1. Build 2 subnets including one public subnet, one private subnet.

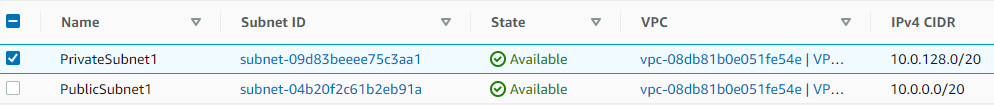


Figure 1. Creation of both Public and Private Subnets

1. Start 2 EC2 instances in public subnet, use t2-micro type to save money. Use userdata to  
   install Apache on both instances.

This step is giving me a little bit of trouble, so I figured I would include screenshots of all my steps. Below you can see I correctly used userdata to install apache on the amazon Linux image, for two ec2 instances within the public subnet, using the public security group which also allows for ssh traffic, so I can try to connect to it and see the “Hello World” line printing.

Graphical user interface, application

Description automatically generated

Figure 2. Using userdata to install apache on 2 pubic ec2 instances

Upon creating, all my instances appear to be running correctly:

Graphical user interface, text, application, email

Description automatically generated

Figure 3. Public instances running

However, I try to connect to them both through the URL provided or through ssh and I get a connection timeout, so I am not sure if this is a firewall issue or something wrong with either my security group or network proxy.

Graphical user interface, text

Description automatically generated

Figure 4. Attempting to connect to the public EC2 instances to prove Apache is installed

1. Start 2 EC2 instances in private subnet, use t2-micro type to save money.

Graphical user interface, application, email

Description automatically generated

Figure 5. Running two EC2 Instances within the Private Subnet

1. Make sure routing table of the router in the public subnet supports both IPv4 and IPv6.

Graphical user interface, text, application, email

Description automatically generated

Figure 6. Public Route Table setup

Graphical user interface, text, application

Description automatically generated

Figure 7. Supporting both IPv4 and IPv6 for the Public Subnet Route Table

1. Add a NAT proxy in the public subnet. (It is up to you to choose if you want to use NAT  
   Gateway).

Graphical user interface, application

Description automatically generated

Figure 8. NAT Gateway in Public Subnet

1. Configure your SG to make sure the public subnet could accept http requests in. Make sure your ACL allow the ingress and egress http traffic.

Graphical user interface, application

Description automatically generated

Figure 9. Security Group in Public Subnet accepting IPV4 and IPV6 Http traffic

Graphical user interface

Description automatically generated

Figure 10. Outbound Rules for Public Security Group

Graphical user interface, application

Description automatically generated

Figure 11. Network ACL Inbound Rules to Ingress HTTP for IPV4 and IPV6

A screenshot of a computer

Description automatically generated with medium confidence

Figure 12. Network ACL Outbound Rules to Egress HTTP for IPV4 and IPV6

1. Configure the routing table of your private subnet to make sure it only allows the 2 EC2  
   instances to access the internet through the NAT proxy in your public subnet.

Graphical user interface, application

Description automatically generated

Figure 13. Private Route Table accessing internet through NAT Proxy in Public Subnet

1. Draw the infrastructure and take the screenshots of your AWS cloud configuration console.

I used the online tool draw.io to create a diagram of the infrastructure, this way I can use professional symbols for the components of the VPC, and so it would look neater,

Diagram

Description automatically generated

Figure 14. Generalized VPC Diagram