**Subnets**

**Default subnets** – a default subnet is a public subnet, because the main route table sends the subnet's traffic that is destined for the internet to the internet gateway.

Instances that you launch into a default subnet receive both a public IPv4 address and a private IPv4 address, and both public and private DNS hostnames.

**Non-Default subnets –** Instances that you launch into a nondefault subnet in a default VPC don't receive a public IPv4 address or a DNS hostname.

You can delete default subnet and VPC but then you must create non-default subnet and VPC to provison nodes. OR you can again create default ones.

## **Creating a Default Subnet -** When you create a default subnet in an Availability Zone that does not have one, it is created with a size /20 IPv4 CIDR block in the next available contiguous space in your default VPC. The following rules apply:

* You cannot specify the CIDR block yourself.
* You cannot restore a previous default subnet that you deleted.
* You can have only one default subnet per Availability Zone.
* You cannot create a default subnet in a nondefault VPC.

# **IP Addressing in Your VPC**

Your VPC can operate in dual-stack mode: your resources can communicate over IPv4, or IPv6, or both. IPv4 and IPv6 addresses are independent of each other; you must configure routing and security in your VPC separately for IPv4 and IPv6.

A private IP address remains associated with the network interface when the instance is stopped and restarted, and is released when the instance is terminated.

# **Security Groups for Your VPC**

Instances associated with a security group can't talk to each other unless you add rules allowing it (exception: the default security group has these rules by default).

## **Default Security Group for Your VPC** - Your VPC automatically comes with a default security group.

The following table describes the default rules for a default security group.

|  |  |  |  |
| --- | --- | --- | --- |
| **Inbound** | | | |
| **Source** | **Protocol** | **Port Range** | **Comments** |
| The security group ID (sg-xxxxxxxx) | All | All | Allow inbound traffic from instances assigned to the same security group. |
| **Outbound** | | | |
| **Destination** | **Protocol** | **Port Range** | **Comments** |
| 0.0.0.0/0 | All | All | Allow all outbound IPv4 traffic. |
| ::/0 | All | All | Allow all outbound IPv6 traffic. This rule is added by default if you create a VPC with an IPv6 CIDR block or if you associate an IPv6 CIDR block with your existing VPC. |

**Non-Default security group**

* When you create a security group, it has no inbound rules. Therefore, no inbound traffic originating from another host to your instance is allowed until you add inbound rules to the security group.
* By default a custom security group includes an outbound rule that allows all outbound traffic.

# **Network ACLs**

## **Network ACL Basics** - The following are the basic things that you need to know about network ACLs:

* Your VPC automatically comes with a modifiable default network ACL. By default, it allows all inbound and outbound IPv4 traffic and, if applicable, IPv6 traffic.
* By default, each custom network ACL denies all inbound and outbound traffic until you add rules.
* In practice, to cover the different types of clients that might initiate traffic to public-facing instances in your VPC, you can open ephemeral ports 1024-65535. However, you can also add rules to the ACL to deny traffic on any malicious ports within that range. Ensure that you place the DENY rules earlier in the table than the ALLOW rules that open the wide range of ephemeral ports.

## **Default Network ACL-** The default network ACL is configured to allow all traffic to flow in and out of the subnets with which it is associated. Each network ACL also 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.

The following is an example default network ACL for a VPC that supports IPv4 only.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Inbound** | | | | | |
| **Rule #** | **Type** | **Protocol** | **Port Range** | **Source** | **Allow/Deny** |
| 100 | All IPv4 traffic | All | All | 0.0.0.0/0 | ALLOW |
| \* | All IPv4 traffic | All | All | 0.0.0.0/0 | DENY |
| **Outbound** | | | | | |
| **Rule #** | **Type** | **Protocol** | **Port Range** | **Destination** | **Allow/Deny** |
| 100 | All IPv4 traffic | All | All | 0.0.0.0/0 | ALLOW |
| \* | All IPv4 traffic | All | All | 0.0.0.0/0 | DENY |

# **ROUTES TABLE**

### Route Tables for Transit Gateways - When you attach a VPC to a transit gateway, you need to add a route for traffic to route through the transit gateway.

