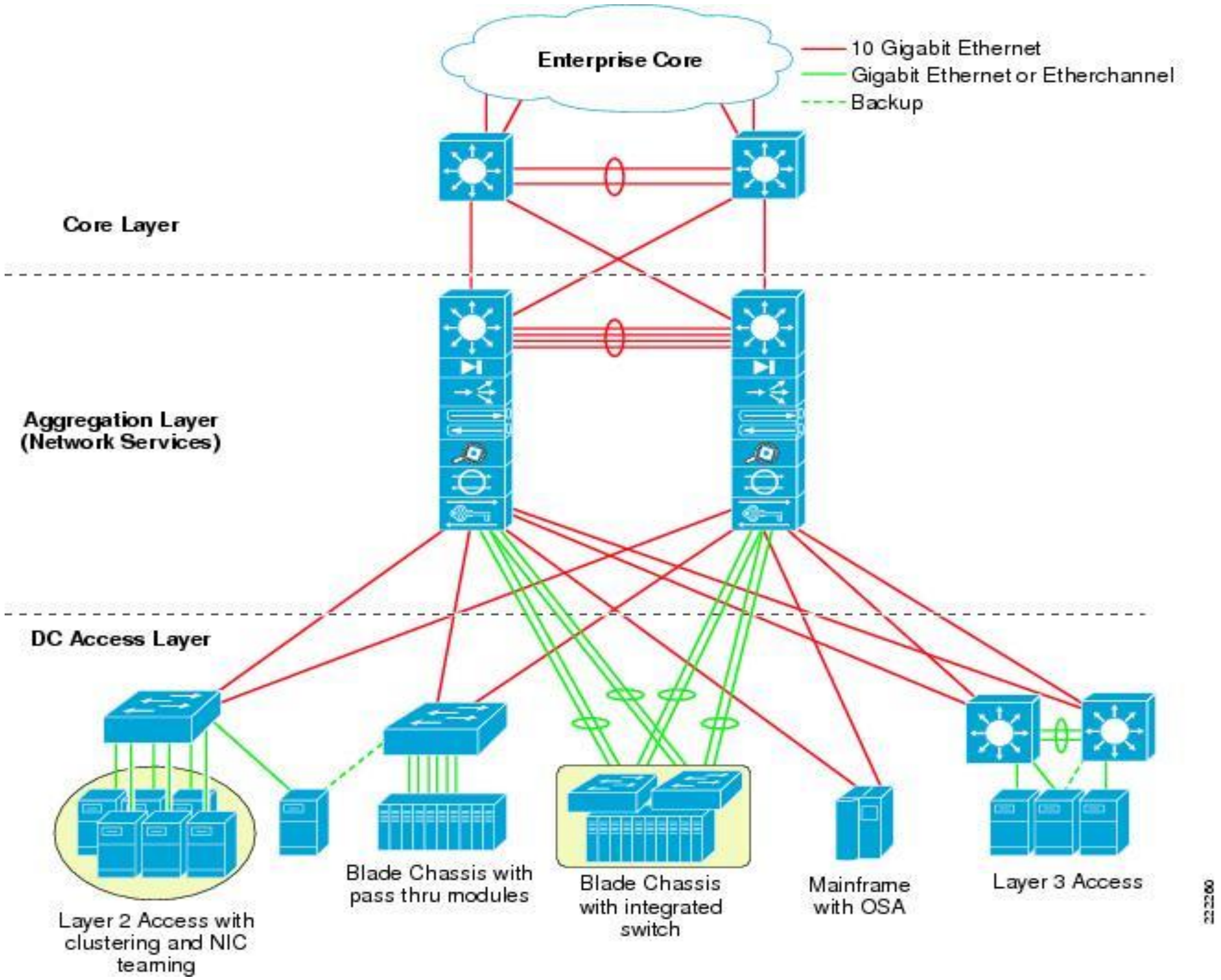
Module 1: Architecture and Design Networking in AWS

Mohanraj Shanmugam

AWS Networking

- AWS Networking will give a overview of how we network the AWS compute (EC2), Storage, Database and Other Services.
- All Amazon network are virtualized and It works same way as our Virtual network in our datacenter.
- Amazon Virtual Network is as scalable as other AWS Resources
- Lets understand how to setup a virtual Enterprise network in a Datacenter before starting the AWS Networking

Traditional Physical Datacenter Network



Enterprise Core
10 Gigabit Ethernet
Gigabit Ethernet or Etherchannel
Backup
Core Layer
Aggregation Layer (Network Services)
DC Access Layer
Layer 2 Access with clustering and NIC teaming
Blade Chassis with pass thru modules
Blade Chassis with integrated switch
Mainframe with OSA
Layer 3 Access

- All Servers are connected to Access layer switch
- Each server will have multiple NIC cards, Team or bond Two or more NIC Cards for High Availability and Link aggregation is configured at Access switch level to support Team or Bond.
- Each server requires multiple networks like Production, Backup , Clustering and Management Networks.
- Each network connected to different NIC cards or Bonds and wired separately
- Each network is Segmented by VLAN so that it can communicate within the segmented server
- All access layer switches are connected to Aggregation layer switch where all segmentation and routing takes place across networks
- All Network Functions like Load balancer, Firewall, VPN, Proxy and Intrusion prevention system is connected to aggregation layer
- All external networks like Internet and leased line and WAN network Terminate at Aggregation Layer
- Core or Metro layer connects between Data centers and Provide Layer 3 and Layer 2 high speed routing between Datacenters

Virtualize Enterprise Network

- The network team is being bombarded with configuration requests that can take days or weeks to handle, There are emerging Network Technologies which will automate and move towards the Programmable Network to provide as a service
- The Three Main Categories of Virtual Enterprise Network are:
  - Network Virtualization
  - Network Function Virtualization
  - Software Defined Networking
- Lets understand each of this in detail

Network Virtualization

- Network virtualization is the process of combining hardware network resources and software network resources into a single administrative unit.
- The goal of network virtualization is to provide systems and users with efficient, controlled, and secure sharing of the networking resources.
- There are two types of Network Virtualization
  - Internal virtualization
  - External virtualization

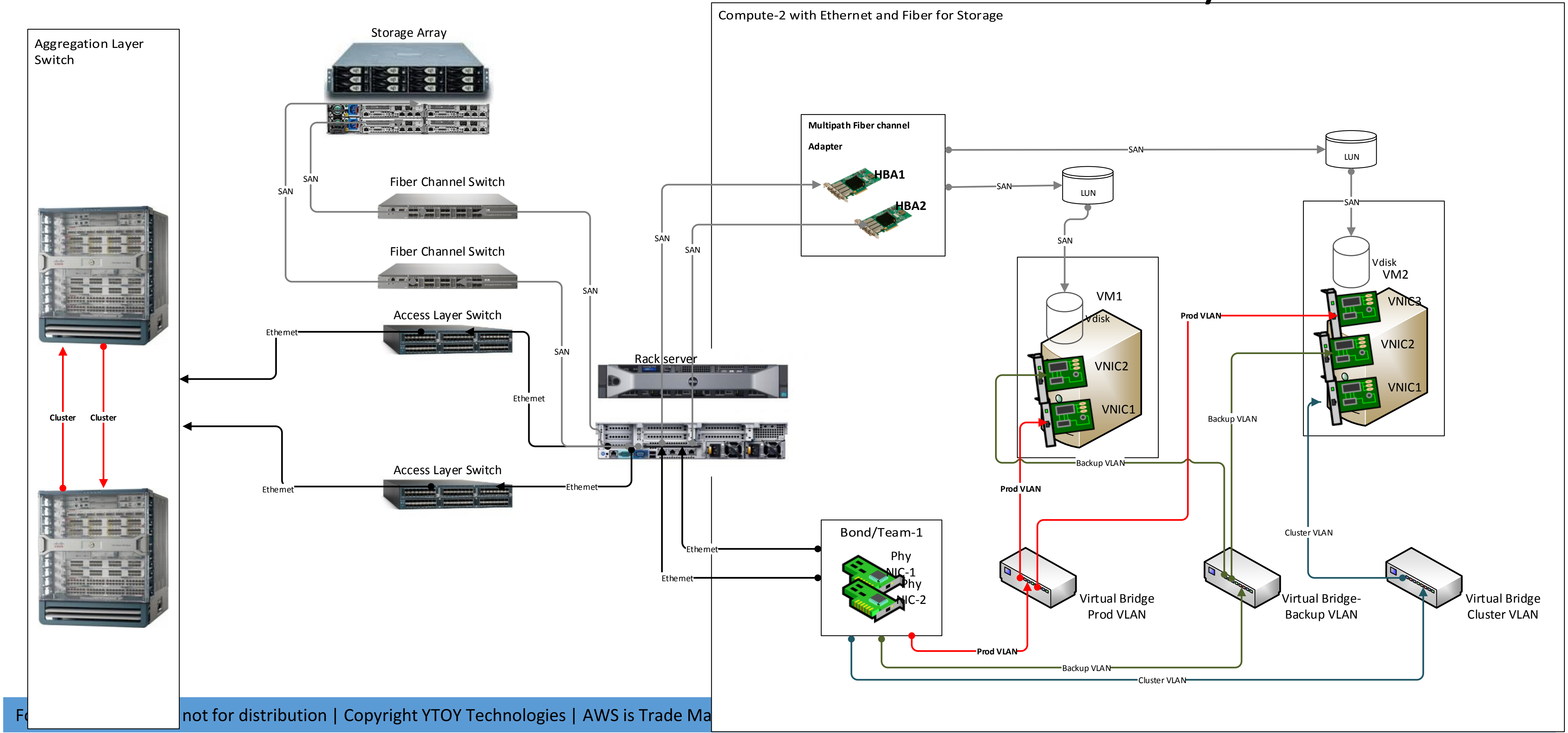
External virtualization

- External network virtualization combines or subdivides one or more local area networks (LANs) into virtual networks to improve a large network's or data center's efficiency.
- External virtual networks are administered by software as a single entity.
- Examples of external virtual networks include large corporate networks and data centers.
- Components of External Virtualization
    - Access Switches
    - Fiber Switches
    - Convergent Switches
    - Aggregate switches
    - VLAN
    - Physical Server Adapters

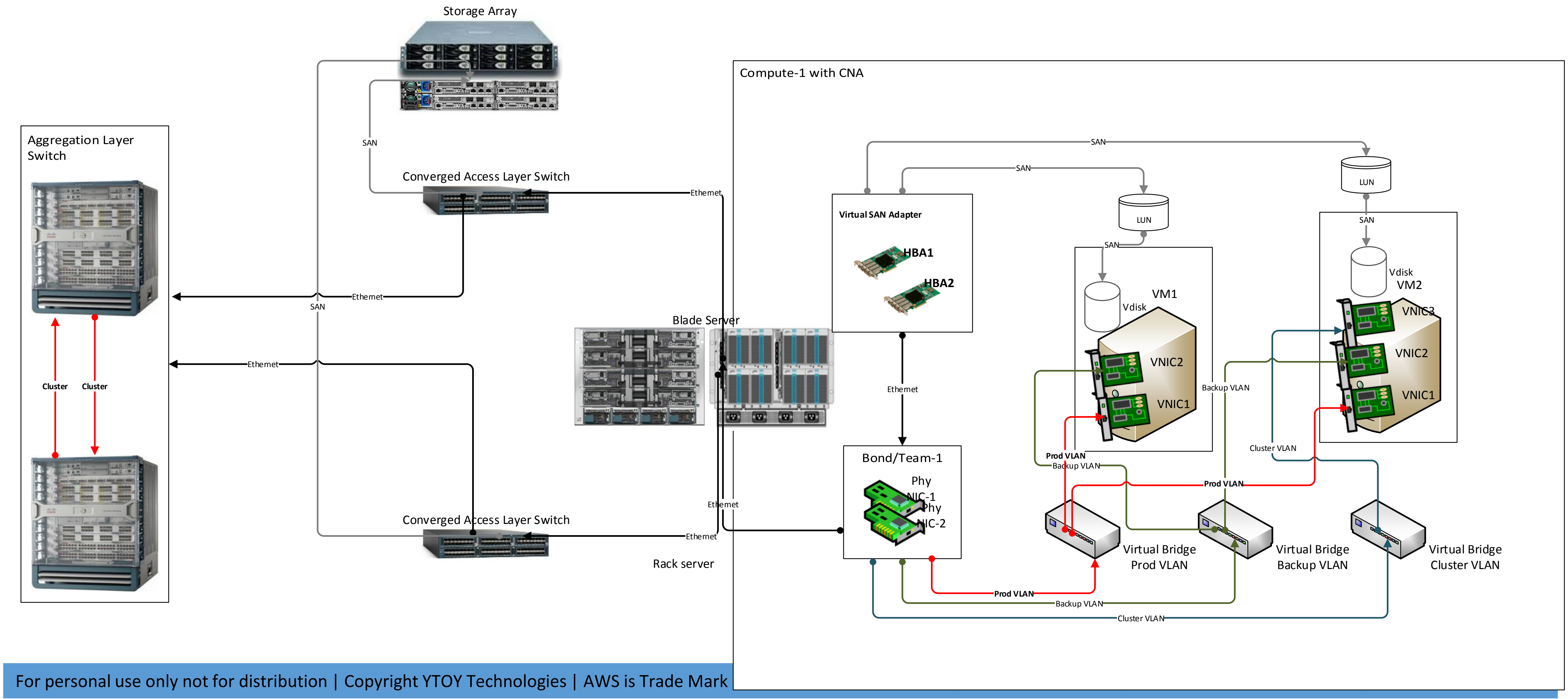
Internal Virtualization

- Internal Network Virtualization provides network functionality purely based on software.
- An example of is the network topology used by common virtualization produces such as KVM or VMWARE ESX.
- In these you use existing network in your environment and present it to the virtual machines using a simple bridged or NAT based networking.
- Components of Internal Networking
  - Vitual NIC
  - Virtual SAN Adapter
  - Virtual Bridge or Switch
  - Physical NIC

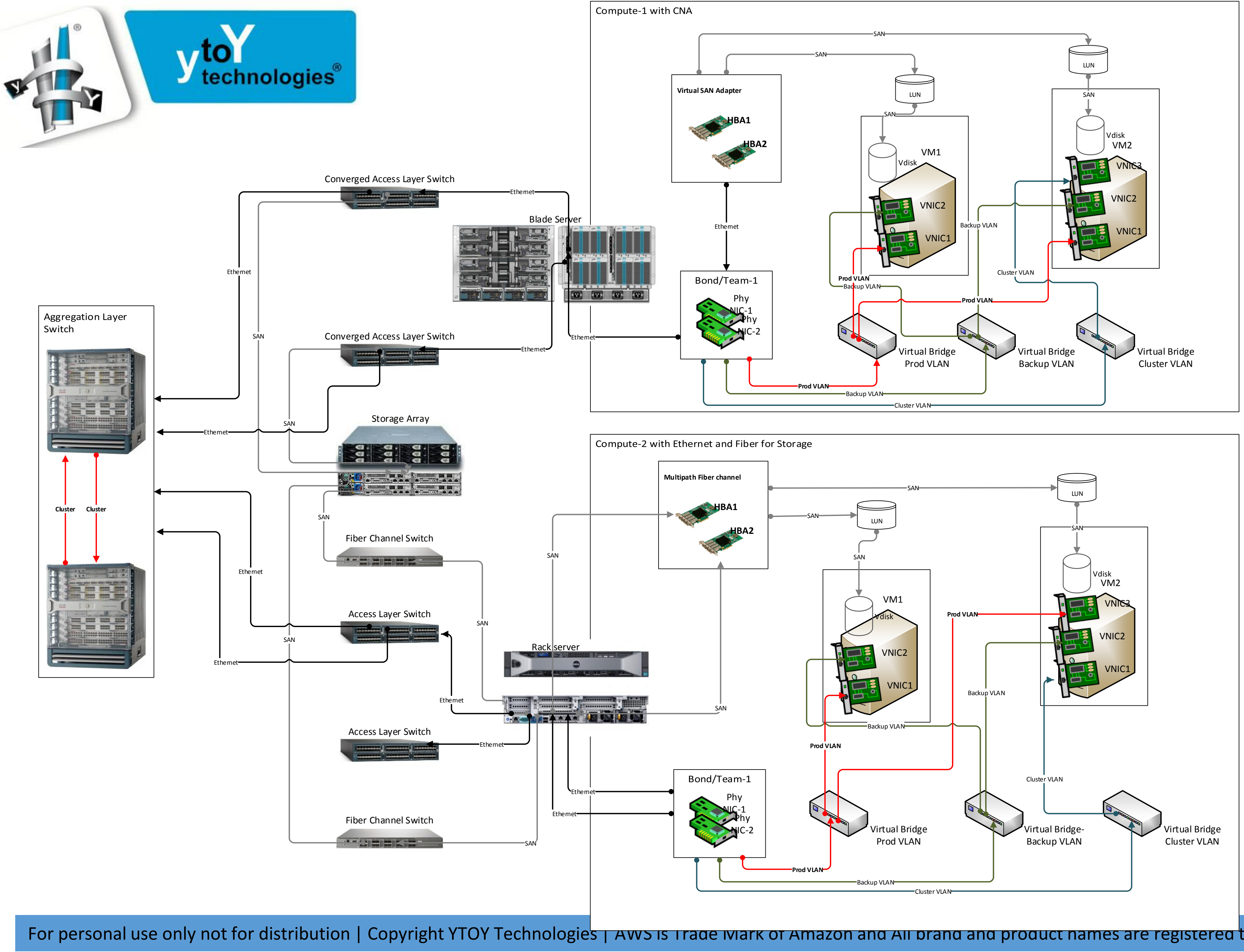
Network Virtualization – Traditional Physical Server



