Advanced Networking on Cloud

René Serral <rene.serral@upc.edu>

Agenda

- Introduction
- Evolution of network topologies
- AWS VPN
- VPC Peering
- Transit Gateway

Agenda

- **Introduction**
- Evolution of network topologies
- AWS VPN
- VPC Peering
- Transit Gateway

Introduction

- Company networks are complex
  - They many involve hybrid scenarios
  - Employees working from home
  - ...
- The network needs to adapt to diverse business needs
- Often networks have complex routing policies

Introduction

- Public cloud providers offer predefined networking solutions to ease adoption
- Configuration automation
- With ease of use
- Simple yet powerful to guarantee easy transition to cloud

Agenda

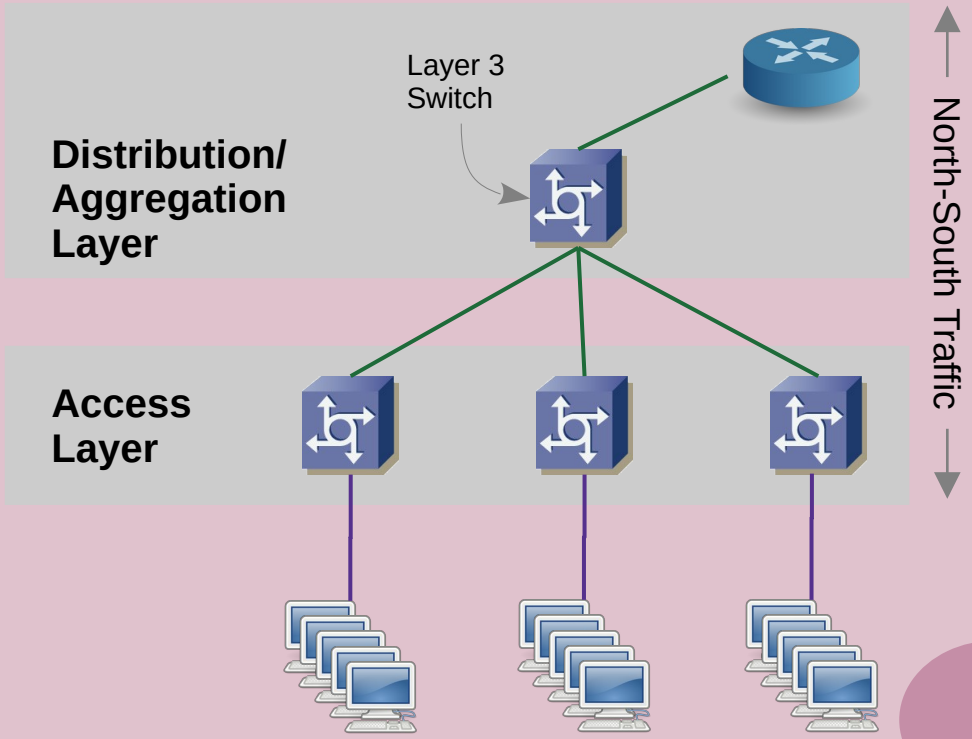
- Introduction
- **Evolution of network topologies**
- AWS VPN
- VPC Peering
- Transit Gateway

