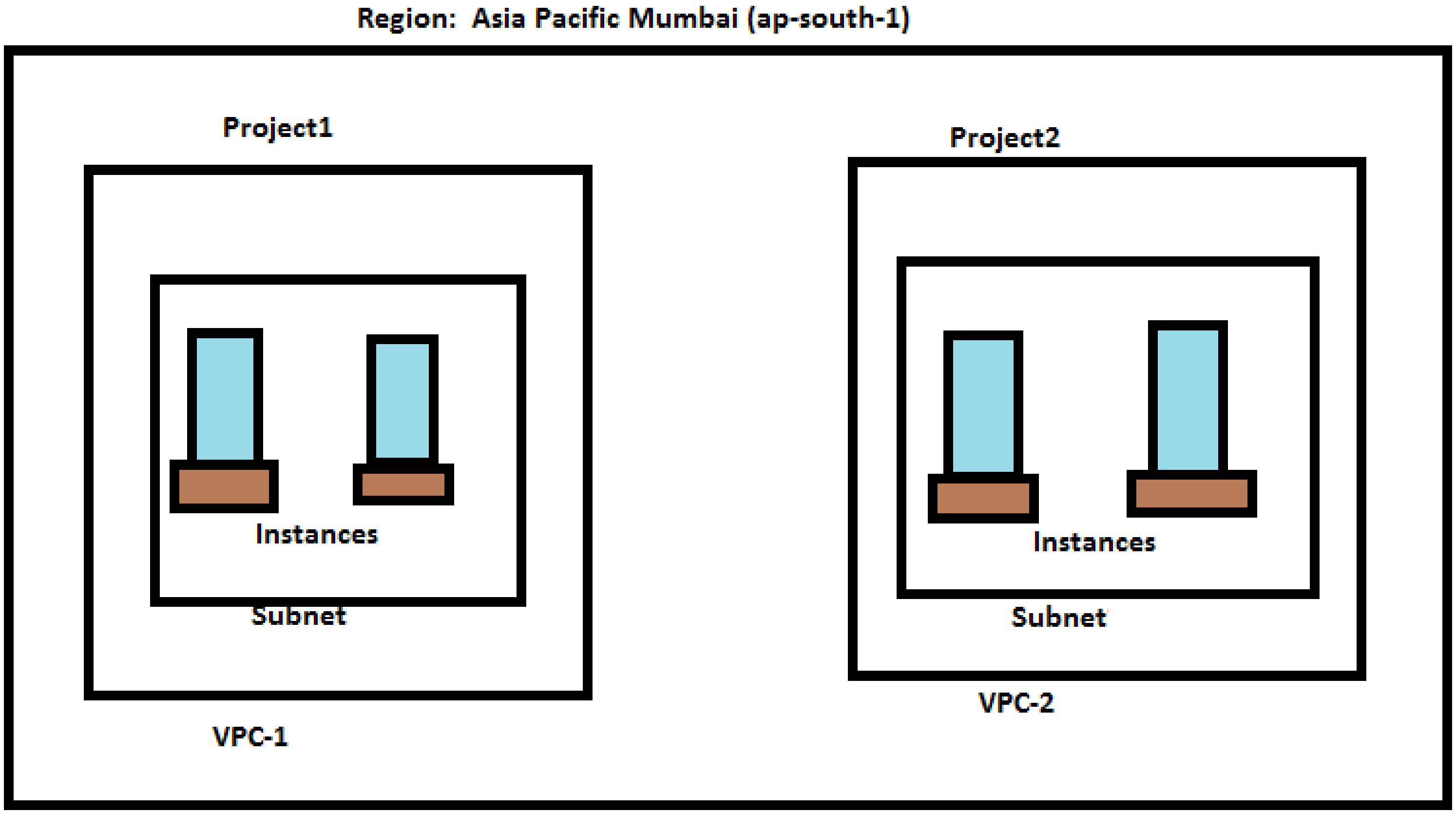
Amazon
VPC
amazon
web services


Region: Asia Pacific Mumbai (ap-south-1)
Project1
Instances
Subnet
VPC-1
Project2
Instances
Subnet
VPC-2

VPC Lab

1) Create 2 VPC with one subnet in Mumbai.

2) Launch one windows server in both VPC and try to connect internally each other using RDP and HTTP.

3) VPC peering same region.

4) VPC peering different region different account.

5) Transit gateway

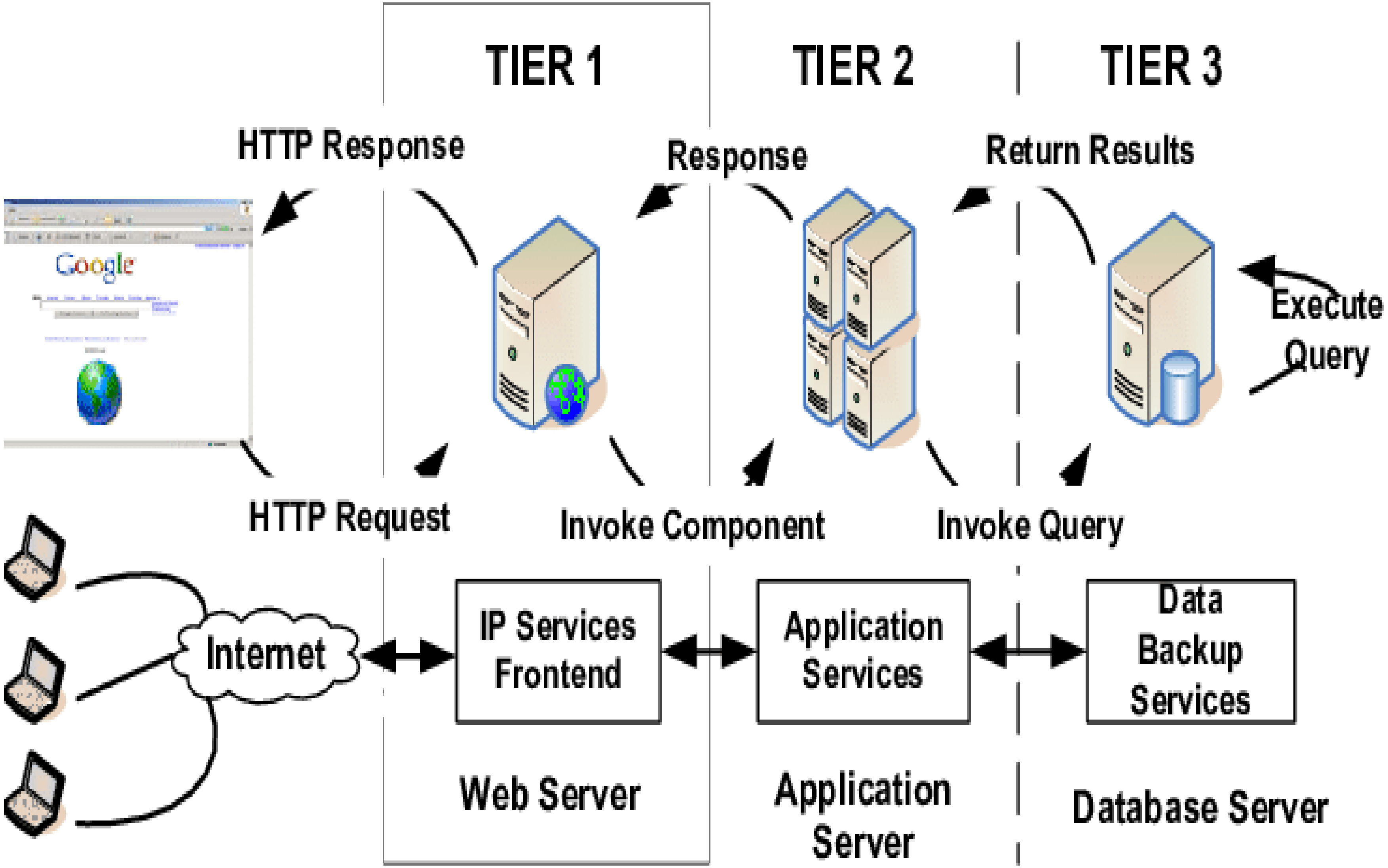
6) Create public and private Network.

