

**Postgraduate Diploma in Computing in DevOps**

|  |  |
| --- | --- |
| **Student Name:** | Anuradha Goli |
| **Student Number:** | L00177576 |
| **Module:** | IaC for DevOps Pipelines |
| **Title of Assignment:** | Assignment 1 – VPC CloudFormation |

Student Name: **ANURADHA GOLI**

Date Submitted: **07 November 2022**

Table of Contents

[GitHub Repository: 3](#_Toc118737089)

[Conclusion: 3](#_Toc118737090)

[YAML Template generation: 3](#_Toc118737091)

[CloudFormation 5](#_Toc118737092)

[Reference Links: 8](#_Toc118737093)

# GitHub Repository:

<https://github.com/L00177576/IDP-Assignment1>

The GitHub repository is created as a Private repository, with Ruth Lennon (ruth.lennon@lyit.ie) added as one of the collaborators.

Assignment template for creating VPC in Cloud Formation: <https://github.com/L00177576/IDP-Assignment1/blob/master/L00177576_VPC-advanced.yaml>

# Conclusion:

## YAML Template generation:

For building the VPC with the described architecture, I have followed the below steps:

1. Create VPC

Using the sample template defined in (*AWS::EC2::VPC - AWS CloudFormation*, no date), I have defined the VPC with Ipv4 address.

1. Create Internet Gateway

Using the template defined in (*AWS::EC2::InternetGateway - AWS CloudFormation*, no date), I have created the Internet gateway, which will be attached to VPC. If a resource in the public subnets has a public IPv4 address or an IPv6 address, an internet gateway enables the resource to connect to the internet.

1. Attach the Internet Gateway to the VPC

Using the template defined in (*AWS::EC2::VPCGatewayAttachment - AWS CloudFormation*, no date), VPCGatewayattachment is declared to attach an internet gateway to VPC, which enables the connectivity between the VPC and the internet.

1. Create the Public Subnets

Resources can be created and attached to subnets. The subnet is the range of IP addresses within the VPC. For resources connected to the internet, a public subnet will be used. Each subnet resides in one availability zone. Using the template defined in (*AWS::EC2::Subnet - AWS CloudFormation*, no date), I have created 2 Public subnets. Instead of selecting a specific Availability zone, I have used the Intrinsic function 'GetAZs' to select the indexed one from the list of availability zones.

1. Create Public Route Table

The route table contains a set of routes to determine where the traffic from every subnet is directed. I have created a route table for the Public subnet using the template defined in (AWS::EC2::RouteTable - AWS CloudFormation, no date).

1. Add Public Route to the Public Route Table

An internet gateway must be attached to VPC, and the route table associated with the public subnet must route to an internet gateway. Using the template defined in(*AWS::EC2::Route - AWS CloudFormation*, no date), I have created a public route table to public route table and associated it with the Internet gateway.

1. Associate the Public Subnets with the Public Route Table

Using the template defined in (*AWS::EC2::SubnetRouteTableAssociation - AWS CloudFormation*, no date), the subnet is associated with the route table, and they both must be in the same VPC. The Route table associations were defined for both public subnets.

1. Create the Private Subnets

Using the template defined in (*AWS::EC2::Subnet - AWS CloudFormation*, no date), I have created 2 Private subnets. Similar to creating public subnets, I have used the Intrinsic function 'GetAZs' to select the indexed one from the list of availability zones. The private subnet is not accessible via the internet using an Internet gateway; instead, a NAT device needs to be used.

1. Create the Security Groups

Each subnet's AWS resources can be safeguarded using a variety of security measures, such as security groups. Using the rules defined in (*Security group rules for different use cases - Amazon Elastic Compute Cloud*, no date), I have defined four security groups

* + 1. Application Load balancing - Enable HTTP/HTTPS access on port 80/443.
    2. SSH – allow unrestricted inbound access on port 22
    3. Web Server - Enable HTTP/HTTPS via port 80/443 for the Load balancer security group + SSH access via port 22 for the SSH security group
    4. Database - Open DB for access- 3306 is the default port to access MySQL.

