**VIRTUAL PRIVATE CLOUD**

A Virtual Private Cloud is a Virtual Network that closely resembles a traditional Networking that you operate in your own data center, with the benefits of using the Scalable Infrastructure of AWS.

[or]

VPC is a Virtual Network or datacentre inside AWS for one client.

**WHY VPC?**

The three main reasons why we are using VPC are Security, Privacy & Prevention of loss of data

If we are not comfortable when we need to use a public network of a cloud service like AWS, in order to use their services, The VPC provides an extra layer of security so they can decide their own network. So here it gives only security but not privacy. When we launch an instance they will provide VPC by default, but in real-time we need to customize our VPC for getting more security, privacy, and prevention of loss of data.

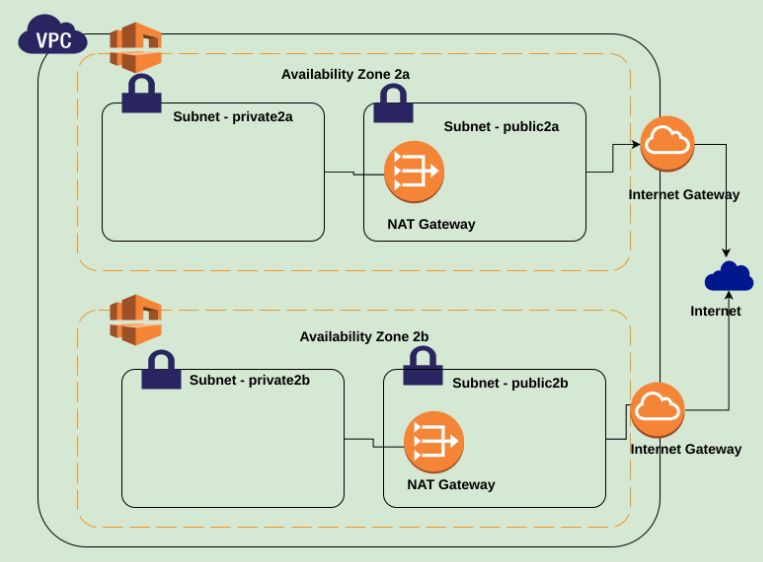
**FEATURES OF VPC:**

* It is logically isolated from other virtual N/W in the AWS Cloud.
* We can create 5 VPCs for each region and 200 subnets in 1 ACCOUNT.
* We can allocate a max 5 Elastic IP.
* Once we created VPC, DHCP(Dynamic Host Control Protocol), NACL(Network Access Control List) and Security Group will be automatically created.
* A VPC is confined to an AWS Region and does not extend between regions.
* VPCs are always created in Regions, not in Availability zones.
* Once the VPC is created, you cannot change its CIDR Block Range.
* If you need a different CIDR Size, Create a new VPC.
* You can expand your VPC CIDR by adding New / Extra by adding new and extra IP addresses Except for Govt cloud and AWS China.

**TYPES OF VPC:**

1. DEFAULT VPC: Create in each Region when we created an AWS account. It has default CIDR, Security Group, NACL, Internet gateway & Routing table settings.
2. CUSTOM VPC: It is a kind of VPC created by the AWS account owner. AWS user creating the custom VPC cab decided on the CIDR. It has its own Security Groups, Network ACL, and Route Table. It doesn’t have any Internet Gateways by default, we need to create it if we needed.

**VPC ARCHITECTURE :**



In this VPC, we have 2 Availability Zones (2a, 2b)

Each Availability has its own subnet.