7) Accessing DBserver using Webserver.

8) Provide outbound internet connection to private subnet(NAT gateway and NAT Instance)

9) VPC Endpoint

10) NACL


TIER 1
TIER 2
TIER 3
HTTP Response
Response
Return Results
Google
Execute Query
HTTP Request
Invoke Component
Invoke Query
Internet
IP Services Frontend
Application Services
Data Backup Services
Web Server
Application Server
Database Server

Subnet Classification

Public Subnet – Frontend network – internet facing Subnet

Private Subnet – Backend Network – No Internet facing Subnet

Default	Non-Default VPC
If your account supports the EC2-VPC platform only, it comes with a default VPC that has a default subnet in each Availability Zone.	You can create your own non-default VPC, and configure it as you need. Subnets that you create in your non-default VPC and additional subnets that you create in your default VPC are called non-default subnets.
Your default VPC includes an **internet gateway,** which allows your instances to communicate with the internet, and each default subnet is a **public subnet.**	Instances can communicate with each other, but can't access the internet. You can enable internet access for an instance launched into a non-default subnet by attaching an internet gateway and associating an **Elastic IP address** with the instance.
Each instance that you launch into a default subnet has a **private IPv4 address and a public IPv4 address.**	By default, each instance that you launch into a non-default subnet has a private IPv4 address, but no public IPv4 address, unless you specifically assign one at launch, or you modify the subnet's public IP address attribute.
To allow an instance in your VPC to initiate outbound connections to the internet but prevent unsolicited inbound connections from the internet, you can use a **network address translation** (NAT) device for IPv4 traffic.	To allow an instance in your VPC to initiate outbound connections to the internet but prevent unsolicited inbound connections from the internet, you can use a **network address translation** (NAT) device for IPv4 traffic.
You can optionally associate an **Amazon-provided IPv6 CIDR block** with your VPC and assign IPv6 addresses to your instances. IPv6 traffic is separate from IPv4 traffic; your **route tables** must include separate routes for IPv6 traffic.	You can optionally associate an **Amazon-provided IPv6 CIDR block** with your VPC and assign IPv6 addresses to your instances. IPv6 traffic is separate from IPv4 traffic; your **route tables** must include separate routes for IPv6 traffic.

Tutorials Dojo

VPC

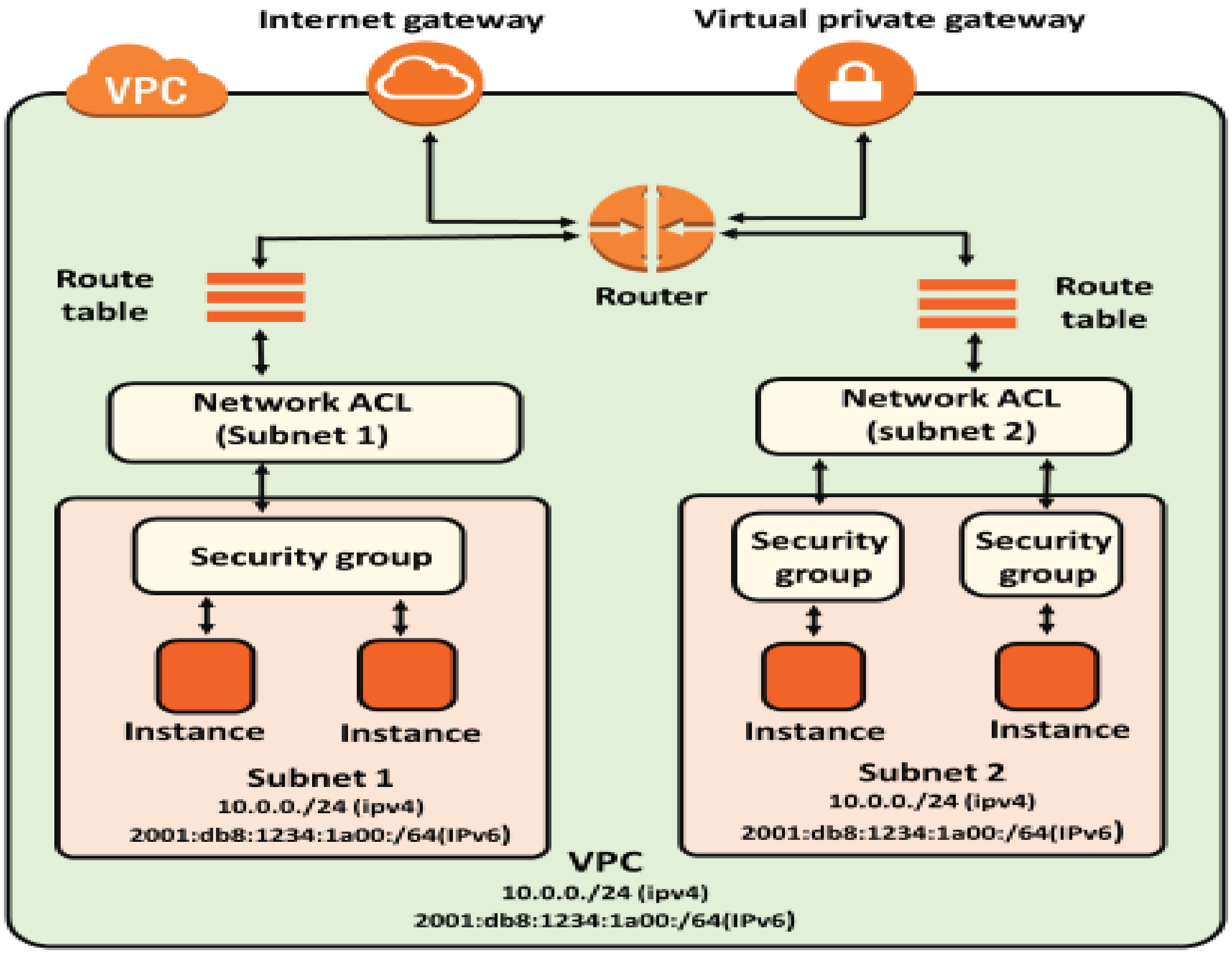
- Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define.

- You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways. You can use both IPv4 and IPv6 in your VPC for secure and easy access to resources and applications.

- You can easily customize the network configuration of your Amazon VPC. For example, you can create a public-facing subnet for your web servers that have access to the internet. You can also place your backend systems, such as databases or application servers, in a private-facing subnet with no internet access. You can use multiple layers of security, including security groups and network access control lists, to help control access to Amazon EC2 instances in each subnet.