1. Create Instances to link with Subnets

One Instance is created in each subnet total of 4 Instances. Mappings were created to look up the AMI from the mappings dictionary to define the values in Key-value pairs for the AWS regions. Each Instance is associated with a different security group to make it associated with subnets.

Once all the resources are created, metadata info is added to ensure all the parameters are in the defined sequence from the template. Finally, the YAML template creates a stack in CloudFormation designer (L00177576\_VPC-advanced.YAML).

## CloudFormation

* Using the template generated YAML in the previous section (<https://github.com/L00177576/IDP-Assignment1/blob/master/L00177576_VPC-advanced.yaml>), I have uploaded the template in the CloudFormation and created Stack

***Preview in Template Designer :***

Diagram

Description automatically generated

* Stack is created called L00177576-VPC

Graphical user interface, text, application, email, Teams

Description automatically generated

* Resources created when creating the stack

Graphical user interface, application, table, Excel

Description automatically generated

* Parameters defined in Stack

Graphical user interface, application, Teams

Description automatically generated

* The Stack is successfully created with no issues.

Graphical user interface, text, application, email, website

Description automatically generated

## GitHub Scripts

### **Push initial files to the Empty GitHub repository:**

git init

git add .

git status

git commit -a -m "Template files added"

git remote add origin <https://github.com/L00177576/IDP-Assignment1.git>

git push -f origin master

Text

Description automatically generated

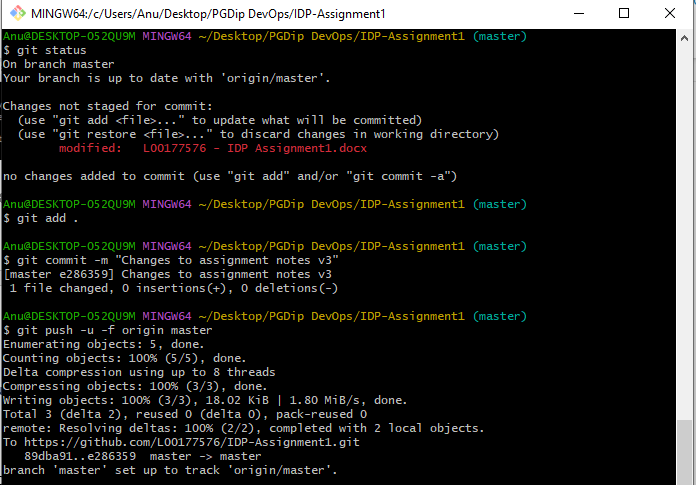
### **Push updated files to the Existing GitHub repository:**

git status

git add .

git commit -m "Changes to existing files."

git push -u -f origin master



# Reference Links:

*AWS::EC2::InternetGateway - AWS CloudFormation* (no date). Available at: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-internetgateway.html (Accessed: 7 November 2022).

*AWS::EC2::Route - AWS CloudFormation* (no date). Available at: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-route.html (Accessed: 7 November 2022).

*AWS::EC2::RouteTable - AWS CloudFormation* (no date). Available at: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-routetable.html (Accessed: 7 November 2022).

*AWS::EC2::Subnet - AWS CloudFormation* (no date). Available at: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-subnet.html (Accessed: 7 November 2022).

*AWS::EC2::SubnetRouteTableAssociation - AWS CloudFormation* (no date). Available at: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-subnetroutetableassociation.html (Accessed: 7 November 2022).

*AWS::EC2::VPC - AWS CloudFormation* (no date). Available at: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-vpc.html (Accessed: 7 November 2022).

*AWS::EC2::VPCGatewayAttachment - AWS CloudFormation* (no date). Available at: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-vpc-gateway-attachment.html (Accessed: 7 November 2022).

*Security group rules for different use cases - Amazon Elastic Compute Cloud* (no date). Available at: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/security-group-rules-reference.html (Accessed: 7 November 2022).