**COMPONENTS OF VPC:**

* **SUBNETS**: A region contains three or more availability zones. A VPC contains subnets that are used to logically separate resources inside a region. A subnet cannot span across multiple availability zones. A subnet can either be a private subnet or a public subnet based on its accessibility from outside of VPC and if it can access resources outside of VPC.
* **ROUTING TABLE**:, it essentially facilitates traffic in and out of a software-defined network. This traffic needs to know where to go, and this is achieved via route tables. A routing table in VPC has rules or routes defined for the flow of traffic. Every VPC has a default route table that is known as the main route table. You can modify this main route table and you can create additional route tables.
* **NAT GATEWAYS:** An Natgateway allows communication between resources such as EC2 and RDS instances in your VPC and the Internet. It is highly available, redundant, and horizontally scalable; that is, you do not need to attach more than one internet gateway to your VPC in order to support an increase in traffic.
* **SECURITY GROUPS**: It consists set of firewalls rules that control the traffic for your sample. You can have a single security group associated with multiple instances.
* **NETWORK ACL**: It is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. It adds an additional layer of security to your VPC.
* **VIRTUAL PRIVATE GATEWAY**: is the VPN(Virtual Private Network) hub on the Amazon side of the VPN connection to have a secure transaction. Users can attach it to the VPC from which they want to create the VPN connection.
* **PEERING CONNECTORS**: You can connect your VPC with one or more VPCs in the same/diff region through the VPCs peering option. This connection enables you to communicate with other VPCs using private IPv4 or private IPv6 addresses.
* **ELASTIC IP ADDRESS:** An Elastic IP Address is a public IPv4, static address that can be associated with anyone instance or one network interface at a time within any VPC in your AWS account.

**CREATE A VPC MANUALLY:**

1. **VPC:**

Name: vpc-name

CIDR: 10.2.0.0/16

1. **SUBNETS:**

Name: subnet-name

VPC: attach our vpc

CIDR: 10.2.0.0/24

1. **INTERNET GATEWAYS:**

Name: ig-name

Select and attach our-vpc

1. Select route tables and add route
2. Select subnet and modify auto assign IP

**PEERING:**

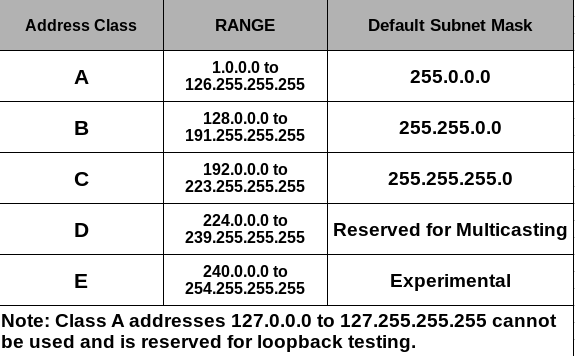
1. Create 2 vpcs and create 2 instances on each one of them with enabling public ip.
2. Create peering connection
3. Name -- > acceptor -- > requestor
4. Actions on perring connections -- > Accept Request
5. Go to route tables and add their each cidr blocks with target of peering connection
6. Give one cidr block to another cidr block
7. Actin -- > edit routes give one to two and two to one
8. Connect to one instance and ping with private address.

IPV4 IPV6

32 BIT 128 BIT

. :