Security Group and NACL

Firewall concept in VPC



NACL

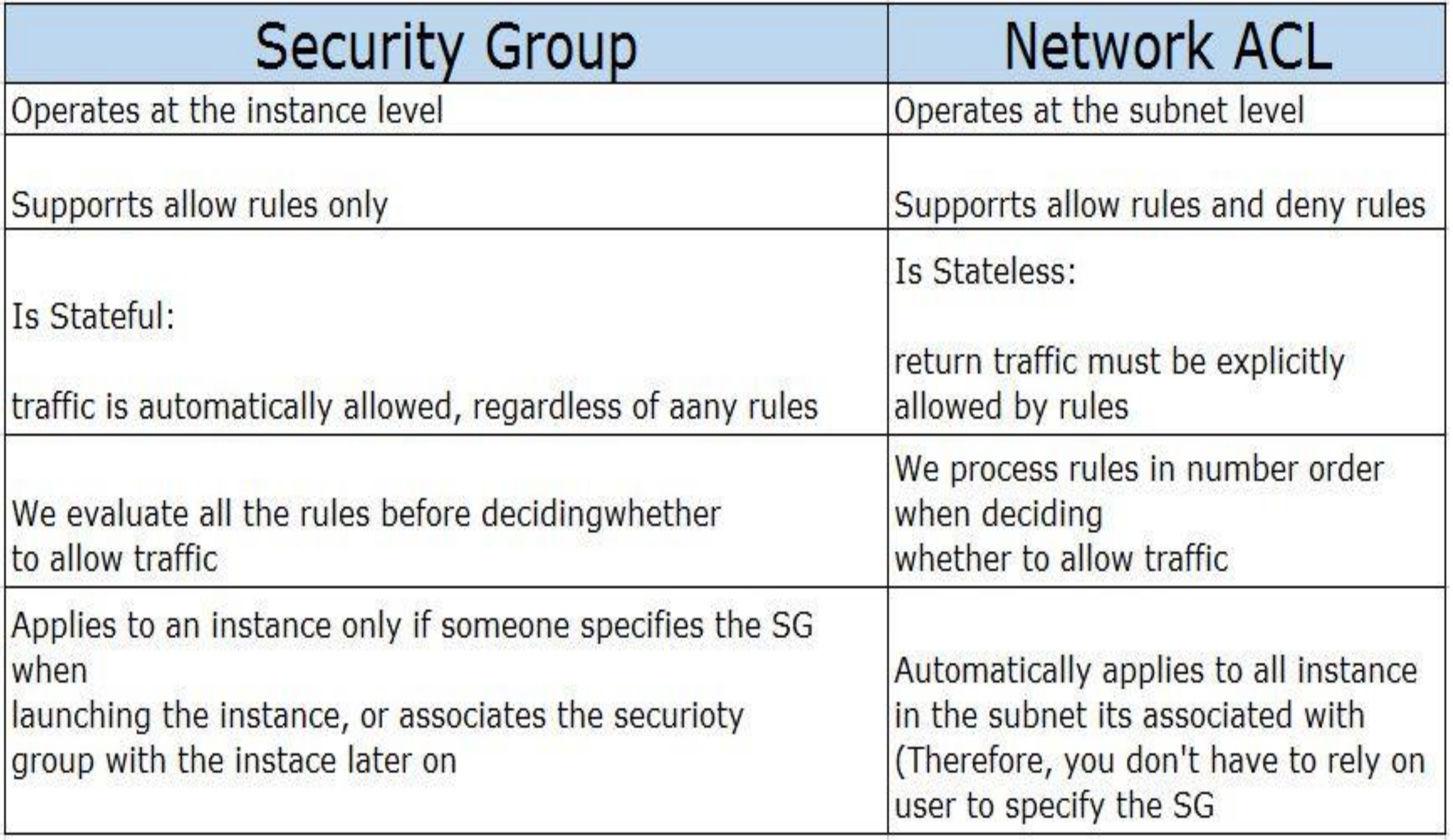
NACL also adds an additional layer of security associated with subnets that control both inbound and outbound traffic at the subnet level.

Maximum number of rules that exist per NACL: 20

Maximum number of rules that can exist per Security Group: 50

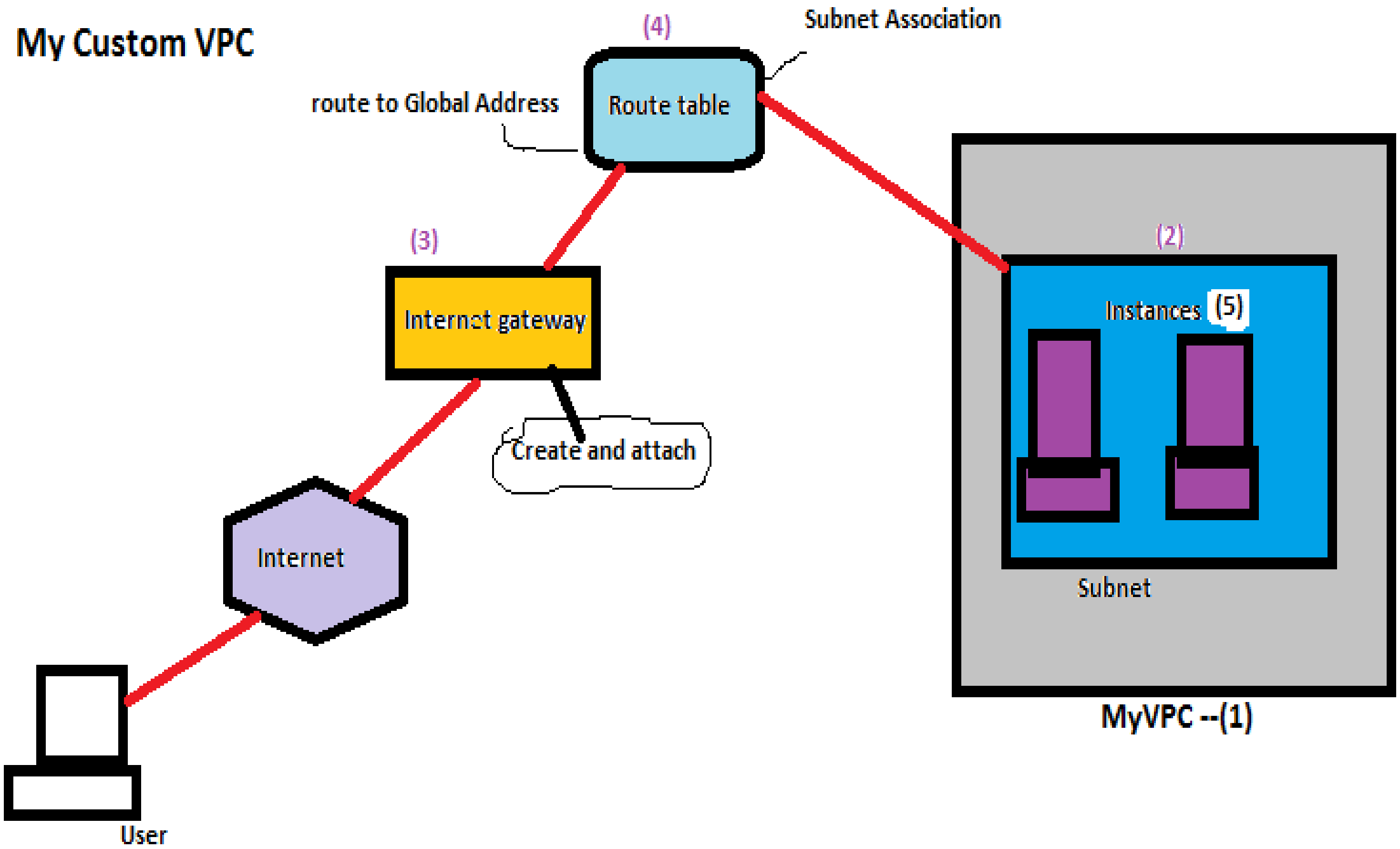
Maximum number of Security Groups that can exist per instance: 5

