

AWS CloudFormation Lab 2 Report

Samuel Kojo Andam Quansah

University of Maryland Global Campus

CCS 625: Network Engineering- Project 2

Instructor Name: Dr. Lipton

May 15,2025

Introduction

In this lab, I utilized AWS CloudFormation to design and deploy a virtual private cloud (VPC) environment consisting of a VPC and an Internet Gateway. Using CloudFormation Designer, I visually created the resources, manually edited the underlying JSON template to add CIDR blocks and tags, established relationships between the components, and provisioned the environment as a stack. This report outlines my step-by-step implementation and key takeaways from the lab.

Step-by-Step Methodology

Step 1: Exploring AWS CloudFormation Dashboard and Designer

After logging into the AWS Console using the provided UMGC credentials, I navigated to **Services > CloudFormation**. From the CloudFormation console, I selected **Create Stack > With new resources (standard)**. Then, I selected **Design template** to launch the CloudFormation Designer, which opened with a visual canvas on top and a JSON editor at the bottom.

Step 2: Designing the CloudFormation Stack

From the Resource Types panel, I dragged the **AWS::EC2::VPC** resource onto the canvas. I then clicked on the JSON editor and added the following properties to define the CIDR block and tags:



Figure 1 defining CIDR block

Step 3: Creating the Internet Gateway

I dragged the **AWS::EC2::InternetGateway** resource onto the canvas and edited its properties as follows:

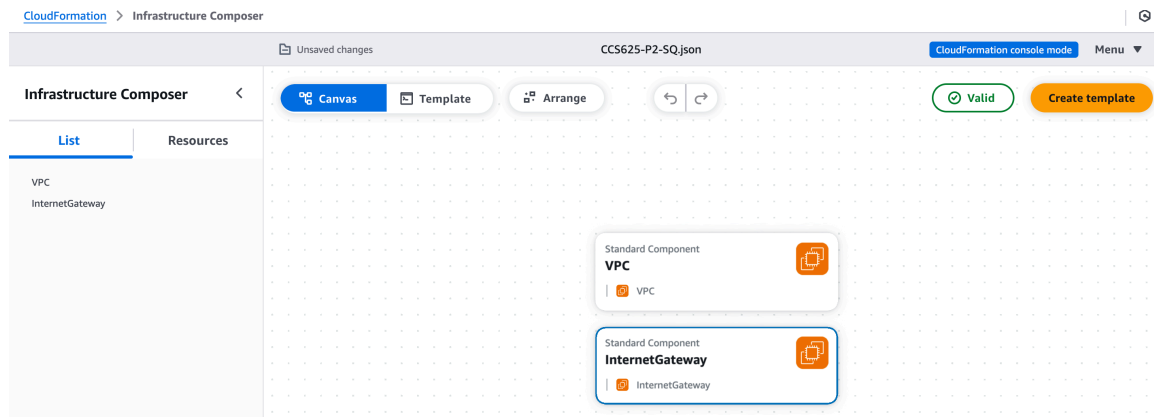


Figure 2 CloudFormation Designer with the full diagram

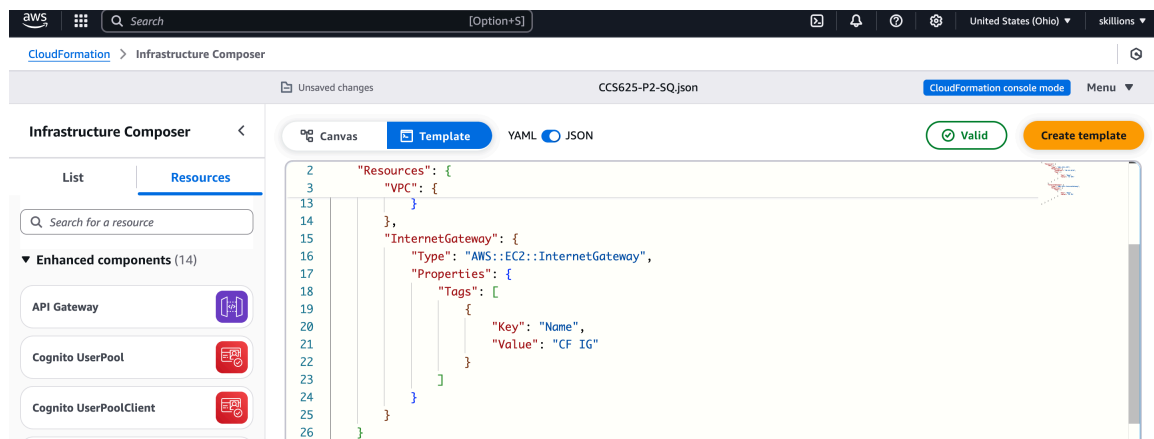


Figure 3 defining gateway properties

Step 4: Validating connection of Internet Gateway to the VPC

With both VPC and internet gateway configured in the template, the connection is automatically made. I validated the connection by navigating to my VPC dashboard to observe the attachment

as green.

The screenshot shows the AWS VPC dashboard for the VPC `vpc-0709f2c3943a8190d`. The left sidebar contains navigation links for VPC dashboard, Virtual private cloud, and Security. The main content area displays the VPC details, including VPC ID, State (Available), Block Public Access (Off), DNS resolution (Enabled), Main network ACL, IPv6 CIDR, Tenancy (default), Default VPC (No), Network Address Usage metrics (Disabled), DHCP option set, IPv4 CIDR (10.0.0.0/16), Route 53 Resolver DNS Firewall rule groups, DNS hostnames (Disabled), Main route table, and IPv6 pool. Below the details, there are tabs for Resource map, CIDRs, Flow logs, Tags, and Integrations. The CIDRs tab is active, showing a table with columns for Address family, CIDR, and Status. The table contains one entry for IPv4 with CIDR 10.0.0.0/16 and Status Associated.