Network Virtualization- Blade Server and CNA



Network Virtualization



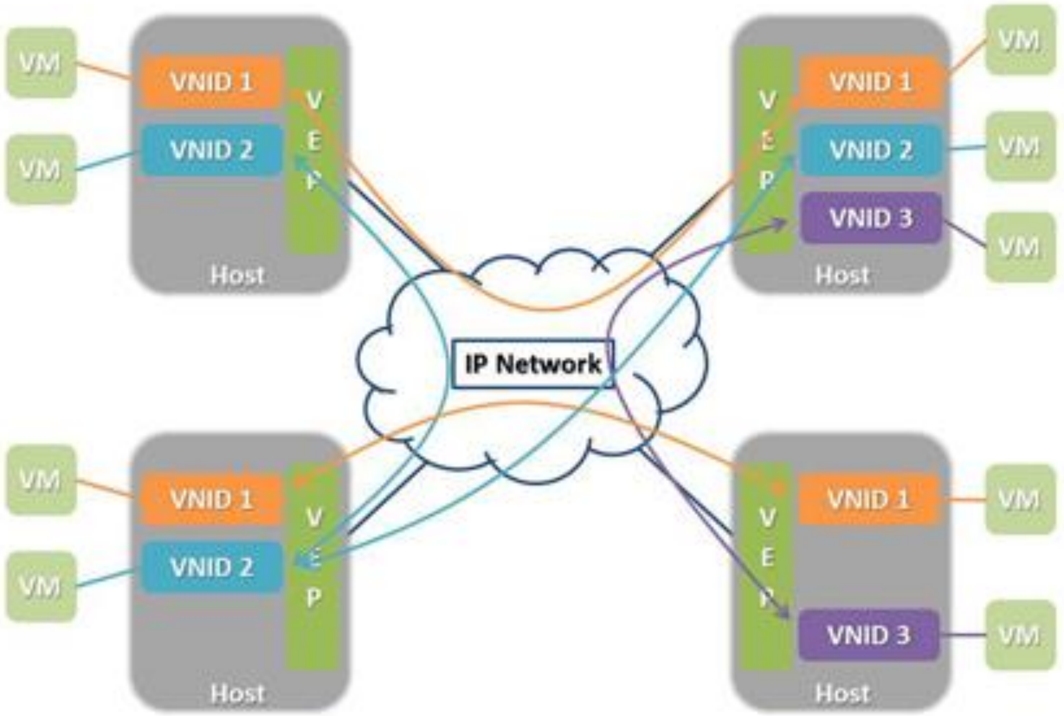
Network Overlays

- The 802.1q standard defines the VLAN tag as a 12-bit space, providing for a max of 4,096 VLANs .
- This is an easily reachable ceiling in multitenant environments where multiple internal or external customers will request multiple subnets.
- A physical server now has multiple Virtual Machines (VMs) each with its own Media Access Control (MAC) address. This requires larger MAC address tables in the switched Ethernet network due to potential attachment of and communication among hundreds of thousands of VMs.
- To Overcome that we use Overlay, overlay is used to carry the MAC traffic from the individual VMs in an encapsulated format over a logical "tunnel".
- The popular Standards of Overlay are:

  Virtual Extensible LAN (VXLAN),

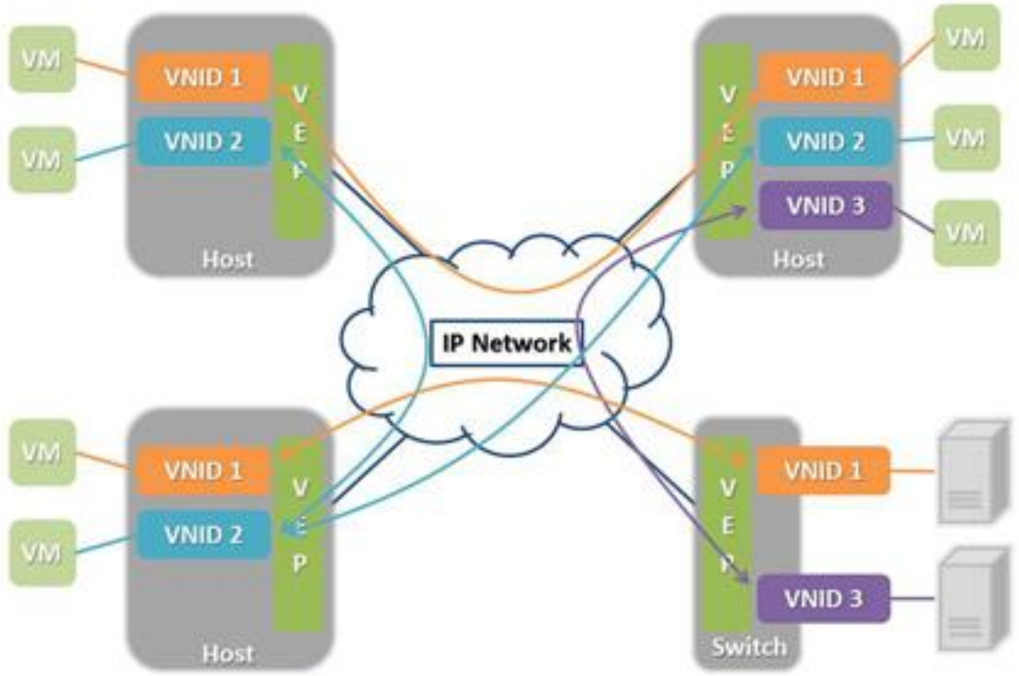
  Network Virtualization using Generic Routing Encapsulation(NVGRE),

Network Overlays



Endpoints are assigned to a virtual network via a Virtual Network ID (VNID).
These endpoints will belong to that virtual network regardless of their location on the underlying physical IP network.
In diagram 1 there are four virtual hosts connected via an IP network.
Each host contains a Virtual End Point (VEP), which is a virtual switch capable of acting as the encapsulation/de-encapsulation point for the virtual networks (VNIDs.)
Each host has two or more VNIDs operating on it and each workload assigned to a given VNID can communicate with other workloads in the same VNID, while maintaining separation from workloads in other VNIDs on the same or other hosts.

Network Overlays



- Depending on the chosen encapsulation and configuration method, hosts that do not contain a given VNID will either never see packets destined for that VNID, or will see them and drop them at ingress. This ensures the separation of tenant traffic.
- The same concept would apply if using a physical switch with the VEP functionality. This would allow physical devices to be connected to the overlay network
- With a physical switch capable of acting as the tunnel end-point, you can add both physical servers and appliances (firewalls, load balancers, and so on) to the overlay.
- This model is key to a cohesive deployment in mixed workload environments common in today's data centers.

Virtual Extensible LAN (VXLAN),

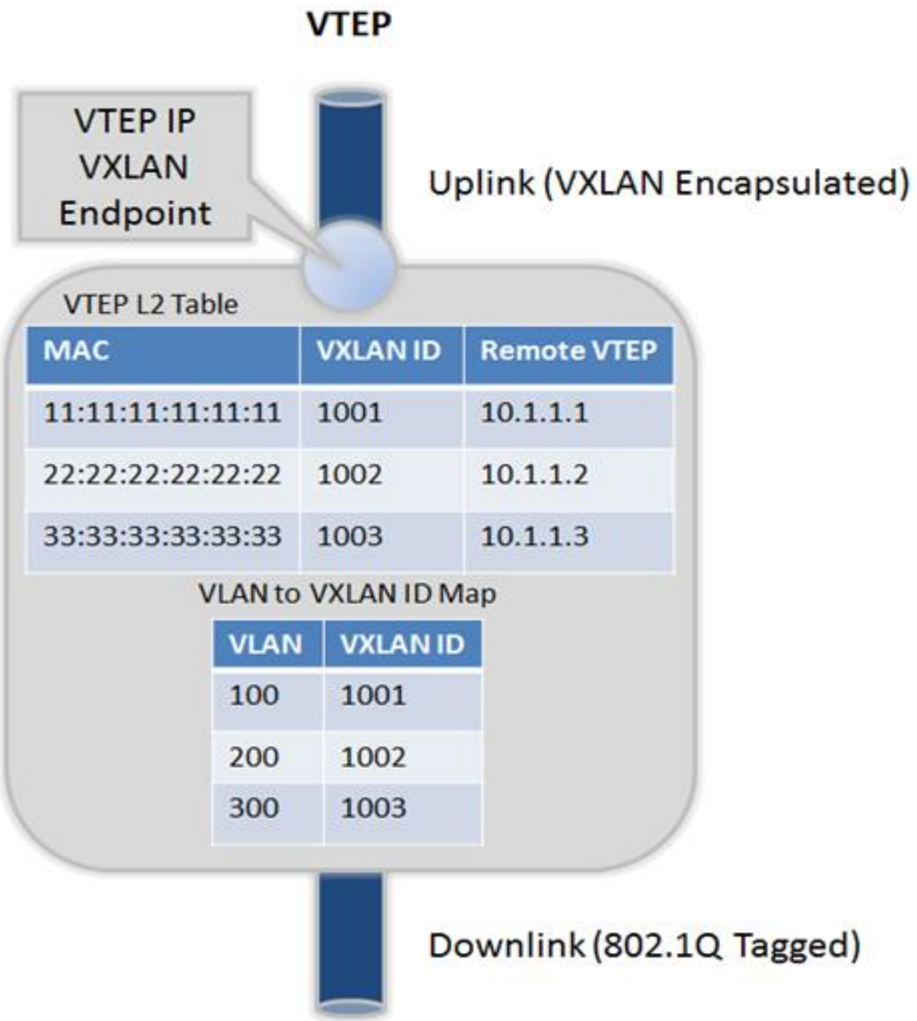
- **Virtual Extensible LAN** (**VXLAN**) is majorly backed up by Cisco and Vmware
- The standard for VXLAN is under the scope of the IETF NVO3 working group.
- VXLAN’s goal is allowing dynamic large scale isolated virtual L2 networks to be created for virtualized and multi-tenant environments.
- It does this by encapsulating frames in VXLAN packets.
- VXLAN utilizes a 24-bit VXLAN header, shown in the diagram, to identify virtual networks.
- This header provides for up to 16 million virtual L2 networks.
- Frame encapsulation is done by an entity known as a VXLAN Tunnel Endpoint (VTEP.)

Outer MAC SA	Outer MAC DA	Outer 802.1Q	Outer IP DA	Outer IP SA	Outer UDP	VXLAN ID(24 Bit)	Inner MAC DA	Inner MAC SA	Optional Inner 802.1q	Original Ethernet Payload

Virtual Extensible LAN (VXLAN),

- A VTEP has two logical interfaces: an uplink and a downlink.
- The uplink is responsible for receiving VXLAN frames and acts as a tunnel endpoint with an IP address used for routing VXLAN encapsulated frames.
- These IP addresses are infrastructure addresses and are separate from the tenant IP addressing for the nodes using the VXLAN fabric.
- VTEP functionality can be implemented in software such as a virtual switch or in the form a physical switch.
- VXLAN frames are sent to the IP address assigned to the destination VTEP; this IP is placed in the Outer IP DA.
- The IP of the VTEP sending the frame resides in the Outer IP SA.
- Packets received on the uplink are mapped from the VXLAN ID to a VLAN and the Ethernet frame payload is sent as an 802.1Q Ethernet frame on the downlink.
- During this process the inner MAC SA and VXLAN ID is learned in a local table. Packets received on the downlink are mapped to a VXLAN ID using the VLAN of the frame.
- A lookup is then performed within the VTEP L2 table using the VXLAN ID and destination MAC; this lookup provides the IP address of the destination VTEP. The frame is then encapsulated and sent out the uplink interface.

Virtual Extensible LAN (VXLAN),



VTEP L2 Table

MAC	VXLAN ID	Remote VTEP
11:11:11:11:11:11	1001	10.1.1.1
22:22:22:22:22:22	1002	10.1.1.2
33:33:33:33:33:33	1003	10.1.1.3

VLAN to VXLAN ID Map

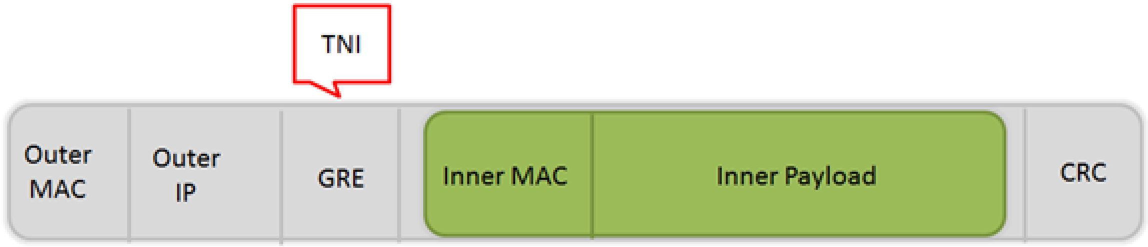
VLAN	VXLAN ID
100	1001
200	1002
300	1003

- Using the diagram above for reference a frame entering the downlink on VLAN 100 with a destination MAC of 11:11:11:11:11:11 will be encapsulated in a VXLAN packet with an outer destination address of 10.1.1.1. The outer source address will be the IP of this VTEP (not shown) and the VXLAN ID will be 1001.
- In a traditional L2 switch a behavior known as flood and learn is used for unknown destinations (i.e. a MAC not stored in the MAC table. This means that if there is a miss when looking up the MAC the frame is flooded out all ports except the one on which it was received. When a response is sent the MAC is then learned and written to the table. The next frame for the same MAC will not incur a miss because the table will reflect the port it exists on. VXLAN preserves this behavior over an IP network using IP multicast groups.
- Each VXLAN ID has an assigned IP multicast group to use for traffic flooding (the same multicast group can be shared across VXLAN IDs.) When a frame is received on the downlink bound for an unknown destination it is encapsulated using the IP of the assigned multicast group as the Outer DA; it's then sent out the uplink. Any VTEP with nodes on that VXLAN ID will have joined the multicast group and therefore receive the frame. This maintains the traditional Ethernet flood and learn behavior.
- VTEPs are designed to be implemented as a logical device on an L2 switch. The L2 switch connects to the VTEP via a logical 802.1Q VLAN trunk. This trunk contains an VXLAN infrastructure VLAN in addition to the production VLANs. The infrastructure VLAN is used to carry VXLAN encapsulated traffic to the VXLAN fabric. The only member interfaces of this VLAN will be VTEP's logical connection to the bridge itself and the uplink to the VXLAN fabric. This interface is the 'uplink' described above, while the logical 802.1Q trunk is the downlink.

Network Virtualization using Generic Routing Encapsulation(NVGRE)

- NVGRE uses Generic Routing Encapsulation (GRE) to tunnel layer 2 packets over layer 3 networks.
- Its principal backer is Microsoft. Other companies supporting the development of NVGRE include Chelsio Communications, F5 Networks, AristaNetworks, Mellanox, Broadcom, Dell, Emulex, Intel a nd Hewlett Packard.
- It uses the lower 24 bits of the GRE header to represent the Tenant Network Identifier (TNI.)   Like VXLAN this 24 bit space allows for 16 million virtual networks.

NVGRE Encapsulation



www.definethecloud.com

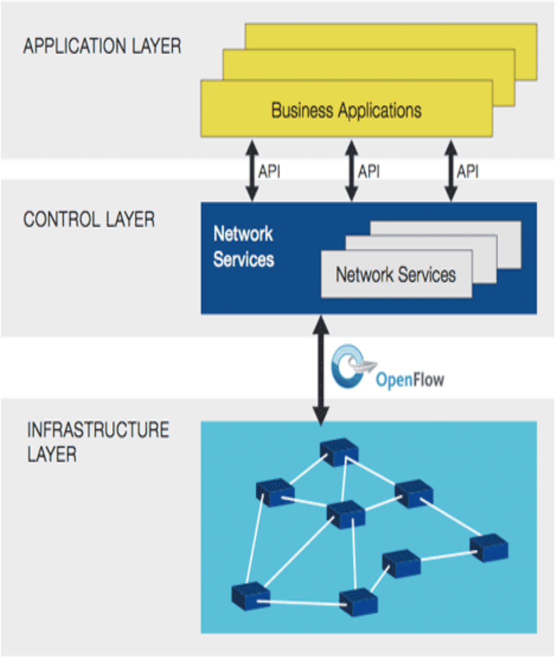
VXLAN Vs NVGRE

- NVGRE provides optional support for broadcast via IP multi-cast
- It supports multi-pathing capabilities.
- It Supports Jumbo frames

Software Defined Network (SDN)

- Software-Defined Networking (SDN) is an emerging architecture that is dynamic, manageable, cost-effective, and adaptable, making it ideal for the high-bandwidth, dynamic nature of today's applications.
- This architecture decouples the network control (Control Pane) and forwarding functions (Data Pane)
- IT enables the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services.
- The OpenFlow® protocol is a foundational element for building SDN solutions.

Software Defined Network (SDN)



The SDN architecture is:

Directly programmable: Network control is directly programmable because it is decoupled from forwarding functions.

Agile: Abstracting control from forwarding lets administrators dynamically adjust network-wide traffic flow to meet changing needs.

Centrally managed: Network intelligence is (logically) centralized in software-based SDN controllers that maintain a global view of the network, which appears to applications and policy engines as a single, logical switch.

Programmatically configured: SDN lets network managers configure, manage, secure, and optimize network resources very quickly via dynamic, automated SDN programs, which they can write themselves because the programs do not depend on proprietary software.

Open standards-based and vendor-neutral: When implemented through open standards, SDN simplifies network design and operation because instructions are provided by SDN controllers instead of multiple, vendor-specific devices and protocols.

Software Defined Network (SDN)

- SDN addresses the fact that the static architecture of conventional networks is ill-suited to the dynamic computing and storage needs of today's data centers, campuses, and carrier environments. The key computing trends driving the need for a new network paradigm include:
- **Changing traffic patterns**: Applications that commonly access geographically distributed databases and servers through public and private clouds require extremely flexible traffic management and access to bandwidth on demand.
- **The "consumerization of IT"**: The Bring Your Own Device (BYOD) trend requires networks that are both flexible and secure.
- **The rise of cloud services**: Users expect on-demand access to applications, infrastructure, and other IT resources.
- **"Big data" means more bandwidth**: Handling today's mega datasets requires massive parallel processing that is fueling a constant demand for additional capacity and any-to-any connectivity.

OpenFlow

- OpenFlow® is the first standard communications interface defined between the control and forwarding layers of an SDN architecture.
- OpenFlow® allows direct access to and manipulation of the forwarding plane of network devices such as switches and routers, both physical and virtual (hypervisor-based).
- OpenFlow-based SDN technologies enable IT to address the high-bandwidth, dynamic nature of today's applications, adapt the network to ever-changing business needs, and significantly reduce operations and management complexity.