NUMBERS ALPHA & NUMBERS



1. VPC -- > CREATE -- > VPC ONLY -- > NAME: --- > CIDR : 10.0.0.0/16 -- > CREATE
2. SUBNET -- > VPC -- > NAME: WEB SERVER -- > CIDR: 10.0.0.0/24 -- > CRAETE
3. SUBNET -- > VPC -- > NAME: APP SERVER -- > CIDR: 10.0.1.0/24 -- > CRAETE
4. IGW -- > NAME: -- > CREATE -- > ACTIONS -- > ATTACH -- > VPC
5. ROUTE TABLE -- > CREATE -- > NAME -- > VPC -- > CREATE -- > ROUTES -- > EDIT -- > 0.0.0.0/0 -- > IGW -- > SAVE -- > SUBNET ASSOCIATION -- > WEB SERVER -- > CREATE
6. EC2 -- > CREATE -- > NAME -- > VPC -- > SUBNET (WEB SERVER) -- > ENABLE IP : ENABLE -- > SG: NEW TAB -- > SG : WEB -- > INBOUND: SSH & IPV4 -- > OUTBOUND ALL TRAFFIC -- > SAVE -- > SELECT THIS SG -- > LAUNCH
7. EC2 -- > CREATE -- > NAME -- > VPC -- > SUBNET (APP SERVER) -- > ENABLE IP : DISABLE -- > SG: LAST ONE -- > LAUNCH
8. NOW CONNECT TO WEB SERVER BY USING TERMINAL OR BROWSER
9. NOW IF YOU WANT TO ACCESS APP SERVER IT WONT CONNECT, IF YOU WANT TO CONNECT TO APP SERVER WE CAN CONNECT FROM WEBSERVER
10. VIM ABC.PEM -- > COPY THE PEM CONTENT -- > SAVE -- > CHMOD 400 ABC.PEM -- >
11. ssh ec2-user@10.0.1.40 -i abc.pem
12. PING 8.8.8.8, IT WONT PING BECAUSE WE NEED TO ENABLE NAT GATEWAY
13. VPC -- > NAT GATEWAY-- > NAME -- > SUBNET: WEBSERVER -- > ALLOCATE ELASTIC IP -- > CREATE
14. ROUTE TABLE -- > CREATE -- > NAME: APP-RT -- > VPC -- > CREATE
15. APP-RT -- > ROUTES -- > 0.0.0.0/0 -- > NAT (VPC NAME) -- > SAVE
16. SUBNET ASSOCIATION -- > EDIT -- > APP SERVER -- > SAVE
17. NOW TRY TO PING IT WILL COME
18. VPC -- > CREATE -- > VPC ONLY -- > NAME: --- > CIDR : 10.0.0.0/16 -- > CREATE
19. SUBNET -- > VPC -- > NAME: WEB SERVER -- > CIDR: 10.0.0.0/24 -- > CRAETE
20. SUBNET -- > VPC -- > NAME: APP SERVER -- > CIDR: 10.0.1.0/24 -- > CRAETE
21. IGW -- > NAME: -- > CREATE -- > ACTIONS -- > ATTACH -- > VPC
22. ROUTE TABLE -- > CREATE -- > NAME -- > VPC -- > CREATE -- > ROUTES -- > EDIT -- > 0.0.0.0/0 -- > IGW -- > SAVE -- > SUBNET ASSOCIATION -- > WEB SERVER -- > CREATE
23. EC2 -- > CREATE -- > NAME -- > VPC -- > SUBNET (WEB SERVER) -- > ENABLE IP : ENABLE -- > SG: NEW TAB -- > SG : WEB -- > INBOUND: SSH & IPV4 -- > OUTBOUND ALL TRAFFIC -- > SAVE -- > SELECT THIS SG -- > LAUNCH
24. EC2 -- > CREATE -- > NAME -- > VPC -- > SUBNET (APP SERVER) -- > ENABLE IP : DISABLE -- > SG: LAST ONE -- > LAUNCH
25. NOW CONNECT TO WEB SERVER BY USING TERMINAL OR BROWSER
26. NOW IF YOU WANT TO ACCESS APP SERVER IT WONT CONNECT, IF YOU WANT TO CONNECT TO APP SERVER WE CAN CONNECT FROM WEBSERVER
27. VIM ABC.PEM -- > COPY THE PEM CONTENT -- > SAVE -- > CHMOD 400 ABC.PEM -- >
28. ssh ec2-user@10.0.1.40 -i abc.pem
29. PING 8.8.8.8, IT WONT PING BECAUSE WE NEED TO ENABLE NAT GATEWAY
30. VPC -- > NAT GATEWAY-- > NAME -- > SUBNET: WEBSERVER -- > ALLOCATE ELASTIC IP -- > CREATE
31. ROUTE TABLE -- > CREATE -- > NAME: APP-RT -- > VPC -- > CREATE
32. APP-RT -- > ROUTES -- > 0.0.0.0/0 -- > NAT (VPC NAME) -- > SAVE
33. SUBNET ASSOCIATION -- > EDIT -- > APP SERVER -- > SAVE
34. NOW TRY TO PING IT WILL COME