Address family	CIDR	Status
IPv4	10.0.0.0/16	Associated

Figure 4 vpc dashboard (validating IGW attachment)

Step 5: Validating and Saving Template

I clicked on the **Validate Template** (checkmark icon) to ensure there were no syntax errors. Then, I saved the template locally as CCA625-P2-SQ.json.

The screenshot shows the AWS CloudFormation console for the template `CCS625-P2-SQ.json`. The left sidebar contains navigation links for CloudFormation, Infrastructure Composer, and Resources. The main content area displays the template code in JSON format. The code defines two resources: a VPC and an InternetGateway. The VPC resource has properties for Type, Properties, CidrBlock, and Tags. The InternetGateway resource has properties for Type, Properties, and Tags. The template is validated, and the status is "Valid".

```
1 {
2   "Resources": {
3     "VPC": {
4       "Type": "AWS::EC2::VPC",
5       "Properties": {
6         "CidrBlock": "10.0.0.0/16",
7         "Tags": [
8           {
9             "Key": "Name",
10            "Value": "CF VPC"
11          }
12        ]
13      }
14    },
15    "InternetGateway": {
16      "Type": "AWS::EC2::InternetGateway",
17      "Properties": {
18        "Tags": [
19          {
20            "Key": "Name",
21            "Value": "CF IG"
22          }
23        ]
24      }
25    }
26  }
27 }
```

Figure 5 validating json code

Step 6: Creating the CloudFormation Stack

1. Back in the CloudFormation dashboard, I selected **Create Stack > With new resources (standard)** and uploaded my saved template file.
2. I named the stack CCA625-P2-SQ and proceeded with default stack configuration settings.
3. On the review page, I verified the resources listed and clicked **Create Stack**.
4. I monitored the stack creation process under the **Events** tab until the status changed to **CREATE_COMPLETE**.

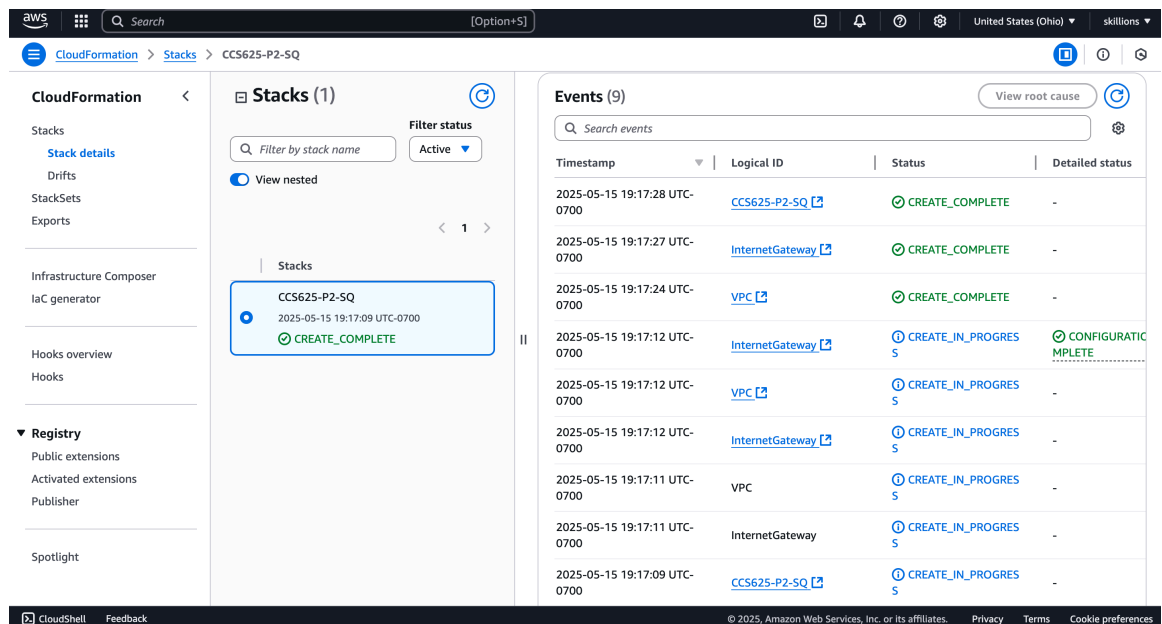


Figure 6 stack formation completion

Verification of Resources

- I went to the **VPC Dashboard** and confirmed that a VPC named "CF VPC" with the CIDR 10.0.0.0/16 was successfully created.
- Under **Internet Gateways**, I confirmed that the gateway named "CF IG" was created and attached to the correct VPC.
- Template "CCA625-P2-SQ" attached to lab submission

Conclusion

This lab provided hands-on experience with AWS CloudFormation to define and deploy infrastructure as code. By using the Designer tool and manual JSON editing, I created a properly structured VPC and Internet Gateway, linked them, and deployed them using a CloudFormation stack. This process emphasized the importance of understanding AWS resources, dependency management, and the flexibility of Infrastructure as Code (IaC). Furthermore, adapting outdated instructions to the current AWS UI sharpened my problem-solving and cloud navigation skills.