Consider the following scenario where you have three VPCs that are attached to a transit gateway. In this scenario, all attachments are associated with the transit gateway route table and propagate to the transit gateway route table. Therefore, all attachments can route packets to each other, with the transit gateway serving as a simple layer 3 IP hub.

# **Internet Gateways**

An internet gateway serves two purposes: to provide a target in your VPC route tables for internet-routable traffic, and to perform network address translation (NAT) for instances that have been assigned public IPv4 addresses.

## **Enabling Internet Access** - To enable access to or from the internet for instances in a VPC subnet, you must do the following:

* Attach an internet gateway to your VPC.
* Ensure that your subnet's route table points to the internet gateway.
* Ensure that instances in your subnet have a globally unique IP address (public IPv4 address, Elastic IP address, or IPv6 address).
* Ensure that your network access control and security group rules allow the relevant traffic to flow to and from your instance.

**Internet Access for Default and Nondefault VPCs** - The following table provides an overview of whether your VPC automatically comes with the components required for internet access over IPv4 or IPv6.

|  |  |  |
| --- | --- | --- |
| **Component** | **Default VPC** | **Nondefault VPC** |
| Internet gateway | Yes | Yes, if you created the VPC using the first or second option in the VPC wizard. Otherwise, you must manually create and attach the internet gateway. |
| Route table with route to internet gateway for IPv4 traffic (0.0.0.0/0) | Yes | Yes, if you created the VPC using the first or second option in the VPC wizard. Otherwise, you must manually create the route table and add the route. |
| Route table with route to internet gateway for IPv6 traffic (::/0) | No | Yes, if you created the VPC using the first or second option in the VPC wizard, and if you specified the option to associate an IPv6 CIDR block with the VPC. Otherwise, you must manually create the route table and add the route. |
| Public IPv4 address automatically assigned to instance launched into subnet | Yes (default subnet) | No (nondefault subnet) |
| IPv6 address automatically assigned to instance launched into subnet | No (default subnet) | No (nondefault subnet) |

# **NAT GATEWAY**

**Note for NAT / NAT Gateways -** If you have resources in multiple Availability Zones and they share one NAT gateway, in the event that the NAT gateway’s Availability Zone is down, resources in the other Availability Zones lose internet access. To create an Availability Zone-independent architecture, create a NAT gateway in each Availability Zone and configure your routing to ensure that resources use the NAT gateway in the same Availability Zone.

If you no longer need a NAT gateway, you can delete it. Deleting a NAT gateway disassociates its Elastic IP address, but does not release the address from your account.

### NAT Gateway Rules and Limitations - A NAT gateway has the following characteristics and limitations:

* A NAT gateway supports 5 Gbps of bandwidth and automatically scales up to 45 Gbps. If you require more, 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. You cannot disassociate an Elastic IP address from a NAT gateway after it's created. 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.
* 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. A NAT gateway uses ports 1024–65535. For more information, see [Network ACLs](https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html).
* When a NAT gateway is created, it receives a 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. For more information, see [Viewing Details about a Network Interface](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html#view_eni_details). You cannot modify the attributes of this network interface.
* A NAT gateway cannot be accessed by a ClassicLink connection associated with your VPC.
* A NAT gateway can support up to 55,000 simultaneous connections to each unique destination. This limit also applies if you create approximately 900 connections per second to a single destination (about 55,000 connections per minute). If the destination IP address, the destination port, or the protocol (TCP/UDP/ICMP) changes, you can create an additional 55,000 connections. For more than 55,000 connections, there is an increased chance of connection errors due to port allocation errors. These errors can be monitored by viewing the ErrorPortAllocation CloudWatch metric for your NAT gateway.

### Best Practice When Sending Traffic to Amazon S3 or DynamoDB in the Same Region

To avoid data processing charges for NAT gateways when accessing Amazon S3 and DynamoDB that are in the same Region, set up a gateway endpoint and route the traffic through the gateway endpoint instead of the NAT gateway. There are no charges for using a gateway endpoint.