Maximum number of rules that can exist per instance: 5*50 + 20 = 270



Security Group	Network ACL
Operates at the instance level	Operates at the subnet level
Supporrts allow rules only	Supporrts allow rules and deny rules
Is Stateful: traffic is automatically allowed, regardless of aany rules	Is Stateless: return traffic must be explicitly allowed by rules
We evaluate all the rules before decidingwhether to allow traffic	We process rules in number order when deciding whether to allow traffic
Applies to an instance only if someone specifies the SG when launching the instance, or associates the securioty group with the instace later on	Automatically applies to all instance in the subnet its associated with (Therefore, you don't have to rely on user to specify the SG

My Custom VPC



(4)

Subnet Association

route to Global Address

Route table

(3)

(2)

Internet gateway

Instances (5)

Create and attach

Internet

Subnet

MyVPC --(1)

User

VPC Configuration Steps

Region	Mumbai	Singapore
VPC ID	10.100.0.0/16	10.200.0.0/16
Subnet 1 ID	10.100.1.0/24	10.200.1.0/24
Subnet 2 ID	10.100.2.0/24	10.200.2.0/24

In Mumbai

1) Open AWS Console –Services – VPC – Your VPC – Create VPC- Type name : project1-vpc – IP CIDR block -10.100.0.0/16 – Create VPC

2) Subnets – Create Subnets –Select VPC ID – subnet name: project1-subnet1 – Availiblity zone : ap-south-1a – IPV4 CIDR block:10.100.1.0/24 – Create Subnet

VPC Configuration Steps

3) Internet gateway – Create Internet gateway –Tag – project1-int-gtw -- Create Internet gateway

Then go to action –Attach to VPC – Available VPCs –select project1-vpc – Attach Internet gateway

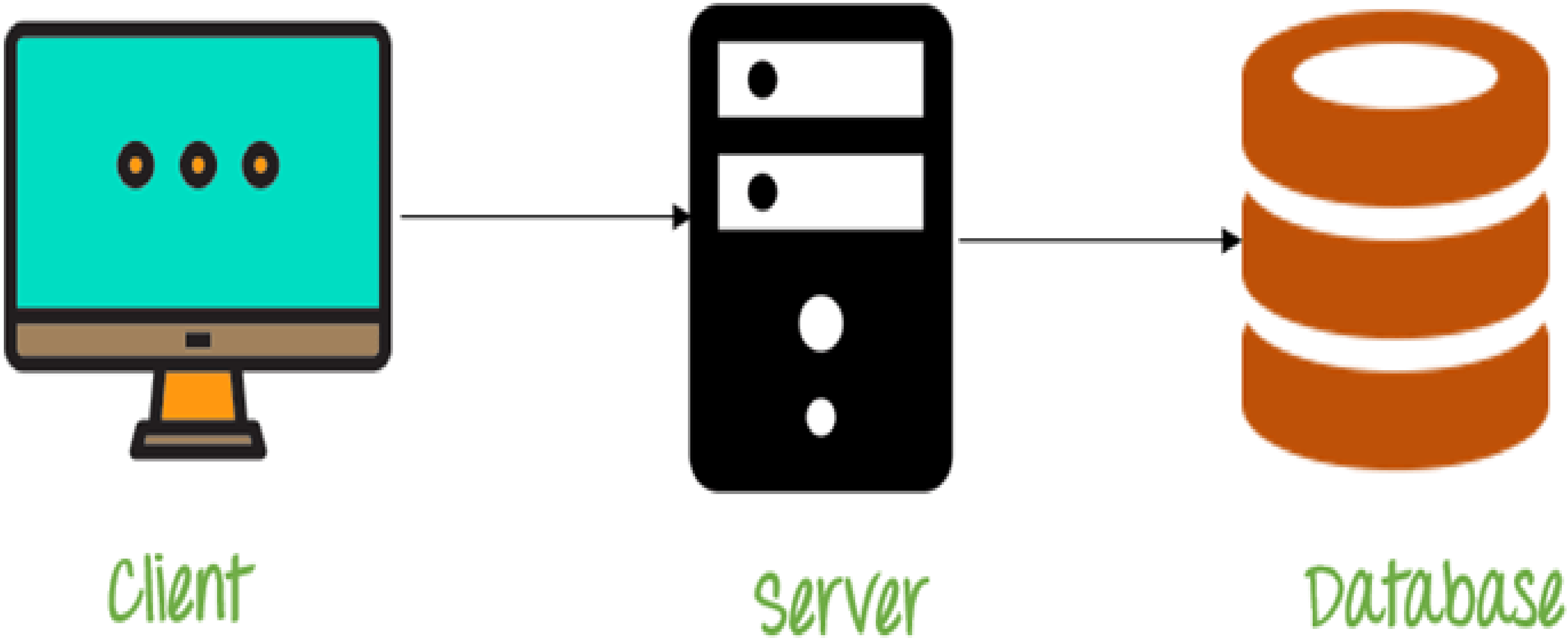
4) Route table –Create Route table – Name tag: Project1-RT1 – VPC - project1-vpc – Create

After creating select it – subnet association –edit –select project1-subnet1 ---save

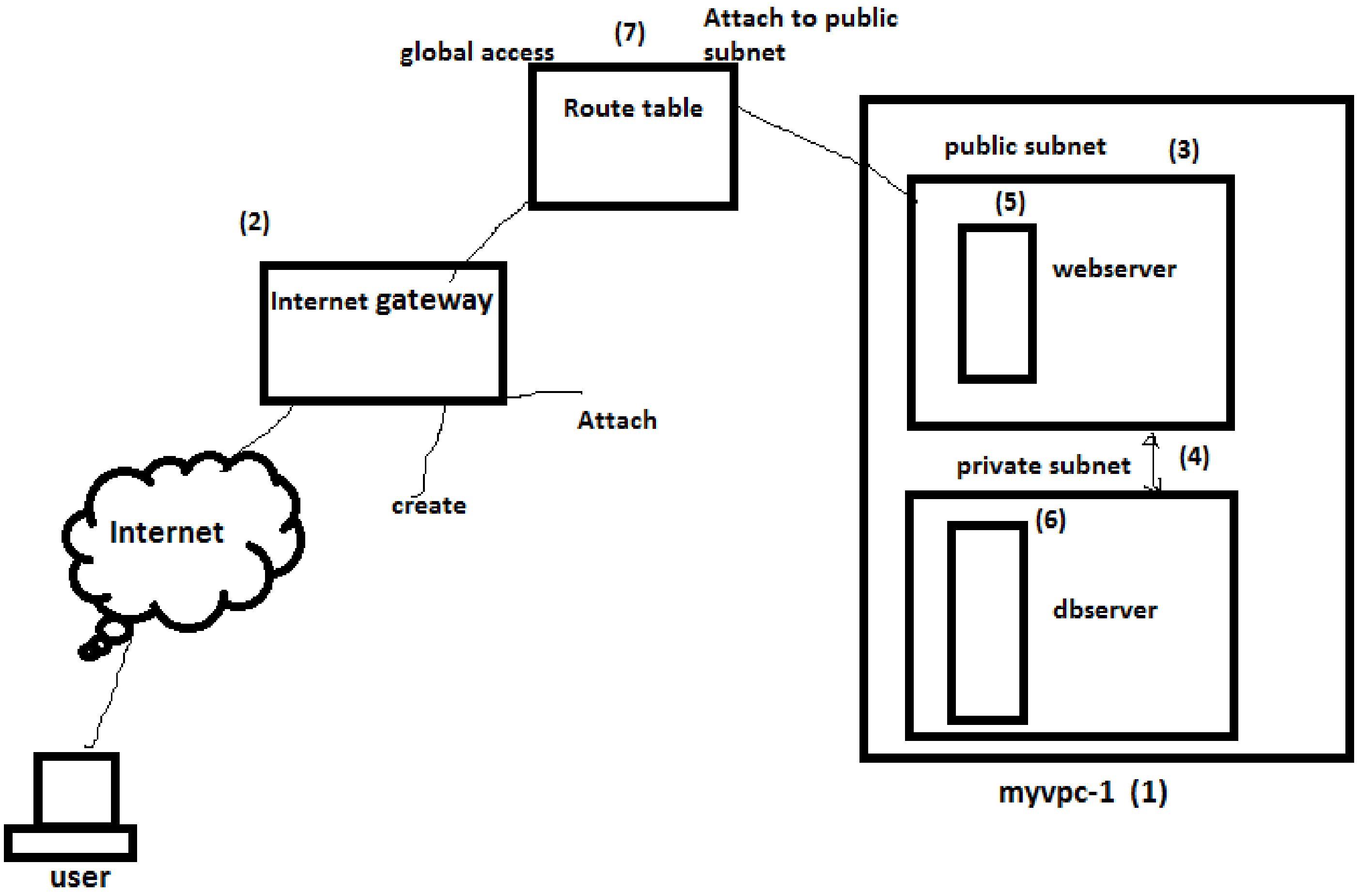
Go to Routes –Edit –Add route – 0.0.0.0/0 --- Target – Internet gateway - project1-int-gtw – save routes

