Getting started with AWS VPC

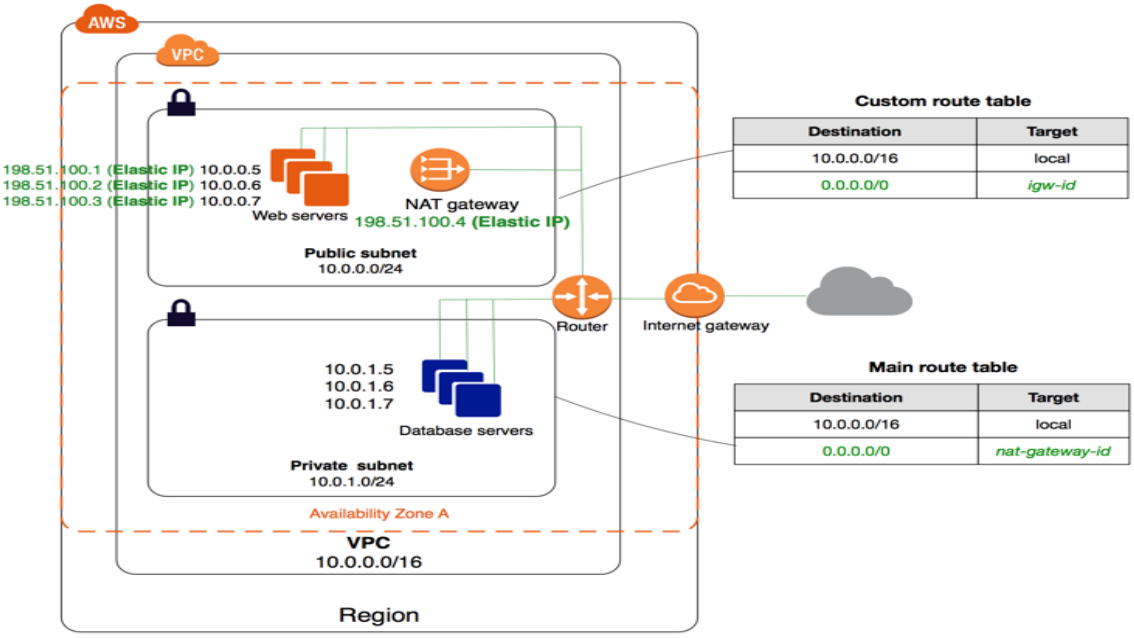
Amazon Virtual Private Cloud (Amazon VPC) enables you to launch Amazon Web Services (AWS) resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

A *virtual private cloud* (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS cloud. You can launch your AWS resources, such as Amazon EC2 instances, into your VPC. You can configure your VPC; you can select its IP address range, create subnets, and configure route tables, network gateways, and security settings.

A *subnet* is a range of IP addresses in your VPC. You can launch AWS resources into a subnet that you select. Use a public subnet for resources that must be connected to the Internet, and a private subnet for resources that won't be connected to the Internet.

A. VPC with Public and Private subnets

1. The scenario you are asked to implement is depicted below. Please note that for this basic exercise you are not required to deploy an actual webserver and MySQL Database server – you are just required to configure the infrastructure that could support the application environment.



Full details can be found here.

http://docs.amazonwebservices.com/AmazonVPC/latest/UserGuide/VPC_Scenario2.html

2. You will need to create the following security groups
    a. WebServerSG—For the web servers in the public subnet
    b. DBServerSG—For the database servers in the private subnet
3. From the Your VPC screen note the details for your VPC – VPC ID, DHCP Options set, Main Route table, Default Network ACL. Also note the Subnets, Internet Gateways and Elastic IPs that have been created for your VPC. You should clearly name your VPC resources.
4. You can choose yourself whether you want to work with Windows or Linux machines or a mixture of both. In keeping with previous exercises we suggest you launch a free-tier Linux AMI in the **Public** subnet in the VPC. You do not need to specify an IP address e.g. 10.205.0.10 /24 as the DHCP server should assign an appropriate address. Make sure you enable Auto-Assign Public IP address.

5. You should put in some meaningful details in the Instance details tags key – value screen e.g. Webserver
6. Launch the server in the relevant Security Group e.g. WebServerSG
7. Login to your Linux instance You will see both the Private and Public IP addresses assigned to this server. As in previous exercises you can install and configure an nginx webserver and connect to the Public IP address from your own desktop.
8. Now you can launch another Linux instance – you can choose a basic AMI - this instance must be launched in the private subnet. This Server should be launched into the DBServerSG.
    a. You **DO NOT** want to Auto-Assign a Public IP address to this server. If you enable/allow ssh from the WebServerSG to the DBServerSG you will be able to login from the Server in the Public subnet to the server in the Private subnet. You could do this by copying the pem key for the dbserver instance up to your webserver instance and using ssh to login to that instance however this is **NOT** recommended. You could enable ssh agent forwarding as described in this post - https://aws.amazon.com/blogs/security/securely-connect-to-linux-instances-running-in-a-private-amazon-vpc/ .
    b. The first step in using SSH agent forwarding with EC2 instances is to configure a bastion in your VPC. The instance you use for your bastion should be purpose-built and use it only as a bastion and not for anything else. The bastion should also be set up with a security group that's configured to listen only on the SSH port (TCP/22) and accept connections from trusted IP addresses. Never place your SSH private keys on the bastion instance. Instead, use SSH agent forwarding to connect first to the bastion and from there to other instances in private subnets. This lets you keep your SSH private key just on your computer. Always have more than one bastion. You should have a bastion in each availability zone (AZ) where your instances are. Configure Linux instances in your VPC to accept SSH connections only from bastion instances.
    c. Once you ssh from your webserver instance (or Bastion host if you have configured one) to your dbinstance you can check your public IP address using

        wget http://ipinfo.io/ip -qO -

    d. What is the Public IP address of the server in your Private Network ? What does it correspond with?

NOTE: The Scenario 2 wizard in this exercise creates a NAT Gateway (the recommended solution from AWS although you can use a NAT instance also). The NAT Gateway does cost approx. €1.50 per day so you should make sure you delete this instance when you are finished the exercise. When you are working on your assignment using a VPC you can use the same Elastic IP address for your NAT Gateway each time you create a new NAT Gateway and this should not require you to modify route tables.