Legacy Topologies

- 2-Tier topology
- 3-Tier topology
- Spine-Leaf topology

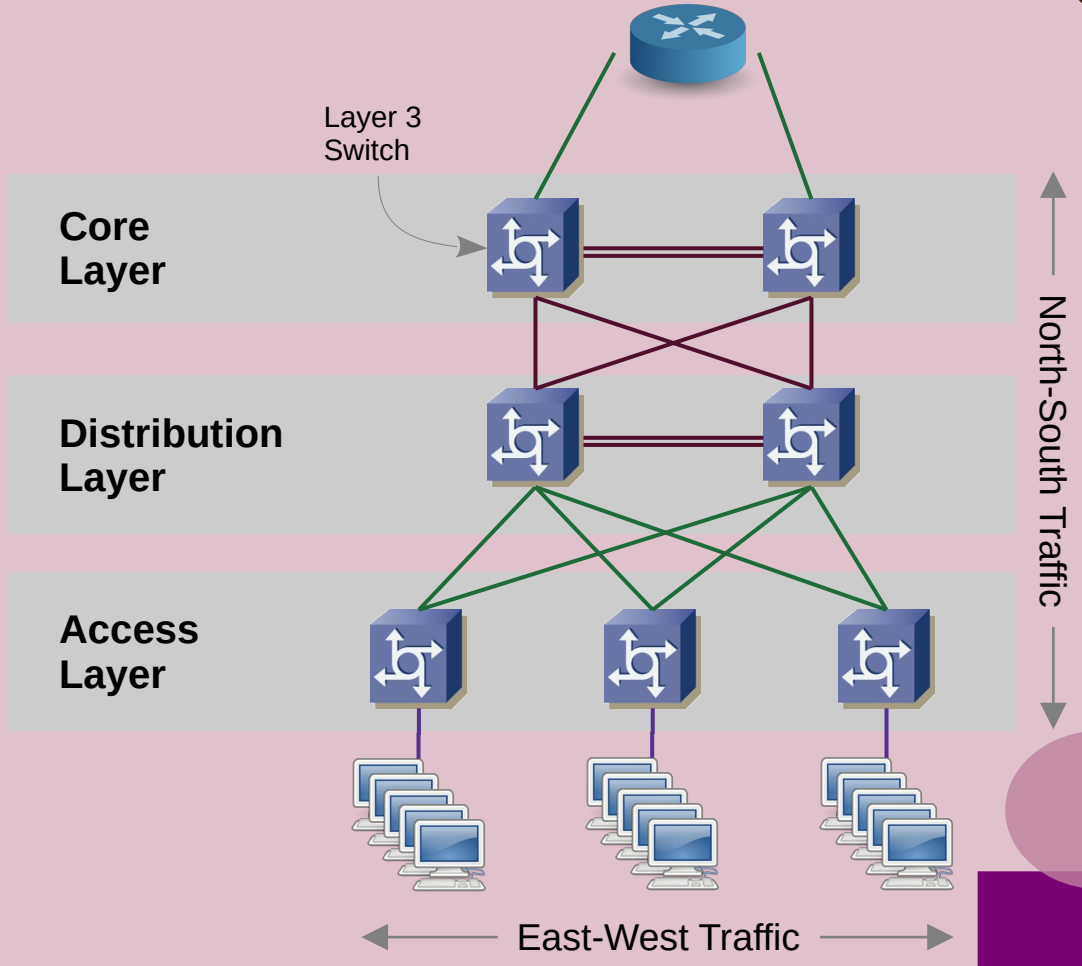
2-Tier Topology

- Easy and cheap topology
- Single point of failure on the distribution layer
- Good for in/out traffic on the data-center



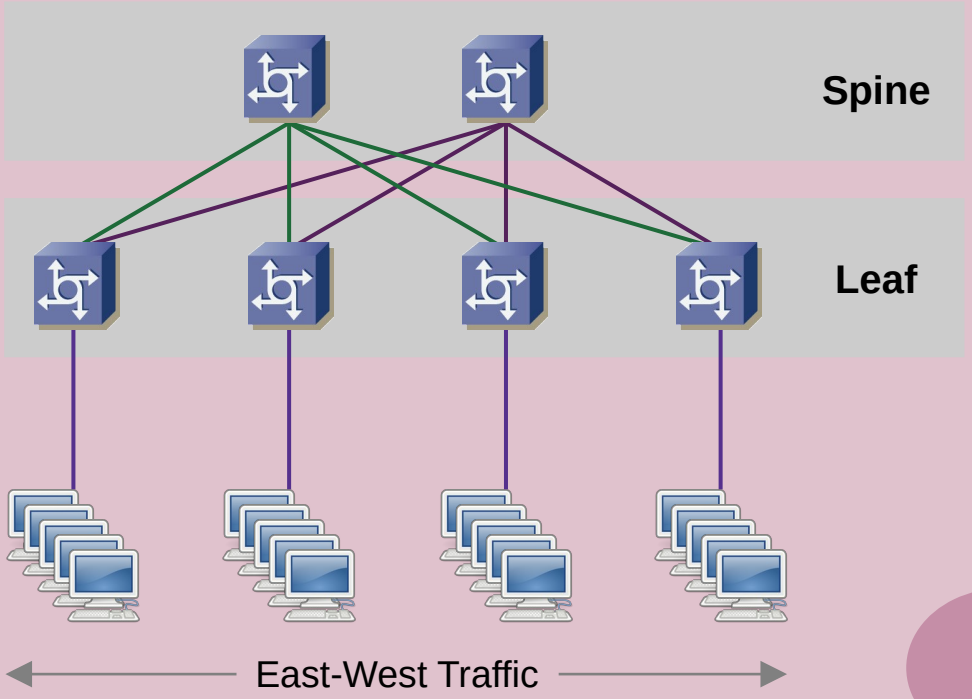
3-Tier Topology

- More complex and expensive
- Better reliability through redundancy
- Uses spanning tree protocol



Spine-Leaf Topology

- Good Scalability
- Good compromise
- Uses SBP or TRILL and ECMP
- Normally Speed Ratio of 3:1



Cloud Topologies

- Which is the goal of topologies in the cloud?

