Cornell AWS Direct Connect Routing Diagrams

Introduction

This page shows and discusses the different routing options over the Cornell Direct Connect to AWS VPCs.

Glossary

Term	Definition
Asymmetric Routing	Condition that presents itself when network traffic between a client and its destination follows different paths inbound and outbound. This results in the client sending packets to one IP address but receiving responses from a potentially *different* IP address, preventing client and server from properly establishing two-way communication.
AWS Private Subnet	Subnet in an AWS VPC that has no direct access to the Internet.
AWS Public Subnet	Subnet in an AWS VPC that has direct Internet access by way of a configured Internet gateway (IGW).
Cornell Private Network	Private IPv4 address range 10.0.0.0/8, defined in RFC 1918 for use on private/internal networks. Addresses in this range **are not allowed** to leave the Cornell network and route directly over the Internet.
Cornell Public Network	Cornell's publicly routable IPv4 address ranges.
Direct Connect	Dedicated network connection between Cornell and Amazon Web Services via AWS peering partners. Direct Connect should be treated as if it were a campus network, including leveraging transport encryption for sensitive data. See also Cornell AWS Direct Connect.
Internet Gateway (IGW)	AWS-managed VPC routing device that provides inbound and outbound access from a subnet to the Internet. Allows use of public IP addresses (Elastic IP) on EC2 Instances.
Transit Gateway (TGW)	AWS-managed routing device that can cross-connect VPCs and Direct Connect resources. Transit Gateways are an integral component in the Cornell AWS Direct Connect Architecture.
Direct Connect Gateway (DCGW)	AWS-managed routing device that can connect VPCs and Transit Gateways in multple regions to Direct Connect connections. Direct Connect Gateways are an integral component in the Cornell AWS Direct Connect Architecture.
TCGW-DCGW infrastructure	The infrastructure components in the Cornell AWS Direct Connect Architecture that lie between the campus network and an AWS VPC using Direct Connect.

Secondary CIDR (in VPCs)	The secondary CIDR block is an extra, small CIDR block added to VPCs for use by the TGW-DCGW infrastructure. This CIDR block should not be used for any purpose other than for utility subnets.
Utility Subnets	Utility subnets are small (`/28`) subnets whose only purpose is to contain attachment points for a TGW. The utility subnets in a Direct Connect-connected VPC utilize CIDR blocks within the secondary CIDR block assigned to the VPC.

Exclusions

The discussion and examples below focus on traffic between private and public network segments in AWS, and private and public network segments on campus. Although the Cornell AWS Direct Connect Architecture also links AWS VPCs to Cornell's private network segments in Azure, traffic between AWS and Azure is not included in this article.

Direct Connect Routing Options

Private Network Extension

This is the **preferred** routing configuration for VPCs that have no specific requirements to directly address Cornell Public Network addresses via Direct Connect. It is the easiest to understand and troubleshoot.

In this configuration, the Cornell campus network will route network traffic to the VPC's private address space over the Direct Connect. The DCGW+TGW infrastructure connected to the AWS VPC will route any Cornell Private Network traffic not destined for Cornell VPCs in AWS back to the Cornell campus network via Direct Connect. This effectively leverages the Direct Connect as an *extension* of the Cornell Private Network.

Traffic from the VPC destined for other Cornell AWS VPCs will transit the DCGW+TGW infrastructure or dedicated peering connections, without exiting AWS.