**Disabling Source/Destination Checks (NAT INSTANCE)**

Each EC2 instance performs source/destination checks by default. This means that the instance must be the source or destination of any traffic it sends or receives. However, a NAT instance must be able to send and receive traffic when the source or destination is not itself. Therefore, you must disable source/destination checks on the NAT instance.

**DNS**

## **DNS Support in Your VPC -** Your VPC has attributes that determine whether instances launched in the VPC receive public DNS hostnames that correspond to their public IP addresses, and whether DNS resolution through the Amazon DNS server is supported for the VPC.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| enableDnsHostnames | Indicates whether instances with public IP addresses get corresponding public DNS hostnames.  If this attribute is true, instances in the VPC get public DNS hostnames, but only if the enableDnsSupport attribute is also set to true. |
| enableDnsSupport | Indicates whether the DNS resolution is supported.  If this attribute is false, the Amazon-provided DNS server 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 IPv4 network range plus two will succeed. For more information, see [Amazon DNS Server](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_DHCP_Options.html#AmazonDNS). |

If both attributes are set to true, the following occurs:

* Instances with a public IP address receive corresponding public DNS hostnames.
* The Amazon-provided DNS server can resolve Amazon-provided private DNS hostnames.

If either or both of the attributes is set to false, the following occurs:

* Instances with a public IP address do not receive corresponding public DNS hostnames.
* The Amazon-provided DNS server cannot resolve Amazon-provided private DNS hostnames.
* Instances receive custom private DNS hostnames if there is a custom domain name in the [DHCP options set](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_DHCP_Options.html). If you are not using the Amazon-provided DNS server, your custom domain name servers must resolve the hostname as appropriate.

By default, both attributes are set to true in a default VPC or a VPC created by the VPC wizard. By default, only the enableDnsSupport attribute is set to true in a VPC created any other way.

**Important -** If you use custom DNS domain names defined in a private hosted zone in Amazon Route 53, or use private DNS with interface VPC endpoints (AWS PrivateLink), you must set the enableDnsHostnames and enableDnsSupport attributes to true.

## **Using Private Hosted Zones -** If you want to access the resources in your VPC using custom DNS domain names, such as example.com, instead of using private IPv4 addresses or AWS-provided private DNS hostnames, you can create a private hosted zone in Route 53. A private hosted zone is a container that holds information about how you want to route traffic for a domain and its subdomains within one or more VPCs without exposing your resources to the Internet. You can then create Route 53 resource record sets, which determine how Route 53 responds to queries for your domain and subdomains. For example, if you want browser requests for example.com to be routed to a web server in your VPC, you'll create an A record in your private hosted zone and specify the IP address of that web server.

To access resources using custom DNS domain names, you must be connected to an instance within your VPC. From your instance, you can test that your resource in your private hosted zone is accessible from its custom DNS name by using the ping command; for example, ping mywebserver.example.com. (You must ensure that your instance's security group rules allow inbound ICMP traffic for the ping command to work.)

**Elastic IP Addresses**

Note that the advantage of associating the Elastic IP address with the network interface instead of directly with the instance is that you can move all the attributes of the network interface from one instance to another in a single step.

We currently do not support Elastic IP addresses for IPv6.

An Elastic IP address is accessed through the Internet gateway of a VPC. If you have set up a AWS Site-to-Site VPN connection between your VPC and your network, the VPN traffic traverses a virtual private gateway, not an Internet gateway, and therefore cannot access the Elastic IP address.

