VPC [VIRTUAL PRIVATE CLOUD]

* It lets you define your own network in aws. You can use vpc to create your own network and use aws resources like EC2 in that network.
* You have the complete control of that network like giving the ip address range you want, block suspicious ip (or) ip add range, configuring route tables, create subnets, igw.
* The first 3 and last ipaddress are not available. Because, it is used for router, dns, broadcast and network address.
* You can create public and private facing subnets. Public-subnets for web servers to access the internet, private-subnets for databases which you don’t want internet access.
* You can have multiple layers of security like security groups and Network Acl’s to control inbound and outbound traffic.
* If you launch an instance in public-subnet, you can access internet directly by attaching IGW. If you launch instance in private-subnet, you can access internet through NAT. so that your private ip won’t show to public.
* Route table is nothing but a set of rules, called routes which determines which determines where the network traffic is directed.
* Each subnet in your VPC must be associated with the route table. The table controls routing for the subnet.
* When you launch an instance in a new vpc, public **dns** **name** won’t come for that instance. Because, dns hostnames is disabled by default in newly created VPC.
* To enable Dns host name,

**Go to vpc.**

**Select your vpc.**

**Click actions.**

**Choose edit dns hostnames. Choose Yes.**

CREATE A VPC

* To Create a VPC,

**Go to VPC.**

**Select your vpc’s option.**

**Click Create VPC.**

**Type a name to vpc.**

**CIDR block (192.160.0.0/16).**

**Click Create**.

* Create two subnets, public and private. Public will have internet access and private have NAT instance to access to internet.
* Create a **public** subnet with CIDR **192.168.10.0/24.**

**Go to VPC.**

**Click on Subnet.**

**Click create subnet.**

**Type a name(public).**

**Select your vpc.**

**Select AZ.**

**CIDR block (192.168.10.0/24).**

**Click Create.**

* Create a **private** subnet with CIDR **192.168.20.0/24.**

**Click subnet.**

**Click create subnet.**

**Type a name (private).**

**Select your vpc.**

**Select AZ.**

**CIDR block (192.168.20.0/24).**

**Click Create.**

* Now we have two subnets, if you launch any instance in these two subnets, you won’t able to access internet. Because, internet gateway is not attached to these subnets. To access internet, you have to attach an IGW to the subnet.
* To create an IGW,

**Go to internet gateway.**

**Click create gateway.**

**Select your vpc.**

**Click create.**

* Now we created a IGW and attached it to VPC.
* Now, you have to associate a subnet to route table and add a rule in that route table with IGW.
* To Create a route table,

**Go to Route table.**

**Click Create route table.**

**Select your VPC.**

**Click Create.**

* By default, a default route table will be created automatically while creating a vpc. If you want, you can attach igw to that default route table (or) if you want to create a new route table, create and attach igw to it.
* Go to route tables,

**Select your new route table.**

**On Association tab.**

**Click edit.**

**Select your public subnet and save.**

* On routes tab,

**Click edit.**

**Choose add another route.**

**Give Destination (0.0.0.0/0).**

**Select your igw and save.**

* 0.0.0.0/0 – destination for all networks.
* Now, you can launch instances in this subnet to connect to internet.

NAT INSTANCE

* By default, instances in pvt-subnet can't access internet directly like pub-subnet. They need to access internet through NAT instance.
* So that, first we have to create an NAT instance in pub-subnet, add a route in the route table of pvt-subnet with Nat instance ID.
* Go to pvt subnet route table,

**Click add another rule.**

**Destination = 0.0.0.0/0.**

**Target = NAT instance ID.**

**Click, Save**

* Disable the **source and destination check** of the NAT instance.
* Login into NAT and from there login into instance which is in private subnet with its **private** **ip** and ec2 **key**-**pair.**
* Now, test the connection by pinging another instance from private-facing instance.

**NAT GATEWAY**

* Aws gave an alternative for NAT instance called NAT gateway.
* In the nat instance, we have to configure everything. But, in nat gateway, aws will manage everything.
* You just have to create a NAT gateway in public subnet and add a route in private subnet’s route table to route traffic to private instance from NAT gateway.
* NAT gateway is costlier than NAT instance.

