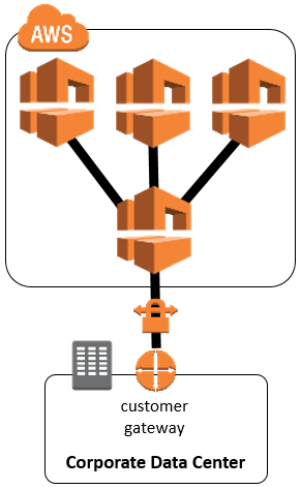
MULTIPLE-VPC VPN CONNECTION SHARING *"How do I share a single VPN connection with multiple VPCs?"*

Overview

Amazon Virtual Private Cloud (Amazon VPC) provides customers with the ability to create as many virtual networks as they need. However, when connecting on-premises infrastructure to these networks, they must determine how many remote network connections they need to create. Some AWS customers establish a single shared connection for multiple VPCs to minimize the number of remote connections they need to configure. Configurations like these can be beneficial because they save time and effort, and simplify network management. For example, each connection to AWS requires infrastructure changes to on-premises production network equipment that, for some customers, can involve lengthy change-approval processes that take weeks or months to implement. In these situations, it is often preferable to reduce the number of network connections to establish in order to free up development teams for more innovative projects that take full advantage of the flexibility and capabilities of the AWS platform. Additionally, the proliferation of IT-as-a-Service offerings requires companies to manage an ever-growing number of remote connections to their technology partners' networks, which can be daunting and complicated.

When connecting multiple VPCs to on-premises networks, AWS recommends leveraging AWS Direct Connect because a single physical connection can map logical virtual interfaces to hundreds of VPCs.[1] For VPN-based customers, AWS recommends creating a separate VPN connection for each customer VPC. However some customers would like to share a single VPN configuration for the reasons mentioned above or as an interim bridge for several existing VPCs while they provision AWS Direct Connect. In the latter situation, after an AWS Direct Connect connection is established, you can easily migrate traffic from VPN connections to AWS Direct Connect virtual interfaces by simply changing route advertisements to your spoke VPCs.[2]



The following sections address key considerations and recommendations for connecting multiple VPCs to an on-premises network using a single VPN connection,[1] and assume basic knowledge of highly available remote-network connectivity,[2] IPsec VPNs, network addressing, subnetting, and routing. The solutions in this document assume a typical hub-and-spoke network topology where remote VPCs access a corporate network over a VPN connection established in a shared VPC, as depicted in the diagram to the right.

General Best Practices

When configuring VPN connections to any computer network, there are some universal network-design principles to consider. For example, whenever possible, limit the amount of traffic that must traverse VPN connections. This will reduce VPN network contention and latency, which can improve application performance. It is also best to implement non-overlapping network ranges for your private networks to simplify the ability to route between remote networks. Finally, use dynamically routed network connections to create highly available, resilient, more scalable links to resources in your corporate network. With this in mind, consider the following AWS remote-connectivity best practices:

- Replicate latency-sensitive or critical shared services to AWS to help improve application performance and reduce application dependencies.
- Ensure that your VPC network ranges (CIDR blocks) do not overlap one another or other private network ranges.
- Leverage multiple dynamically routed, rather than statically routed, VPN connections to AWS. This will allow your network infrastructure to automatically failover between available VPN connections as necessary.

[1] For more information about connecting multiple VPCs to an on-premises network using a physical AWS Direct Connect connection, please see the AWS Direct Connect product documentation: http://docs.aws.amazon.com/directconnect/latest/UserGuide/WorkingWithVirtualInterfaces.html

[2] See the *Resources* section for relevant Solution Briefs.

Application on the AWS Platform

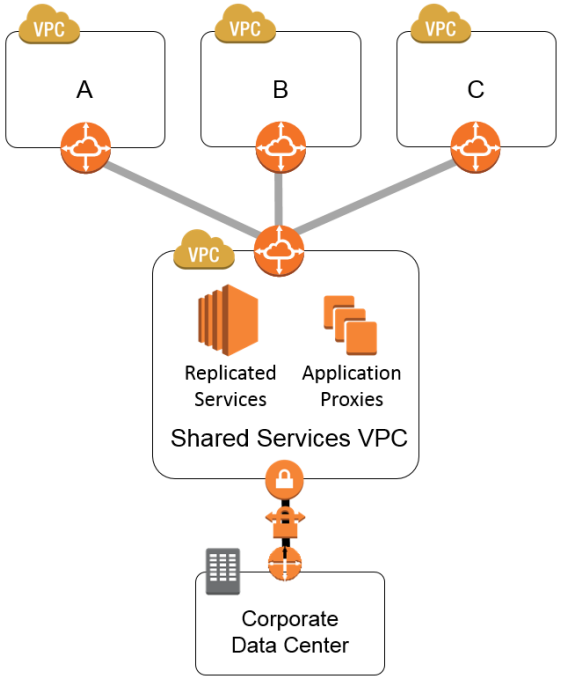
There are two distinct approaches that AWS customers typically take to limit network complexity when accessing on-premises networks through a single VPN connection. The first approach proxies all on-premises network requests through a shared services VPC and the second approach creates a dedicated transit VPC to directly route multi-VPC traffic over a VPN connection. The following sections discuss these two approaches, and also a combined approach for customers who need the flexibility of both designs.