**VPC ENDPOINTS**

**Gateway Endpoint Limitations -** To use gateway endpoints, you need to be aware of the current limitations:

* You cannot use a prefix list ID in an outbound rule in a network ACL to allow or deny outbound traffic to the service specified in an endpoint. If your network ACL rules restrict traffic, you must specify the CIDR block (IP address range) for the service instead. You can, however, use a prefix list ID in an outbound security group rule. For more information, see [Security Groups](https://docs.aws.amazon.com/vpc/latest/userguide/vpc-endpoints-access.html#vpc-endpoints-security-groups).
* Endpoints are supported within the same region only. You cannot create an endpoint between a VPC and a service in a different region.
* Endpoints support IPv4 traffic only.
* You cannot transfer an endpoint from one VPC to another, or from one service to another.
* You have a limit on the number of endpoints you can create per VPC. For more information, see [VPC Endpoints](https://docs.aws.amazon.com/vpc/latest/userguide/amazon-vpc-limits.html#vpc-limits-endpoints).
* Endpoint connections cannot be extended out of a VPC. Resources on the other side of a VPN connection, VPC peering connection, AWS Direct Connect connection, or ClassicLink connection in your VPC cannot use the endpoint to communicate with resources in the endpoint service.
* You must enable DNS resolution in your VPC, or if you're using your own DNS server, ensure that DNS requests to the required service (such as Amazon S3) are resolved correctly to the IP addresses maintained by AWS.

**Unsupported VPC Peering Configurations**

The following VPC peering connection configurations are not supported.

**Invalid Configurations**

* [Overlapping CIDR Blocks](https://docs.aws.amazon.com/vpc/latest/peering/invalid-peering-configurations.html#overlapping-cidr)
* [Transitive Peering](https://docs.aws.amazon.com/vpc/latest/peering/invalid-peering-configurations.html#transitive-peering)
* [Edge to Edge Routing Through a Gateway or Private Connection](https://docs.aws.amazon.com/vpc/latest/peering/invalid-peering-configurations.html#edge-to-edge-vgw)

## **Edge to Edge Routing Through a Gateway or Private Connection**

If either VPC in a peering relationship has one of the following connections, you cannot extend the peering relationship to that connection:

* A VPN connection or an AWS Direct Connect connection to a corporate network
* An internet connection through an internet gateway
* An internet connection in a private subnet through a NAT device
* A VPC endpoint to an AWS service; for example, an endpoint to Amazon S3.
* (IPv6) A ClassicLink connection. You can enable IPv4 communication between a linked EC2-Classic instance and instances in a VPC on the other side of a VPC peering connection. However, IPv6 is not supported in EC2-Classic, so you cannot extend this connection for IPv6 communication.

For example, if VPC A and VPC B are peered, and VPC A has any of these connections, then instances in VPC B cannot use the connection to access resources on the other side of the connection. Similarly, resources on the other side of a connection cannot use the connection to access VPC B.

**Example: Edge to Edge Routing Through a VPN Connection or an AWS Direct Connect Connection**

You have a VPC peering connection between VPC A and VPC B (pcx-aaaabbbb). VPC A also has a VPN connection or an AWS Direct Connect connection to a corporate network. Edge to edge routing is not supported; you cannot use VPC A to extend the peering relationship to exist between VPC B and the corporate network. For example, traffic from the corporate network can’t directly access VPC B by using the VPN connection or the AWS Direct Connect connection to VPC A.


                Edge to edge routing through a VPN
            

**Example: Edge to Edge Routing Through an InternetGateway**

You have a VPC peering connection between VPC A and VPC B (pcx-abababab). VPC A has an internet gateway; VPC B does not. Edge to edge routing is not supported; you cannot use VPC A to extend the peering relationship to exist between VPC B and the internet. For example, traffic from the internet can’t directly access VPC B by using the internet gateway connection to VPC A.


                Edge to edge routing through an internet gateway
            

Similarly, if VPC A has a NAT device that provides internet access to instances in private subnets in VPC A, instances in VPC B cannot use the NAT device to access the internet.

**Example: Edge to Edge Routing Through a VPC Endpoint**

You have a VPC peering connection between VPC A and VPC B (pcx-aaaabbbb). VPC A has a VPC endpoint that connects it to Amazon S3. Edge to edge routing is not supported; you cannot use VPC A to extend the peering relationship to exist between VPC B and Amazon S3. For example, VPC B can't directly access Amazon S3 using the VPC endpoint connection to VPC A.


       Edge to edge routing through a VPC endpoint
      

# **What is AWS Site-to-Site VPN?**

By default, instances that you launch into an Amazon VPC can't communicate with your own (remote) network. You can enable access to your remote network from your VPC by attaching a virtual private gateway to the VPC, creating a custom route table, updating your security group rules, creating an AWS Site-to-Site VPN (Site-to-Site VPN) connection, and configuring routing to pass traffic through the connection.