OpenFlow

Programmability

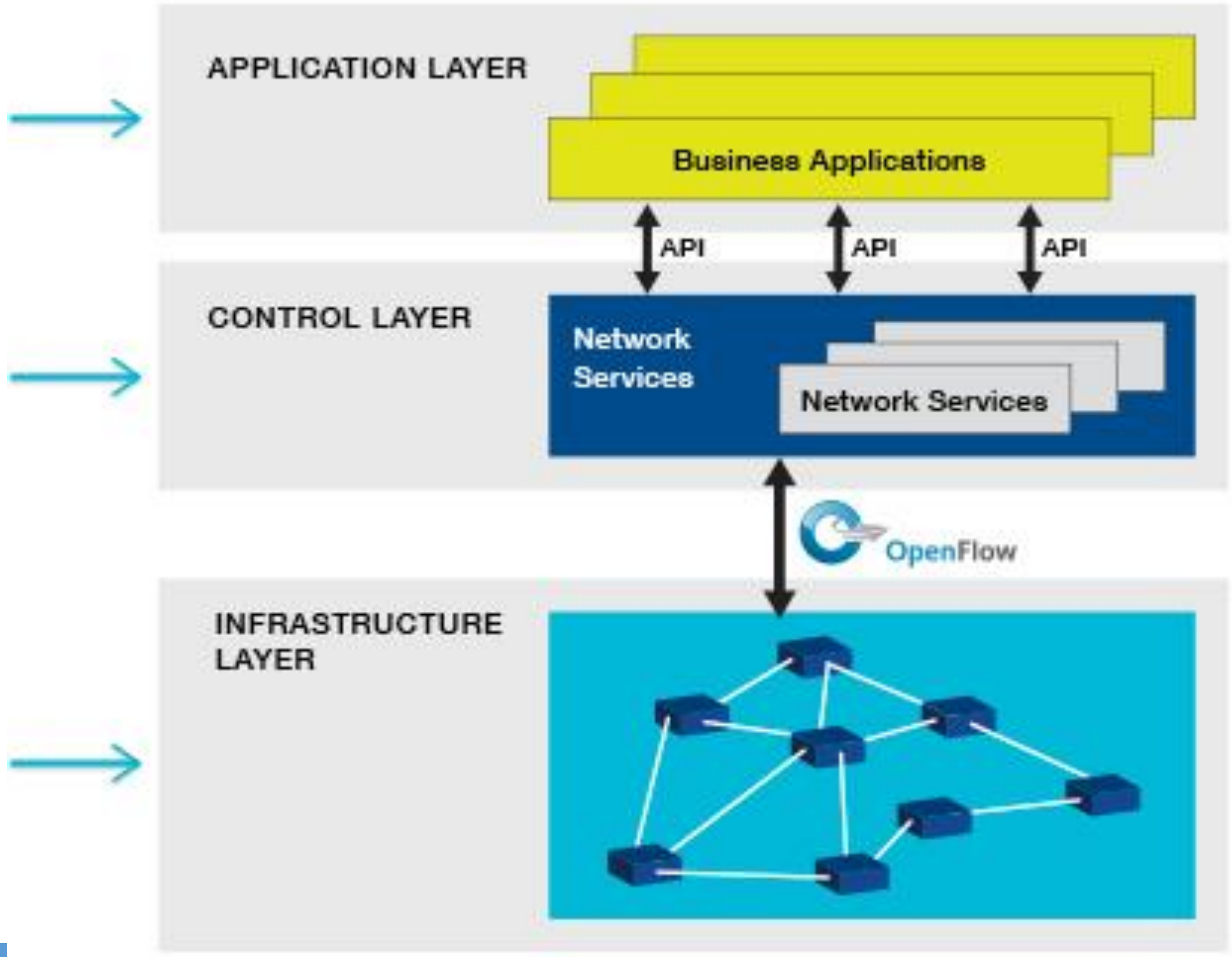
- Enable innovation/differentiation
- Accelerate new features and services introduction

Centralized Intelligence

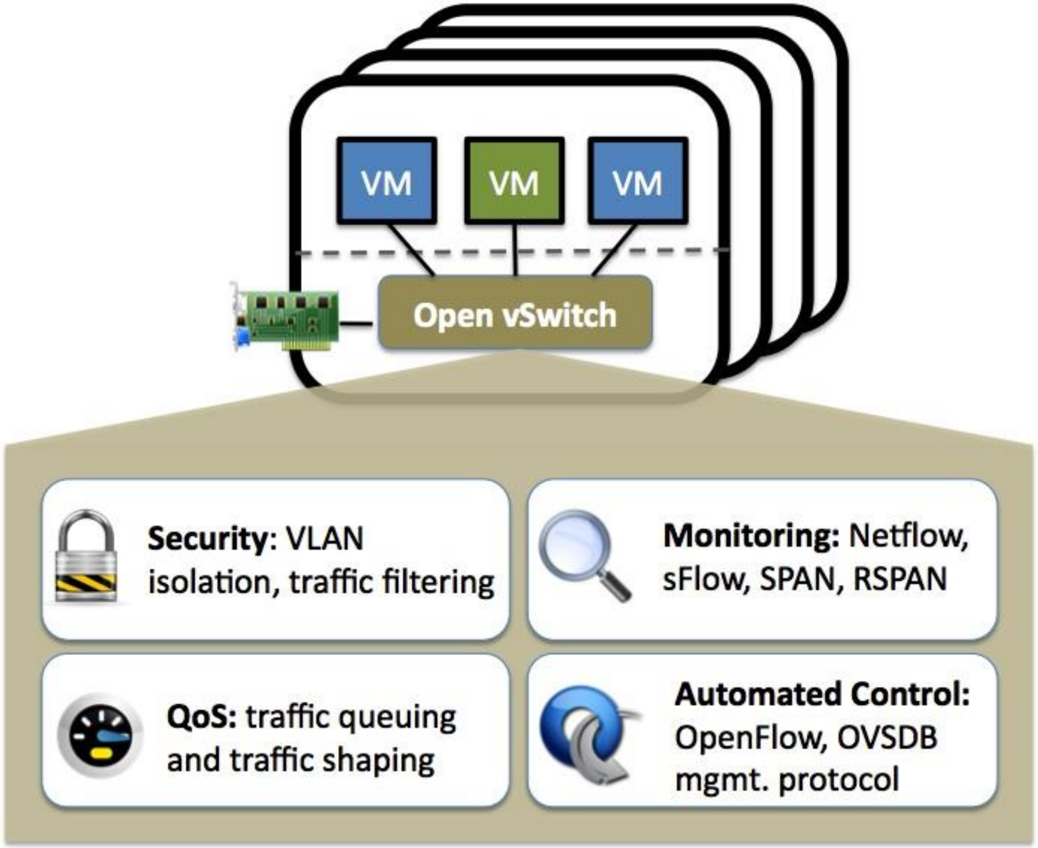
- Simplify provisioning
- Optimize performance
- Granular policy management

Abstraction

- Decouple:
  - Hardware & Software
  - Control plane & forwarding
  - Physical & logical config.



Open VSwitch


VM
VM
VM
Open vSwitch
Security: VLAN isolation, traffic filtering
Monitoring: Netflow, sFlow, SPAN, RSPAN
QoS: traffic queuing and traffic shaping
Automated Control: OpenFlow, OVSDB mgmt. protocol

- Open vSwitch is a production quality, multilayer virtual switch licensed under the open source **Apache 2.0** license.
- It is designed to enable massive network automation through programmatic extension, while still supporting standard management interfaces and protocols (e.g. NetFlow, sFlow, IPFIX, RSPAN, CLI, LACP, 802.1ag).
- In addition, it is designed to support distribution across multiple physical servers similar to VMware's vNetwork distributed vswitch or Cisco's Nexus 1000V.

Network Function Virtualization

- **Network-Function Virtualization** (NFV) is a network architecture concept that uses the technologies of IT virtualization to virtualize entire classes of network node functions into building blocks that may connect, or chain together, to create communication services.
- NFV is virtualizing Layer 4-7 functions such as virtualized load balancers, firewalls, intrusion detection devices and WAN accelerators, Routers, Proxies provided by a software by a inside a VM or a group of VMs.
- For Example providing a Load balancer service using HAProxy in a Linux VM

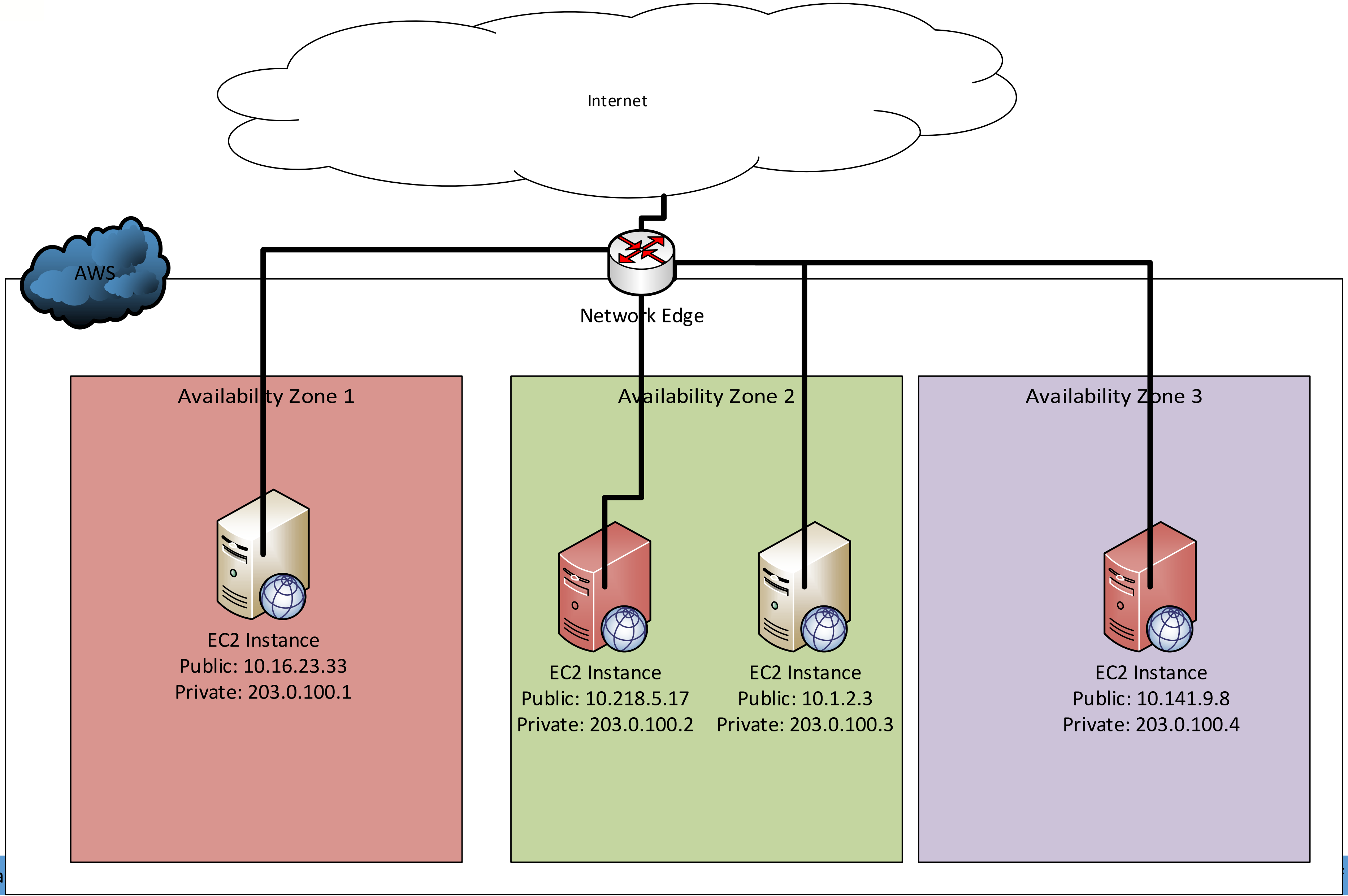
Cloud Networking

- Cloud networking is a new networking paradigm for building and managing secure private networks over the public Internet by utilizing global cloud computing infrastructure.
- In cloud networking, We Create Virtual Networks using Network Virtualization.
- Use SDN to make it more programmable networks
- Use Overlays to connect within or multiple cloud networks
- Use Network Function Virtualization to consume Network Functions as a Service and scale them as and when required.

AWS EC2 Classic Network

- EC2 Classic network is introduced initial period of AWS. It is available for Account only created before 12/4/2013
- Your instance receives a public IP and Private IP automatically assigned by Default
- AWS select a single private IP address for your instance; multiple IP addresses are not supported
- An EIP is disassociated from your instance when you stop it.
- DNS hostnames are enabled by default.
- A security group can reference security groups that belong to other AWS accounts.
- You can create up to 500 security groups in each region.
- You can assign an unlimited number of security groups to an instance when you launch it. You can't change the security groups of your running instance.
- Your instance can access the Internet directly through the AWS network edge.

AWS EC2 Classic



Internet

AWS

Network Edge

Availability Zone 1

EC2 Instance
Public: 10.16.23.33
Private: 203.0.100.1

Availability Zone 2

EC2 Instance
Public: 10.218.5.17
Private: 203.0.100.2

EC2 Instance
Public: 10.1.2.3
Private: 203.0.100.3

Availability Zone 3

EC2 Instance
Public: 10.141.9.8
Private: 203.0.100.4

Amazon VPC

- Amazon Virtual Private Cloud (Amazon VPC) enables you to launch Amazon Web Services (AWS) resources into a virtual network that you've defined.
- This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.
- Amazon VPC let you to create Virtual networks with in the cloud with Network Virtualization, SDN and Network Function Virtualization

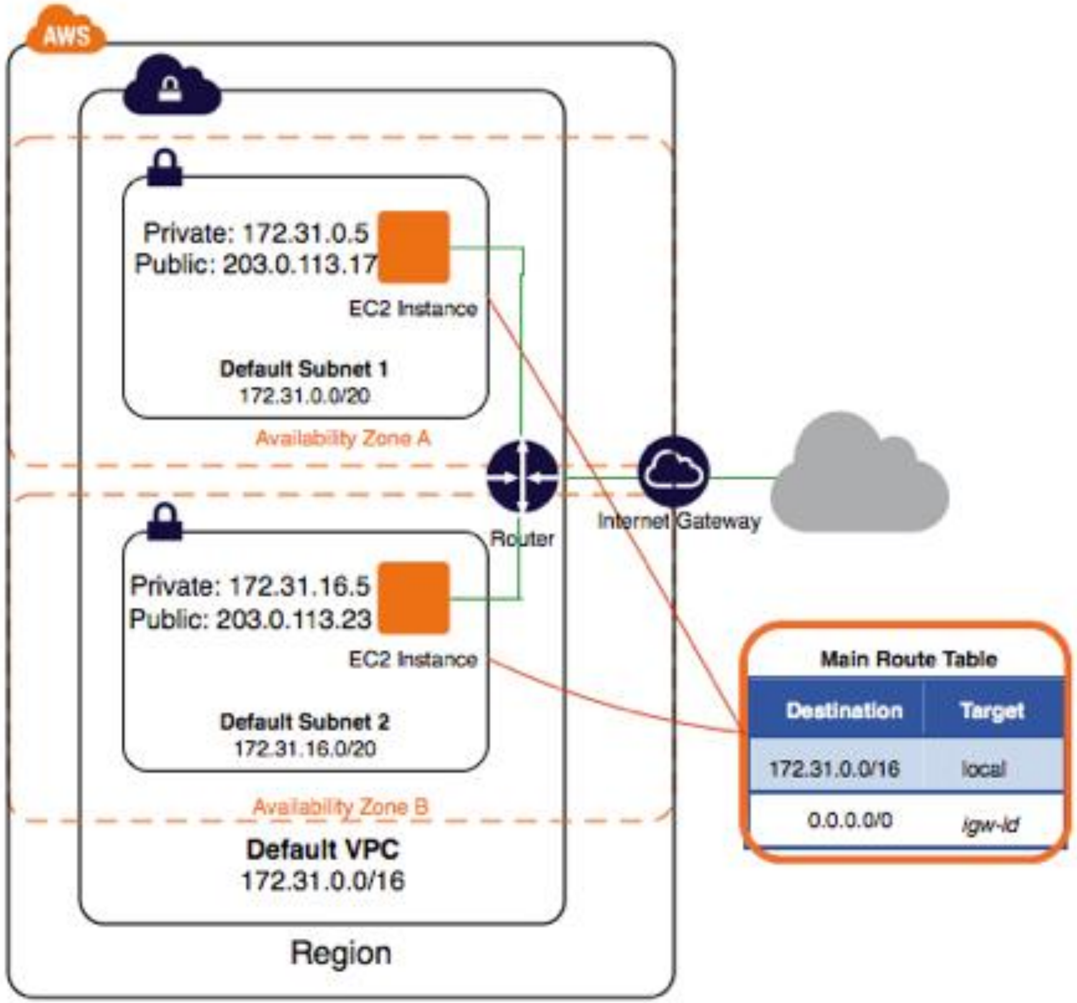
Amazon VPC

- A *virtual private cloud* (VPC) is a virtual network dedicated to your AWS account.
- It is logically isolated from other virtual networks in the AWS cloud.
- You can launch your AWS resources, such as Amazon EC2 instances, into your VPC.
- You can configure your VPC; you can select its IP address range, create subnets, and configure route tables, network gateways, and security settings.

Amazon VPC - Subnet

- A *subnet* is a range of IP addresses in your VPC. You can launch AWS resources into a subnet that you select.
- Use a public subnet for resources that must be connected to the Internet, and a private subnet for resources that won't be connected to the Internet.
- To protect the AWS resources in each subnet, you can use multiple layers of security, including security groups and network access control lists (ACL).

AWS Default Virtual Private Cloud



Main Route Table

Destination	Target
172.31.0.0/16	local
0.0.0.0/0	*igw-id*

- If your account created after 12/4/2013 it comes with Default VPC
- A *default VPC* that has a *default subnet* in each Availability Zone.
- If you have a default VPC and don't specify a subnet when you launch an instance, the instance is launched into your default VPC.
- Create an Internet gateway and connect it to your default VPC.
- Create a main route table for your default VPC with a rule that sends all traffic destined for the Internet to the Internet gateway.
- Create a default security group and associate it with your default VPC.
- Create a default network access control list (ACL) and associate it with your default VPC.
- Associate the default DHCP options set for your AWS account with your default VPC
- You can use Advances VPC features as and when required.

