======
VPC
=====

=> VPC stands for Virtual Private Cloud.

=> VPC provides Virtual Network Environment for the AWS cloud resources.

=> A VPC allows users to create and manage their own isolated virtual networks within the cloud.

=> VPC provides flexible and secured network environment to manage our resources in AWS cloud.

================
VPC Terminology
================

1) Isolated network

2) CIDR blocks

3) Subnets

4) Route Tables

5) Internet Gateway

6) NAT Gateway

7) VPC Peering

8) Security Groups

9) NACL

===============
Types of IP's
===============

1) IPv4

2) IPv6

=========
IPV4
=========

=> It represents 32 bit numeric IP address.

=> It contains 4 octets seperated by period (.)

        Ex : 172.168.98.101

Note: Each octet can contain upto 8 bits

=> It is most widely used IP version in the market.

=> It supports approximatley 4.3 billion devices

===========
IPV6
===========

=> It represents 128-bit alpha numeric values address.

=> It contains 8 octets seperated by colon (:)

Ex: 2001:0db8:85a3:0000:0000:8a2e:0370:7334

Note: Each octet can contain upto 16 bits

=> IPv6 provides a significantly larger address space than IPv4

=> It supports approximately 340 undecillion unique addresses

Note: IPv6 introduced to overcome the IPv4 address exhaustion issue and support the growing number of internet-connected devices.

============
VPC Sizing
============

=> The process of deciding no.of IPs required for VPC and no.of IPs required for Subnets is called as VPC Sizing.

-> VPC Sizing will be calculated in 2 power

-> CIDR defines IP address range for our VPC (ex: /16, /24 etc...)

CIDR = Classless Inter-Domain Routing

Note: AWS allows VPC CIDR blocks between /16 and /28

10.0.0.0/16 => 2 power (32-16) => 2 power 16 => 65, 536

10.0.0.0/32 => 2 power (32-32) => 2 power 0 => 1

10.0.0.0/31 => 2 power (32-31) => 2 power 1 => 2

10.0.0.0/30 => 2 power (32-30) => 2 power 2 => 4

10.0.0.0/29 => 2 power (32-29) => 2 power 3 => 8

10.0.0.0/24 => 2 power (32-24) => 2 power 8 => 256

Note: For VPC we will use /16 and for subnet we will use /24 as CIDR range.

================
What is Subnet
================

=> VPC is a big network

=> A VPC is divided into mulitple smaller networks called as Subnets.

=> We can create 2 types of subnets

Public Subnet : Connected to the internet via an Internet Gateway.

Private Subnet : No direct internet access, used for internal resources.

======================
What is Route Table
======================

=> It defines how traffic is routed with in VPC

=> Each Subnet is associated with a Route Table.

==========
What is Internet Gateway (IGW)
==========

=> It is used to allow the resources to connect with Internet.

=> For Every VPC one Internet Gateway is required.

Note: If we attach IGW to Subnet then that is called as Public Subnet.

==========
What is NAT Gateway (NGW)
==========

=> NAT Gateway enables outbound internet access for private subnets.

Ex : NATGateway we should create in public subnet and we will attach that to private subnet.

==========
VPC Lab Task
==========

Step-1) Create VPC (Name : ashokit-vpc)

 Use VPC CIDR as : 10.0.0.0/16

Note : It will create one Route Table by default. Rename it as "ashokit-private-rt"

Step-2) Create Internet Gateway and Attach to our ashokit-VPC

Step-3) Create 2 Subnets (Public and Private Subnets)

        public Subnet CIDR : 10.0.0.0/24 (256 ips)

        private Subnet CIDR : 10.0.1.0/24 (256 ips)

Step-4) Create one new Route Table (Name it as public Route Table)

Step-5) Peform Route Tables Association with Subnets

                => public-rt => associate with public-subnet

                => private-rt => associate with private-subnet

Step-6) Attach IGW to Public Route Table in Routes, so that subnet will become public-sn (internet will be available).

Step-7) Create One EC2 VM in public subnet and another EC2 vm in private subnet.

Step-8) Test connectivity of both vms using SSH client.

==========
Step - 9 : Connect with 'private-sn ec2 vm' from 'public-sn ec2 vm' using 'ssh' connection
==========

Note: As both Ec2 instances are available under same VPC, we should be able to access one machine from another machine.