Although the term *VPN connection* is a general term, in this documentation, a VPN connection refers to the connection between your VPC and your own on-premises network. Site-to-Site VPN supports Internet Protocol security (IPsec) VPN connections.

**Components of Your Site-to-Site VPN -** A Site-to-Site VPN connection offers two VPN tunnels between a virtual private gateway or transit gateway on the AWS side and a customer gateway on the remote (customer) side.

**Virtual Private Gateway -** A *virtual private gateway* is the VPN concentrator on the Amazon side of the Site-to-Site VPN connection. You create a virtual private gateway and attach it to the VPC from which you want to create the Site-to-Site VPN connection.

**AWS Transit Gateway -** You can modify the target gateway of AWS Site-to-Site VPN connection from a virtual private gateway to a transit gateway. A transit gateway is a transit hub that you can use to interconnect your virtual private clouds (VPC) and on-premises networks.

**Customer Gateway -** A *customer gateway* resource in AWS, which provides information to AWS about your [Customer Gateway Device](https://docs.aws.amazon.com/vpn/latest/s2svpn/how_it_works.html#CustomerGatewayDevice).

**Customer Gateway Device -** A *customer gateway device* is a physical device or software application on your side of the Site-to-Site VPN connection.

# **Site-to-Site VPN Single and Multiple Connection Examples**

## **Single Site-to-Site VPN Connection**

The VPC has an attached virtual private gateway, and your remote network includes a customer gateway, You set up the routing so that any traffic from the VPC bound for your network is routed to the virtual private gateway.


        VPN layout
      

## **Single Site-to-Site VPN Connection with a Transit Gateway**

The VPC has an attached transit gateway, and your remote network includes a customer gateway, You set up the routing so that any traffic from the VPC bound for your network is routed to the transit gateway.


        Single Site-to-Site VPN Connection with a Transit Gateway
      

## **Multiple Site-to-Site VPN Connections**

When you create multiple Site-to-Site VPN connections to a single VPC, you can configure a second customer gateway to create a redundant connection to the same external location. You can also use it to create Site-to-Site VPN connections to multiple geographic locations.


        Multiple Site-to-Site VPN layout
      

## **Multiple Site-to-Site VPN Connections with a Transit Gateway**

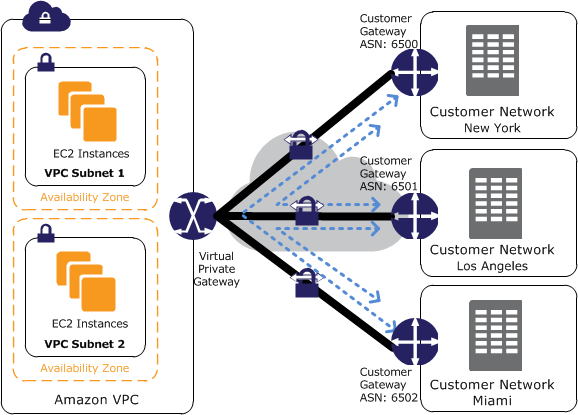
When you create multiple Site-to-Site VPN connections to a single VPC, you can configure a second customer gateway to create a redundant connection to the same external location. You can also use it to create Site-to-Site VPN connections to multiple geographic locations.


        Multiple Site-to-Site VPN connections with a Transit Gateway
      

# **Providing Secure Communication Between Sites Using VPN CloudHub**

If you have multiple AWS Site-to-Site VPN connections, you can provide secure communication between sites using the AWS VPN CloudHub. This enables your remote sites to communicate with each other, and not just with the VPC. The VPN CloudHub operates on a simple hub-and-spoke model that you can use with or without a VPC. This design is suitable for customers with multiple branch offices and existing internet connections who'd like to implement a convenient, potentially low-cost hub-and-spoke model for primary or backup connectivity between these remote offices.