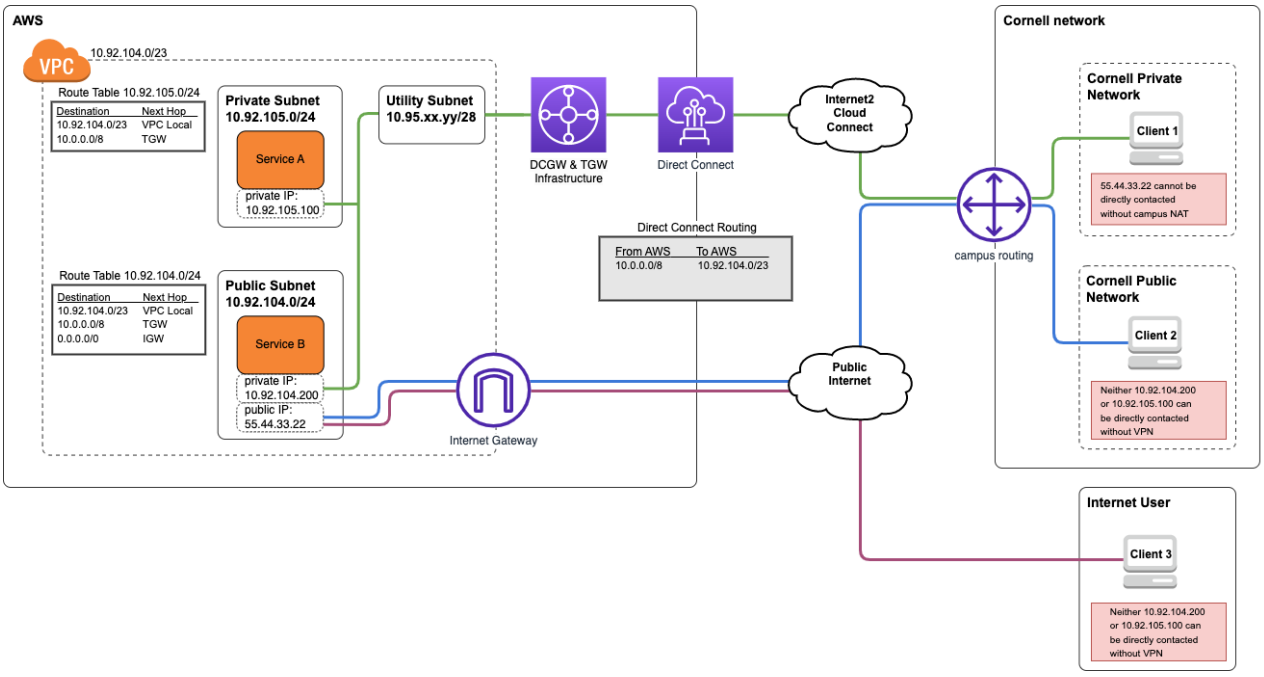
Design Decisions

When using the Private Network Extension model:

- All VPC subnets that use the DCGW+TGW infrastructure will need a static route for 10.0.0.0/8 defined to send traffic bound for other Cornell private network segments through the DCGW+TGW infrastructure.
- Private and Public subnets in the AWS VPC will be **unable** to address services or clients in Cornell Public Space directly over Direct Connect.
    - AWS Public Subnets may still be able to reach Cornell Public Network services over the Internet via the IGW.
        - Cornell network ACL or Managed Firewall policy updates may be required.
        - Consideration should be taken when transferring sensitive data over the public Internet. Use of transport encryption is **strongly suggested** and may be required by policy.
- Services deployed in the VPC should be configured to use AWS public addresses (e.g., EIP, public load balancer).
    - Exposing production services to clients over Direct Connect is **not advised** and **will not work** for clients in Cornell Public Network space.

Network Diagram

In the diagram below, Client 2 (Cornell Public Network) and Client 3 (Internet User) cannot reach Service A or Service B via their Cornell Private Network (10.0.0.0/8) addresses **without** use of a Cornell departmental VPN. Leveraging a Cornell departmental VPN connection would give either client an IP address and routing configuration for Cornell Private Network space, allowing them to directly contact the private IP addresses of Service A and Service B. This configuration is **not** shown in the diagram.



draw.io source: private-network-extension.v2.drawio

Hybrid Routing

This is our **preferred** routing configuration for VPCs that have a requirement for AWS Private Subnets to directly address Cornell Public Network addresses via Direct Connect.

In this configuration, the Cornell campus network will route network traffic to the VPC's private address space over the Direct Connect. The DCGW+TGW infrastructure connected to the AWS VPC will route traffic from *Private* AWS subnets to both *Private* and *Public* Cornell network segments back to campus via Direct Connect. For *Public* AWS subnets, the DCGW+TGW infrastructure will route traffic only to *Private* Cornell network segments back to campus via Direct Connect.

As in the **Private Network Extension** configuration discussed earlier, local VPC traffic (i.e. destined for the VPC itself), traffic to peered AWS VPCs, and traffic to other Cornell VPCs using Direct Connect in AWS remains within AWS and is **not** sent back to campus over the Direct Connect.

To prevent asymmetric routing from occurring, this configuration leverages a distinct set of routing tables in the AWS VPC for Public and Private subnets.