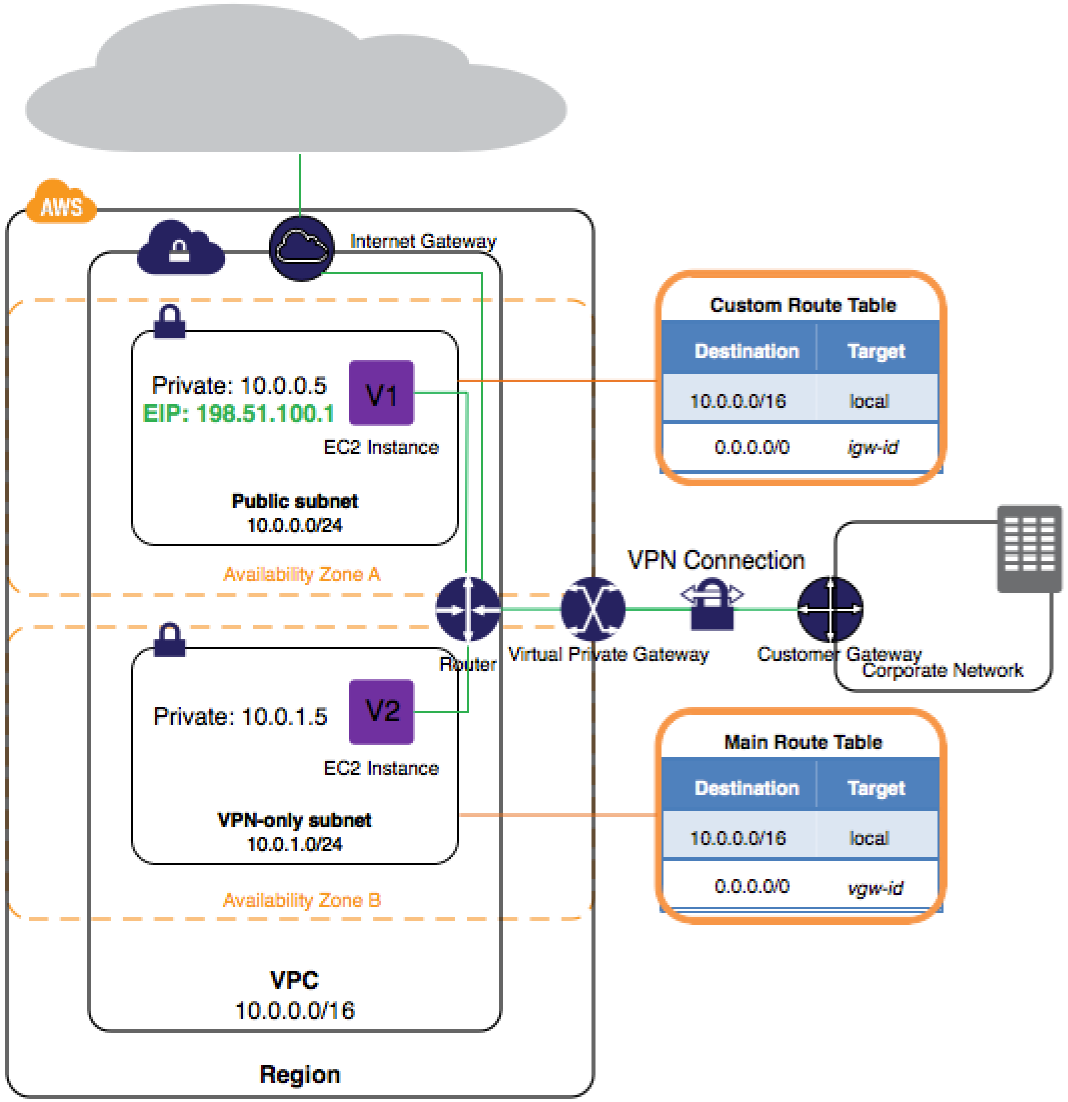
AWS Default Virtual Private Cloud

- Instances that you launch into a default subnet receive both a public IP address and a private IP address.
- Instances in a default subnet also receive both public and private DNS hostnames.
- Instances that you launch into a non default subnet in a default VPC don't receive a public IP address or a DNS hostname.
- You can change your subnet's default public IP addressing behavior.
- you can add subnets, modify the main route table, add additional route tables, associate additional security groups, update the rules of the default security group, and add VPN connections.
- You can use a default subnet as you would use any other subnet; you can add custom route tables and set network ACLs. You can also specify a default subnet when you launch an EC2 instance.

Non Default VPC

- You can create non default VPC any time
- VPC gives you advanced networking features such as :
  - Elastic Network Interface (ENIs)
  - Multiple Ips
  - Routing Tables
  - Egress Security Groups
  - Network ACLS
  - Private Connectivity
  - Enhanced Networking
  - Connectivity to your corporate or Datacenter

Route Tables



Custom Route Table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*igw-id*

Main Route Table

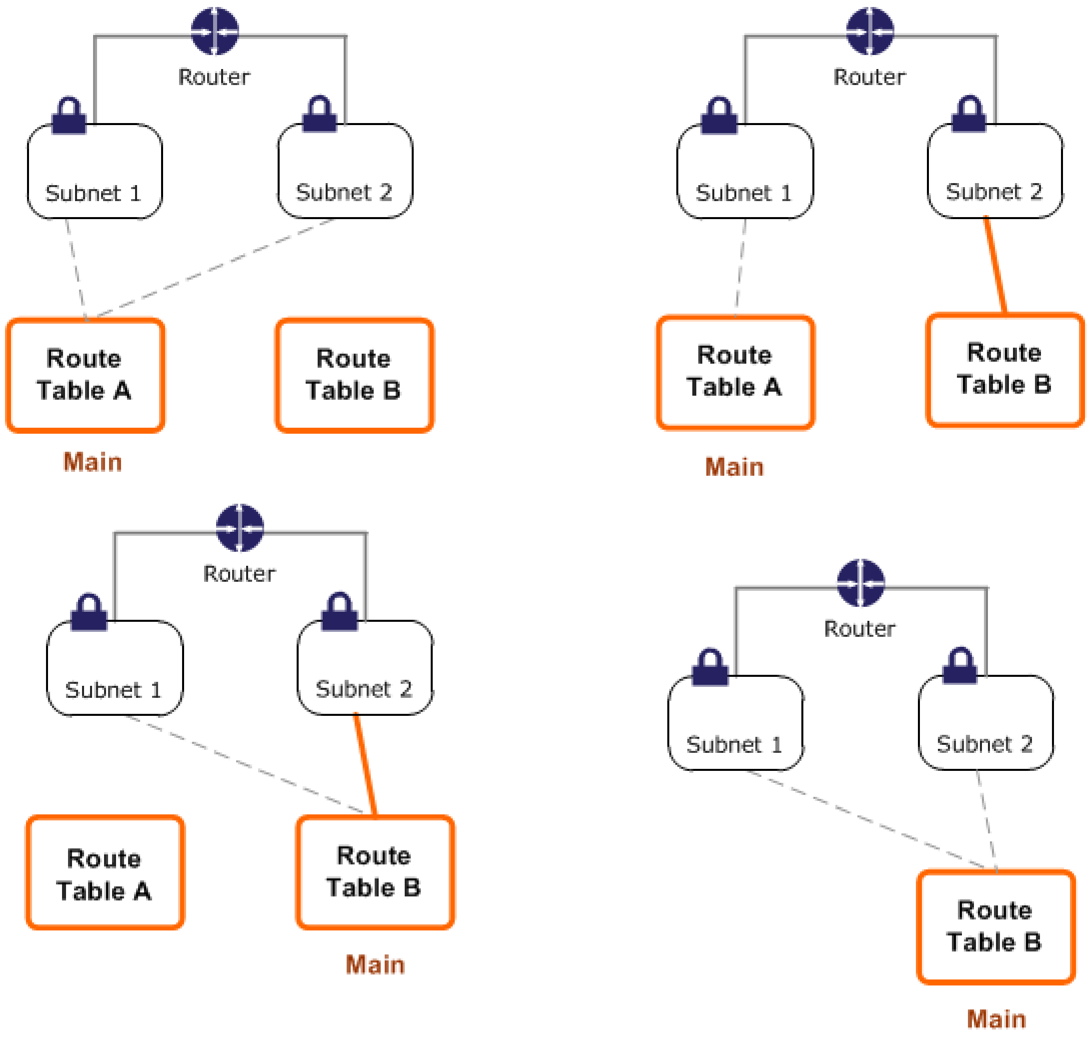
Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*vgw-id*

- 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.
- A subnet can only be associated with one route table at a time, but you can associate multiple subnets with the same route table.

Route Table Basics

- Your VPC has an implicit router.
- Your VPC automatically comes with a main route table that you can modify.
- You can create additional custom route tables for your VPC.
- Each subnet must be associated with a route table, which controls the routing for the subnet. If you don't explicitly associate a subnet with a particular route table, the subnet is implicitly associated with the main route table.
- You can replace the main route table with a custom table that you've created (so that this table is the default table each new subnet is associated with).
- Each route in a table specifies a destination CIDR and a target (for example, traffic destined for 172.16.0.0/12 is targeted for the virtual private gateway). We use the most specific route that matches the traffic to determine how to route the traffic.
- Every route table contains a local route that enables communication within a VPC. You cannot modify or delete this route.
- When you add an Internet gateway, a virtual private gateway, a NAT device, a peering connection, or a VPC endpoint in your VPC, you must update the route table for any subnet that uses these gateways or connections.
- There is a limit on the number of route tables you can create per VPC, and the number of routes you can add per route table.
  - Route tables per VPC: 100
  - Routes per route table : 50
  - BGP advertised routes per route table :100

Route Table Association



- The VPC console shows the number of subnets explicitly associated with each route table and provides information about subnets that are implicitly associated with the main route table.
- Subnets can be implicitly or explicitly associated with the main route table. Subnets typically won't have an explicit association to the main route table, although it might happen temporarily if you're replacing the main route table.
- You might want to make changes to the main route table, but to avoid any disruption to your traffic, you can first test the route changes using a custom route table. After you're satisfied with the testing, you then replace the main route table with the new custom table.

Route Priority

Destination	Target
10.0.0.0/16	Local
172.31.0.0/16	pcx-1a2b1a2b
0.0.0.0/0	igw-11aa22bb

- We use the most specific route in your route table that matches the traffic to determine how to route the traffic (longest prefix match).
- . For example, the following route table has a route for Internet traffic (`0.0.0.0/0`) that points to an Internet gateway, and a route for`172.31.0.0/16` traffic that points to a peering connection (`pcx-1a2b3c4d`). Any traffic from the subnet that's destined for the `172.31.0.0/16` IP address range uses the peering connection, because this route is more specific than the route for Internet gateway.
- Any traffic destined for within the VPC (10.0.0.0/16)is covered by local route

Route Priority

- If you've attached a virtual private gateway to your VPC and enabled route propagation on your route table, routes representing your VPN connection automatically appear as propagated routes in your route table’s list of routes.
- If these routes overlap with existing static routes and longest prefix match cannot be applied, then we prioritize the routes as follows in your VPC, from most preferred to least preferred:
  - Local routes for the VPC
  - Static routes whose targets are an Internet gateway, a virtual private gateway, a network interface, an instance ID, a VPC peering connection, or a VPC endpoint
  - Any propagated routes from a VPN connection or AWS Direct Connect connection

Route Priority

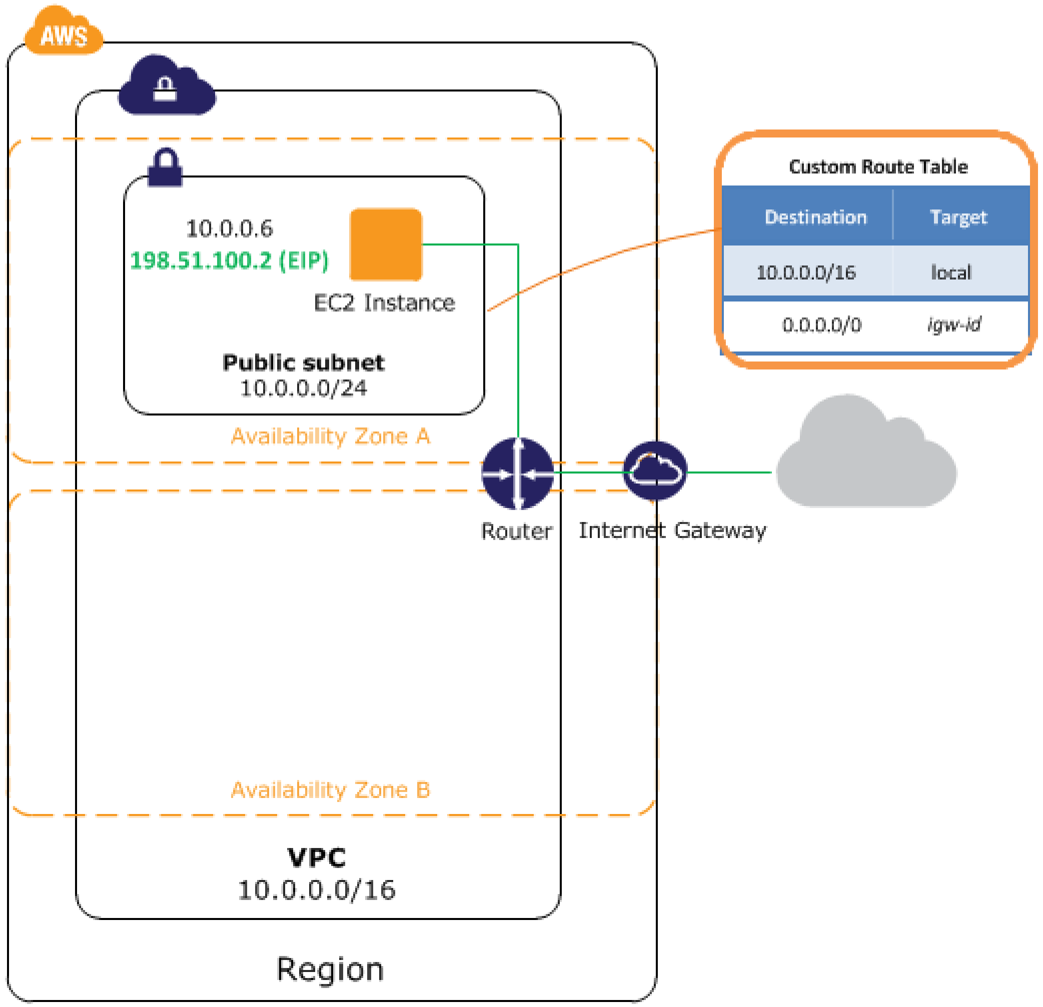
- If you have overlapping routes within a VPN connection and longest prefix match cannot be applied, then we prioritize the routes as follows in the VPN connection, from most preferred to least preferred:
  - BGP propagated routes from an AWS Direct Connect connection
  - Manually added static routes for a VPN connection
  - BGP propagated routes from a VPN connection

Destination	Target
10.0.0.0/16	Local
172.31.0.0/24	vgw-1a2b3c4d (propagated)
172.31.0.0/24	igw-11aa22bb

Routing Options

- The Following routing Option are enabled in VPC
  - Internet Gateway
  - NAT Device
  - Virtual Private Gateway
  - VPC Peering Connections
  - VPC endpoint

Internet Gateways



Custom Route Table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*igw-id*

- An Internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the Internet.
- It therefore imposes no availability risks or bandwidth constraints on your network traffic.
- An Internet gateway serves two purposes:
  - to provide a target in your VPC route tables for Internet-routable traffic,
  - To perform network address translation (NAT) for instances that have been assigned public IP addresses.

Internet Gateway Lifecycle

- Create Internet Gateway
- Attaching a Internet gateway to a VPC
- Update Route table for Internet gateway
- Allow Security Group for Instance access to internet
- Assaign Public IP or Elastic IP to Instance
- Delete Route table for Internet Gateway
- Detach Internet Gateway to VPC
- Delete Internet Gateway

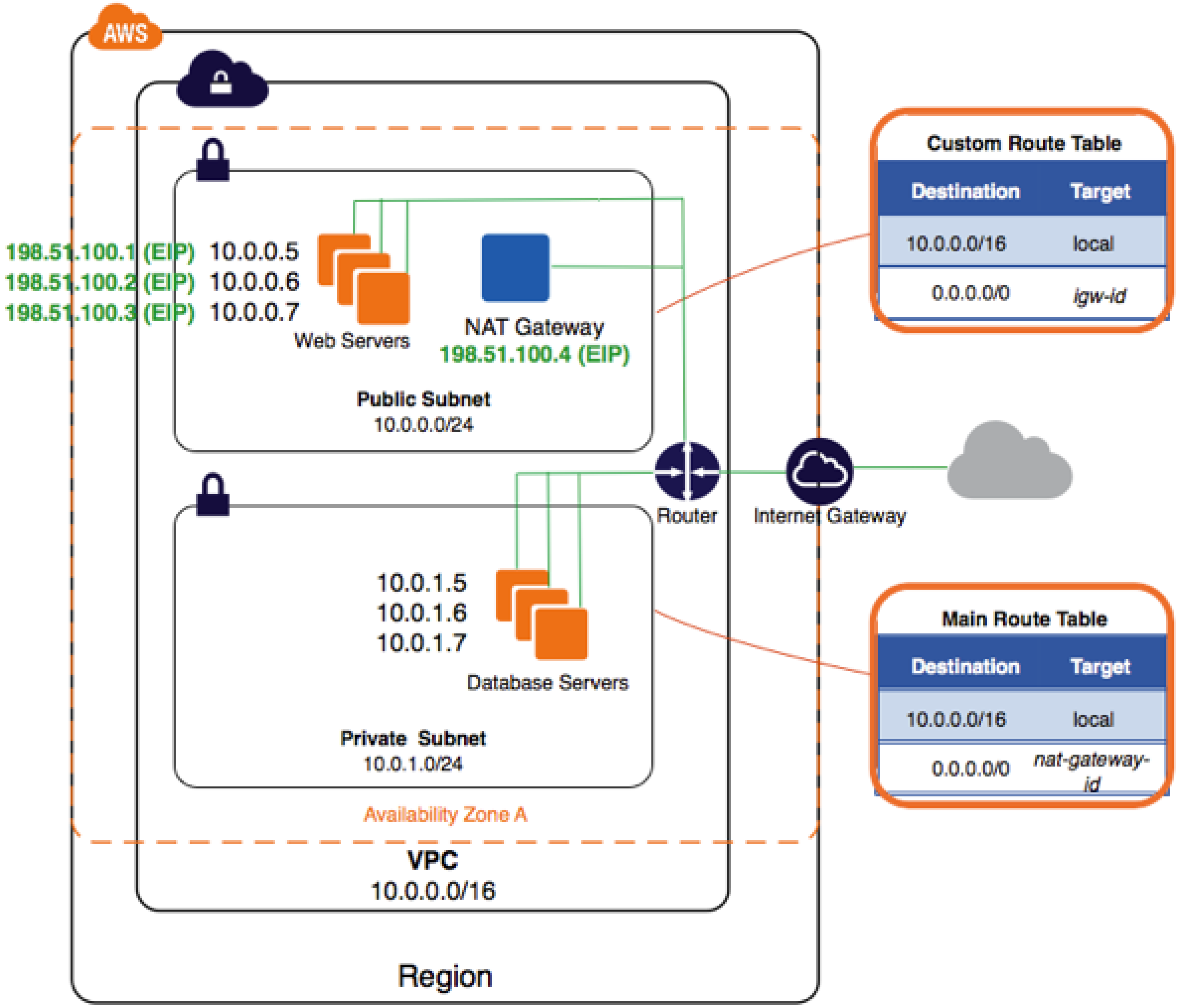
NAT Devices

- AWS offers two kinds of NAT devices
  - *NAT gateway*
  - *NAT instance*.

NAT Gateways

- 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.
- You are charged for creating and using a NAT gateway in your account. NAT gateway hourly usage and data processing rates apply. Amazon EC2 charges for data transfer also apply.