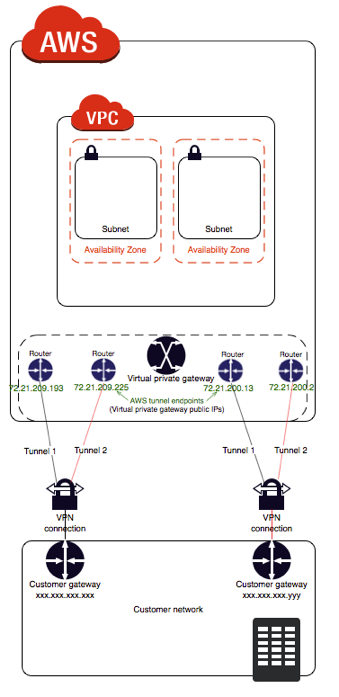
The following diagram shows the VPN CloudHub architecture, with blue dashed lines indicating network traffic between remote sites being routed over their Site-to-Site VPN connections.



# **Using Redundant Site-to-Site VPN Connections to Provide Failover**

As described earlier, a Site-to-Site VPN connection has two tunnels to help ensure connectivity in case one of the Site-to-Site VPN connections becomes unavailable. To protect against a loss of connectivity in case your customer gateway becomes unavailable, you can set up a second Site-to-Site VPN connection to your VPC and virtual private gateway by using a second customer gateway. To establish redundant Site-to-Site VPN connections and customer gateways on your remote network, you need to set up a second Site-to-Site VPN connection. The customer gateway IP address for the second Site-to-Site VPN connection must be publicly accessible.

The following diagram shows the two tunnels of each Site-to-Site VPN connection and two customer gateways.



# **What is a Transit Gateway?**

A transit gateway is a network transit hub that you can use to interconnect your virtual private clouds (VPC) and on-premises networks.

## Transit Gateway Concepts

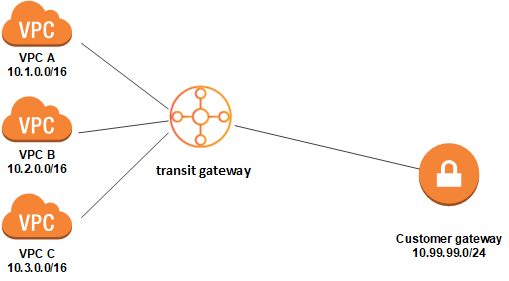
The following are the key concepts for transit gateways:

* **attachment —** You can attach a VPC, an AWS Direct Connect gateway, or a VPN connection to a transit gateway.
* **transit gateway route table** — A transit gateway has a default route table and can optionally have additional route tables. A route table includes dynamic and static routes that decide the next hop based on the destination IP address of the packet. The target of these routes could be a VPC or a VPN connection. By default, the VPCs and VPN connections that you attach to a transit gateway are associated with the default transit gateway route table.
* **associations** — Each attachment is associated with exactly one route table. Each route table can be associated with zero to many attachments.
* **route propagation** — A VPC or VPN connection can dynamically propagate routes to a transit gateway route table. With a VPC, you must create static routes to send traffic to the transit gateway. With a VPN connection, routes are propagated from the transit gateway to your on-premises router using Border Gateway Protocol (BGP).

# **Transit Gateway Scenario: Centralized Router**

You can configure your transit gateway as a centralized router that connects all of your VPCs, AWS Direct Connect, and AWS Site-to-Site VPN connections. In this scenario, all attachments are associated with the transit gateway default route table and propagate to the transit gateway default route table. Therefore, all attachments can route packets to each other, with the transit gateway serving as a simple layer 3 IP router.

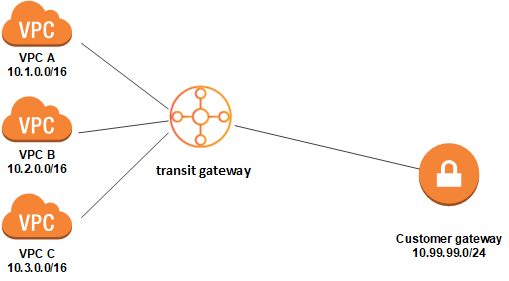
## **Overview** - Packets from the subnets in VPC, A, VPC B, and VPC C that have the internet as a destination, route first through the transit gateway and then route to the VPN. Packets from one VPC that have a destination of a subnet in another VPC, for example from 10.1.0.0 to 10.2.0.0, route through the transit gateway.



