Cornell AWS Direct Connect Architecture

Introduction

Cornell migrated to the 2023 Direct Connect architecture (aka v2 architecture) in January 2023.

We use the following terminology in this document:

- I2CC – Internet2 Cloud Connect
- DC – Direct Connect
- TGW – Transit Gateway
- VPC – Virtual Private Cloud

Internet 2 Architecture

Cornell uses the Internet2 Cloud Connect (I2CC) service to provide private connectivity of Cornell networks to Azure and AWS. Cornell has multiple 100Gbps connections to Internet2. In turn I2CC has multiple 5Gbps (as of 16 Feb 2023) connections to the major cloud vendors.

The I2CC service offers several benefits:

- Consolidating and simplifying configuration and management of Direct Connect for Cornell AWS accounts (compared to the previous on-campus Direct Connect architecture)
- Improving flexibility and bandwidth of Direct Connect connectivity
- Allowing private Cornell network traffic in AWS and Azure to flow between those clouds without transiting campus

- Cornell campus is connected to NYSERNET/I2 via two separate physical links, each 100Gbps. One link runs to Buffalo, and one runs to New York City.
- I2 to Azure linkage is dual connectivity, with a physical maximum of 5Gbps. The actual capability of the I2-Azure link is limited by Express Route bandwidth configuration.
- As of January 2023, Cornell does not have a I2CC link to GCP.
- The 5Gbps connectivity between I2 and Azure/AWS is infrastructure shared with other I2CC customers.
- See also the physical configuration details here: https://mapper.cit.cornell.edu/g46967a52/document/_/!index.html

? Unknown Attachment

Cross-Account Architecture

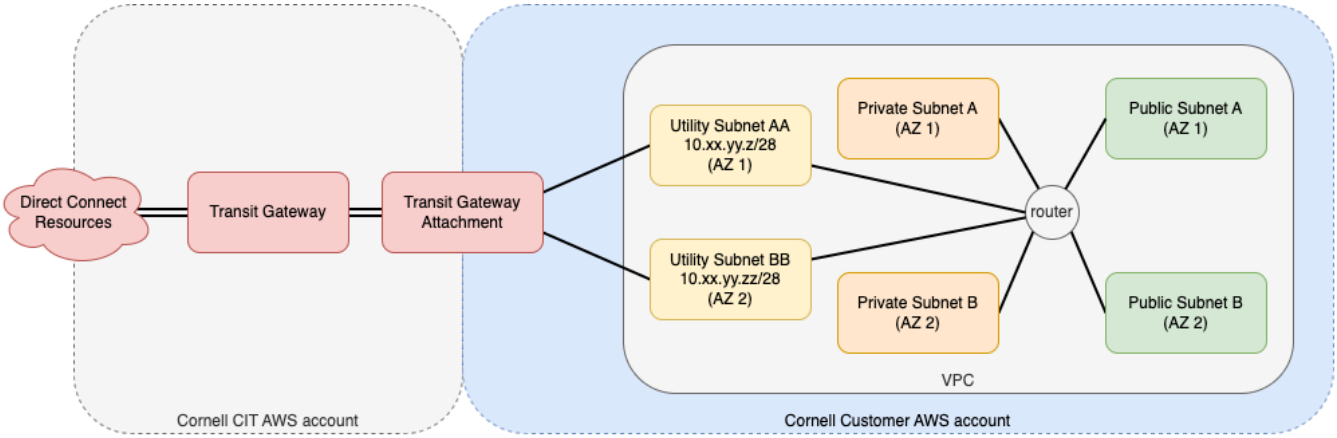
The architecture used to provide Direct Connect service to Cornell AWS accounts utilizes AWS Transit Gateways (one per AWS region) in a central AWS account (cu-cit-network) to which VPCs in Cornell AWS accounts are attached. Multiple VPCs in a single AWS account can be attached to Direct Connect in this way.

Each VPC connected to this architecture has full connectivity to all other VPCs connected to the architecture, without need for VPC-to-VPC peering.

? Unknown Attachment

Account (VPC) Architecture

See Direct Connect Resources in Cornell AWS Accounts for details about the DC-related resources shown below.



Example Public VPC Route Table
0.0.0.0/0 → Internet Gateway
10.92.aa.bb/22 → local (primary subnets)
10.xx.yy.z/28 → local (utility)
10.xx.yy.zz/28 → local (utility)
10.0.0.0/8 → TGW Attachment

Example Private VPC Route Table
0.0.0.0/0 → NAT Gateway
10.92.aa.bb/22 → local (primary subnets)
10.xx.yy.z/28 → local (utility)
10.xx.yy.zz/28 → local (utility)
10.0.0.0/8 → TGW Attachment
128.84.0.0/16 → TGW Attachment
128.253.0.0/16 → TGW Attachment
132.236.0.0/16 → TGW Attachment
192.35.82.0/24 → TGW Attachment
192.122.235.0/24 → TGW Attachment
192.122.236.0/24 → TGW Attachment

draw.io source: dc-arch-2023.customer.10.0.0.8.v2.drawio

Paths and Traffic Filtering

Inbound Traffic – From TGW to EC2 Instance

	Resource	Filtering	Notes
Source	TGW	—	
	TGW Attachment	—	
	TGW Attachment Elastic Network Interface	—	
	NACL of Subnet attached to TGW	outbound rules of NACL attached to utility subnet	The NACL bound to the utility subnets allow all traffic in and out.
	Route Table of Subnet attached to TGW	—	
	NACL of Subnet containing EC2 instance	inbound rules of NACL for destination subnet	
	EC2 Instance Security Group	inbound rules of SG	
Destination	EC2 Instance Elastic Network Interface	—	

