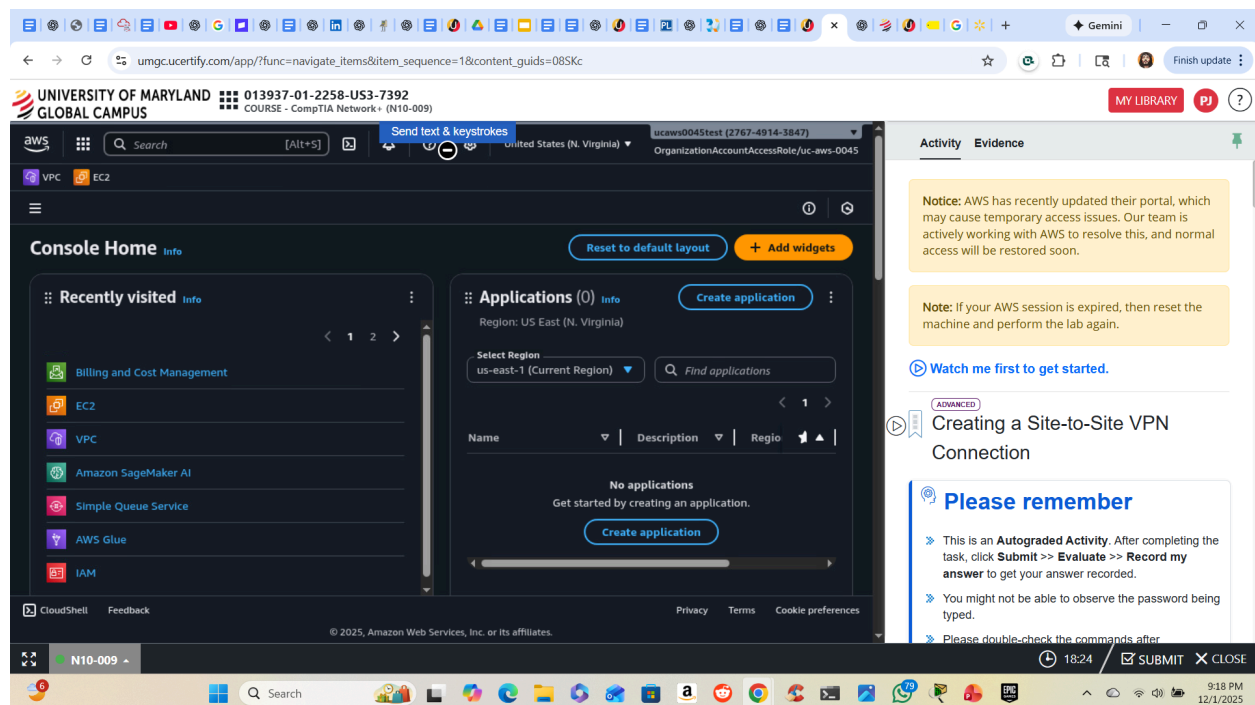
Platform: AWS (uCertify Lab Environment)

Project

The objective of this lab was to create a secure site to site VPN connection between an on-premises network (simulated Customer Gateway) and an AWS Virtual Private Cloud (VPC). Through this project, I learned how organizations interconnect distributed networks over the public internet using encrypted tunnels.

By the end of the lab, I was able to:

- Create a Virtual Private Gateway (VGW) for an AWS VPC
- Create a Customer Gateway (CGW) representing an on-premises router
- Configure a Site-to-Site VPN connection using IPSec tunneling
- Customize tunnel settings including Pre-Shared Keys and Inside CIDR ranges
- Verify successful deployment of the VPN connection in AWS