5) Do the same VPC Setup in Singapore Region with different VPC ID

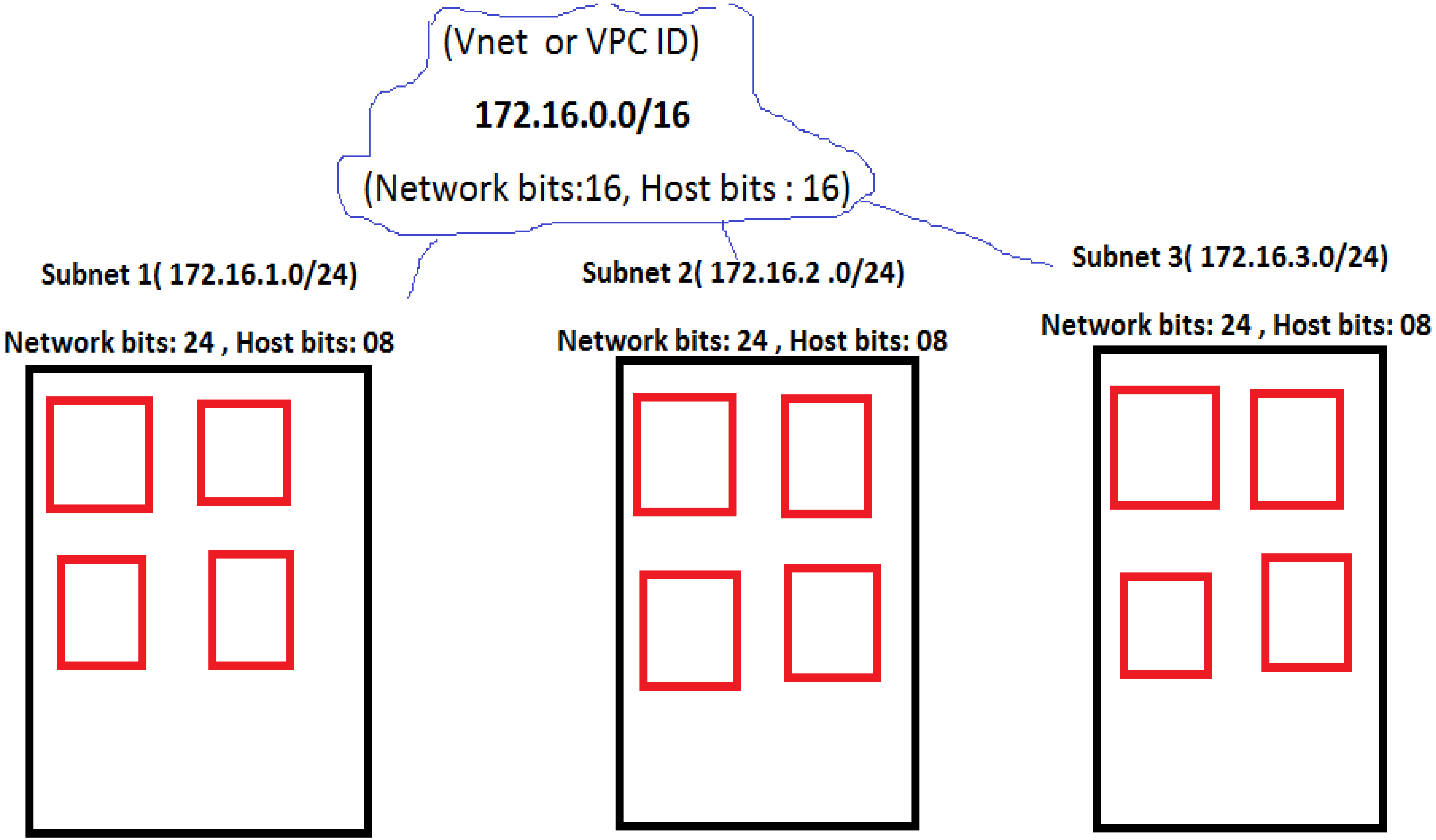
Three Tier Architecture



Hands on – Configure custom VPC with public and private network



Private IP address range		
Class	Starting	Ending
A	10.0.0.0	10.255.255.255
B	172.16.0.0	172.31.255.255
C	192.168.0.0	192.168.255.255



(Vnet or VPC ID)

172.16.0.0/16

(Network bits:16, Host bits : 16)

Subnet 1(172.16.1.0/24)