# **Transit Gateway Scenario: Isolated VPCs**

You can configure your transit gateway as multiple isolated routers. This is similar to using multiple transit gateways, but provides more flexibility in cases where the routes and attachments might change. In this scenario, each isolated router has a single route table. All attachments associated with an isolated router propagate and associate with its route table. Attachments associated with one isolated router can route packets to each other, but cannot route packets to or receive packets from the attachments for another isolated router.

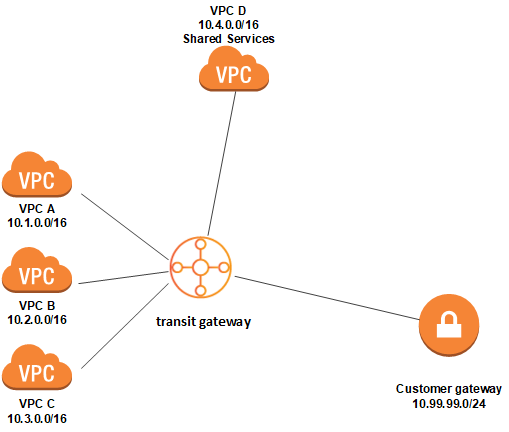
## **Overview** - The following diagram shows the key components of the configuration for this scenario. Packets from VPC A, VPC B, and VPC C route to the transit gateway. Packets from the subnets in VPC, A, VPC B, and VPC C that have the internet as a destination, route first through the transit gateway and then route to the Site-to-Site VPN. Packets from one VPC that have a destination of a subnet in another VPC, for example from 10.1.0.0 to 10.2.0.0, route through the transit gateway, where they are blocked because there is no route for them in the transit gateway route table.



# **Transit Gateway Scenario: Isolated VPCs with Shared Services**

You can configure your transit gateway as multiple isolated routers that use a shared service this is similar to using multiple transit gateways, but provides more flexibility in cases where the routes and attachments might change. In this scenario, each isolated router has a single route table. All attachments associated with an isolated router propagate and associate with its route table. Attachments associated with one isolated router can route packets to each other, but cannot route packets to or receive packets from the attachments for another isolated router. Attachments can route packets to or receive packets from the shared services. You can use this scenario when you have groups that need to be isolated, but use a shared service, for example a production system.

**Overview** - The following diagram shows the key components of the configuration for this scenario. Packets from VPC A, VPC B, and VPC C route to the transit gateway. Packets from the subnets in VPC, A, VPC B, and VPC C that have the internet as a destination, route first through the transit gateway and then route to the Site-to-Site VPN. Packets from one VPC that have a destination of a subnet in another VPC, for example from 10.1.0.0 to 10.2.0.0, route through the transit gateway, where they are blocked because there is no route for them in the transit gateway route table. Packets from VPC A, VPC B, and VPC C that have VPC D as the destination route through the transit gateway.



## **AWS VPN CloudHub and Redundant Customer Gateways**

You can establish multiple VPN connections to a single virtual private gateway from multiple customer gateways. This configuration can be used in different ways.

* You can have redundant customer gateways between your data center and your VPC,
* or you can have multiple locations connected to the AWS VPN CloudHub.

If you have redundant customer gateways, If one customer gateway fails, the virtual private gateway directs all traffic to the working customer gateway.

If you use the AWS VPN CloudHub configuration, multiple sites can access your VPC or securely access each other using a simple hub-and-spoke model.

## **Configuring Multiple VPN Connections to Your VPC** - You can create up to ten VPN connections for your VPC. You can use multiple VPN connections to link your remote offices to the same VPC. For example, if you have offices in Los Angeles, Chicago, New York, and Miami, you can link each of these offices to your VPC. You can also use multiple VPN connections to establish redundant customer gateways from a single location.

If you need more than ten VPN connections, complete the [Request to Increase Amazon VPC Limits](http://aws.amazon.com/contact-us/vpc-request/) form to request an increased limit.

The following diagram shows the configuration of multiple VPNs.


     Multiple VPN layout
    