Tool and Service Used

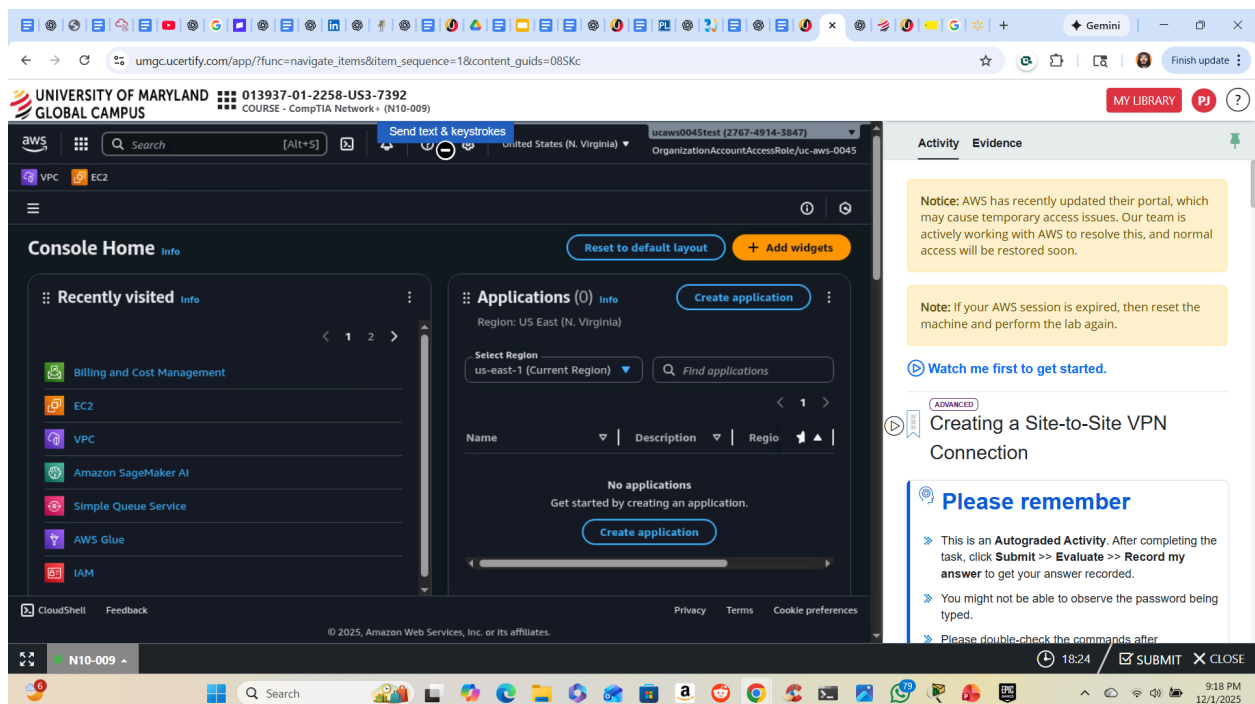
- **AWS Console:** Management interface for creating network resources
- **Amazon VPC:** Provides isolated networking infrastructure in AWS
- **Virtual Private Gateway (VGW):** AWS-side VPN endpoint
- **Customer Gateway (CGW):** Represents the on-premises router
- **Site-to-Site VPN service:** Establishes encrypted IPSec connections
- **IPSec Tunnels:** Mechanism used to create secure encrypted communications

Step by Step Project Procedure:

Step 1:

Navigate to AWS Console Home

You start on the AWS Console. Ensure your Region is set to US-East-1 (N. Virginia). This region selection matters because your VPC and VPN resources must all exist in the same region for the lab to work.



Step 2:

Create a Virtual Private Gateway (VGW)

Left menu → *Virtual Private Network (VPN)* → Virtual Private Gateways
Click Create Virtual Private Gateway

Input:

- **Name tag:** ucertify-VPG
- **Type:** ipsec.1 (Default)

The screenshot displays the AWS Management Console interface for the 'Virtual private gateways' page. The left-hand navigation pane is expanded, showing the 'Virtual private network (VPN)' section with 'Virtual private gateways' selected. The main content area indicates that there are no virtual private gateways in the current region (United States (N. Virginia)) and provides a 'Create virtual private gateway' button. The right-hand pane contains a 'Caution' box and a series of steps (STEP 1 to STEP 6) for creating a VPG. The bottom of the screen shows a Windows taskbar with various application icons and a system clock.

Activity Evidence

STEP 1
Click on the "Click here to start" button on the left pane.

Caution
On the navigation bar, verify that the Region is selected as N.Virginia. If it is not selected, then select it from the list. [show figure](#)

STEP 2
On the navigation bar, click Services.

STEP 3
In the left pane, scroll down and click Networking & Content Delivery.

STEP 4
In the right pane, scroll down if needed and click VPC.

STEP 5
In the left pane, scroll down and under Virtual private network (VPN), click Virtual private gateways.

STEP 6
In the right pane, click Create virtual private gateway.

After creation, the VGW appears with:

- **State:** Pending
- **Attachment:** Detached (because not yet attached to a VPC)

What is a VGW?

A VGW is the AWS endpoint of a VPN connection. It works like the “router” sitting on the cloud side of a site-to-site tunnel.

Step 3:

Navigate to VPC Services

Left navigation → Networking & Content Delivery → VPC

The VPC dashboard is where AWS networking components like subnets, routing tables, gateways, and firewalls live.