NAT Gateway Basics



Custom Route Table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*igw-id*

Main Route Table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*nat-gateway-id*

- To create a NAT gateway, you must specify the public subnet in which the NAT gateway will reside.
- You must also specify an Elastic IP address to associate with the NAT gateway when you create it.
- After you've created a NAT gateway, you must update the route table associated with one or more of your private subnets to point Internet-bound traffic to the NAT gateway.
- This enables instances in your private subnets to communicate with the Internet.
- Each NAT gateway is created in a specific Availability Zone and implemented with redundancy in that zone.
- You can create up to maximum of 5 NAT gateway per AZs
- Always create NAT Gateway in the same AZs of your instance to avoid AZ failure scenario

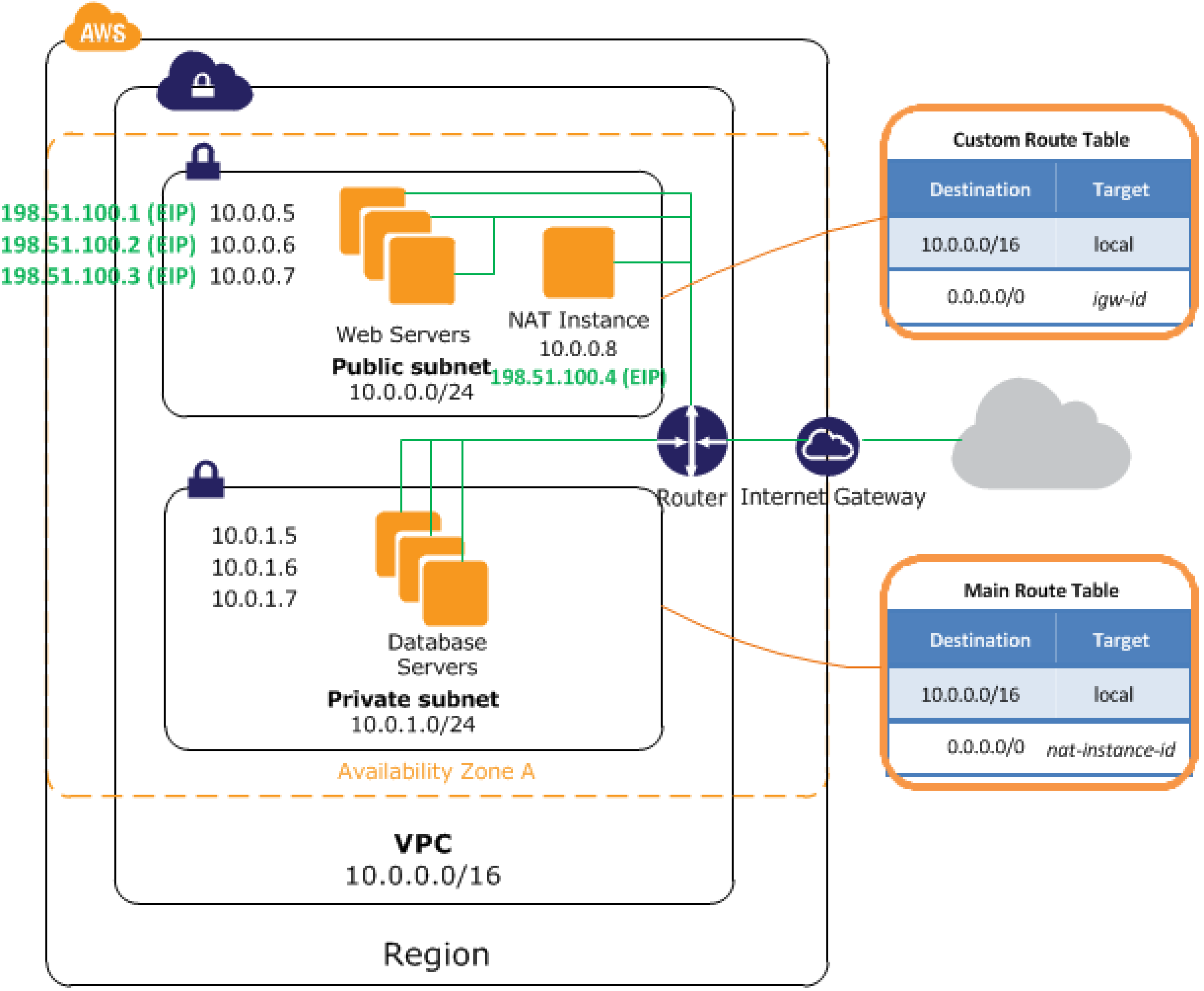
A NAT gateway characteristics:

- A NAT gateway supports bursts of up to 10 Gbps of bandwidth. If you require more than 10 Gbps bursts, you can distribute the workload by splitting your resources into multiple subnets, and creating a NAT gateway in each subnet.
- You can associate exactly one Elastic IP address with a NAT gateway. The association cannot be changed after you've created the NAT gateway. If you need to use a different Elastic IP address for your NAT gateway, you must create a new NAT gateway with the required address, update your route tables, and then delete the existing NAT gateway if it's no longer required.
- A NAT gateway supports the following protocols: TCP, UDP, and ICMP

A NAT gateway characteristics:

- You cannot associate a security group with a NAT gateway. You can use security groups for your instances in the private subnets to control the traffic to and from those instances.
- You can use a network ACL to control the traffic to and from the subnet in which the NAT gateway is located. The network ACL applies to the NAT gateway's traffic.
- When a NAT gateway is created, it receives an elastic network interface that's automatically assigned a private IP address from the IP address range of your subnet. You can view the NAT gateway's network interface in the Amazon EC2 console.

NAT Instances



Custom Route Table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*igw-id*

Main Route Table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*nat-instance-id*

- You can use a network address translation (NAT) instance in a public subnet in your VPC to enable instances in the private subnet to initiate outbound traffic to the Internet or other AWS services, but prevent the instances from receiving inbound traffic initiated by someone on the Internet.
- Amazon provides Amazon Linux AMIs that are configured to run as NAT instances.
- These AMIs include the string `amzn-ami-vpc-nat` in their names, so you can search for them in the Amazon EC2 console. When you launch an instance from a NAT AMI, the following configuration occurs on the instance:
  - •IPv4 forwarding is enabled and ICMP redirects are disabled in `/etc/sysctl.d/10-nat-settings.conf`
  - •A script located at `/usr/sbin/configure-pat.sh` runs at startup and configures iptables IP masquerading

NAT Gateway Vs NAT Instances

Attribute	NAT gateway	NAT instance
Availability	Highly available. NAT gateways in each Availability Zone are implemented with redundancy. Create a NAT gateway in each Availability Zone to ensure zone-independent architecture.	Use a script to manage failover between instances.
Bandwidth	Supports bursts of up to 10Gbps.	Depends on the bandwidth of the instance type.
Maintenance	Managed by AWS.You do not need to perform any maintenance.	Managed by you, for example, by installing software updates or operating system patches on the instance.
Performance	Software is optimized for handling NAT traffic.	A generic Amazon Linux AMI that's configured to perform NAT.
Cost	Charged depending on the number of NAT gateways you use, duration of usage, and amount of data that you send through the NAT gateways.	Charged depending on the number of NAT instances that you use, duration of usage, and instance type and size.
Type and size	Uniform offering; you don’t need to decide on the type or size.	Choose a suitable instance type and size, according to your predicted workload.

NAT Gateway Vs NAT Instances

Attribute	NAT gateway	NAT instance
Public IP addresses	Choose the Elastic IP address to associate with a NAT gateway at creation.	Use an Elastic IP address or a public IP address with a NAT instance. You can change the public IP address at any time by associating a new Elastic IP address with the instance.
Private IP addresses	Automatically selected from the subnet's IP address range when you create the gateway.	Assign a specific private IP address from the subnet's IP address range when you launch the instance.
Security groups	Cannot be associated with a NAT gateway. You can associate security groups with your resources behind the NAT gateway to control inbound and outbound traffic.	Associate with your NAT instance and the resources behind your NAT instance to control inbound and outbound traffic.
Network ACLs	Use a network ACL to control the traffic to and from the subnet in which your NAT gateway resides.	Use a network ACL to control the traffic to and from the subnet in which your NAT instance resides.
Flow logs	Use flow logs to capture the traffic.	Use flow logs to capture the traffic.
Port forwarding	Not supported.	Manually customize the configuration to support port forwarding.
Bastion servers	Not supported.	Use as a bastion server.
Traffic metrics	Not supported.	View CloudWatch metrics

DHCP Options Sets

- The Dynamic Host Configuration Protocol (DHCP) provides a standard for passing configuration information to hosts on a TCP/IP network.
- The Options field of a DHCP message contains the configuration parameters. Some of those parameters are the domain name, domain name server, and the netbios-node-type.
- DHCP options sets are associated with your AWS account so that you can use them across all of your virtual private clouds (VPC).
- When you create a VPC, we automatically create a set of DHCP options and associate them with the VPC. This set includes two options: `domain-name-servers=AmazonProvidedDNS`, and `domain-name=`*`domain-name-for-your-region`*. AmazonProvidedDNS is an Amazon DNS server, and this option enables DNS for instances that need to communicate over the VPC's Internet gateway. The string `AmazonProvidedDNS` maps to a DNS server running on a reserved IP address at the base of the VPC network range, plus two. For example, the DNS Server on a 10.0.0.0/16 network is located at 10.0.0.2.

DNS with Your VPC

- When you launch an instance in Default VPC we provide the instance with public and private DNS hostnames.
- Instances that you launch into a nondefault VPC might have public and private DNS hostnames, depending on the settings you specify for the VPC and for the instance.
- We support the following VPC attributes to control DNS support.

Attribute	Description
enableDnsHostnames	Indicates whether the instances launched in the VPC get DNS hostnames. If this attribute is true, instances in the VPC get DNS hostnames; otherwise, they do not.
enableDnsSupport	Indicates whether the DNS resolution is supported for the VPC. If this attribute is false, the Amazon provided DNS service in the VPC that resolves public DNS hostnames to IP addresses is not enabled. If this attribute is true, queries to the Amazon provided DNS server at the 169.254.169.253 IP address, or the reserved IP address at the base of the VPC network range plus two will succeed.

DHCP Options Sets

DHCP Option Name	Description
domain-name-servers	The IP addresses of up to four domain name servers, or AmazonProvidedDNS. The default DHCP option set specifies AmazonProvidedDNS. If specifying more than one domain name server, separate them with commas.
domain-name	If you're using AmazonProvidedDNS in us-east-1, specify ec2.internal. If you're using AmazonProvidedDNS in another region, specify*region*.compute.internal (for example, ap-northeast-1.compute.internal). Otherwise, specify a domain name (for example, MyCompany.com). **Important** Some Linux operating systems accept multiple domain names separated by spaces. However, other Linux operating systems and Windows treat the value as a single domain, which results in unexpected behavior. If your DHCP options set is associated with a VPC that has instances with multiple operating systems, specify only one domain name.
ntp-servers	The IP addresses of up to four Network Time Protocol (NTP) servers.
netbios-name-servers	The IP addresses of up to four NetBIOS name servers.
netbios-node-type	The NetBIOS node type (1, 2, 4, or 8). We recommend that you specify 2 (broadcast and multicast are not currently supported).

Network ACLs

- A *network access control list (ACL)* is an optional layer of security that acts as a firewall for controlling traffic in and out of a subnet. You might set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your VPC.

Security Group	**Network ACL**
Operates at the instance level (first layer of defense)	Operates at the subnet level (second layer of defense)
Supports allow rules only	Supports allow rules and deny rules
Is stateful: Return traffic is automatically allowed, regardless of any rules	Is stateless: Return traffic must be explicitly allowed by rules
We evaluate all rules before deciding whether to allow traffic	We process rules in number order when deciding whether to allow traffic
Applies to an instance only if someone specifies the security group when launching the instance, or associates the security group with the instance later on	Automatically applies to all instances in the subnets it's associated with (backup layer of defense, so you don't have to rely on someone specifying the security group)

Network ACL Basics

- A network ACL is a numbered list of rules that we evaluate in order, starting with the lowest numbered rule, to determine whether traffic is allowed in or out of any subnet associated with the network ACL. The highest number that you can use for a rule is 32766. We recommend that you start by creating rules with rule numbers that are multiples of 100, so that you can insert new rules where you need to later on.
- A network ACL has separate inbound and outbound rules, and each rule can either allow or deny traffic.
- Your VPC automatically comes with a modifiable default network ACL; by default, it allows all inbound and outbound traffic.
- You can create custom network ACL. Each custom network ACL starts out closed (permits no traffic) until you add rules.
- Each subnet must be associated with a network ACL; if you don't explicitly associate a subnet with a network ACL, the subnet is automatically associated with the default network ACL.
- Network ACLs are stateless; responses to allowed inbound traffic are subject to the rules for outbound traffic (and vice versa).

Network ACL Rules

- Rule number. Rules are evaluated starting with the lowest numbered rule. As soon as a rule matches traffic, it's applied regardless of any higher-numbered rule that may contradict it.
- Protocol. You can specify any protocol that has a standard protocol number, If you specify ICMP as the protocol, you can specify any or all of the ICMP types and codes.
- [Inbound rules only] The source of the traffic (CIDR range) and the destination (listening) port or port range.
- [Outbound rules only] The destination for the traffic (CIDR range) and the destination port or port range.
- Choice of ALLOW or DENY for the specified traffic.

Default Network ACL

Inbound

Rule #	Type	Protocol	Port Range	Source	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	ALLOW
*	All traffic	All	All	0.0.0.0/0	DENY

Outbound

Rule #	Type	Protocol	Port Range	Destination	Allow/Deny
100	All traffic	all	all	0.0.0.0/0	ALLOW
*	All traffic	all	all	0.0.0.0/0	DENY

- The default network ACL is configured to allow all traffic to flow in and out of the subnets to which it is associated. Each network ACL includes a rule whose rule number is an asterisk. This rule ensures that if a packet doesn't match any of the other numbered rules, it's denied. You can't modify or remove this rule.

VPC Flow Logs

- VPC Flow Logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC.
- Flow log data is stored using Amazon CloudWatch Logs. After you've created a flow log, you can view and retrieve its data in Amazon CloudWatch Logs.
- It helps to troubleshoot why specific traffic is not reaching an instance, which in turn can help you diagnose overly restrictive security group rules. You can also use flow logs as a security tool to monitor the traffic that is reaching your instance.

Flow Logs Basics

- You can create a flow log for a VPC, a subnet, or a network interface. If you create a flow log for a subnet or VPC, each network interface in the VPC or subnet is monitored.
- Flow log data is published to a log group in CloudWatch Logs, and each network interface has a unique log stream. Log streams contain *flow log records*, which are log events consisting of fields that describe the traffic for that network interface.
- To create a flow log, Specify the Following
  - Resource
  - type of traffic to capture (accepted traffic, rejected traffic, or all traffic),
  - name of a log group in CloudWatch Logs to which the flow log will be published
  - ARN of an IAM role that has sufficient permission to publish the flow log to the CloudWatch Logs log group.
- You can create flow logs for network interfaces that are created by other AWS services; for example, Elastic Load Balancing, Amazon RDS, Amazon ElastiCache, Amazon Redshift, and Amazon WorkSpaces.
- If you no longer require a flow log, you can delete it. Deleting a flow log disables the flow log service for the resource, and no new flow log records or log streams are created. It does not delete any existing flow log records or log streams for a network interface. To delete an existing log stream, you can use the CloudWatch Logs console. After you've deleted a flow log, it can take several minutes to stop collecting data.

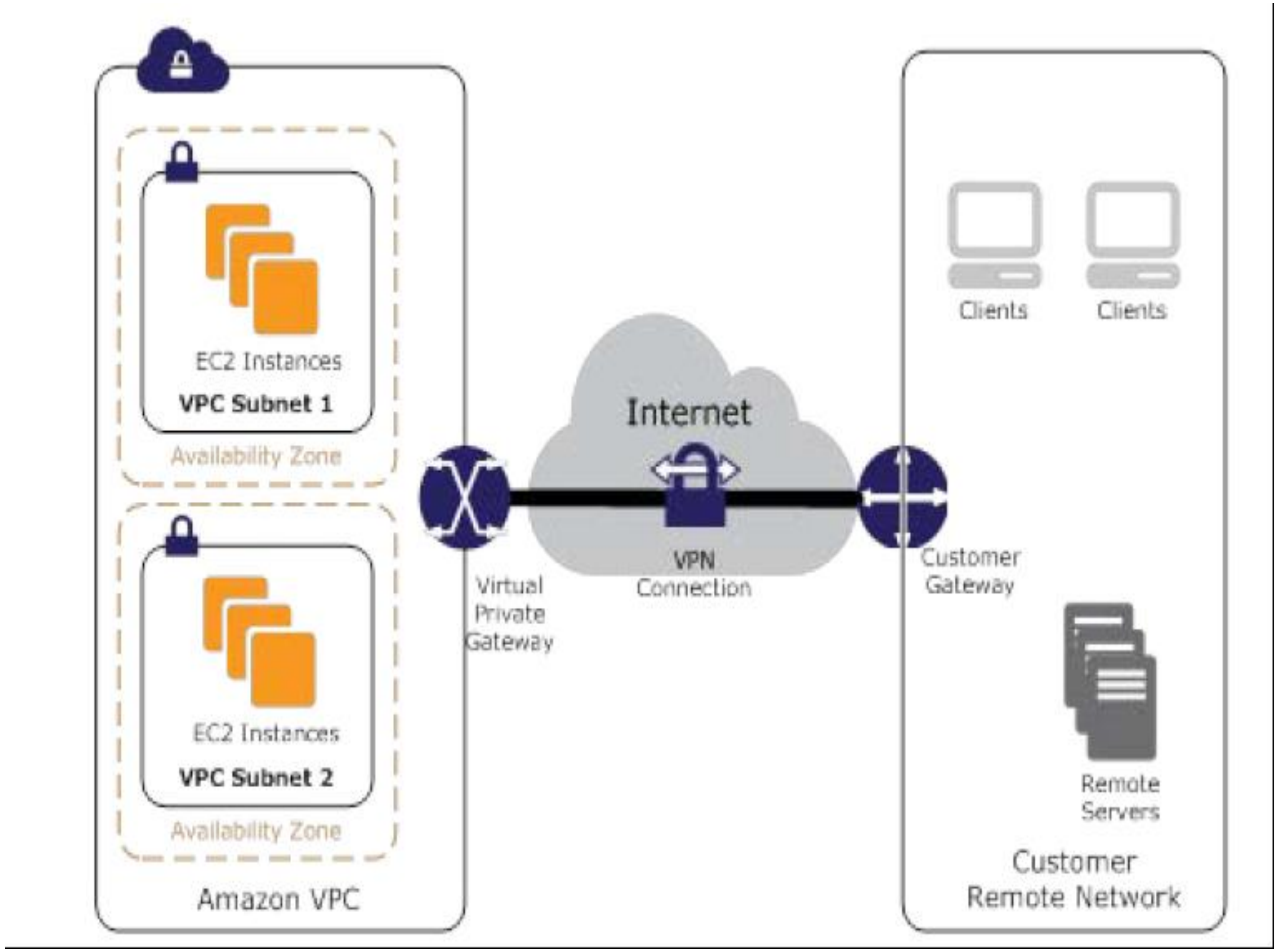
Routing and Private Connections

- Amazon VPC provides multiple network connectivity options for you to leverage depending on your current network designs and requirements.
- These connectivity options include leveraging either the Internet or an AWS Direct Connect connection as the network “backbone” and terminating the connection into either AWS or user managed network endpoints.
- Additionally, with AWS, you can choose how network routing will be delivered between Amazon VPC and your networks, leveraging either AWS or user-managed network equipment and routes.

User Network–to–Amazon VPC Connectivity Options

- Hardware VPN
  - Describes establishing a hardware VPN connection from your network equipment on a remote network to AWS-managed network equipment attached to your Amazon VPC
- AWS Direct Connect
  - Describes establishing a private, logical connection from your remote network to Amazon VPC, leveraging AWS Direct Connect.
- **AWS Direct Connect + VPN**
  - Describes establishing a private, encrypted connection from your remote network to Amazon VPC, leveraging AWS Direct Connect.
- **AWS VPN CloudHub**
  - Describes establishing a hub-and-spoke model for connecting remote branch offices.
- **Software VPN**
  - Describes establishing a VPN connection from your equipment on a remote network to a user-managed software VPN appliance running inside an Amazon VPC.