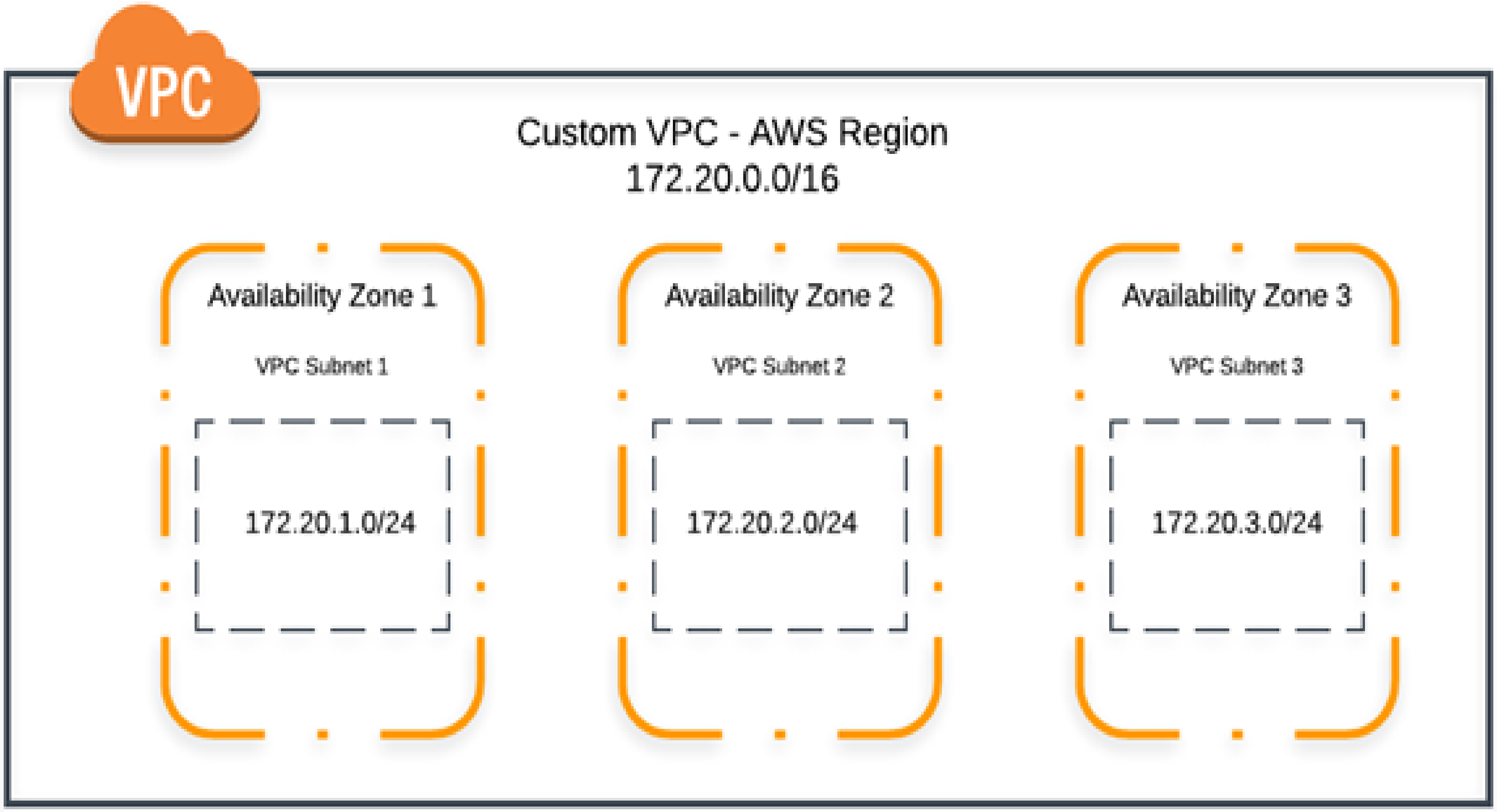
Network bits: 24 , Host bits: 08

Subnet 2(172.16.2 .0/24)

Network bits: 24 , Host bits: 08

Subnet 3(172.16.3.0/24)

Network bits: 24 , Host bits: 08


VPC
Custom VPC - AWS Region
172.20.0.0/16
Availability Zone 1
VPC Subnet 1
172.20.1.0/24
Availability Zone 2
VPC Subnet 2
172.20.2.0/24
Availability Zone 3
VPC Subnet 3
172.20.3.0/24

Reserved IP in VPC --Subnet

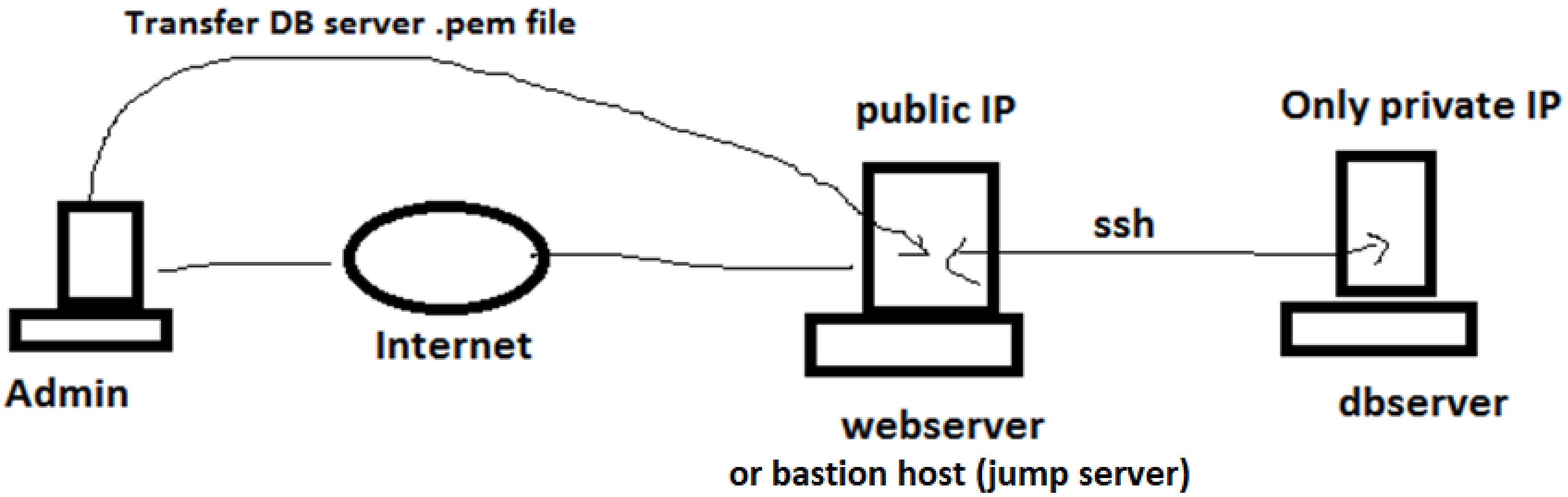
The first four IP addresses and the last IP address in each subnet CIDR block are not available for you to use, and cannot be assigned to an instance.

- 10.0.0.0: Network address.
- 10.0.0.1: Reserved by AWS for the VPC router.
- 10.0.0.2: Reserved by AWS for mapping to the Amazon-provided DNS. (Note that the IP address of the DNS server is the base of the VPC network range plus two. For more information, see Amazon DNS Server.)
- 10.0.0.3: Reserved by AWS for future use.
- 10.0.0.255: Network broadcast address. We do not support broadcast in a VPC, therefore we reserve this address.