Shared Services VPC

This approach creates a shared services VPC which contains replicated services, and also application proxies for requests to remote resources that cannot be directly replicated as a shared service. This approach eliminates the need to create VPN connections for additional VPCs because all required on-premises resources will be accessed either directly or indirectly through the shared services VPC.

This option is best suited for customers with the following use case/requirements:

- The majority of their infrastructure is (or will be) on AWS
- The required on-premises resources are easy to replicate or proxy (e.g., Active Directory)
- They prefer to limit VPN traffic
- Strong security or compliance programs require additional application-level controls and proxy servers between their AWS and on-premises resources (e.g., application-layer firewalls)



Configuration Details

This design pattern connects multiple spoke VPCs to a shared services VPC in the same region using VPC peering, and provides access to remote resources in these VPCs through replicated services and application proxy services. The shared services VPC, in turn, connects to on-premises resources using a standard, dynamically routed AWS VPN connection established with a VGW. Commonly replicated services include infrastructure services such as Active Directory, DNS, and load-balancing services, but they can also include application services such as database replicas or developer source-control, build, and deployment tools. Common application proxies include standard web-reverse proxies such as HAProxy, Nginx, or Apache mod_proxy, but they can also include load balancer virtual IP addresses and SOCKS proxies.

Considerations

Replicated services offer less network latency to spoke VPCs than if these requests had to traverse the VPN. They also reduce the amount of traffic that has to flow over the VPN connection. Application proxies, in addition to facilitating access to on-premises resources, can also provide additional application-level controls and reduce network latency and VPN traffic by caching responses for subsequent requests. For example, application or database firewalls proxy requests while providing additional application-level monitoring or filtering for on-premises resources.

This design requires customers to manage additional replicated infrastructure or proxy server farms on AWS and to configure their applications to use this infrastructure when they need to communicate with on-premises resources. For example, it might be necessary to configure HTTP/S or SOCKS proxy settings, or to direct traffic to load-balancer virtual IP addresses in the shared services VPC rather than to the resource's actual on-premises IP address.

Transit VPC

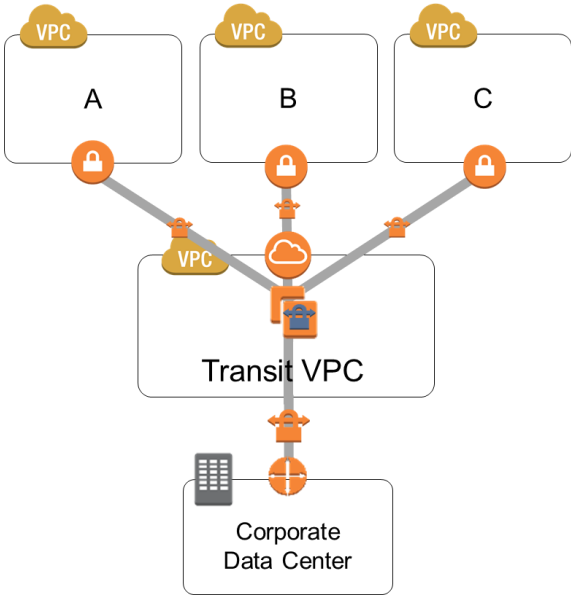
This approach creates a transitive network using host-based VPN appliances on Amazon Elastic Compute Cloud (Amazon EC2) instances in a dedicated VPC to route traffic between multiple VPCs and on-premises resources. AWS highly recommends leveraging virtual network appliances from the AWS Marketplace[3] to significantly reduce the level of effort to establish and maintain these VPN connections.

