**BUILDING A BASIC CLOUD NETWORK**

The most important component of AWS cloud networking is perhaps the Virtual Private Cloud (VPC). Once you create your free AWS account, you are given a default VPC. It is the foundation for anything you do on the cloud. Think of the VPC as a router (refer to materials sent earlier relating to basic networking).

In a traditional network setup process, when you approach an ISP for internet services in your building, you are given a router and an IP address. You must connect a switch to the router and then connect servers/computers to the switch. Similarly, in cloud computing, if you approach AWS for internet services, you’re given a VPC and an IP address. You then install a subnet within the VPC and servers/computers within the subnets.

This writeup will walk you through the sequence of setting up a custom network architecture within your AWS account.

**CREATE A VPC**

* Log into your AWS account with your IAM user account (with Administrator rights)
* On your landing page, type in VPC into the search field.
* Access your VPC home page and click on the ‘create VPC’ button on the left side of your screen.
* From the page that appears, enter a name for your VPC and select the Availability Zone in which you are creating a VPC. (Any AZ or simply click ‘No preference’)
* Select a valid IPv4 CIDR block. Something like 10.1.0.0/16
* Leave everything else as default and click ‘create VPC’.

The IGW is a gateway that allows traffic in and out of the VPC.

**CREATE AN INTERNET GATEWAY (IGW)**

* On the left side of the VPC home page, select the Internet gateway option.
* Click the ‘create internet gateway’ button on the right side of the screen that appears.
* Enter a name for your IGW and click the ‘create internet gateway' button.

**CREATE SUBNETS WITHIN YOUR NEWLY CREATED VPC**

* After creating your VPC and still on the VPC home page, select subnets on the left pane of your screen.
* Click on the ‘Create subnet’ button on the upper right of your screen.
* On the ‘create subnet’ page, select the VPC in which you want to create a subnet.
* Enter a name for your subnet.
* Choose an AZ or ‘No preference' for Availability Zone option.
* Enter a valid IPv4 CIDR block (You can use 10.1.2.0/24 here).
* If you need more subnets, click the ‘add subnet’ button.
* Click on the ‘Create subnet’.

**MAKING YOUR SUBNET ACCESSIBLE PUBLICLY.**

* Select subnets from the left side of the VPC landing home page.
* Identify your public subnet, click the checkbox of the subnet, and click ‘Action’ on the upper right side of your screen.
* Click on the ‘Edit subnet setting’ option.
* From the ‘Edit subnet settings’ page, under the ‘Enable auto-assign IP settings’ option, enable auto-assign public IPv4 address.
* Save your configuration.

After creating your VPC, a route table is created alongside the VPC. The route table defines how internet traffic is routed within your VPC. If you’re building a network that you want to grant free access to the outside internet, you may need to define or associate the route table to your public subnets.

**EXPLICITLY ASSOCIATE YOUR ROUTE TABLE TO YOUR PUBLIC SUBNET**

* From the landing page of your VPC, select ‘route table’ from the left pane of your screen.
* On the right side of your screen, select the route table for your new VPC.
* On the lower part of the screen, select ‘subnet association’.
* Select ‘Edit subnet association’ under ‘Explicit subnet association’.
* From the ‘Edit subnet association’ page, select the subnet you want to associate with the route table and click ‘save association’.

**DEFINE YOUR ROUTE**

* From the route table landing page, select the specific route table, then click route from the lower part of the screen.
* Click the ‘edit route’ button on the lower right side of the screen.
* From the ‘edit route’ page, click the 'Add route' button.
* Define the Destination as 0.0.0.0/0 (meaning you want to receive traffic from anywhere).
* Under Target, select internet gateway and then select the internet gateway you created earlier. Click save changes.
* **BUILD A BASIC CUSTOM CLOUD NETWORK,**
* **INSTALLING A MICROSOFT EC2 INSIDE YOUR NEW NETWORK.**
* **INSTALL A MICROSOFT WEB SERVER WITHIN THE EC2.**
* **HOST A WEBSITE INSIDE THE WEB SERVER.**

1. **VPC and VPC components**

* Create VPC
* Create Internet gateway (IGW)
* Attach IGW to VPC
* Create subnets – public and private subnets.
* Enable auto-assign public IPv4 address.
* Associate subnets to route table
* Define explicit subnet association with rt.
* Associate route table to IGW

1. **INSTALLING AN AWS EC2 - MICROSOFT**

* Install Microsoft Windows machine.
* Install machine within your public subnet.
* Set firewall rules to allow inbound and outbound traffic through http and https.
* Authenticate into your Windows machine through RDP.

1. **VIRTUALIZING INTO YOUR MICROSOFT EC2**

* Use the RDP to remote into your Microsoft EC2

1. **INSTALLING A MICROSOFT SERVER APPLICATION INSIDE THE AWS EC2**

When authenticated into Windows machine,

* Install Microsoft IIS website server.
* Download a website template (from your github account).
* Extract the files.
* Move the files into This PC > inetpub > root [your template]
* Check the website availability with the public IP address of your EC2 windows machine.

1. ***Repeat the above process on Ubuntu machine.***

**MID-TERM PROJECT**

Our team recently got a work order request to create the underlying network infrastructure to launch a public-facing website. The client also requires the website to be hosted on an architecture with minimal resources. This project will be divided into two parts – **The network infrastructure build part** and the **S3/web development part**. To ensure processes are well streamlined, we shall be creating IAM user accounts for everyone on the team. All implementations shall be done from within the same AWS root account. Please reach out to your team mates on slack when you need clarifications. The responsibilities of both teams are detailed below:

**TEAM 1**

Network Infrastructure build. Part A:

* Create a custom VPC. Name it ‘New-Customer’. [stick to the same naming conventions for all components you create within the VPC]
* Create a single public subnet within the VPC named ‘new-customer-public-subnet’.
* Create an Internet Gateway (IGW). Name it ‘New-Customer-IGW’
* Attach your IGW to your custom VPC.
* Associate your subnet to your route table.
* Associate your IGW with your custom VPC.

Network Infrastructure build. Part B:

* Create two EC2 Instances within your custom public subnet.
  + A Windows EC2 Instance
  + An Ubuntu EC2 Instance

Task complete!

**TEAM 2**

S3 Bucket/Web development team. Part A:

* Create an S3 bucket. Name it ‘netxbyte-new-customer-s3’.
* Upload the files from the new client as objects into your S3 bucket.

S3 Bucket/Web development team. Part B:

* Request Microsoft EC2 instance logon credentials from Network Infrastructure build team,
* Virtualize into your Windows EC2 instance using Remote Desktop Protocol (RDP) client application.
* Install a Microsoft IIS server application into your Windows EC2 instance.
* Download the website objects from your S3 bucket and launch the website.

S3 Bucket/Web development team. Part C:

* Request Ubuntu instance logon credentials from Network Infrastructure build team,
* Virtualize into your Ubuntu EC2 instance using Secure Shell (SSH) client application (Use putty).
* Install an Apache2 server application into your Ubuntu EC2 instance.
* Download the website objects from your S3 bucket and launch the website.

**Best of Luck!**

*You may be wondering why they want their website objects stored on S3 bucket and not slack or github: They are looking to build an AWS-native environment for their applications. They hope to start small but build it from down-up. Feel free to reach out to me or Siri for more information.*

The deadline for this project is 10/5/2023 17:59:00PM CDT.