Design Decisions

When using the Hybrid Routing model:

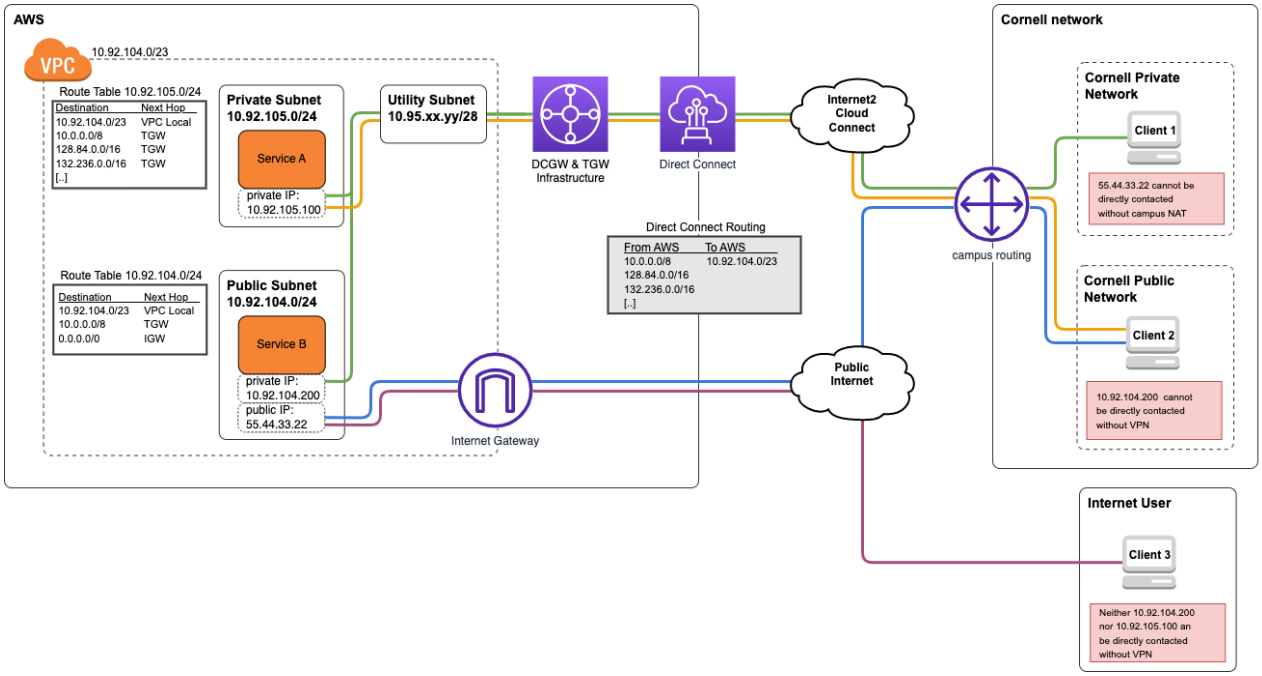
- You will require one route table for Private subnets and another for Public subnets.
  - Subnets can share common routing tables, so multiple "private" or "public" subnets can reference the same routing configuration.
- AWS Private Subnets should use a route table with static routes to both Cornell's Private (10.0.0.0/8) and Public address (e.g., 128.84.0.0/16) space. These routes would designate the TGW as the "next hop".
- AWS Public Subnets should use a route table with one static route designating the TGW as the next hop for Cornell's Private address space (i.e. 10.0.0.0/8).
  - Allowing AWS Public Subnets to send Cornell Public Network traffic over Direct Connect can create asymmetric routing conditions.
- AWS Public subnets will be **unable** to address services or clients in Cornell Public Space directly over Direct Connect
  - AWS Public Subnets may still be able to reach Cornell Public Network services over the Internet via the IGW.
    - Cornell network ACL or Managed Firewall policy updates may be required.
    - Consideration should be taken when transferring sensitive data over the public Internet. Use of transport encryption is **strongly suggested** and may be required by policy.
- Services deployed in the VPC should be configured to use AWS public addresses (e.g., EIP, public load balancer).
  - Exposing production services to clients over Direct Connect is not advised.

Network Diagram

In the diagram below:

- Client 2 (Cornell Public Network) cannot reach Service B via its Cornell Private Network (10.0.0.0/8) address **without** use of a Cornell departmental VPN.
- Client 3 (Internet User) cannot reach Service A or Service B via their Cornell Private Network (10.0.0.0/8) addresses **without** use of a Cornell departmental VPN.