Cloud Topologies

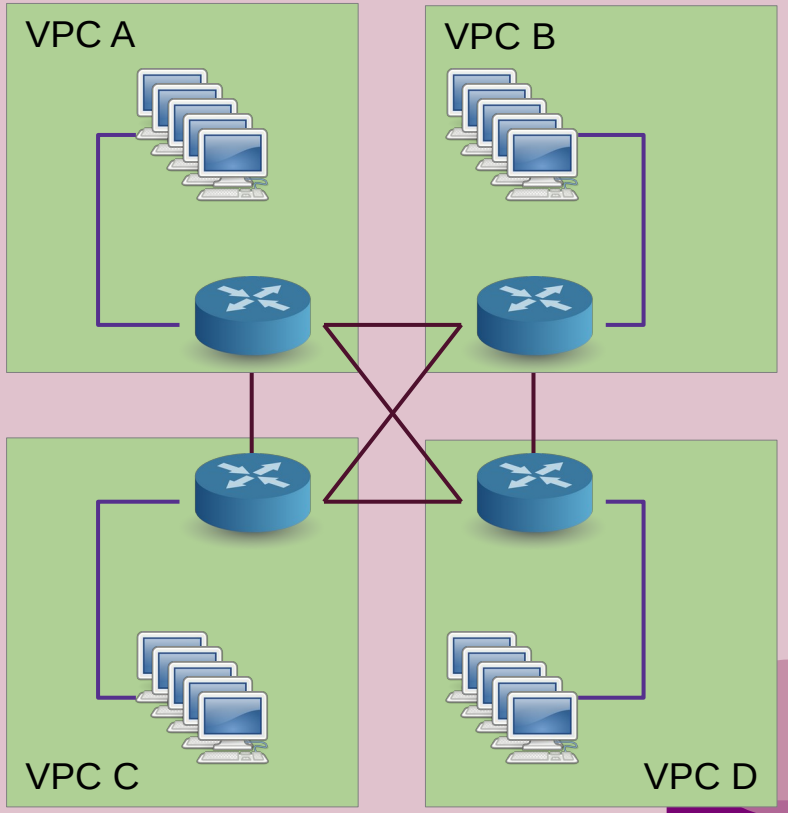
- Which is the goal of topologies in the cloud?
  - Interconnect various VPC
  - Interconnect various Regions
  - Allow hybrid deployments
  - ...

Cloud Topologies

- Full-Mesh
- Partial-Mesh
- Hub-Spoke

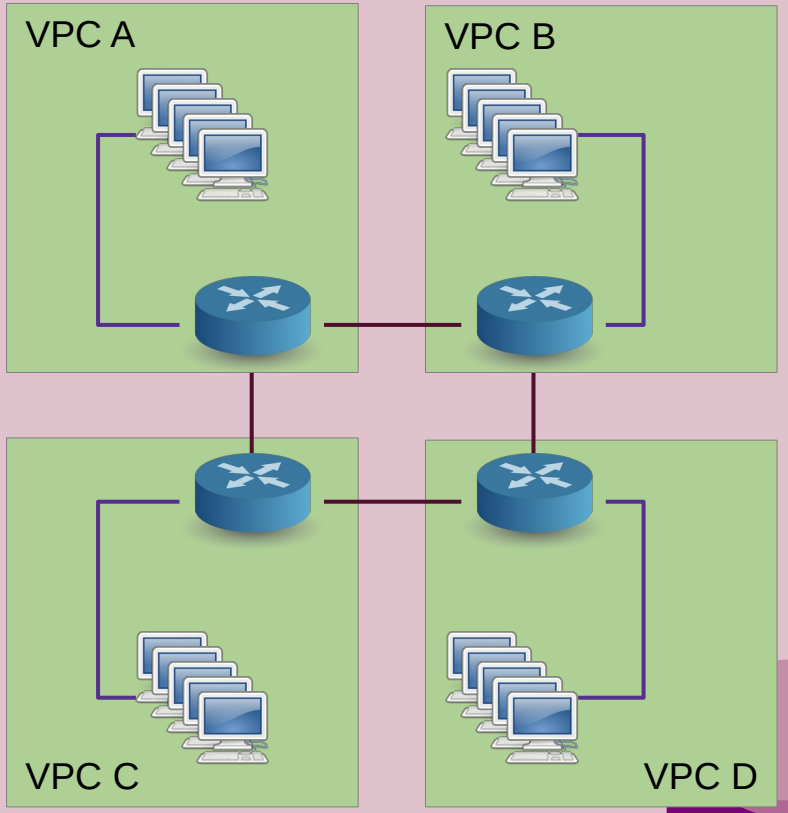
Full-Mesh Topology

- Allows the interconnection among all VPC
- Consistent number of hops to get to the destination
- Costly to maintain (n – 1)
- Security policies are tricky



