# Route Tables

Route tables are the heart of network traffic management within your VPC.

## Advanced Routing Scenarios:

Beyond the basic scenarios of routing traffic to the internet or within the VPC, several advanced routing scenarios require a deeper understanding of route tables:-

### Private Subnets and NAT Gateways/Instances

Private subnets do not have direct access to the internet. To allow instances in private subnets to access the internet (e.g., for software updates), you need to use a NAT Gateway or NAT instance. The route table associated with the private subnet will have a default route (0.0.0.0/0) pointing to the NAT Gateway or NAT instance.

Imagine you have a private subnet with instances running critical applications that need to download security patches. You would create a NAT Gateway in a public subnet and configure the private subnet's route table to route all outbound internet traffic (0.0.0.0/0) to the NAT Gateway.  
For high availability, consider using multiple NAT Gateways in different Availability Zones. However, AWS does not automatically failover between NAT Gateways. You would need to implement custom routing logic (e.g., using Lambda functions and CloudWatch events) to update the route table in case of a NAT Gateway failure.

### Routing Between VPCs using VPC Peering

VPC Peering allows you to connect two VPCs, enabling instances in different VPCs to communicate with each other as if they were in the same network. To enable communication, you need to update the route tables in both VPCs.

Example:

Suppose you have two VPCs: VPC-A (10.0.0.0/16) and VPC-B (10.1.0.0/16). After establishing a VPC Peering connection between them, you need to:

1. In VPC-A's route table, add a route with destination 10.1.0.0/16 and target the VPC Peering connection ID.
2. In VPC-B's route table, add a route with destination 10.0.0.0/16 and target the VPC Peering connection ID.

This configuration allows instances in VPC-A to communicate with instances in VPC-B and vice versa.

Advanced Consideration:

If you have overlapping CIDR blocks between the VPCs, VPC Peering will not work. You would need to use a NAT instance or a Transit Gateway to facilitate communication between the VPCs.

### Routing to Virtual Private Gateways (VPNs)

If you have a Site-to-Site VPN connection to your on-premises network, you need to configure your route tables to route traffic destined for your on-premises network through the Virtual Private Gateway (VGW).

Example:

Assume your on-premises network has a CIDR block of 192.168.0.0/16, and you have a VGW attached to your VPC. You would add a route to your VPC's route table with destination 192.168.0.0/16 and target the VGW.

Advanced Consideration:

For dynamic routing with BGP (Border Gateway Protocol), the routes to your on-premises network will be automatically propagated to your VPC's route table. However, you might need to adjust the route priorities or use route filters to control the flow of traffic.

### Routing to AWS Direct Connect

Similar to VPNs, if you have an AWS Direct Connect connection, you need to configure your route tables to route traffic destined for your on-premises network through the Direct Connect Gateway (DXGW).

Example:

If your on-premises network has a CIDR block of 172.16.0.0/16, and you have a DXGW attached to your VPC, you would add a route to your VPC's route table with destination 172.16.0.0/16 and target the DXGW.

Advanced Consideration:

Direct Connect offers more bandwidth and lower latency compared to VPNs. You can use both Direct Connect and VPN as a backup for each other. In this case, you would configure your route tables to prefer Direct Connect routes over VPN routes.

### Custom Routes for Specific Traffic

You can create custom routes to direct specific traffic to different destinations based on your requirements.

Example:

Let's say you want to inspect all outbound HTTP traffic (port 80) from a specific subnet using a security appliance. You can create a custom route in the subnet's route table with destination 0.0.0.0/0 and port 80, and target the security appliance's Elastic Network Interface (ENI). All HTTP traffic from that subnet will be routed through the security appliance.

Advanced Consideration:

You can use custom routes to implement traffic mirroring, where you copy network traffic from one instance to another for analysis or debugging purposes.

## Custom Route Creation

Creating custom routes involves specifying the destination and target for the traffic. The destination is the CIDR block or IP address range that you want to route, and the target is the resource to which you want to send the traffic.

### Destination CIDR Blocks

The destination CIDR block specifies the range of IP addresses to which the route applies. You can use specific CIDR blocks (e.g., 10.0.1.0/24) or a default route (0.0.0.0/0) to match all traffic.

Example:

* 1. 0.0.0/0: Matches all IPv4 traffic.
* ::/0: Matches all IPv6 traffic.
* 10.0.1.0/24: Matches traffic destined for the 10.0.1.0/24 subnet.
* 192.168.1.10/32: Matches traffic destined for the specific IP address 192.168.1.10.

### Route Targets

The route target specifies the resource to which you want to send the traffic. The available targets depend on the type of route table and the resources in your VPC.

Common Route Targets:

* **Internet Gateway (IGW):** For routing traffic to the internet.
* **Virtual Private Gateway (VGW):** For routing traffic to your on-premises network via a VPN.
* **NAT Gateway:** For routing traffic from private subnets to the internet.
* **Instance:** For routing traffic to a specific EC2 instance.
* **Network Interface:** For routing traffic to a specific network interface.
* **VPC Peering Connection:** For routing traffic to another VPC via VPC Peering.
* **Transit Gateway:** For routing traffic to a Transit Gateway.
* **Gateway Load Balancer Endpoint:** For routing traffic to a Gateway Load Balancer.
* **AWS PrivateLink Endpoint:** For routing traffic to AWS services privately.

### Route Precedence

When multiple routes match a destination IP address, the most specific route takes precedence. This is based on the longest prefix match.

Example:

Consider a route table with the following routes:

* 1. 0.0.0/0 -> IGW

1. 10.0.1.0/24 -> Instance A
2. 10.0.1.5/32 -> Instance B

If traffic is destined for 8.8.8.8, it will match the first route (0.0.0.0/0) and be routed to the IGW. If traffic is destined for 10.0.1.10, it will match the second route (10.0.1.0/24) and be routed to Instance A. If traffic is destined for 10.0.1.5, it will match the third route (10.0.1.5/32) and be routed to Instance B.

### Route Propagation

Route propagation allows you to automatically propagate routes from a VGW or DXGW to your route tables. This simplifies the configuration of routing for VPN and Direct Connect connections.

Example:

If you enable route propagation on a route table associated with a subnet connected to a VGW, any routes advertised by your on-premises network via BGP will be automatically added to the route table.