Leveraging a Cornell departmental VPN connection would give either client an IP address and routing configuration for Cornell Private Network space, allowing them to directly contact the private IP addresses of Service A and Service B. These configurations are **not** shown in the diagram.



draw.io source: hybrid-routing.v2.drawio

"All Campus" Routing

This configuration, though similar to **Hybrid Routing**, is **not preferred** since it allows for the possibility of asymmetric routing on AWS Public Subnets. Given the similarity to **Hybrid Routing** and the potential to serve similar use cases, we *strongly recommend against* using this option.

In this configuration, the Cornell campus network will route network traffic to the VPC's private address space over the Direct Connect. The DCGW+TGW infrastructure connected to the AWS VPC will route traffic from both *Private* and *Public* AWS subnets to both *Private and Public* Cornell network segments back to campus via Direct Connect.

As in the **Private Network Extension** and **Hybrid Routing** configurations discussed earlier, local VPC traffic (i.e. destined for the VPC itself), traffic to peered AWS VPCs, and traffic to other Cornell VPCs using Direct Connect in AWS remains within AWS and is **not** sent back to campus over the Direct Connect.

There is no inherent protection against asymmetric routing from occurring as all advertised routes from Cornell campus are presented in the subnet routing tables.

Design Decisions

When using the "All Campus" Routing model:

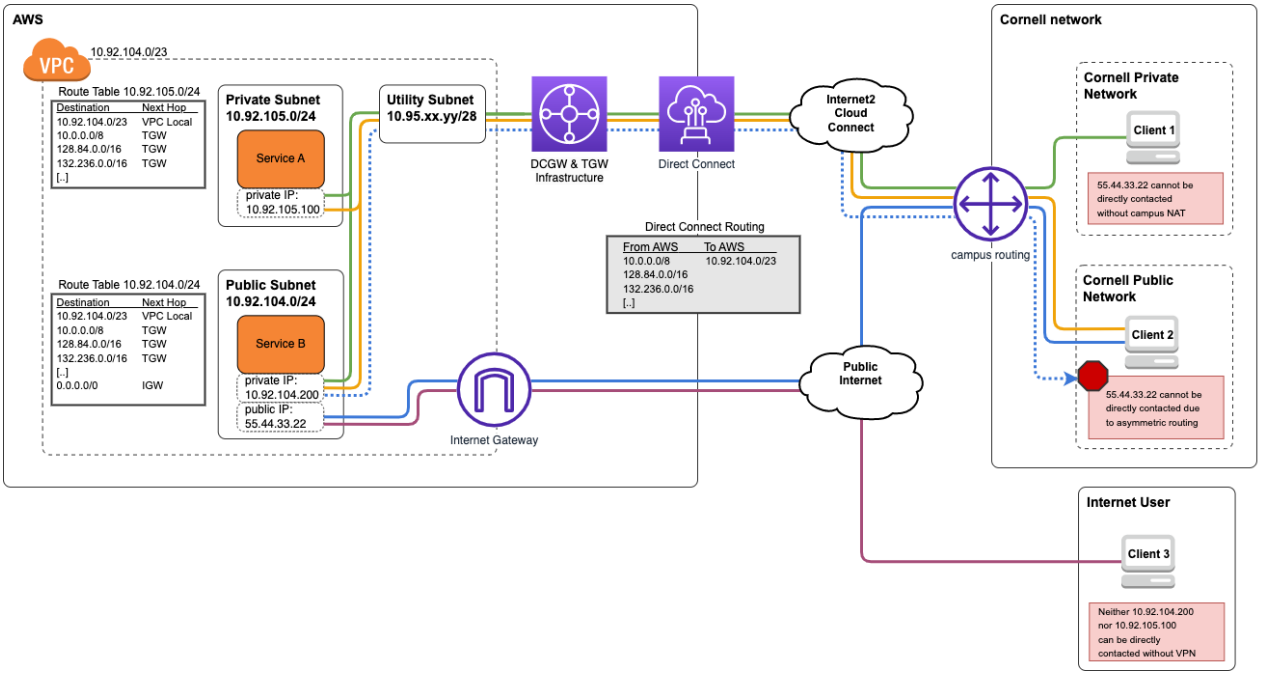
- Both Private and Public AWS Subnets use route tables with static routes to both Cornell's Private (10.0.0.0/8) and Public address (e.g., 128.84.0.0/16) space. These routes would designate the TGW as the "next hop".
- AWS Public Subnets that use the DCGW+TGW infrastructure risk introducing asymmetric routing when presenting services to clients on Cornell Public Networks.
    - Leveraging AWS Load Balancers essentially acting as a proxy may be an acceptable work-around to avoid asymmetric routing.
- AWS Public Subnets will be able to address services or clients in Cornell Public Space directly over Direct Connect.
- Exposing production services to clients over Direct Connect is not advised.