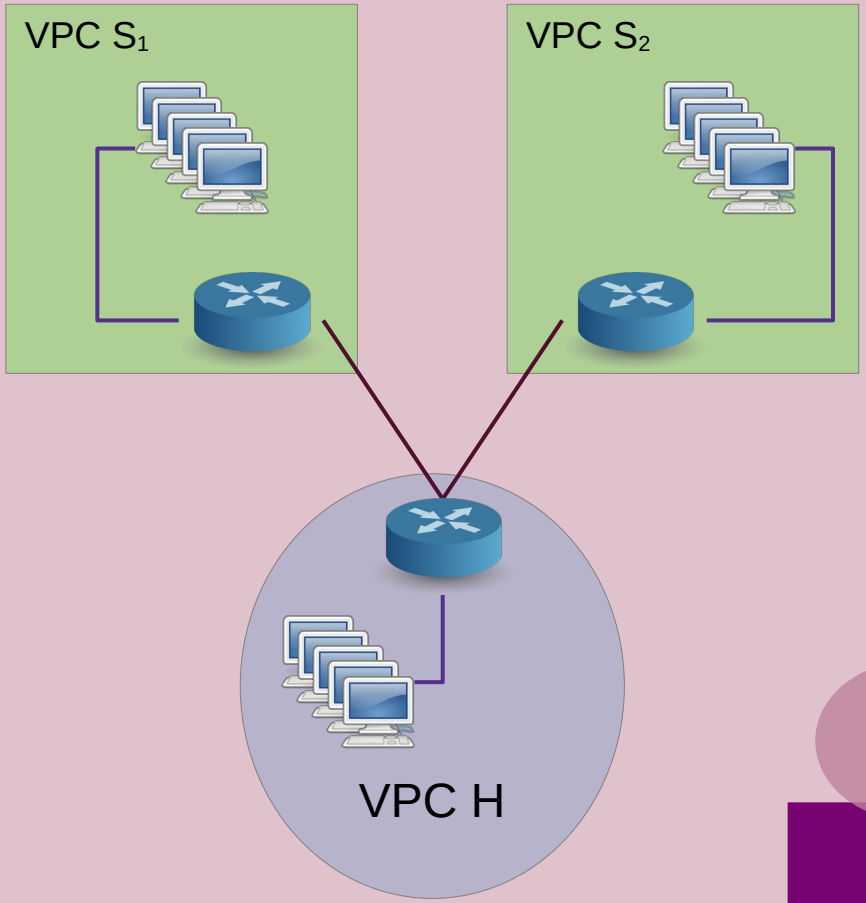
Partial-Mesh Topology

- Easier to maintain
- Variable number of hops to get to the destination
- Complicated routing tables
- Security policies are tricky



Hub and Spoke (Star) Topology

- Easier to maintain
- Variable number of hops to get to the destination
- Simplified routing tables
- Easier Security policies

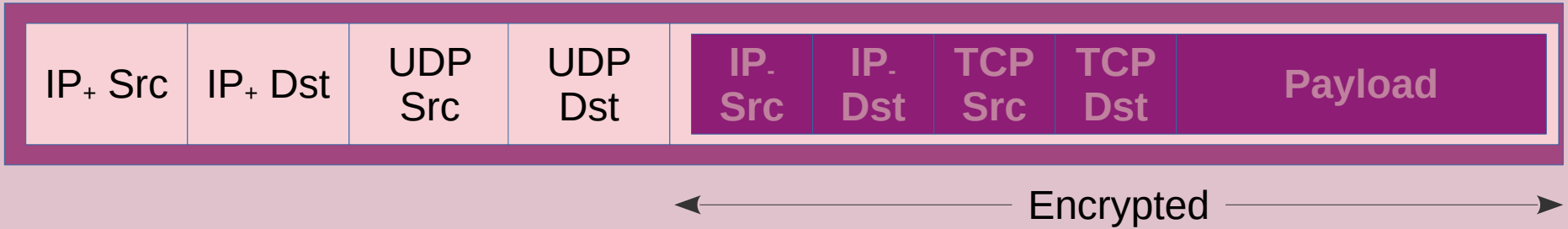


Agenda

- Introduction
- Evolution of network topologies
- **AWS VPN**
- VPC Peering
- Transit Gateway

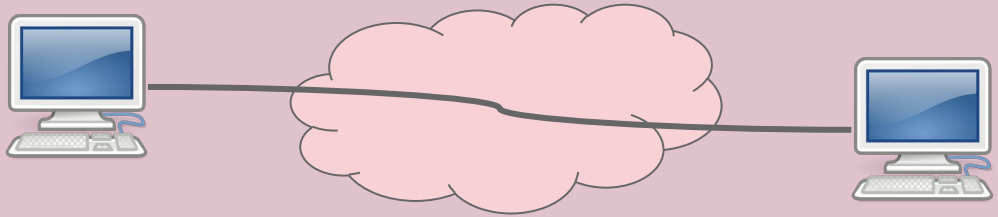
Virtual Private Network (VPN)