Step-1 : Upload pem file into public-sn ec2 vm

Step-2 : Give read permission for pem file

$ chmod 400 <pem-file>

Step-3 : Make SSH connection to private subnet vm using below ssh command

$ ssh -i <pem-file> ec2-user@private-ip

Note: It should establish connection (this is internal connection)

==========
NAT Gateway Lab Task
==========

1) Create NAT gateway in public subnet

2) Add NAT gateway in 'private-subnet-routute-table'

3) After NAT Gateway attached, we should be able to ping google from 'private-sn ec2' also

Note: Delete Elastic IP and NAT Gateway after practise

==========
VPC Peering
==========

=> In General, VPC will provide an isolated network for the resources in aws cloud.

=> If we create resources in private subnet then we can't access them outside.

=> If we have requirement to access one VPC resources in another VPC then we can use VPC peering concept.

Note: VPC peering we can establish between same account vpc's and different account vpc's also.

Note: When we are establishing VPC peering both VPCs CIDR range should be different

My Default VPC CIDR : 172.31.0.0/16

My Custom VPC CIDR : 10.0.0.0/16

Note: We can establish VPC peering for above two VPCs because their IP ranges are diffferent hence no IP collision.

==========
Procedure To Establish VPC Peering
==========

=> Go To VPC -> Select Peering Connections

=> Create VPC Peering request

VPC Peering (Requester) = ashokit__vpc

VPC Peering (Accepter) = default_vpc

=> Now you would see the status Pending Acceptance which means, Requestor has sent a request to the peer now target VPC needs to accept the request.

=> Go to VPC Peering -> Click on Actions -> Accept Request

=> Now we need to make entries in Route Tables

Custom-VPC route-table should allow default-vpc traffic

Default-VPC route-table should allow custom-vpc traffic

=> Now navigate to Route Tables -> Default VPC RT(Route Table) -> Edit routes

Note : By default we will have "local + igw" now we need to add custom vpc cidr)

####### Default VPC Route Table should have below 3 Routes ##########

Local : 172.31.0.0/16

IGW : 0.0.0.0/0

VPC Peering : 10.0.0.0/16

=> Now navigate to Route Tables -> Custom VPC RT(Route Table) -> Edit routes

Note : By default we will have "local + igw" now we need to add default vpc cidr)

####### Custom VPC Route Table should have below 3 Routes ##########

Local : 10.0.0.0/16

IGW : 0.0.0.0/0

VPC Peering : 172.31.0.0/16

########### Allow Traffic in VPC Security Groups ###########

Edit Security Group of Default and Custom VPC to allow traffic from each other

Default VPC Security Group looks like
SSH - 22 - all
All Traffic -> Custom -> 10.0.0.0/16

Custom VPC Security Group would look like
SSH - 22 - all
All Traffic -> Custom -> 172.31.0.0/16

==========Test VPC Peering Connectivity ==========

=> Create one EC2 VM in custom VPC and Create one EC2 VM in default VPC

=> Connect with Custom VPC EC2 VM using ssh client and try to ping default-vpc ec2 vm using private-ip

# Ping default-vpc EC2-VM private IP from ashokit-custom-vpc vm
$ ping <private-ip>

Note: If ping is success that means VPC peering is working as expected.

==========
Q) What is the difference between NACL and Security Groups ?
==========

==========

Security Group
==========

-> Security Group acts as a Firewall to secure our resources

-> Security Group contains Inbound Rules & Outbound Rules

inbound rules ---> controls incoming traffic

outbound rules ---> controls outgoing traffic

-> In One security group we can add 50 Rules.

-> Security Group supports only Allow rules (by default all rules are denied)

-> We can't configure deny rule in security group

Ex : 172.32.31.90 ----> don't accept request from this IP (we can't do this in SG)

-> Security Groups are applicable at the resource level
(manually we have to attach SG to resource)

-> Multiple Security Groups can be attached to single instance &
one instance can have 5 security groups

-> Security Groups are statefull (Any changes applied to incoming rules will be applicable for Outgoing Rules also)

-> Security Group acts as First Level of defense for Outgoing traffic.

======
NACL
======

-> NACL stands for Network Access Control List

-> NACL acts as a firewall for our Subnets in VPC

-> NACL applicable at the subnet level

-> NACL rules are applicable for all the resources which are part of that Subnet

-> NACL rules are stateless (Any changes applied to incoming rules will not be applicable for outgoing rules, we need to do that manually).

-> In NACL we can configure both Allow & Deny rules

Ex: We can block particual IP address (192.168.2.4) to connect with EC2 instance

-> One subnet can have only one NACL

Note: One NACL can be added to multiple subnets

-> NACL acts as first level of Defense for Incoming Traffic