Step 4:

Create a Site-to-Site VPN Connection

Left menu → Site-to-Site VPN Connections

Click Create VPN Connection

Fill out the Details section:

- **Name tag:** ucertify-VPN
- **Target Gateway Type:** Virtual Private Gateway
- **Virtual Private Gateway:** select your newly created VGW (ucertify-VPGW)
- **Customer Gateway:**
 - Select New
 - Enter provided Customer Gateway Public IP: 54.23.15.23 (given by lab)

The screenshot shows the AWS Management Console interface for creating a VPN connection. The 'Details' section is highlighted, showing the following configuration:

- Name tag - optional:** ucertify-VPN
- Target gateway type:** Virtual private gateway
- Virtual private gateway:** vgw-043ecf2143db40b7
- Customer gateway:** Existing
- Customer gateway ID:** vgw-043ecf2143db40b7

The 'Routing options' section is also visible, showing the 'Route table' dropdown menu.

On the right side of the console, there is a 'Figure E: The Details section' which provides instructions for the steps:

- STEP 2:** In the right pane, click **Create VPN connection**.
- STEP 3:** On the **Create VPN connection** page, under **Details**, in the **Name tag - optional** text box, type **ucertify-VPN** and from the **Virtual private gateway** list, select **ucertify-VPGW**.
- STEP 4:** Scroll down, under **Customer gateway**, select the **New** radio button and type **IP address** as **54.23.15.23**.

Step 5:

Configure Tunnel Options (Tunnel 2 as required)

Scroll down → Expand Tunnel 2 Options

Enter:

- **Inside IPv4 CIDR:** 169.254.8.0/30
- **Pre-shared Key:** uc_123456

The screenshot displays the AWS Management Console interface for configuring a VPN connection. The left pane shows the 'VPN connections' page with the 'Tunnel 2 options' section expanded. The 'Inside IPv4 CIDR' is set to 169.254.8.0/30, and the 'Pre-shared key for tunnel 2' is set to uc_123456. The right pane shows a 'Lab Summary' with steps 6, 7, and 8. Step 6 instructs to scroll down and expand Tunnel 2 options, and type the following details: a. Inside IPv4 CIDR for tunnel 2: 169.254.8.0/30, and b. Pre-shared key for tunnel 2: uc_123456. Step 7 instructs to keep all the remaining details as default, scroll down, and then click the 'Create VPN connection' button. Step 8 instructs to observe the VPN connection with the name ucertify-VPN. The bottom of the console shows a 'Lab Summary' section with a 'SUBMIT' button and a 'CLOSE' button.

Explanation:

- **Inside CIDR:** small network range for the encrypted tunnel interface
- **Pre-shared key:** used to authenticate both sides of the tunnel during IPSec negotiation

Step 6:

Create the VPN Connection

Scroll down → Click **Create VPN connection**

You will see a confirmation banner:

"You successfully created vpn-05a0ef42209cd9a88 / ucertify-VPN"

The screenshot displays the AWS Management Console interface. At the top, a green banner confirms the successful creation of the VPN connection: "You successfully created vpn-05a0ef42209cd9a88 / ucertify-VPN". Below this, the "VPN connections" table lists the newly created connection:

Name	VPN ID	State	Virtual private gateway	Transit gateway
ucertify-VPN	vpn-05a0ef42209cd9a88	Pending	vpgw-020daa173576f7a9f	-

The right-hand pane shows the "Activity" tab with instructions for Step 6, Step 7, and Step 8. Step 6 instructs to expand Tunnel 2 options and type specific details. Step 7 instructs to keep details as default and click the "Create VPN connection" button. Step 8 instructs to observe the VPN connection with the name "ucertify-VPN". Below the instructions is a small thumbnail image of the console showing the newly created VPN connection.

The VPN will show:

- **State:** Pending
- **Virtual Private Gateway:** Attached
- **Tunnel 1 / Tunnel 2:** Pending creation

This is expected in a sandbox environment

Key Networking Concepts Explained

- **Site-to-Site VPN:** A secure, encrypted connection between two private networks over the public internet.
- **IPSec:** The encryption protocol suite used to secure VPN tunnels.

- **Virtual Private Gateway:** The AWS endpoint of the VPN. Acts like your “cloud router.”
- **Customer Gateway:** Represents the on-premises router or firewall. Uses its own public IP to establish the encrypted tunnel.
- **Tunnel Inside CIDR:** The inner communication network used strictly by the IPSec tunnel endpoints.

Lessons Learned

- I learned how organizations securely connect remote networks and branch offices using AWS VPN technologies.
- I gained hands-on understanding of Virtual Private Gateways, Customer Gateways, and IPSec tunnel configurations.
- This lab helped me understand how cloud networking mirrors real on-prem infrastructure but with more flexibility and automation.