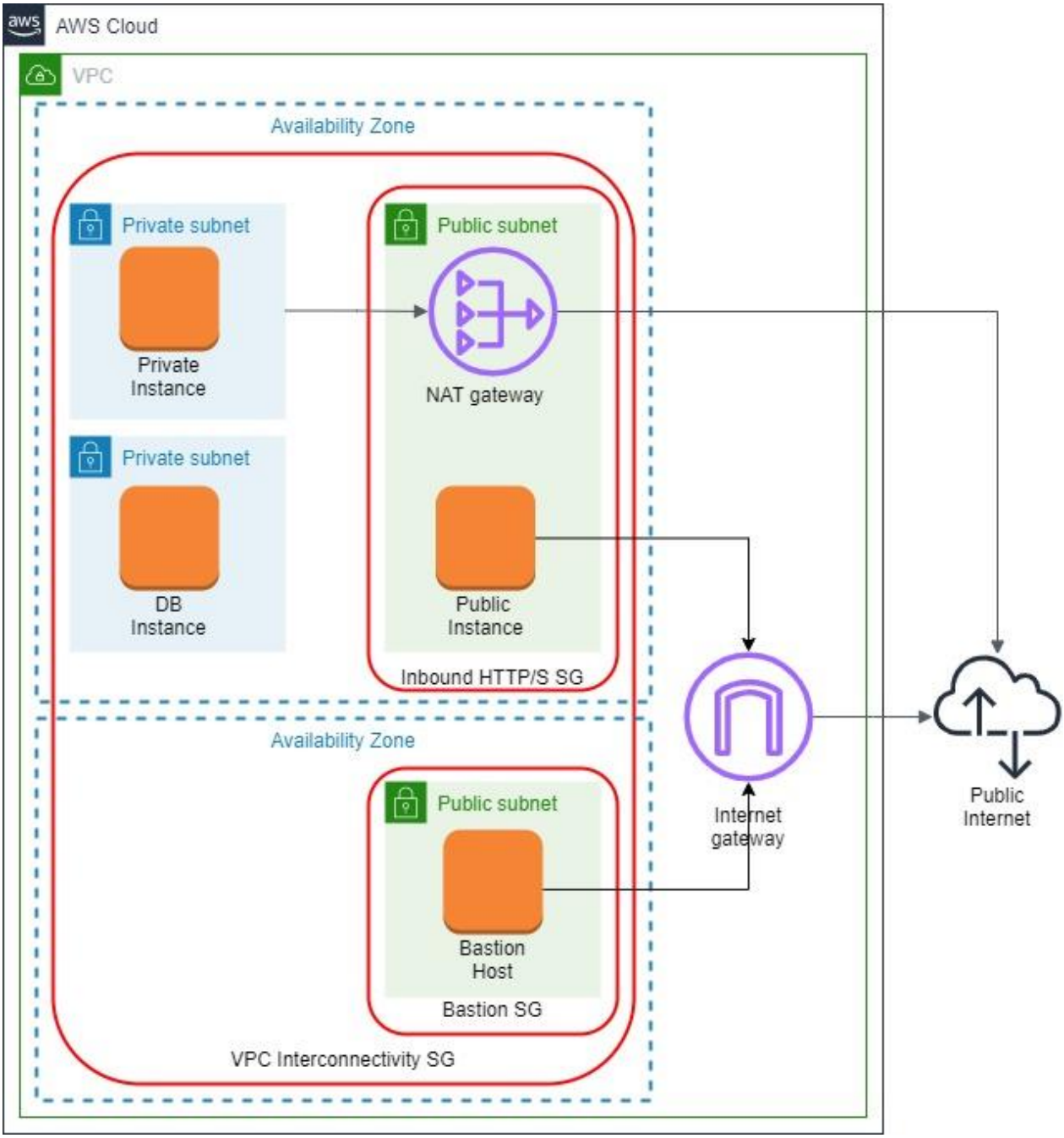
Task #6 – VPC

What to do

Create a VPC which fulfills the following architecture (AWS diagram notation):



1. The VPC should have a name following this convention *<ProjectName>-Network* and a CIDR block of *10.0.0.0/16*.
2. Create an internet gateway named *<ProjectName>-IGW* and attach it to the VPC.
3. Create two public subnets in the VPC:
    a. *<ProjectName>-PublicSubnet-A* in the first AZ with a CIDR block of *10.0.11.0/24*.

    b. *<ProjectName>-PublicSubnet-B* in the second AZ with a CIDR block of *10.0.21.0/24*.
    c. Make them public and choose *Auto-Assign Public IP*.
    d. Create a new route table named *<ProjectName>-PublicRouteTable*. Add a *10.0.0.0/16 – Local* route and a route to the *<ProjectName> IGW* to it.
    e. Associate the subnets with the new route table.
4. Create private subnet in the VPC:
    a. *<ProjectName>-PrivateSubnet-A* in the first AZ with a CIDR block of *10.0.12.0/24*.
    b. Create new route table named *<ProjectName>-PrivateRouteTable-A*. Add a *10.0.0.0/16 – Local* route to it.
    c. Associate private subnet with the new route table.
5. Create DB subnet in the VPC:
    a. *<ProjectName>-DbSubnet-A* in the first AZ with a CIDR block of *10.0.13.0/24*.
    b. Create a new route table named *<ProjectName>-DbRouteTable*. Add a *10.0.0.0/16 – Local* route to it.
    c. Associate DB subnet with the new route table.
6. Create NAT gateway for the private subnet in public subnet:
    a. *<ProjectName>-NatGateway-A* with an elastic IP for the subnet *<ProjectName>-PrivateSubnet-A*.
    b. Add the gateway A the route table *<ProjectName>-PrivateRouteTable-A*.

    NOTE: Be aware about NAT costs 1$ per day. Remove if not using right now.

7. Create a bastion host in the public subnet in the second AZ.
8. Create EC2 instance in the public subnet in the first AZ. Install the application developed in module 3 on the public instance.
9. Create one EC2 instance in the private subnet and one EC2 instance in the DB subnet. The instances do not have to have any special contents.
10. Create security groups:
    a. To allow inbound SSH traffic only from your IP address. Apply security group to the bastion host.
    b. To allow inbound HTTP/S traffic from anywhere. Apply security group to the public instance.
    c. To allow all inbound traffic from other instances associated with this security group. The security group should specify itself as a source security group in its inbound rules. Apply security group to all the instances.
11. Ensure:
    a. the application on the public instance is available from anywhere
    b. the private and DB instances are available from the bastion ONLY when you're connected to it over SSH (use the *ping* command or also *ssh* them)
    c. the bastion host and public instance have access to the Internet (*ping* Google, for example)
    d. the private instance has access to the Internet (*ping* Google, for example)
    e. the private and public instances have access to the DB instance (*ping* again)
    f. the DB instance doesn't have Internet access

What should I remember?

1. **Once you create AWS Account -> Setup Multi-factor Authentication**
2. **Do NOT share your account**
3. **Do NOT commit your account Credentials into the Git**
4. **Terminate/Remove all created resources/services once you finish Module**
5. **Please Do not forget to delete NAT Gateway if you used it.**
6. **Do NOT keep instance running if you don’t use it**
7. **Carefully keep track of billing and working instances so you don't exceed limits**