Outbound Traffic – From EC2 Instance to TGW

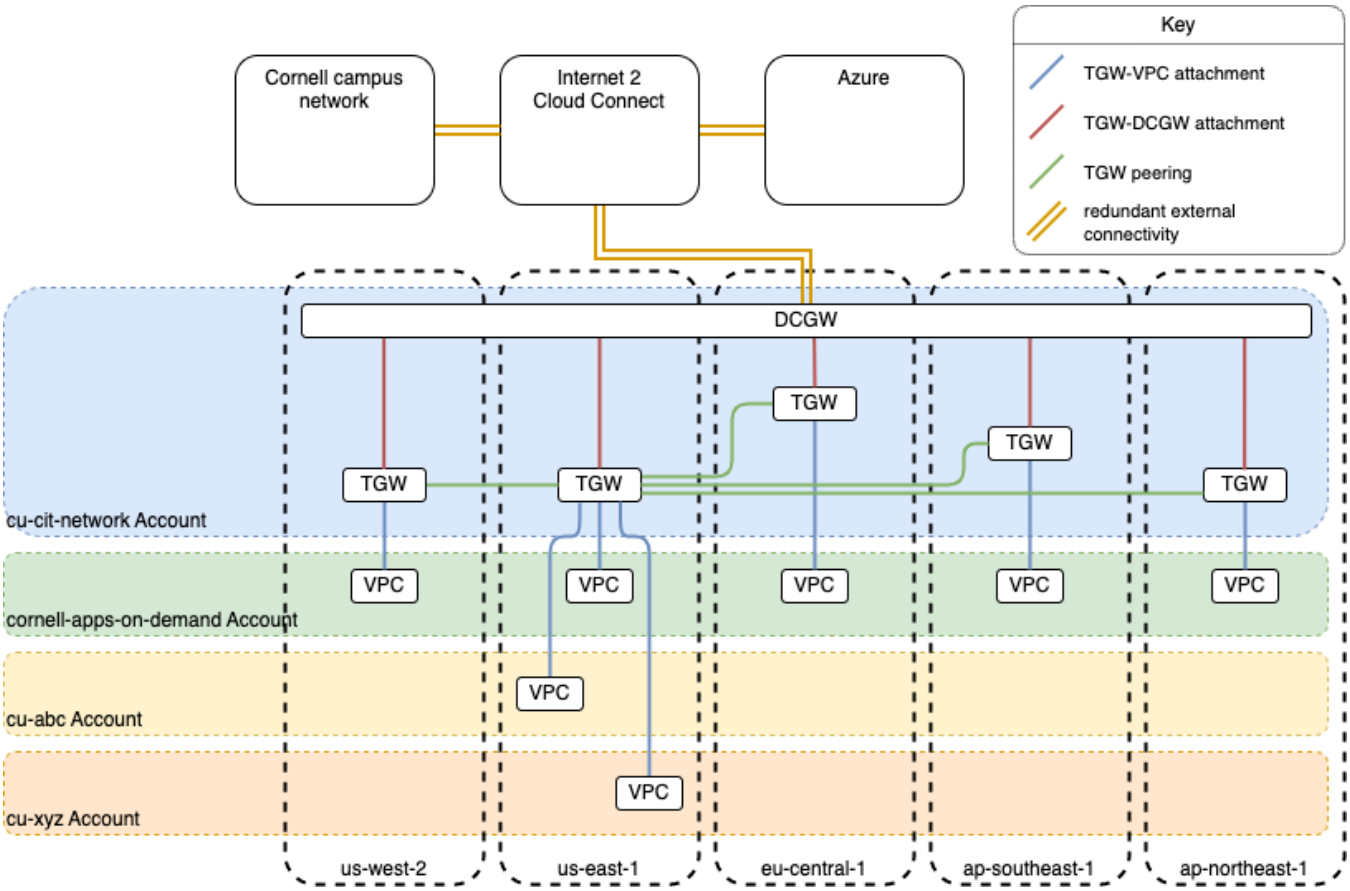
	Resource	Filtering	Notes
Source	EC2 Instance Elastic Network Interface	—	
	EC2 Instance Security Group	outbound rules of SG	
	NACL of Subnet containing EC2 instance	outbound rules of NACL for source subnet	
	Route Table of Subnet containing EC2 instance	—	
	NACL of Subnet attached to TGW	inbound rules of NACL attached to utility subnet	The NACL bound to the utility subnets allow all traffic in and out.
	TGW Attachment Elastic Network Interface	—	
	TGW Attachment	—	
Destination	TGW	—	

Multi-Region Architecture

The 2023 Direct Connect architecture supports DC connectivity in multiple, but limited AWS regions. Transit Gateways are are regional, but TGWs in different regions can be peered. We use the TGW in us-east-1 as a "hub" and consider the TGWs in other regions as "spokes". This allows any VPC connected to any TGW to reach any other connected VPC. The TGWs in each region receive Direct Connect connectivity by connecting to a single Direct Connect Gateway (DCGW) which has a global footprint and can support TGW connections in any region. Technically, each DCGW is limited to attaching to no more than 3 TGWs. However, Cornell has received a special allowance that allows 5-6 TGW attachments per DCGW.

The cost to Cornell of supporting TGWs in each region is about $864 region/yr.

As of January 2023, this multi-region capability exists primarily because of Cornell Apps on Demand requirements. No other Cornell AWS accounts have expressed the need to utilize Direct Connect in regions other than us-east-1.



draw.io source: tgw-peering.v2.drawio