- Secure (encrypted) communications
- Overlay network that allows the *“direct”* interconnection between two Internet locations
- Traffic encapsulation



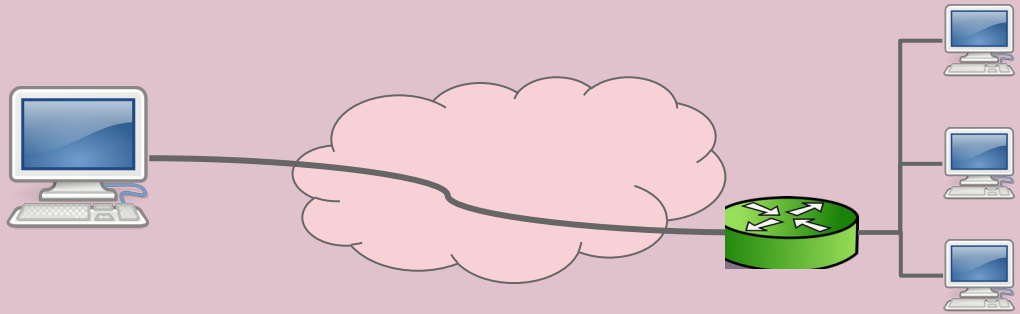
Virtual Private Network (VPN)

- Secure (encrypted) communications
- Different topologies
  - **From Host to Host**
  - From Host to Site
  - From Site to Site



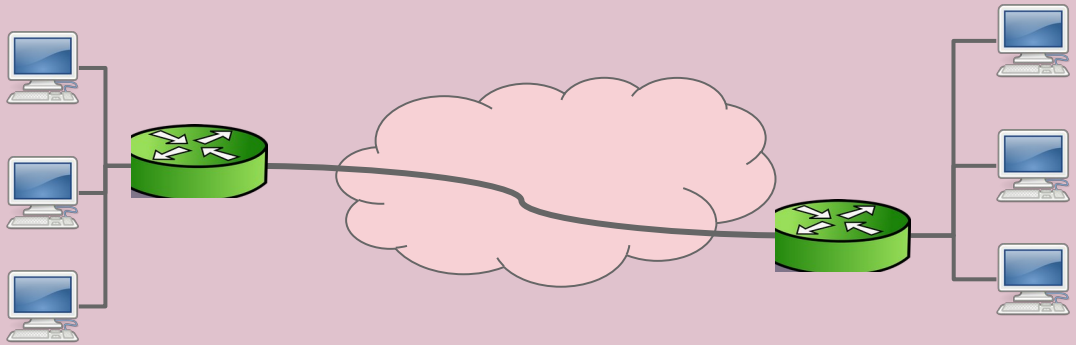
Virtual Private Network (VPN)

- Secure (encrypted) communications
- Different topologies
  - From Host to Host
  - **From Host to Site**
  - From Site to Site



Virtual Private Network (VPN)

- Secure (encrypted) communications
- Different topologies
  - From Host to Host
  - From Host to Site
  - **From Site to Site**