This option is best suited for customers with the following use case/requirements:

- AWS resources in spoke VPCs need access to a wide variety of on-premises infrastructure
- The required on-premises resources are extremely difficult to replicate or proxy (e.g., proprietary mainframe protocols)
- They are implementing a hybrid architecture with complex network-routing requirements
- Their security or compliance programs require additional network-based monitoring or filtering between AWS and on-premises resources (e.g., Network Intrusion Detection Systems or next-generation firewalls)

Configuration Details

This design pattern creates dynamically routed VPN connections between spoke VPC VGWs and VPN appliances in the transit VPC, and again between these appliances and on-premises network equipment. Note that in the diagram to the right, all communication with the VPN appliances (including the VPN connection between the corporate data center and the transit VPC) uses the transit VPC Internet Gateway and Elastic IP addresses. This design uses VPN connections, rather than VPC peering, to connect to the transit VPC because VPC peering does not support transitive routing. The best practice for making this transit network highly available and scalable is to use dynamically routed VPN connections. Additionally, AWS highly recommends the use of Auto Recovery for EC2 or Auto Scaling for automatic recovery of failed EC2-based VPN instances.



In addition to providing direct network routing between VPCs and on-premises networks, this design also allows the transit VPC to implement more complex routing rules, such as network address translation between overlapping network ranges, or to add additional network-level packet filtering or inspection.

Considerations

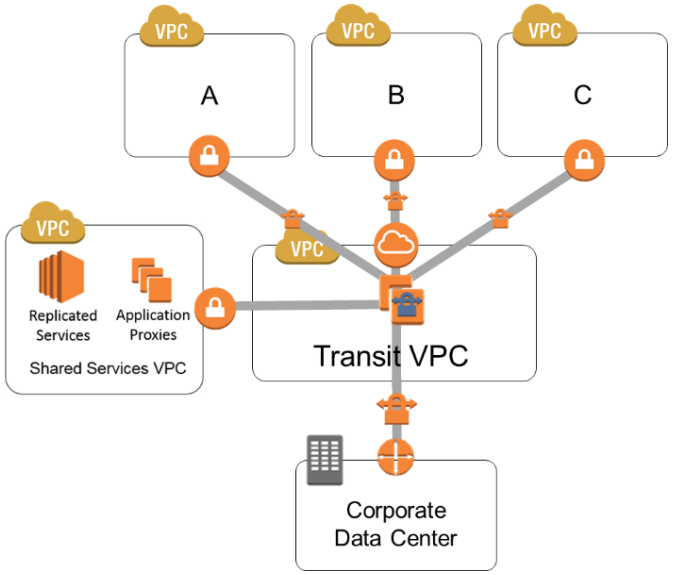
This design supports any IP-based connectivity requirements between Amazon VPCs and remote resources with minimal on-premises network changes. It also provides an opportunity to select products available on the AWS Marketplace that integrate seamlessly with AWS-provided VPN connections, without the need to deploy these products into existing data centers. However, it does require the customer to configure and manage the EC2-based VPN instances deployed in the transit VPC. This will result in additional EC2 and, potentially, third-party license charges. Also, be aware that this design will generate additional data-transfer charges for traffic traversing the transit VPC: data is charged when it is sent from a spoke VPC to the transit VPC, and again from the transit VPC to the on-premises network.

[3] For recommended products, search AWS Marketplace for one the following terms: Cisco CSR 1000V, Fortinet FortiGate, Palo Alto Networks, Sophos UTM, Vyatta

Shared Service and Transit VPCs

As mentioned previously, these approaches are not mutually exclusive. A shared services VPC provides lower-latency access to replicated services or proxy-controlled access to on-premises resources, while a transit VPC is appropriate for services that do not make sense to replicate or proxy. Although a transit VPC can access shared services, a combined design that leverages VPC peering connections will improve network connectivity and reduce network-transfer costs (note that peering connections between spoke VPCs and the shared services VPC are not depicted in the diagram to the right for simplicity).

Please note that although the transit VPC can be classified as a shared service, AWS recommends creating a separate transit VPC. This approach greatly simplifies VPC routing tables since it allows all VPCs to route through VPC peering and VGW VPN connections, rather than introducing instance-based routing in some VPCs but not others. This also avoids introducing a single point of failure associate with instance-specific routes.



Resources

Amazon VPC Documentation	https://aws.amazon.com/documentation/vpc/ AWS webpage with links to VPC technical documentation, including introductory material (*Getting Started Guide*), component and strategy overviews (*User Guide*), and more robust technical documentation (*Network Administrator Guide*).
Auto Recovery for EC2	http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-recover.html EC2 product documentation describing the Auto Recovery for EC2 feature.
Auto Scaling Website	https://aws.amazon.com/autoscaling/
Single Data Center HA Network Connectivity	https://d0.awsstatic.com/aws-answers/AWS_Single_Data_Center_HA_Network_Connectivity.pdf AWS Solution Brief describing options for creating highly available connections from a single data center to AWS.
Multiple Data Center HA Network Connectivity	https://d0.awsstatic.com/aws-answers/AWS_Multiple_Data_Center_HA_Network_Connectivity.pdf AWS Solution Brief describing options for creating highly available connections from multiple data centers to AWS.