SECURITY GROUPS

* Security groups are like firewall for your instances to control the inbound and outbound traffic. Security acts at instance level.
* You have to select security groups while creating ec2. If you didn’t selected any security group, vpc automatically selects default one.
* For each security group, you add rules to control inbound and outbound traffic. Like protocol, port, ip etc.
* To create a security group,

**Go to Security groups**

**Click create a security group**

**Type a name to security group.**

**Select your vpc and choose Create.**

* On Inbound rules tab,

**Click edit, select rules you want (ssh, http, https).**

**Enter source as 0.0.0.0/0 (or) any ipaddress and Click Save**.

NETWORK ACL’S

* These are the additional security layer for vpc which acts as firewall for controlling traffic in and out.
* It provides same security as security groups, which also controls the inbound and outbound traffic.
* The main difference between them is .. Security groups acts in instance level and acl acts in subnet level.
* Security groups are stateful, where acls are stateless.

**Stateful** = if you add an inbound rule, outbound rule will be added automatically.

**Stateless** = you have add both type of rules manually.

* Whatever the traffic comes, first it will first face the acls because its subnet level security. if port 80 is blocked in acl, it won't allow even it is allowed in security groups.
* Acl's have rules same as security groups. But with an extra field called **RULE#**. These rules are numbers which arranged in ascending order. if you give a port and protocol with rule number = 50, it will apply that rule before the rule which has no=100 (1-32766).
* To create an NACL,

**Go to VPC.**

**On left pane, select network Acl.**

**Click create ACL.**

**Type a name and select vpc.**

**Click subnet associations.**

**Click edit, select your subnets and click save.**

* The subnets you selected will use NACL for all inbound and outbound traffic.
* Go to inbound tab,

**Click edit and choose add rule**

**Type rule no, choose protocol, choose port, type an ip (or) network.**

**Choose Allow (or) Deny for that specific n/w, Click Save.**

* Go to outbound tab,

**Type a rule no, add details just as inbound rule, Click save.**

* EX : rule=50 : protocol=ssh : port=22 : source=192.168.10.222/32 : deny : save.
* In the above example, we added a rule for ip 192.168.10.222/32 to deny traffic from ssh with port 22. ACL will block that ip from port 22, even it is allowed in security group.

VPC PEERING

* It is used to make communication and connection between two vpcs that enables you to route traffic between them using private ip.
* With Peering, you can communicate with other vpc’s which are in your own aws account and other aws accounts.
* You have to create an peering connection and send request to other vpc and tell them to accept the request to make connection between them.

CREATE AND ACCEPT A VPC PEERING

* Go to VPC,

**Click peering connection,**

**Click create connection.**

**Type a name to peering.**

**Select your vpc requester to connect.**

* If you want to connect to other aws account,

**Select another account,**

**Type acc ID and select vpc.**

**Select vpc requester and vpc accepter(another acc).**

**Click Create.**

* Select your peering, click actions and choose accept request.
* If you peer to other aws account, other user should accept the request in his account.
* If you don’t want a peering connection, you can reject the connection.
* Now, you have to update the route table with this subnet associated with peering to send traffic from instance to other instance in peer VPC.
* Route table must add in both sides to send traffic between peering connections.
* To update the route table.

Create two route tables for requester and accepter.

* Go to requester route table,

**choose edit.**

**choose add another route.**

**Destination = accepter subnet.**

**select peering in target and save.**

* Go to accepter route table,

**click edit.**

**choose add another route.**

**Destination = requester subnet.**

**select peering in target and save**.

* You can use vpc peering between two (or) more vpcs.

EX: VPC A, VPC B, VPC C - You want communication between these 3 vpcs.

Create a vpc peering between VPC A & VPC B. Add both peer in A & B.

Create a vpc peering between VPC B & VPC C. Add both peer in B & C.

Create a vpc peering between VPC A & VPC C. Add both peer in A & C.