Hardware VPN



Amazon VPC provides the option of creating an IPsec, hardware VPN connection between remote customer networks and their Amazon VPC over the Internet

Consider taking this approach when you want to take advantage of an AWS-managed VPN endpoint that includes automated multi–data center redundancy and failover built into the AWS side of the VPN connection.

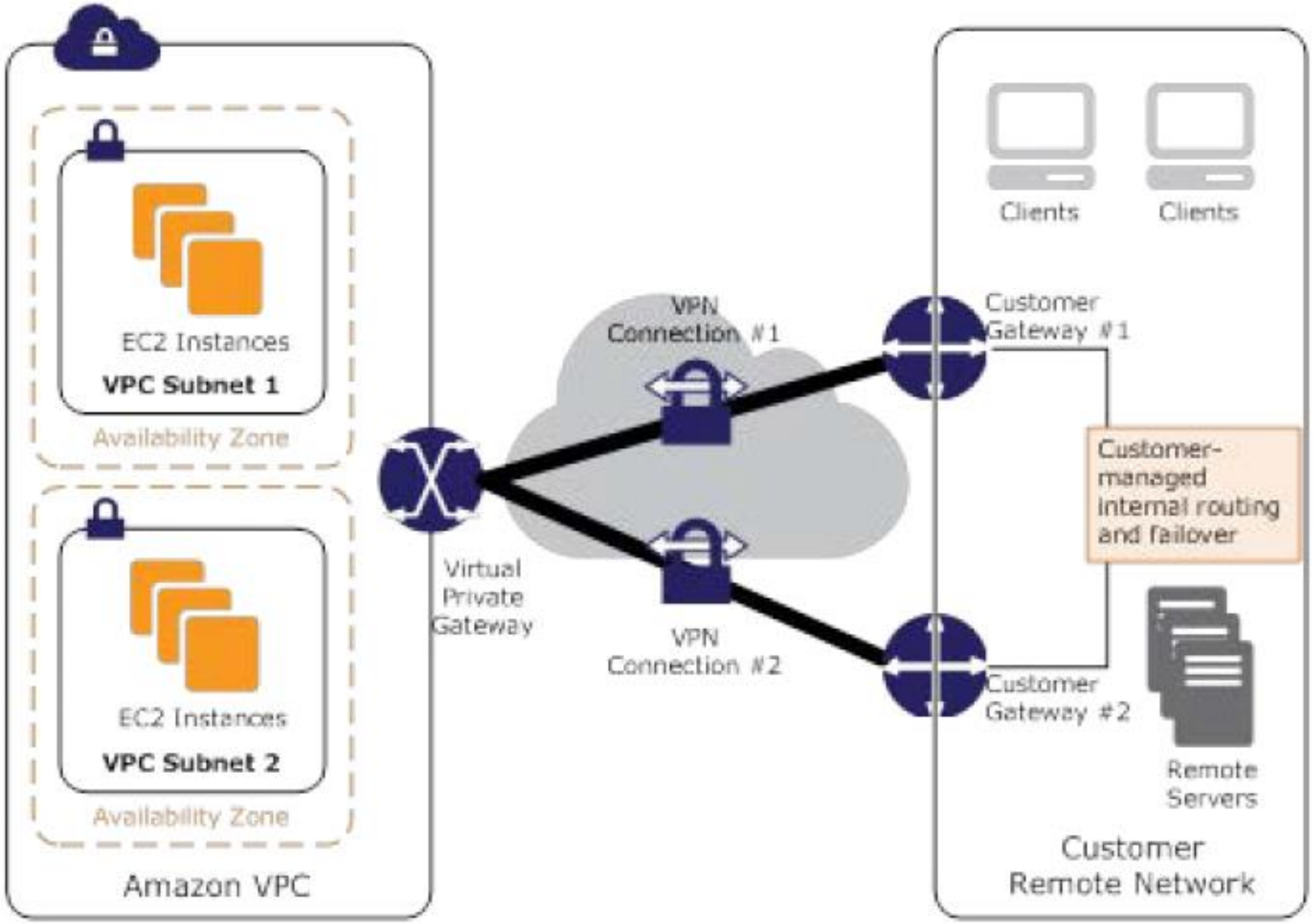
Reuse existing VPN equipment and processes

Reuse existing Internet connections

AWS-managed endpoint includes multidata center redundancy and automated failover

Supports static routes or dynamic Border Gateway Protocol (BGP) peering and routing policies

Hardware VPN



- The Amazon virtual private gateway (VGW) represents two distinct VPN endpoints, physically located in separate data centers to increase the availability of your VPN connection.
- The VGW also supports and encourages multiple user gateway connections so you can implement redundancy and failover on your side of the VPN connection
- Both dynamic and static routing options are provided to give you flexibility in your routing configuration.
- Both dynamic and static routing options are provided to give you flexibility in your routing configuration.
- With dynamic routing, you can also specify routing priorities, policies, and weights (metrics) in your BGP advertisements and influence the network path between your network(s) and AWS.
- It is important to note that when BGP is used, both the IPSec and the BGP connections must be terminated on the same user gateway device, so it must be capable of terminating both IPSec and BGP connections.

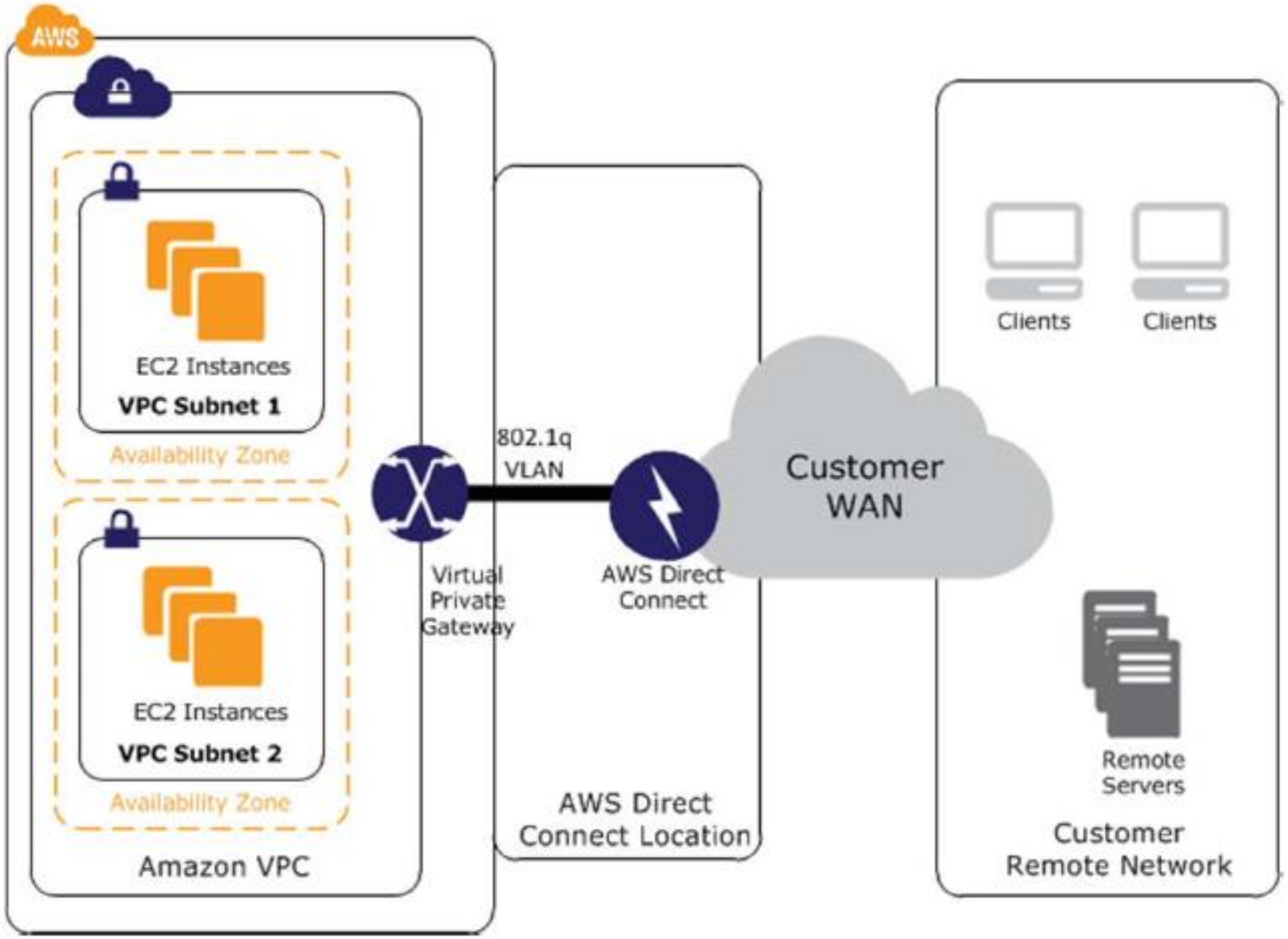
Limitation of Hardware VPN

- Network latency, variability, and availability are dependent on Internet conditions
- Customer-managed is responsible for implementing redundancy and failover (if required)
- Customer device must support single-hop BGP (when leveraging BGP for dynamic routing)

AWS Direct Connect

- AWS Direct Connect makes it easy to establish a dedicated connection from an on-premises network to Amazon VPC.
- Using AWS Direct Connect, you can establish private connectivity between AWS and your data center, office, or colocation environment.
- This private connection can reduce network costs, increase bandwidth throughput, and provide a more consistent network experience than Internet-based connections.
- AWS Direct Connect lets you establish 1 Gbps or 10 Gbps dedicated network connections (or multiple connections) between AWS networks and one of the AWS Direct Connect locations.

AWS Direct Connect



- It uses industry-standard VLANs to access Amazon Elastic Compute Cloud (Amazon EC2) instances running within an Amazon VPC using private IP addresses.
- You can choose from an ecosystem of WAN service providers for integrating your AWS Direct Connect endpoint in an AWS Direct Connect location with your remote networks.
- AWS Direct connect Location

https://aws.amazon.com/directconnect/faqs/

AWS Direct connect Partner Locations

https://aws.amazon.com/directconnect/partners/

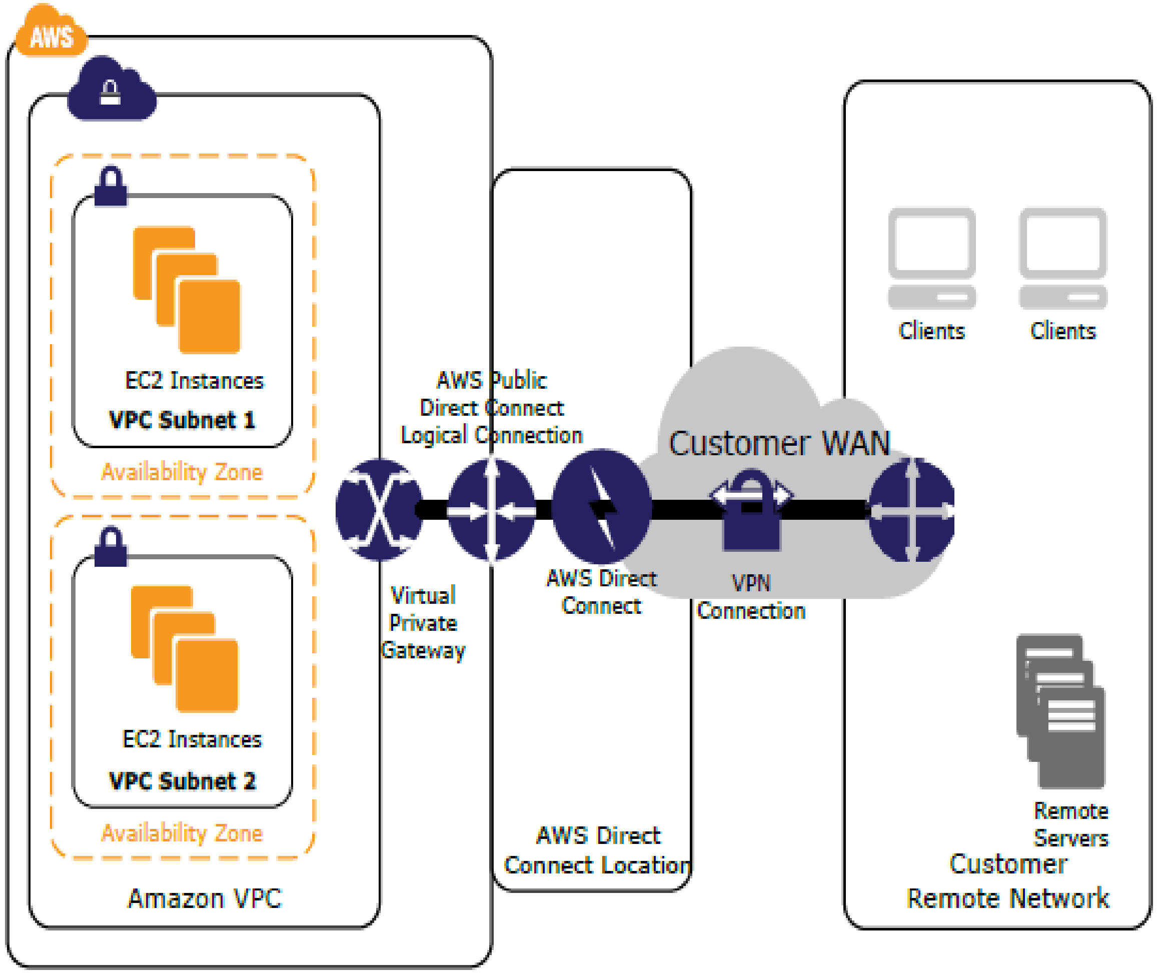
AWS Direct Connect

- When to use AWS Direct Connect
  - Working with Large Data Sets
  - Real-time Data Feeds
  - Hybrid Environments
- Pricing
  - Pricing is per port-hour for all AWS Direct Connect locations.
  - Data Transfer In is $0.00 per GB in all locations.
  - Data Transfer Out is cost per GB per month
  - http://aws.amazon.com/directconnect/pricing/

AWS Direct Connect

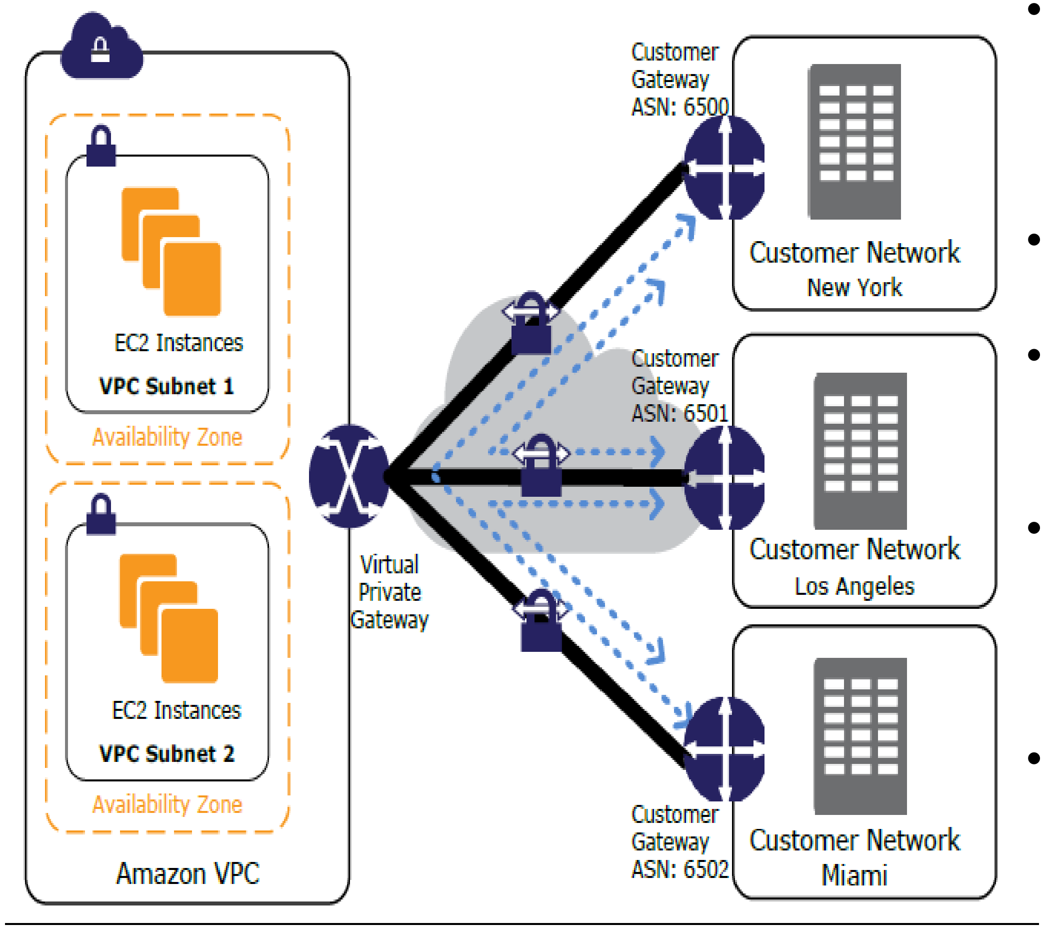
- To use AWS Direct Connect, you simply:
  - Decide on an AWS Direct Connect location, how many connections you would like to use, and the port size. Multiple ports can be used simultaneously for increased bandwidth or redundancy.
  - Use the AWS Management Console to create your connection request(s).
  - If you are connecting from a remote location, you can work with an APN Partner supporting Direct Connect or a network carrier of your choice.
  - Once your request is confirmed, you will receive an email which contains a Letter of Authorization – Connecting Facility Assignment (LOA-CFA).
  - Provide the LOA-CFA to an APN Partner or your service provider who will establish the connection on your behalf.
  - Once the connection is up, use the AWS Management Console to configure one or more virtual interfaces to establish network connectivity.

