Lab 01

Setting Up and Testing Amazon Web Services (AWS) Virtual Private Cloud (VPC)

The objective of Lab 01 was to learn to setup AWS and create a VPC. The task that led to this lab was the setup of AWS account, which we did last week. As for the lab itself, we selected a Ubuntu Server virtual machine instance. However, before adding the server image, we also setup the networking components like CIDR, DNS, subnet and gateway. After setting up the image, we also setup security group as a way for us to securely connect from our computers to the server.

During the lab, first we learn about the AWS console interface. We learn to add the most commonly used features like VPC and EC2 to the shortcut bar. We then make sure, we are at proper Availability Zones for faster access to the server. Amazon has the services served from different global regions called Availability Zones. I selected N. Virginia for my setup.

Next in the lab we create Virtual Private Cloud and enable domain name server (DNS). We then create subnets to allow a number of different machines (in our case VM instances) to connect to the internet through a common gateway to the internet. For classless inter-domain routing (CIDR) we choose 173.1.0.0/16 as the IP address. We also made sure we get an auto-assigned public IP when we connect. One convention we used during our lab was we choose lastname<component descriptor> mechanism to name different components. E.g., In my case a subnet would be named NepalPublicSubnet and a VPC would be named NepalVPC and so on.

For the gateway we allow all networks to be able to connect to the VPC, provided they have proper credentials setup. For that we choose 0.0.0.0/0 as the destination IP in the routing table.

Above steps were performed to setup the VPC. Next part was setting up a VM. For that we used EC2 Dashboard from the shortcut bar, where we had put the EC2 icons in previous steps. We first create an instance and select Ubuntu Server 14.04 LTS image. During the network setup step, while creating the image, we select the VPC we created earlier. We also make sure, we have correct subnet selected. As for the primary ip we select 173.1.1.10 as the static IP, although the operating system thinks it is obtaining the IP address dynamically. AWS does that for us. For the storage size, we select the default size of 8 GB.

Finally, we create a security group named using above naming scheme. E.g., NepalSG. As mentioned above, this helps us communicate with the server securely. We can add different rules for connection to different services. AWS has SSH rule added by default. To connect through windows computers we add RDP as one more rule. If we decide to add another service that requires a dedicated port, we can add that as a rule as well. After the setup is completed, we start the VM using the Launch button.

To connect through SSH, we create a key file and download it to our disk. We then use PuttyGen program to convert the key file to a format that Putty understands (i.e., convert from pem format to ppk). We use this file to connect to our server's command line using Putty. In Putty, we basically select the key file and connect to our server after specifying the ip address (using the IP address provided by the server instance we created above).

After we connect, we ping to an internet website to see if the internet was setup properly (during the network configuration steps we performed above). Lastly, we shutdown the server from both command line and EC2 interface.

And in this way, we were able to setup VPC and OS instance in AWS. In my opinion, this lab is a setup for labs that will follow, where we will require similar instances to learn about different OS specific system administration commands and activities that needs to be performed.