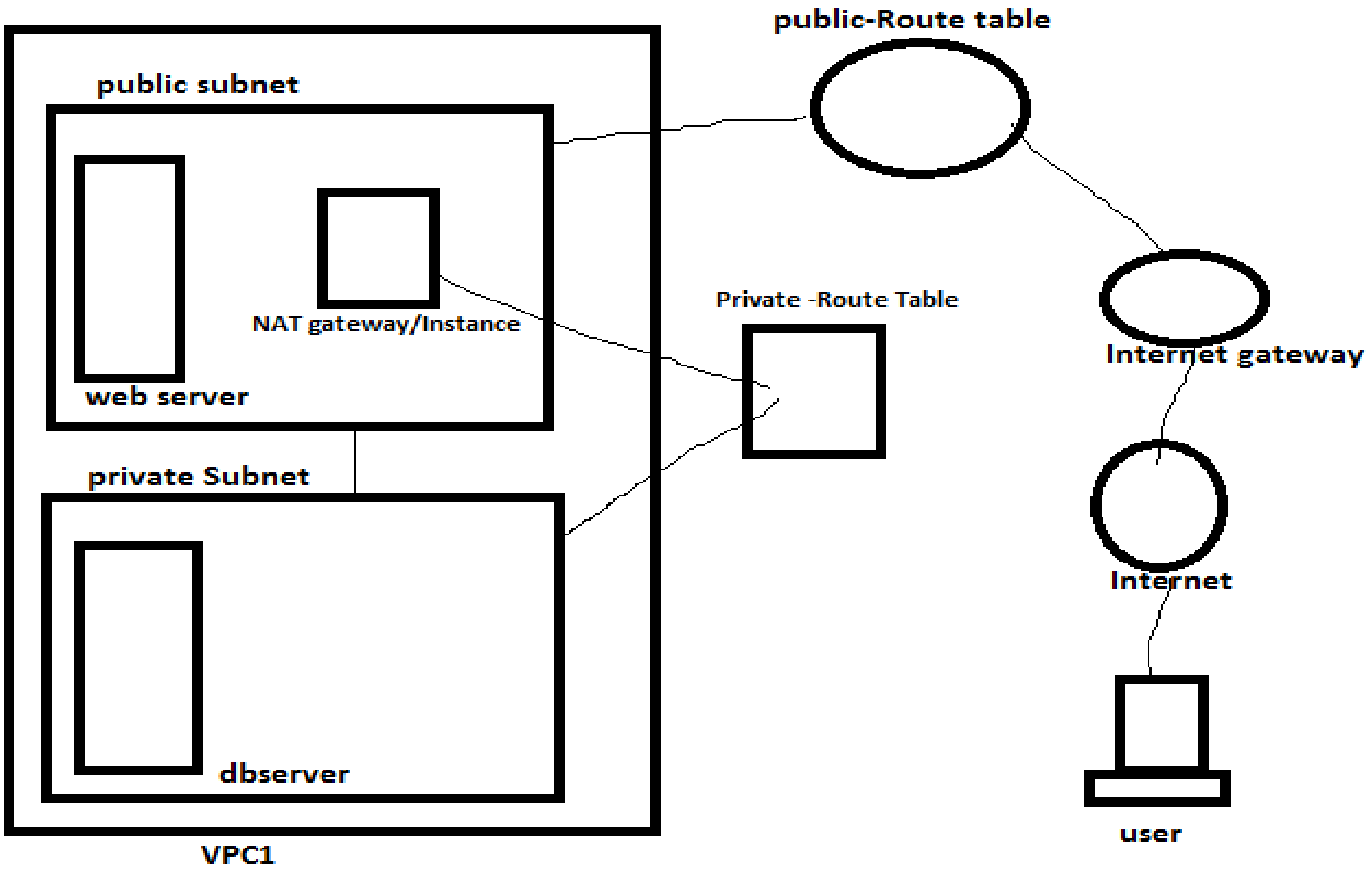
DB Server Security

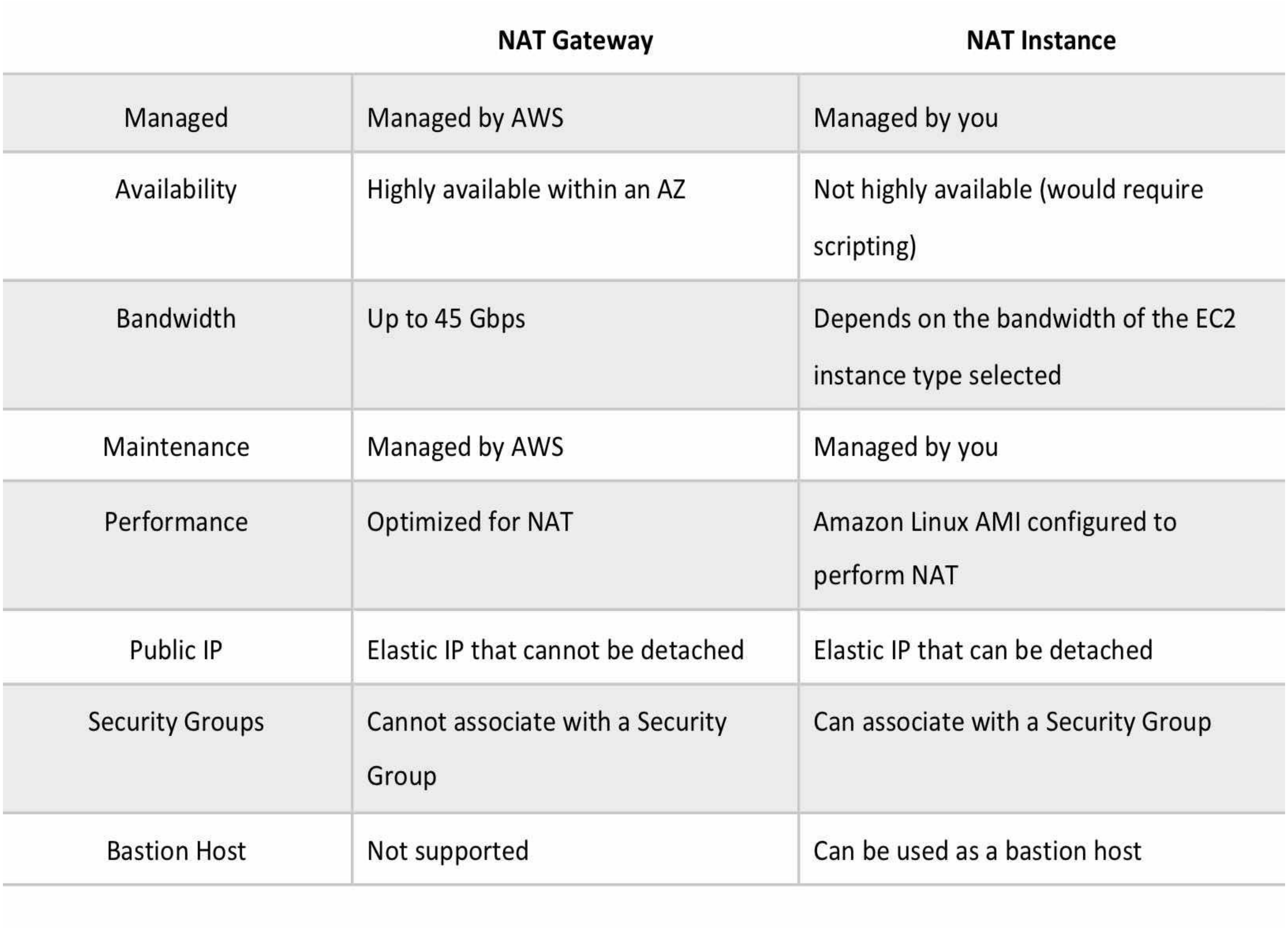
1) No public IP
2) Security group--SSH –mapped—webserver-SG
3) No route table configuration

Connecting Dbserver through Webserver



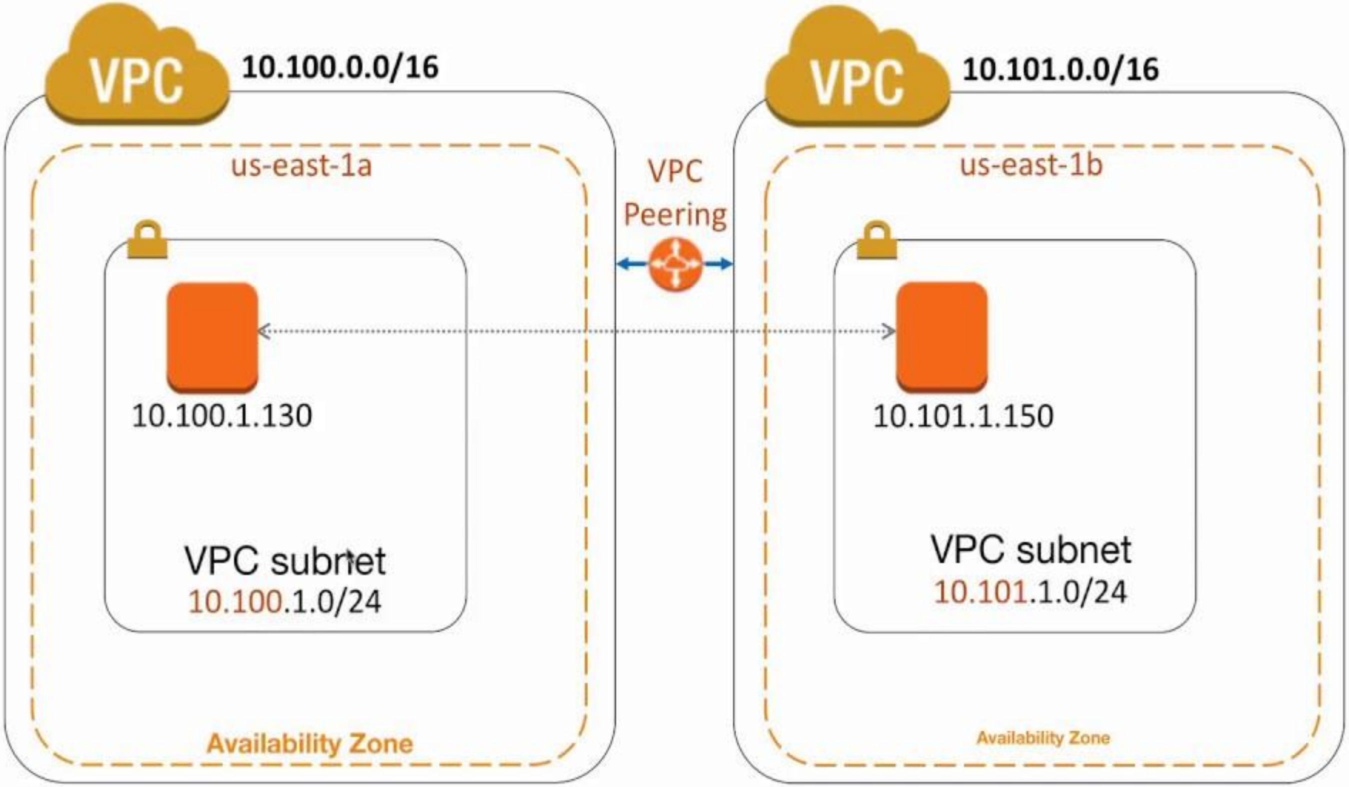
Provide Internet Connectivity (Outbound) to Private Subnet