AWS Direct Connect + VPN



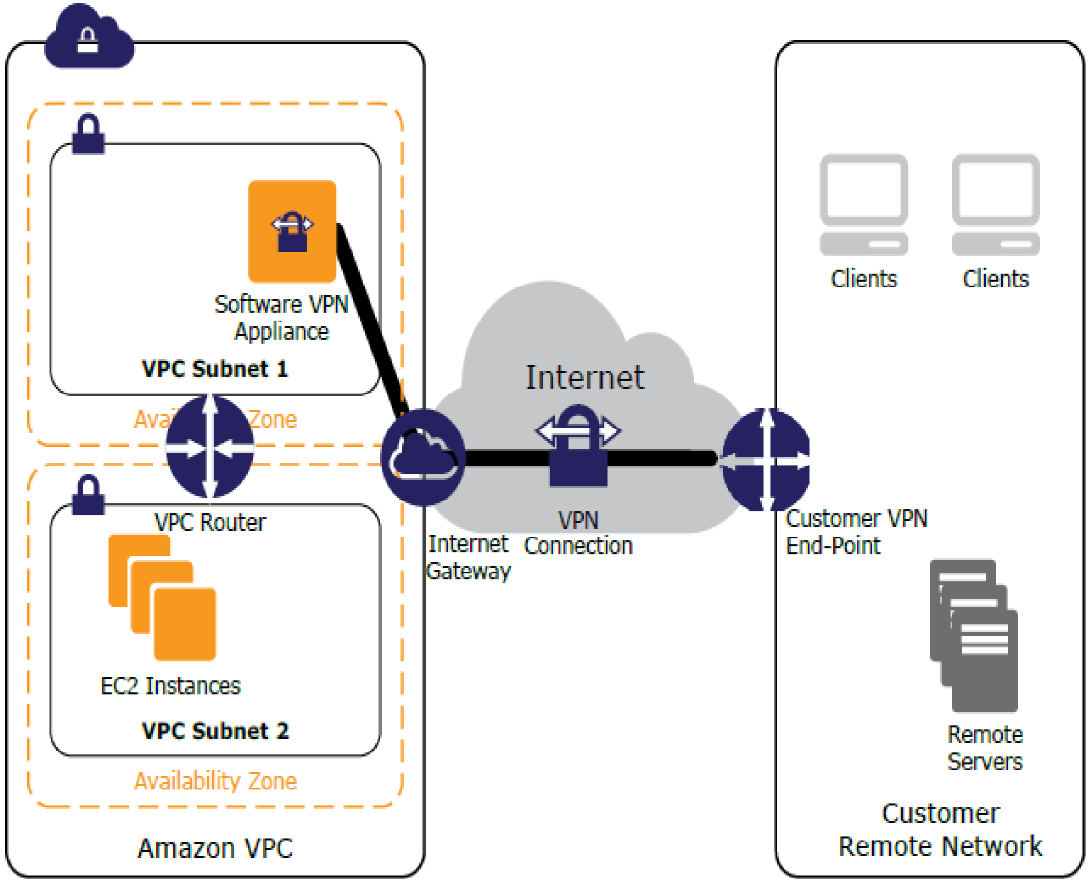
- AWS Direct Connect + VPN, you can combine one or more AWS Direct Connect dedicated network connections with the Amazon VPC hardware VPN.
- This combination provides an IPsec-encrypted private connection that also reduces network costs, increases bandwidth throughput, and provides a more consistent network experience than Internet-based VPN connections.
- This solution combines the AWS-managed benefits of the hardware VPN solution with low latency, increased bandwidth, more consistent benefits of the AWS Direct Connect solution, and an end-to-end, secure IPsec connection.

AWS VPN CloudHub



- You can securely communicate from one site to another using the AWS VPN CloudHub.
- The AWS VPN CloudHub operates on a simple hub-and-spoke model that you can use with or without a VPC.
- Use this design if you have multiple branch offices and existing Internet connections and would like to implement a convenient, potentially low cost hub-and-spoke model for primary or backup connectivity between these remote offices.
- blue dashed lines indicating network traffic between remote sites being routed over their AWS VPN connections.
- AWS VPN CloudHub leverages an Amazon VPC virtual private gateway with multiple gateways, each using unique BGP autonomous system numbers (ASNs).
- Your gateways advertise the appropriate routes (BGP prefixes) over their VPN connections. These routing advertisements are received and readvertised to each BGP peer so that each site can send data to and receive data from the other sites.
- The remote network prefixes for each spoke must have unique ASNs, and the sites must not have overlapping IP ranges. Each site can also send and receive data from the VPC as if they were using a standard VPN connection.

Software VPN



- Amazon VPC offers you the flexibility to fully manage both sides of your Amazon VPC connectivity by creating a VPN connection between your remote network and a software VPN appliance running in your Amazon VPC network.
- This option is recommended if you must manage both ends of the VPN connection either for compliance purposes or for leveraging gateway devices that are not currently supported by Amazon VPC's hardware VPN solution.
- You can choose VPN products from well-known security companies like Check Point, Astaro, OpenVPN Technologies, and Microsoft, as well as popular open source tools like OpenVPN, Openswan, and IPsec-Tools.
- It is your responsibility for you to manage the software appliance, including configuration, patches, and upgrades.
- Implement in a HA Architecture to avoid any single point of Failure

Amazon VPC-to-Amazon VPC Connectivity Options

- Use these design patterns when you want to integrate multiple Amazon VPCs into a larger virtual network.
- This is useful if you require multiple VPCs due to security, billing, presence in multiple regions, or internal charge-back requirements to more easily integrate AWS resources between Amazon VPCs.
- You can also combine these patterns with the UsCustomer Network-–to–-Amazon VPC Connectivity Options for creating a corporate network that spans remote networks and multiple VPCs.
- VPC connectivity between VPCs is best achieved when using nonoverlapping IP ranges for each VPC being connected. For example, if you’d like to connect multiple VPCs, make sure each VPC is configured with unique Classless Inter-Domain Routing (CIDR) ranges. Therefore, we advise you to allocate a single, contiguous, nonoverlapping CIDR block to be used by each VPC.

Amazon VPC-to-Amazon VPC Connectivity Options

- VPC Peering
- Software VPN
- Software-to-hardware VPN
- Hardware VPN
- AWS Direct Connect

VPC Peering

- A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IP addresses.
- Instances in either VPC can communicate with each other as if they are within the same network.
- You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account within a single region.
- AWS uses the existing infrastructure of a VPC to create a VPC peering connection
- it is neither a gateway nor a VPN connection, and does not rely on a separate piece of physical hardware. There is no single point of failure for communication or a bandwidth bottleneck.

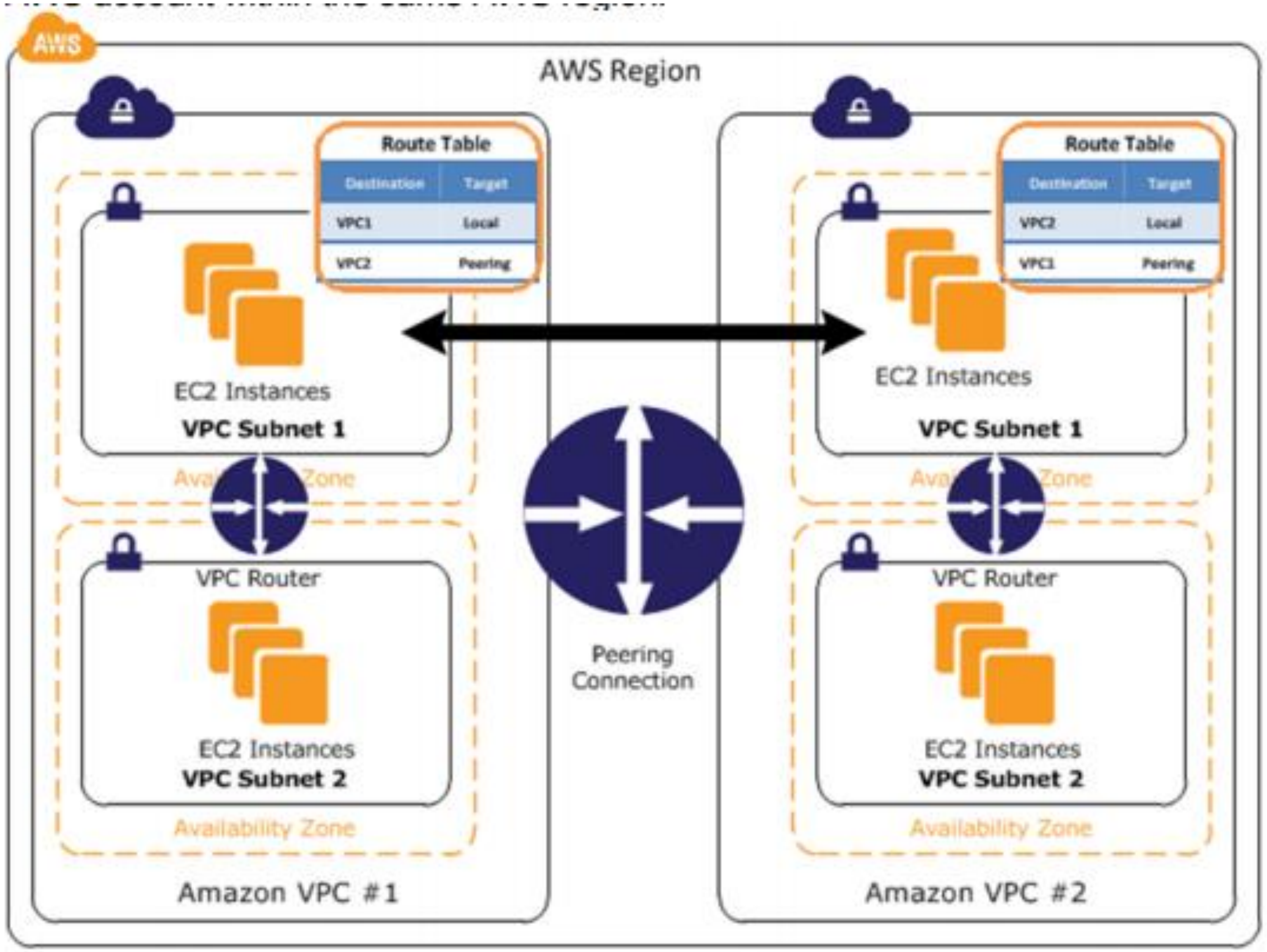
VPC Peering

- A VPC peering connection can help 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.

VPC Peering Basics

- To establish a VPC peering connection, the owner of the *requester VPC* (or *local VPC*) sends a request to the owner of the *peer VPC* to create the VPC peering connection.
- . The peer 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.
- The owner of the peer VPC has to accept the VPC peering connection request to activate the VPC peering connection.
- To enable the flow of traffic between the peer VPCs using private IP addresses, add a route to one or more of your VPC's route tables that points to the IP address range of the peer VPC.
- The owner of the peer VPC adds a route to one of their VPC's route tables that points to the IP address range of your VPC.
- You may also need to update the security group rules that are associated with your instance to ensure that traffic to and from the peer VPC is not restricted.

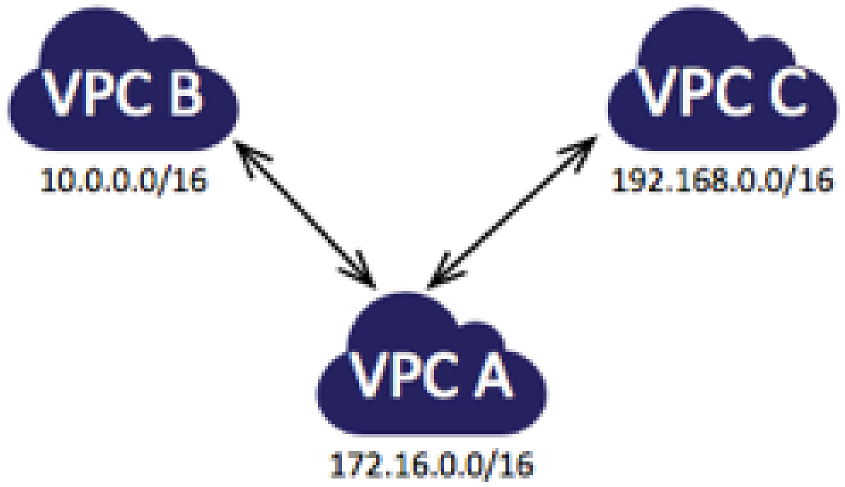
VPC Peering Basics



Destination	Target
VPC1	Local
VPC2	Peering

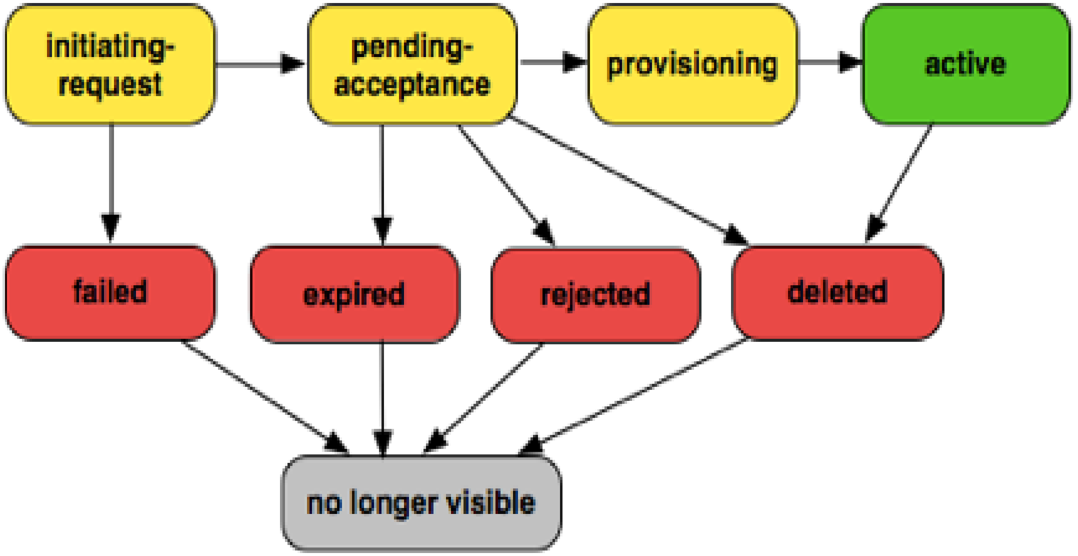
Destination	Target
VPC2	Local
VPC1	Peering

VPC Peering Basics



- A VPC peering connection is a one to one relationship between two VPCs.
- You can create multiple VPC peering connections for each VPC that you own, but transitive peering relationships are not supported
- you will not have any peering relationship with VPCs that your VPC is not directly peered with.
- example of one VPC peered to two different VPCs. There are two VPC peering connections: VPC A is peered with both VPC B and VPC C. VPC B and VPC C are not peered, and you cannot use VPC A as a transit point for peering between VPC B and VPC C. If you want to enable routing of traffic between VPC B and VPC C, you must create a unique VPC peering connection between them.

VPC Peering Connection Lifecycle



- A VPC peering connection goes through various stages starting from when the request is initiated. At each stage, there may be actions that you can take, and at the end of its lifecycle, the VPC peering connection remains visible in the VPC console and API

VPC Peering Connection Lifecycle

- **Initiating-request**: A request for a VPC peering connection has been initiated. At this stage, the peering connection may fail or may go to `pending-acceptance`.
- **Failed**: The request for the VPC peering connection has failed. During this state, it cannot be accepted or rejected. The failed VPC peering connection remains visible to the requester for 2 hours.
- **Pending-acceptance**: The VPC peering connection request is awaiting acceptance from the owner of the peer VPC. During this state, the owner of the requester VPC can delete the request, and the owner of the peer VPC can accept or reject the request. If no action is taken on the request, it will expire after 7 days
- **Expired**: The VPC peering connection request has expired, and no action can be taken on it by either VPC owner. The expired VPC peering connection remains visible to both VPC owners for 2 days.

VPC Peering Connection Lifecycle

•**Rejected**: The owner of the peer VPC has rejected a `pending-acceptance` VPC peering connection request. During this state, the request cannot be accepted. The rejected VPC peering connection remains visible to the owner of the requester VPC for 2 days, and visible to the owner of the peer VPC for 2 hours. If the request was created within the same AWS account, the rejected request remains visible for 2 hours.

•**Provisioning**: The VPC peering connection request has been accepted, and will soon be in the `active` state.

•**Active**: The VPC peering connection is active. During this state, either of the VPC owners can delete the VPC peering connection, but cannot reject it.

•**Deleted**: An `active` VPC peering connection has been deleted by either of the VPC owners, or a `pending-acceptance` VPC peering connection request has been deleted by the owner of the requester VPC. During this state, the VPC peering connection cannot be accepted or rejected. The VPC peering connection remains visible to the party that deleted it for 2 hours, and visible to the other party for 2 days. If the VPC peering connection was created within the same AWS account, the deleted request remains visible for 2 hours.

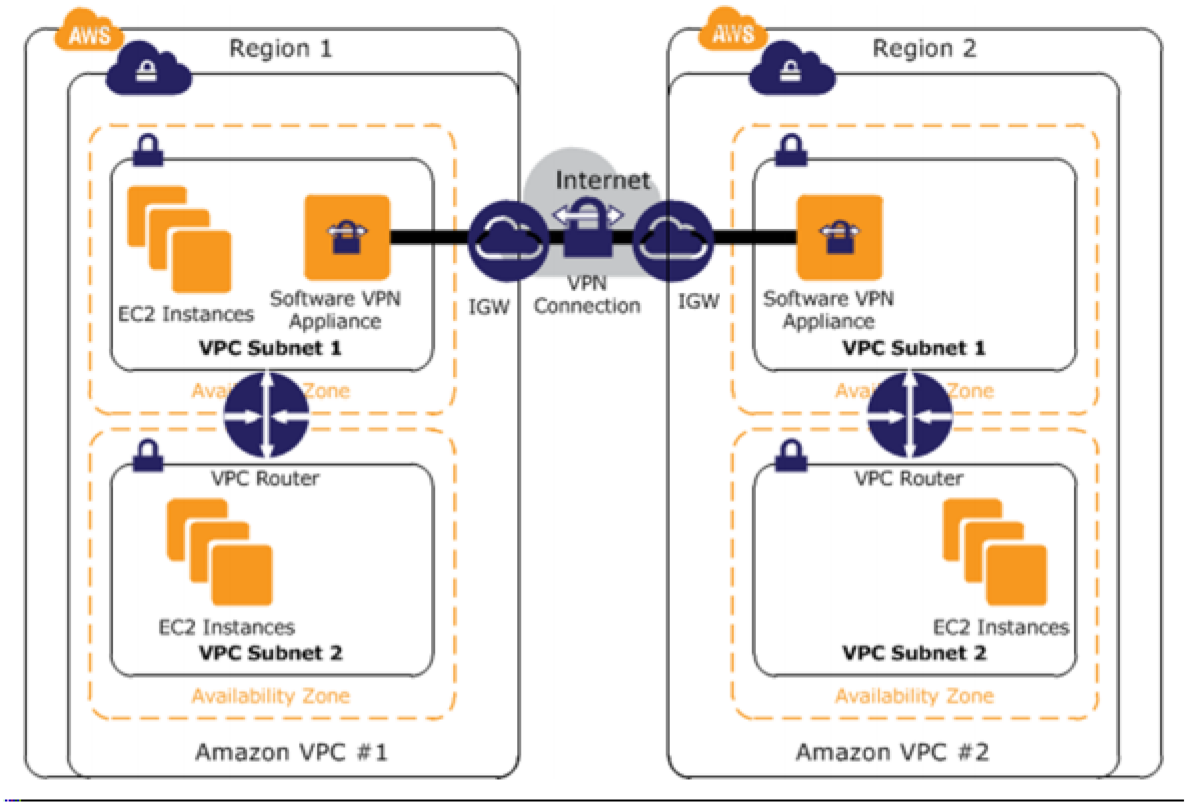
VPC Peering Limitations

- To create a VPC peering connection with another VPC, you need to be aware of the following limitations and rules:
  - You cannot create a VPC peering connection between VPCs that have matching or overlapping CIDR blocks.
  - You cannot create a VPC peering connection between VPCs in different regions.
  - You have a limit on the number active and pending VPC peering connections that you can have per VPC. For more information about VPC limits, see Amazon VPC Limits.
  - VPC peering does not support transitive peering relationships; in a VPC peering connection, your VPC will not have access to any other VPCs that the peer VPC may be peered with. This includes VPC peering connections that are established entirely within your own AWS account.
  - You cannot have more than one VPC peering connection between the same two VPCs at the same time.