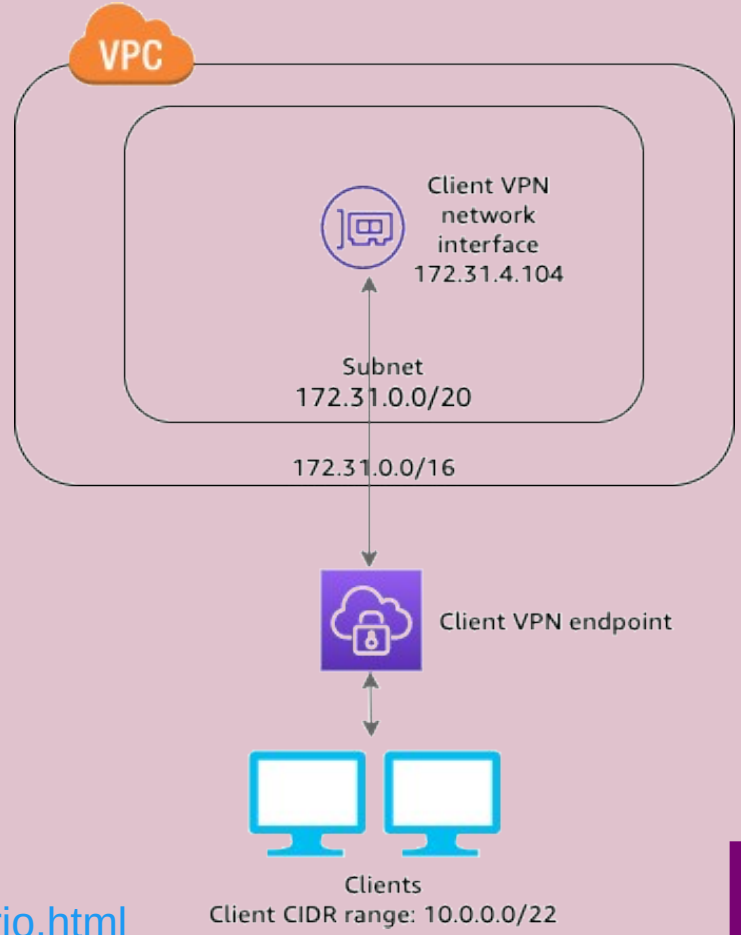


VPN – Protocols

	Managed AWS
IPSec – Oldest reliable solution supported everywhere	☑
OpenVPN – Poor man’s VPN but well supported	☑
Wireguard – New kid on the block	☒

VPN on AWS[1]

- Allow easy connectivity on hybrid deployments
- Uses OpenVPN by default
- Severe bandwidth limits



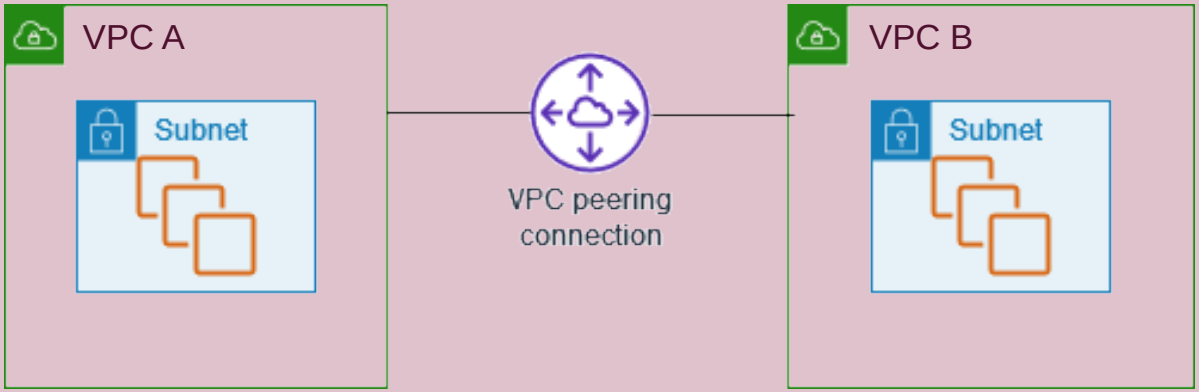
[1] https://docs.aws.amazon.com/vpn/latest/clientvpn-admin/scenario.html

Agenda

- Introduction
- Evolution of network topologies
- AWS VPN
- **VPC Peering**
- Transit Gateway

VPC Peering[1]

- Point-to-Point internal AWS connection
- Interconnects two different VPC
- Building stone for more complex topologies



[1] https://docs.aws.amazon.com/vpc/latest/peering/what-is-vpc-peering.html

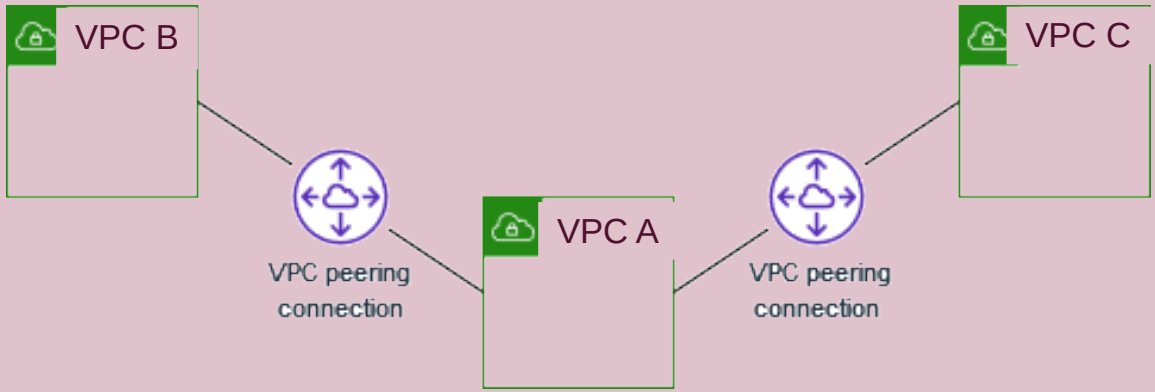
AWS VPC Peering Connections[1]