	NAT Gateway	NAT Instance
Managed	Managed by AWS	Managed by you
Availability	Highly available within an AZ	Not highly available (would require scripting)
Bandwidth	Up to 45 Gbps	Depends on the bandwidth of the EC2 instance type selected
Maintenance	Managed by AWS	Managed by you
Performance	Optimized for NAT	Amazon Linux AMI configured to perform NAT
Public IP	Elastic IP that cannot be detached	Elastic IP that can be detached
Security Groups	Cannot associate with a Security Group	Can associate with a Security Group
Bastion Host	Not supported	Can be used as a bastion host

Set up VPC Peering b/w VPCs in the same Region



VPC 10.100.0.0/16

us-east-1a

10.100.1.130

VPC subnet
10.100.1.0/24

Availability Zone

VPC Peering

VPC 10.101.0.0/16

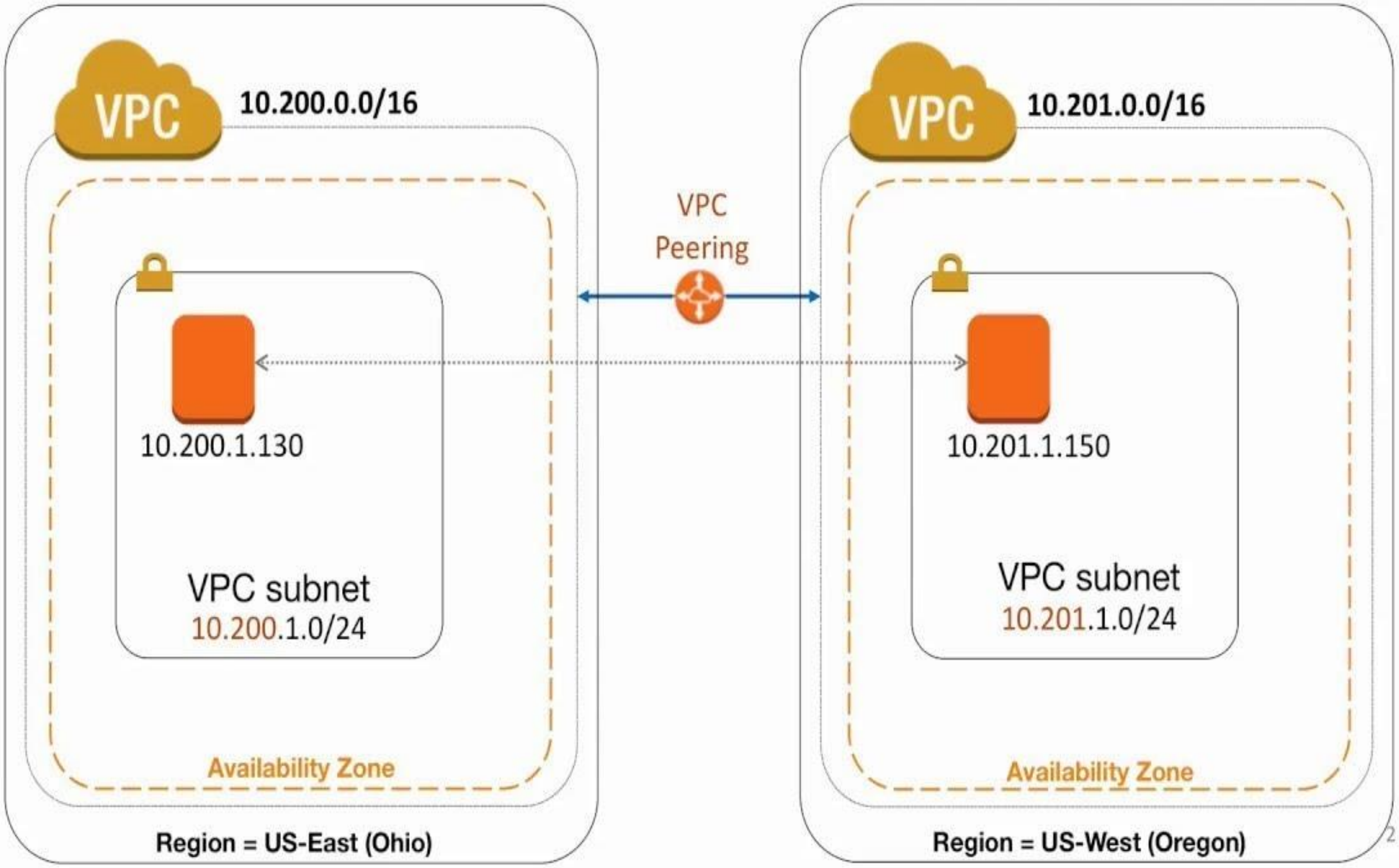
us-east-1b

10.101.1.150

VPC subnet
10.101.1.0/24

Availability Zone

Set up VPC Peering b/w VPCs in the different Regions



VPC peering

Term	VPC1	VPC2
VPC	Project1vpc	Project2vpc
VPC-ID	172.16.0.0./16	172.17.0.0/16
Subnet1	Project1publicsubnet (172.16.1.0/24)	Project2publicsubnet (172.17.1.0/24)
Subnet2	Project1privatesubnet (172.16.2.0/24)	Project2privatesubnet (172.17.2.0/24)
Route Table	Project1-public-rt	Project2-public-rt
Internet gateway	Internet gatewayproject1	Internet gatewayproject2

VPC peering Steps

1) Create 2 VPC with all detail –RT, IG, Subnet etc.

2) Peering Connection –New Peering –Fill the detail—name—vpc1-vpc2 –vpc , Requester –vpc1, Accepter –vpc2

--same account –same region—ok

3) Select created VPC peering –Action –Accept –ok

4) Route Table –select vpc1 route table—routes—edit routes—add route– vpc2 IP –target—peering connection---select: vpc1-vpc2 –ok

5) Route Table –select vpc2 route table—routes—edit routes—add route– vpc1 IP –target—peering connection---select: vpc1-vpc2 –ok