VPC Peering Limitations

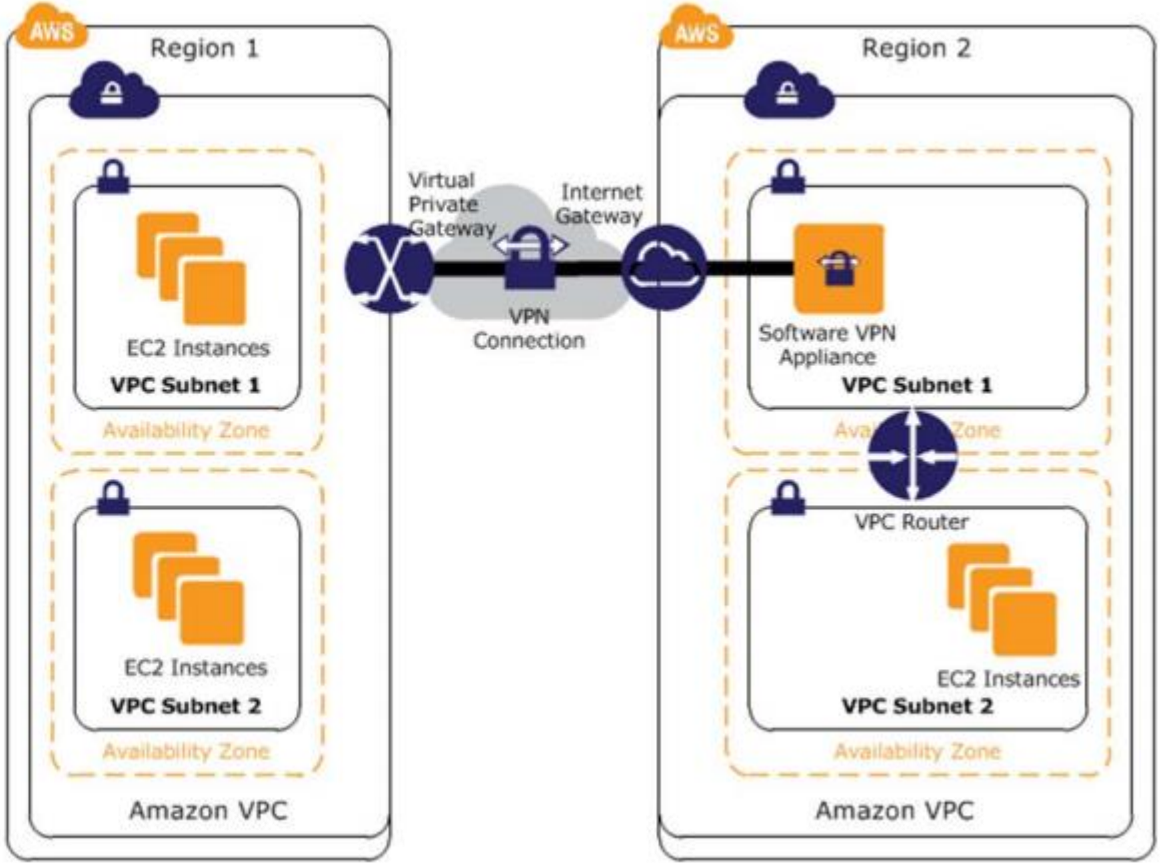
- The Maximum Transmission Unit (MTU) across a VPC peering connection is 1500 bytes.
- A placement group can span peered VPCs; however, you will not get full-bisection bandwidth between instances in peered VPCs. For more information about placement groups, see Placement Groups in the *Amazon EC2 User Guide for Linux Instances*.
- Unicast reverse path forwarding in VPC peering connections is not supported. For more information, see Routing for Response Traffic in the *Amazon VPC Peering Guide*.
- You cannot reference a security group from the peer VPC as a source or destination for ingress or egress rules in your security group. Instead, reference CIDR blocks of the peer VPC as the source or destination of your security group's ingress or egress rules.
- An instance's public DNS hostname will not resolve to its private IP address across peered VPCs.

Software VPN



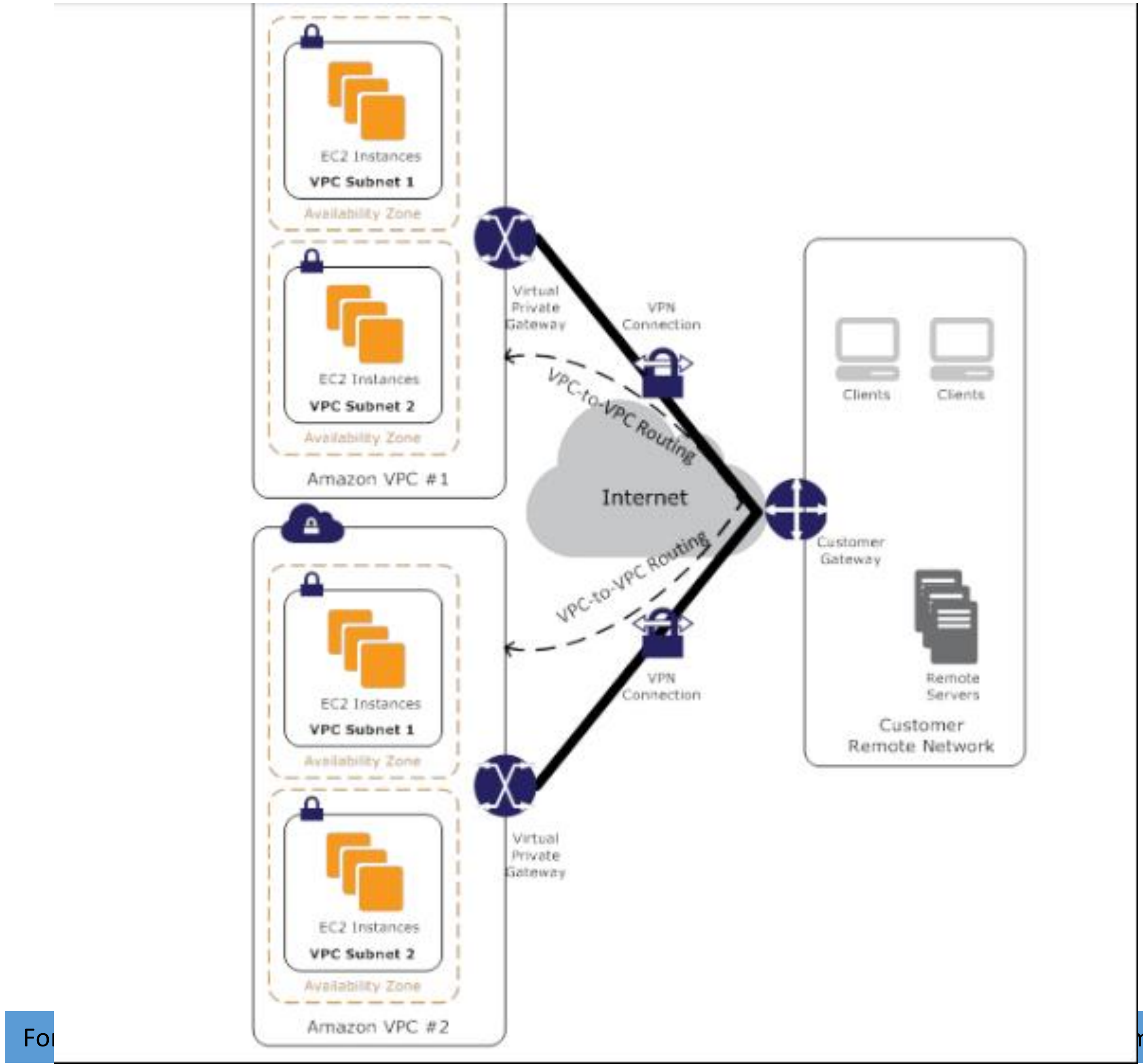
- Amazon VPC provides network routing flexibility.
- This includes the ability to create secure VPN tunnels between two or more software VPN appliances to connect multiple VPCs into a larger virtual private network so that instances in each VPC can seamlessly connect to each other using private IP addresses.
- This option is recommended when you want to connect VPCs across multiple AWS regions and manage both ends of the VPN connection using your preferred VPN software provider
- This option uses an Internet gateway attached to each VPC to facilitate communication between the software VPN appliances

Software-to-Hardware VPN



- Amazon VPC provides the flexibility to combine the hardware VPN and software VPN options to connect multiple VPCs.
- With this design, you can create secure VPN tunnels between a software VPN appliance and a virtual private gateway to connect multiple VPCs into a larger virtual private network, allowing instances in each VPC to seamlessly connect to each other using private IP addresses.
- This option is recommended when you want to connect VPCs across multiple AWS regions and would like to take advantage of the AWS-managed hardware VPN endpoint including automated multidata center redundancy and failover built into the VGW side of the VPN connection.
- This option uses a virtual private gateway in one Amazon VPC and a combination of an Internet gateway and software VPN appliance in another Amazon VPC

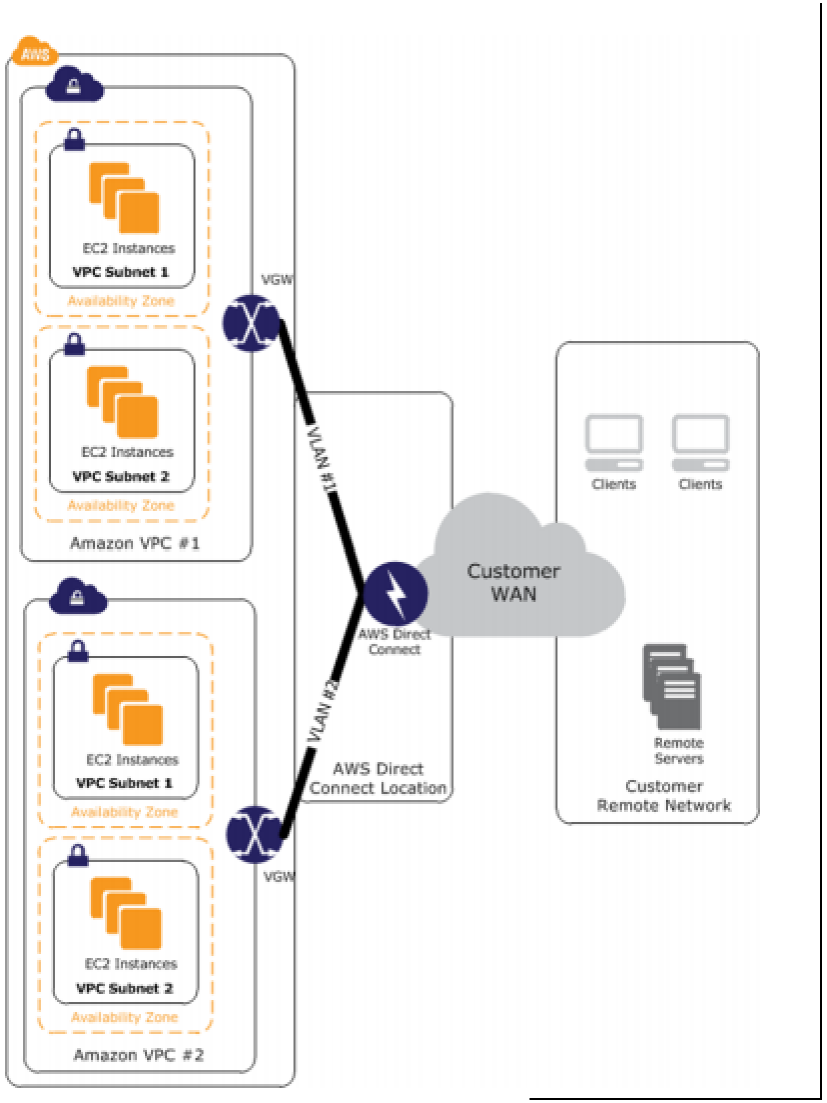
Software-to-Hardware VPN



- Amazon VPC provides the option of creating a hardware IPsec VPN to connect your remote networks with your Amazon VPCs over the Internet. You can leverage multiple hardware VPN connections to route traffic between your Amazon VPCs
- We recommend this approach when you want to take advantage of AWS-managed VPN endpoints including the automated multidata center redundancy and failover built into the AWS side of each VPN connection.
- the Amazon VGW represents two distinct VPN endpoints, physically located in separate data centers to increase the availability of each VPN connection.
- Amazon VGW also supports multiple customer gateway connections allowing you to implement redundancy and failover on your side of the VPN connection
- This solution can also leverage BGP peering to exchange routing information between AWS and these remote endpoints. You can specify routing priorities, policies, and weights (metrics) in your BGP advertisements to influence the network path traffic will take to and from your network(s) and AWS

- This solution can also leverage BGP peering to exchange routing information between AWS and these remote endpoints. You can specify routing priorities, policies, and weights (metrics) in your BGP advertisements to influence the network path traffic will take to and from your network(s) and AWS

AWS Direct Connect

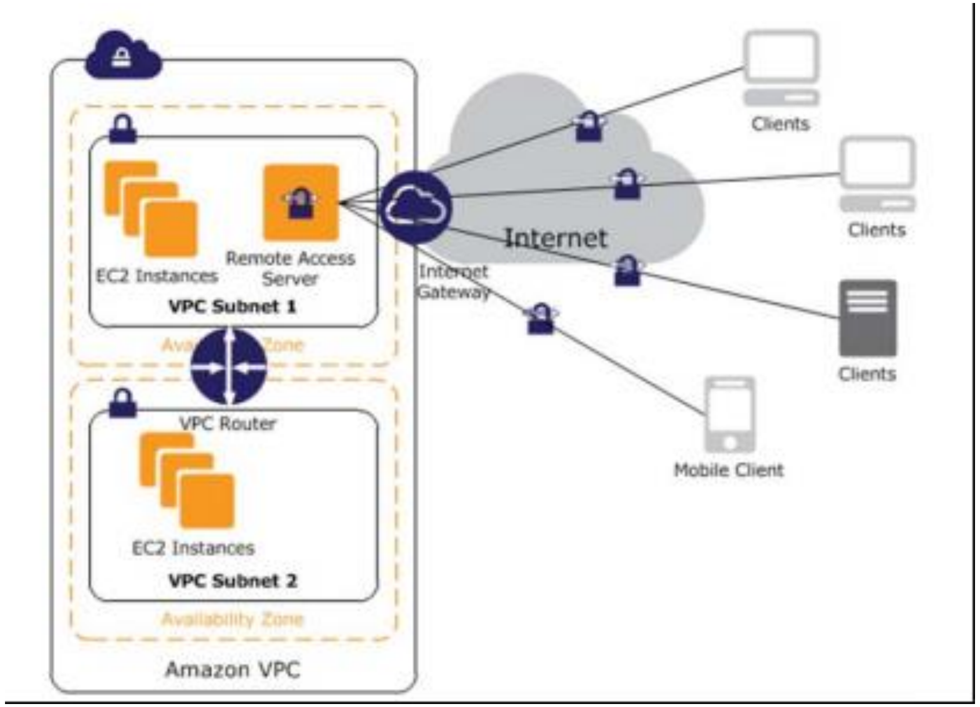


- AWS Direct Connect makes it easy to establish a dedicated network connection from your premises to your Amazon VPC or among Amazon VPCs. This option can potentially reduce network costs, increase bandwidth throughput, and provide a more consistent network experience than the other VPC-to-VPC connectivity options.
- You can divide a physical AWS Direct Connect connection into multiple logical connections, one for each VPC. You can then use these logical connections for routing traffic between VPCs
- you can connect AWS Direct Connect locations in other regions using your existing WAN providers and leverage AWS Direct Connect to route traffic between regions over your WAN backbone network.
- We recommend this approach if you're already an AWS Direct Connect customer or would like to take advantage of AWS Direct Connect's reduced network costs, increased bandwidth throughput, and more consistent network experience
- . AWS Direct Connect can provide very efficient routing since traffic can take advantage of 1 GB or 10 GB fiber connections physically attached to the AWS network in each region. Additionally, this service gives you the most flexibility for controlling and managing routing on your local and remote networks, as well as the potential ability to reuse AWS Direct Connect connections.

Internal User-to-Amazon VPC Connectivity Options

- Internal user access to Amazon VPC resources is typically accomplished either through your network–to-Amazon VPC options or the use of software remote-access VPNs to connect internal users to VPC resources
- With the former option, you can reuse your existing on-premises and remote-access solutions for managing end-user access, while still providing a seamless experience connecting to AWS hosted resources.
- With software remote-access VPN, you can leverage low cost, elastic, and secure Amazon Web Services to implement remote-access solutions while also providing a seamless experience connecting to AWS hosted resources
- In addition, you can combine software remote-access VPNs with your network-to-Amazon VPC options to provide remote access to internal networks if desired. This option is typically preferred by smaller companies with less extensive remote networks or who have not already built and deployed remote access solutions for their employees.

Software Remote-Access VPN



- You can choose from an ecosystem of multiple partners and open source communities that have produced remote-access solutions that run on Amazon EC2.
- Remote-access solutions range in complexity, support multiple client authentication options (including multifactor authentication) and can be integrated with either Amazon VPC or remotely hosted identity and access management solutions (leveraging one of the network-to-Amazon VPC options) like Microsoft Active Directory or other LDAP/multifactor authentication solutions.
- As with the software VPN options, the customer is responsible for managing the remote access software including user management, configuration, patches and upgrades. Additionally, please note that this design introduces a potential single point of failure into the network design as the remote access server runs on a single Amazon EC2 instance

VPC Endpoints

- A VPC endpoint enables you to create a private connection between your VPC and another AWS service without requiring access over the Internet, through a NAT device, a VPN connection, or AWS Direct Connect. Endpoints are virtual devices.
- They are horizontally scaled, redundant, and highly available VPC components that allow communication between instances in your VPC and AWS services without imposing availability risks or bandwidth constraints on your network traffic.
- Currently, we support endpoints for connections with Amazon S3 within the same region only.
- An endpoint enables instances in your VPC to use their private IP addresses to communicate with resources in other services.
- Your instances do not require public IP addresses, and you do not need an Internet gateway, a NAT device, or a virtual private gateway in your VPC. You use endpoint policies to control access to resources in other services. Traffic between your VPC and the AWS service does not leave the Amazon network.