- Private connections within AWS regions
- All services can interact with each other (with exceptions)
- There is no need of using a gateway, VPN connection, or network appliance
- It uses exclusively the private IP space
- All inter-Region traffic is encrypted with no single point of failure, or bandwidth bottleneck
- Traffic always stays on the global AWS backbone

[1] https://docs.aws.amazon.com/vpc/latest/peering/vpc-peering-basics.html

VPC Peering: Limitations

- Max. 1 Peering between the two same VPC
- No access to the DNS of the peer VPC
- No overlaping CIDR blocks
- No transitive peering
- It is not possible to create security group rules referencing the peer’s security rules

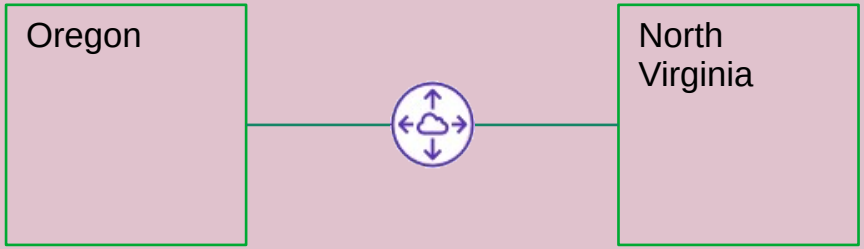


VPC Peering

- Why do you think it is not massively used in AWS?
  - Scalability problems
  - Routing problems
  - Manual configuration
  - All of the above

Lab 1

- Interconnect two AWS regions using an AWS VPC Peering connection



Agenda

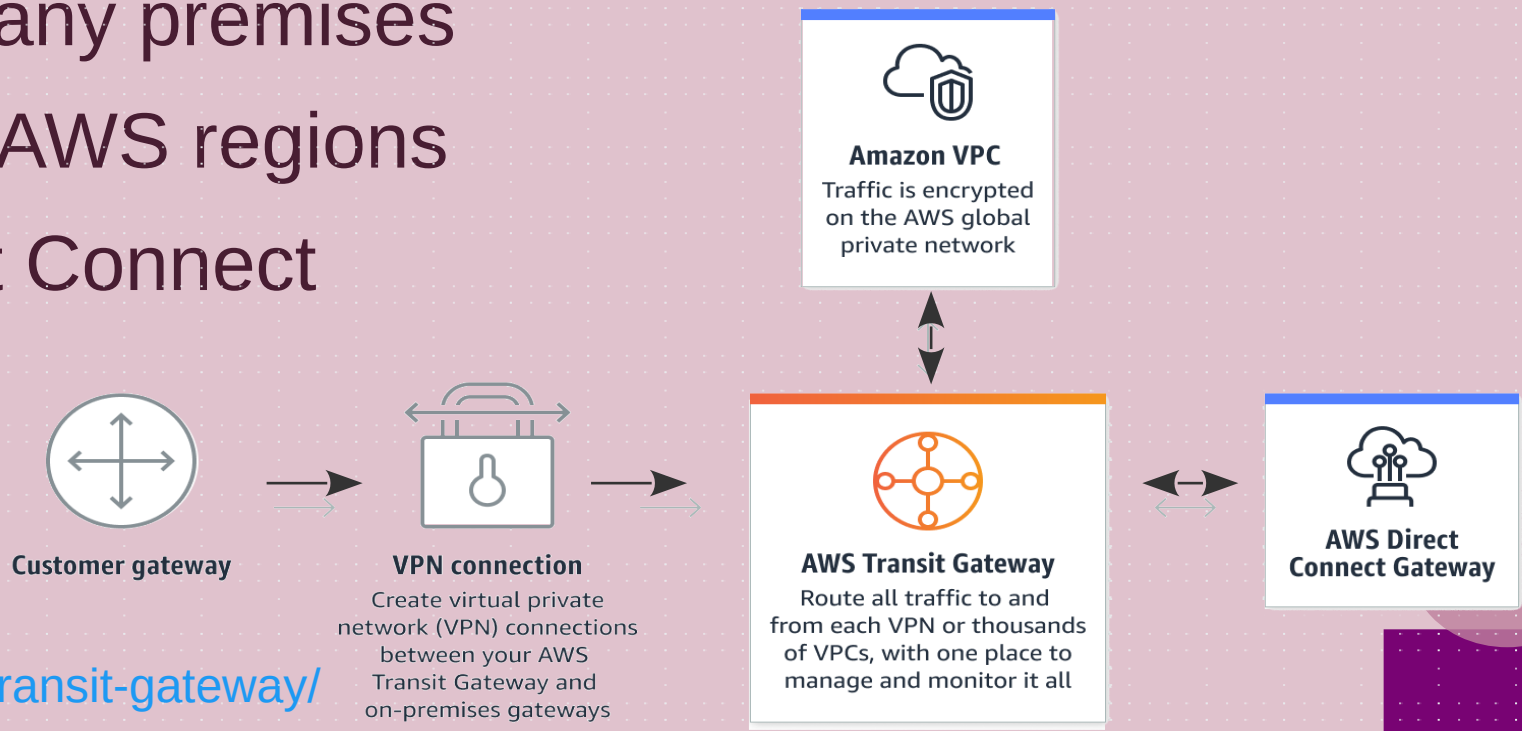
- Introduction
- Evolution of network topologies
- AWS VPN
- VPC Peering
- **Transit Gateway**

What’s the problem with Peering?

