VPC:It is a network inside AWS
1) VPC span with in region only
2) When you create a VPC, you must specify a range of IPv4 addresses for the VPC in the form of a Classless Inter-Domain Routing (CIDR) block;
for example, 10.0.0.0/16.
2) VPC can divided into multiple Subnets
3) When you create a subnet, you specify the CIDR block for the subnet, which is a subset of the VPC CIDR block.
4) Each subnet must reside entirely within one Availability Zone and cannot span zones
5) aws client has full control over the the ec2 instances hosted inside the vpc
6) It is similar to having your own data center inside aws
7) It is Logically isolated from other vpc on aws

There are two type of VPC :
1) Default VPC
2) Custom VPC

Default VPC: For every region AWS will create Default VPC
It consists of
a) default CIRD,
b) security group,
c) NACL
d) RT
e) IGW

For custom VPC AWS will create below comp
a) CIRD,
b) security group,
c) NACL
d) RT

VPC Comp:
==============
1) CIDR and IP address
2) Router: it is logical router, one subnet can connect to other subnet using router
3) Router table
4) Internet gateway
5) Security Group
6) NACL
7) Natgateway

VPC IP Addressing:
When you create a VPC, you must specify a range of IPv4 addresses for the VPC in the form of a Classless Inter-Domain Routing (CIDR) block.

1) Onece the VPC is created ,you can Not changes its CIRD block range
2) Min size of CIRD block is 28
3) max size of CIRD block is 16
4) the different subnets within a VPC Cannot overlap
5) AWS Reserved IPS in each subnets, First 4 and last ip address

Implid Router/Logical Router:
1) It is the central VPC routing function
2) It connects the different AZs together and connects the vpc to the internet
3) Each subnet will have a route table
4) the router uses to forward traffic within the VPC
5) the route table will also have entries to external destinations

RouteTable :
A route table contains a set of rules, called routes, that are used to determine
where network traffic is directed.
Each subnet in your VPC must be associated with a route table; the table controls
the routing for the subnet.

1) In VPC you can have up to 200 RT per VPC
2) You can have up to 50 routes entries per route table
3) Each subnet must be assosiacted with only one RT
4) If you donot specifiy a subnet to RT association ,the subnet will be assosicated with the default RT
5) you can change subnet assosiation with other RT you can edit main RT,but you cannot delete the main RT
6) Every RT in the VPC comes with default rule that allows all the subnets communicate with one anthor.
 you cannot modify/delete this rule

Internet Gateway:
1) by using IGW our network(VPC) connect to the public network.
ex : google.com.facebook.com
2) Only one IGW is attched one VPC. We cannot attch multiple IGW to the VPC.

NACL (Network Access control list):
1) NACL allow/deny request coming from/to the subnet.
2) NACL consist of inbound rules and outbond rules
3) Each subnet accosiate with only one NACL
4) One NACL can be mapped to multiple subnets within a VPC
5) NACL rules are stateless.
6) IN NACL we can define "allow"/"Deny" rules

Security Group:
==============
1) When ever we create VPC defaul security group will get created
2) we control inbound/outbond request to the instances
3) Security groups are stateful

NAT Gateway:
1) You can use a network address translation (NAT) gateway to enable
instances in a private subnet to connect to the internet or other AWS services,
but prevent the internet from initiating a connection with those instances.
2) Alway we need to launch NAT Gateway in Public subnet.
3) When we create NAT Gateway, Elastic Ip will be assigned with it.
4) If we want access internet from private subnet we need to made NAT entry into
 private subnet route table.

==

Creating custom vpc:
1) Create a VPC and divide into 2 subnets
2) make one subnets as public and other one has private
3) launch one ec2 instance in private subnet
4) launch one ec2 instance in public subnet & connet the
   public subnet machine.
5) connect private subnet machine and install
   a java into that machine.
==
Step1:
a) Goto VPC dashboard
   Networking & Content Delivery -> VPC
b) Click on "Your VPCs"
c) click on "Create VPC"
d) give any name to vpc
e) Provide CIDR of VPC
    ex : 10.10.0.0/16

Step2:
part 1:
a) From VPC dashboard click on "Subnets"
b) Click on "CreateSubnet"
c) Fill the required details
    Name tag: subnet1
 VPC: Select any VPC from Dropdown box
 IPv4 CIDR block: 10.10.1.0/24
d) Click on "Yes Create"

part 2:
a) From VPC dashboard click on "Subnets"
b) Click on "CreateSubnet"
c) Fill the required details
    Name tag: subnet1
 VPC: Select any VPC from Dropdown box
 IPv4 CIDR block: 10.10.2.0/24
d) Click on "Yes Create"

Part 3: Create Internet gateway & Attch
a) From VPC dashboard click on "InternetGateways"
b) Click on "CreateInternetGateway"

Step3: Lanch EC2 instance from EC2 dashboard, In EC2 configration details select the VPC & private subnet details
Step4: Lanch EC2 instance from EC2 dashboard, In EC2 configration details select the VPC & public subnet details

===
How to install software updates in Private subnets?
==
By using NAT GATEWAY we are able to access internet in private subnets:

1) Create custom VPC i.e. testvpc
   i.e. 10.0.0.0/16
2) Create two subnets ,one is public ,other one as private
   a)Public subnet creation
  i) create subnet 10.0.1.0/24 i.e SN1
  ii) create new route table i.e RT1
  iii) create internet gateway and attach it to "testvpc" VPC
  iv) edit route table "RT1" and add internet gateway details
  v) attach route table RT1 to subnet SN1
 b)Private subnet creation:
     i) create subnet 10.0.2.0/24 i.e SN2
  ii) create new route table i.e RT2
  iii) attach route table RT2 to subnet SN2
3) Create NAT GATEWAY in Public subnet
   i) Click on NATGATEWAY
   ii) click on "CREATE NAT GATEWAY"
   iii) Select public subnet of "testvpc" i.e SN1
   iv) allocate "Elastic IP address" to NAT GATEWAY.
        "click on create new EIP"
 v) Click on "Create Nat gateway"
4) Attach NAT GATEWAY to private subnet
  i) goto route table page
  ii)edit route table "RT2" and add Nat gateway details
5) Testing:
 ======
 i) Launch one Ec2 instance in "TestVPC" public subnet i.e SN1
   ii) Launch second Ec2 instance in "TestVPC" private subnet i.e SN2
 iii)ssh into step1 ec2 instance
 iv) ssh into step2 ec2 from step1 ec2 instance.
 v) run below command
       sudo apt-get update
==

VPC peering:
============
A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them
using private IPv4 addresses.

You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account.

A VPC peering connection helps you to facilitate the transfer of data. For example, if you have more than one AWS account, you can peer the VPCs across those accounts to create a file sharing network. You can also use a VPC peering connection to allow other VPCs to access resources you have in one of your VPCs.

To establish a VPC peering connection, you do the following:
===
1) The owner of the requester VPC sends a request to the owner of the accepter VPC to create the VPC peering connection. The accepter VPC can be owned by you, or another AWS account, and cannot have a CIDR block that overlaps with the requester VPC's CIDR block.

2) The owner of the accepter VPC accepts the VPC peering connection request to activate the VPC peering connection.

3) To enable the flow of traffic between the VPCs using private IP addresses, the owner of each VPC in the VPC peering connection must manually add a route to one or more of their VPC route tables that points to the IP address range of the other VPC (the peer VPC).

VPC Peering Limitations:
========================
1) You cannot create a VPC peering connection between VPCs that have matching or overlapping IPv4 CIDR blocks.
2) Both the VPC should be in same region.

scenario :
=========
In this scenario, you have two or more VPCs that you want to peer to enable full sharing of resources between all VPCs. The following are some examples:

Your company has a VPC for the finance department, and another VPC for the accounting department. The finance department requires access to all resources that are in the accounting department, and the accounting department requires access to all resources in the finance department.

Your company has multiple IT departments, each with their own VPC. Some VPCs are located within the same AWS account, and others in a different AWS account. You want to peer together all VPCs to enable the IT departments to have full access to each others' resources.

==
Instance Request Flow from/to instance:

S1) AWS instance is accessing google.com
  a) It is outgoing http request from aws instance
  b) It is incoming http request to google server.
Flow:
AWS instance --> SG(OB -> 80) --> NACL(OB --> 80) --> i/B http(80) req to google server NACL ->i/c http req to google server SG
 -->responce will be sent using ephmral port --> O/B ephmral port req to google server NACL --> I/B ephmral port req to AWS NACL
 --> request will be landed on aws instance.

S2) SSH request from intelliq server to AWS SERVER

  a) It is outgoing ssh request to intelliQ instance
  b) It is incoming ssh request to aws server.

Flow:
IntelliQ instance --> SG(OB -> 22) --> NACL(OB --> 22) --> i/B ssh(22) req to AWS server NACL ->i/B SSH req to AWS server SG
 -->responce will be sent using ephmral port --> O/B ephmral port req to AWS server NACL --> I/B ephmral port req to INTELLIQ NACL

--> request will be landed on IntelliQ instance.

===
NACL Configration:
===================
VPC is Having Three subnets,SN1,SN2,SN3

Assumptions:
==============
VPC CIDR: 10.0.0.0/16
SN1 : 10.0.0.0/24
SN2 : 10.0.1.0/24
SN3 : 10.0.2.0/24

1) Allow SSH inbound rules to SN1 from any
    other network
   a) Create NACL
   b) Add ssh inbound rule (Port no 22).
        PortRange no: 22
   Source:0.0.0.0/0
   c) Add ephemral port range inside outbound.
       PortRange :1025-65535
  Destination:0.0.0.0/0
   d) Attach NACL to SN1 Subnet.

2) Allow HTTP outbound rules to 192.168.60.0/24
   network from SN1 Subnet.
   a) Add Http(80) Outbound rule.
       i) Port No :80
  ii) Destination:192.168.60.0/24
   b) add ephemral ports in inbound rules
       PortRange :1025-65535
  Destination:192.168.60.0/24
c) Allow http request to SN3 from SN2
    i) SN3 Subnet NACL Configration
    a) add http inbound from SN2
      i) port no: 80
    ii) Source: 10.0.1.0/24
    b) add ephemral port range in outbound
        i) PortRange: 1025-65535
       ii) Destination :10.0.1.0/24
   ii) SN2 Subnet NACL Configration
     a) add http port range in outbound
     i) Port:80
   ii) Destination: 10.0.2.0/24
     b) add inbound ephemral port range
      i) PortRange:1025-65535
   ii) Source: 10.0.2.0/24

d) Deny All SSH request into SN3 Subnet from SN1
      a) Add SSH Deny Inbound rules in SN3 Subnet NACL

i) Port:22
ii) Source: 10.0.0.0/24
iii) Allow/Deny: deny