Network Diagram

In the diagram below:

- Client 2 (Cornell Public Network):
    - **Can** reach Service B via its Cornell Private Network (10.0.0.0/8).
    - **Can not** reach Service B via its public IP address (return traffic denoted by dashed blue line)
        - Client 2 initiates a connection to Service B at 55.44.33.22. Service B responds, via its Cornell Private Network address over the Direct Connect. Client 2, not expecting return traffic from 10.92.104.200, drops the packets and the connection is never established.
        - Introducing an Elastic Load Balancer, or other proxy/indirection device, instead of directly attaching an AWS public IP address to Service B could potentially work around this problem.

- Client 3 (Internet User) cannot reach Service A or Service B via their Cornell Private Network (10.0.0.0/8) addresses **without** use of a Cornell departmental VPN connection.
    - VPN configuration for Client 3 is **not** shown in the diagram.



draw.io source: all-campus-routing.v2.drawio

Route Table Examples

The examples below shows the VPC using Direct Connect as having 10.92.104.0/23 as primary CIDR and 10.95.32.0/26 as secondary CIDR (used for utility subnets).

Private Network Extension

Private Subnet Route Table

Destination	Target	
0.0.0.0/0	NAT Gateway	public internet
10.92.104.0/23	local	primary VPC CIDR
10.95.32.0/26	local	secondary VPC CIDR
10.0.0.0/8	TGW	Cornell private CIDR

Public Subnet Route Table

Destination	Target	
0.0.0.0/0	IGW	public internet
10.92.104.0/23	local	primary VPC CIDR
10.95.32.0/26	local	secondary VPC CIDR
10.0.0.0/8	TGW	Cornell private CIDR

Hybrid Routing

Private Subnet Route Table

Destination	Target	Notes
0.0.0.0/0	NAT Gateway	public internet
10.92.104.0/23	local	primary VPC CIDR
10.95.32.0/26	local	secondary VPC CIDR
10.0.0.0/8	TGW	Cornell private CIDR
128.84.0.0/16	TGW	Cornell public CIDR
128.253.0.0/16	TGW	Cornell public CIDR
132.236.0.0/16	TGW	Cornell public CIDR
192.35.82.0/24	TGW	Cornell public CIDR
192.122.235.0/24	TGW	Cornell public CIDR
192.122.236.0/24	TGW	Cornell public CIDR

Public Subnet Route Table

Destination	Target	
0.0.0.0/0	IGW	public internet
10.92.104.0/23	local	primary VPC CIDR
10.95.32.0/26	local	secondary VPC CIDR
10.0.0.0/8	TGW	Cornell private CIDR

"All Campus" Routing

Private Subnet Route Table

Destination	Target	Notes
0.0.0.0/0	NAT Gateway	public internet
10.92.104.0/23	local	primary VPC CIDR
10.95.32.0/26	local	secondary VPC CIDR
10.0.0.0/8	TGW	Cornell private CIDR
128.84.0.0/16	TGW	Cornell public CIDR
128.253.0.0/16	TGW	Cornell public CIDR
132.236.0.0/16	TGW	Cornell public CIDR
192.35.82.0/24	TGW	Cornell public CIDR
192.122.235.0/24	TGW	Cornell public CIDR
192.122.236.0/24	TGW	Cornell public CIDR

Public Subnet Route Table

Destination	Target	Notes
0.0.0.0/0	IGW	public internet
10.92.104.0/23	local	primary VPC CIDR
10.95.32.0/26	local	secondary VPC CIDR

10.0.0.0/8	TGW	Cornell private CIDR
128.84.0.0/16	TGW	Cornell public CIDR
128.253.0.0/16	TGW	Cornell public CIDR
132.236.0.0/16	TGW	Cornell public CIDR
192.35.82.0/24	TGW	Cornell public CIDR
192.122.235.0/24	TGW	Cornell public CIDR
192.122.236.0/24	TGW	Cornell public CIDR