- Hard to configure for large networks
- It is not easy to route traffic among the peers – needs to be done manually
- Security becomes a nightmare

Transit Gateway[1]

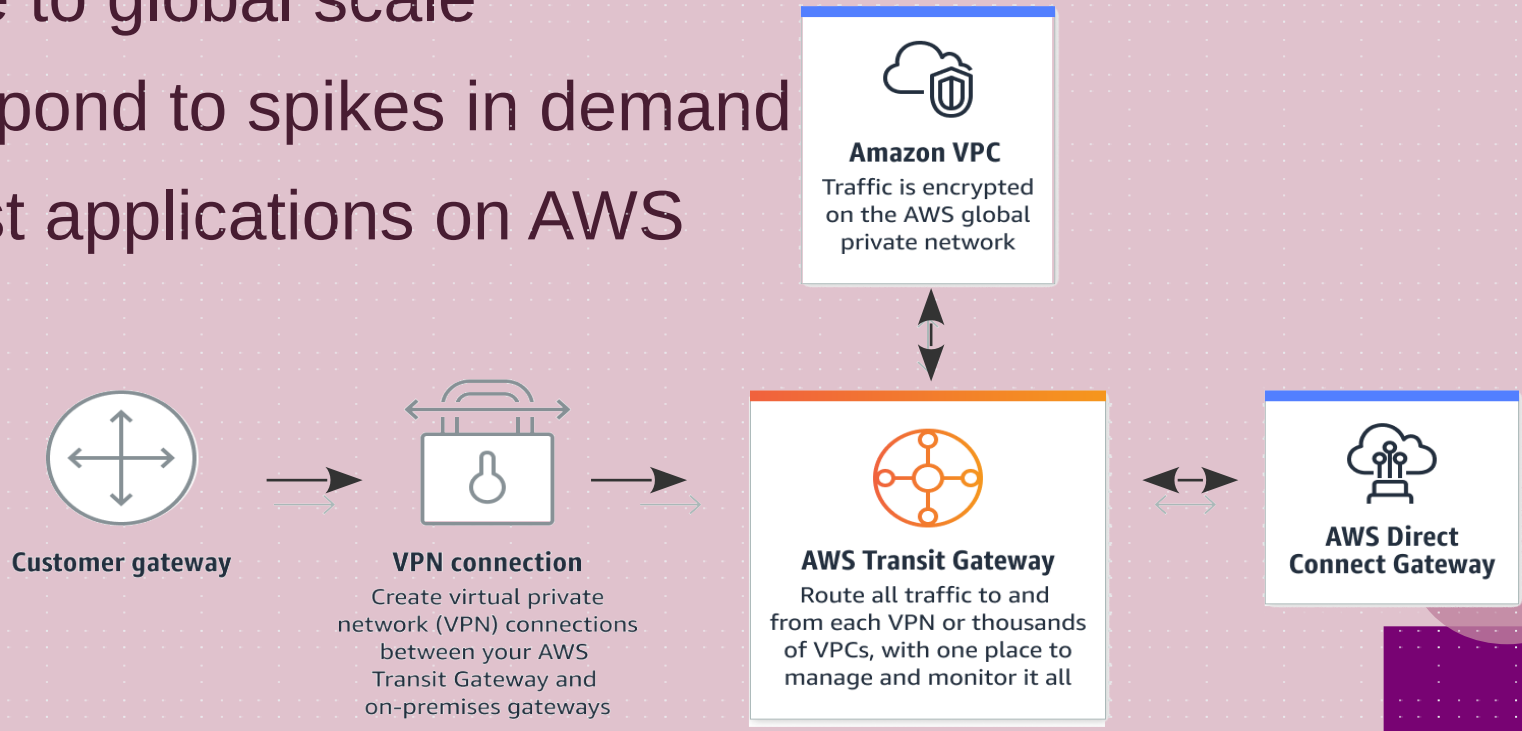
- Ubiquitous interconnection
- From company premises
- From other AWS regions
- From Direct Connect



1 https://aws.amazon.com/transit-gateway/

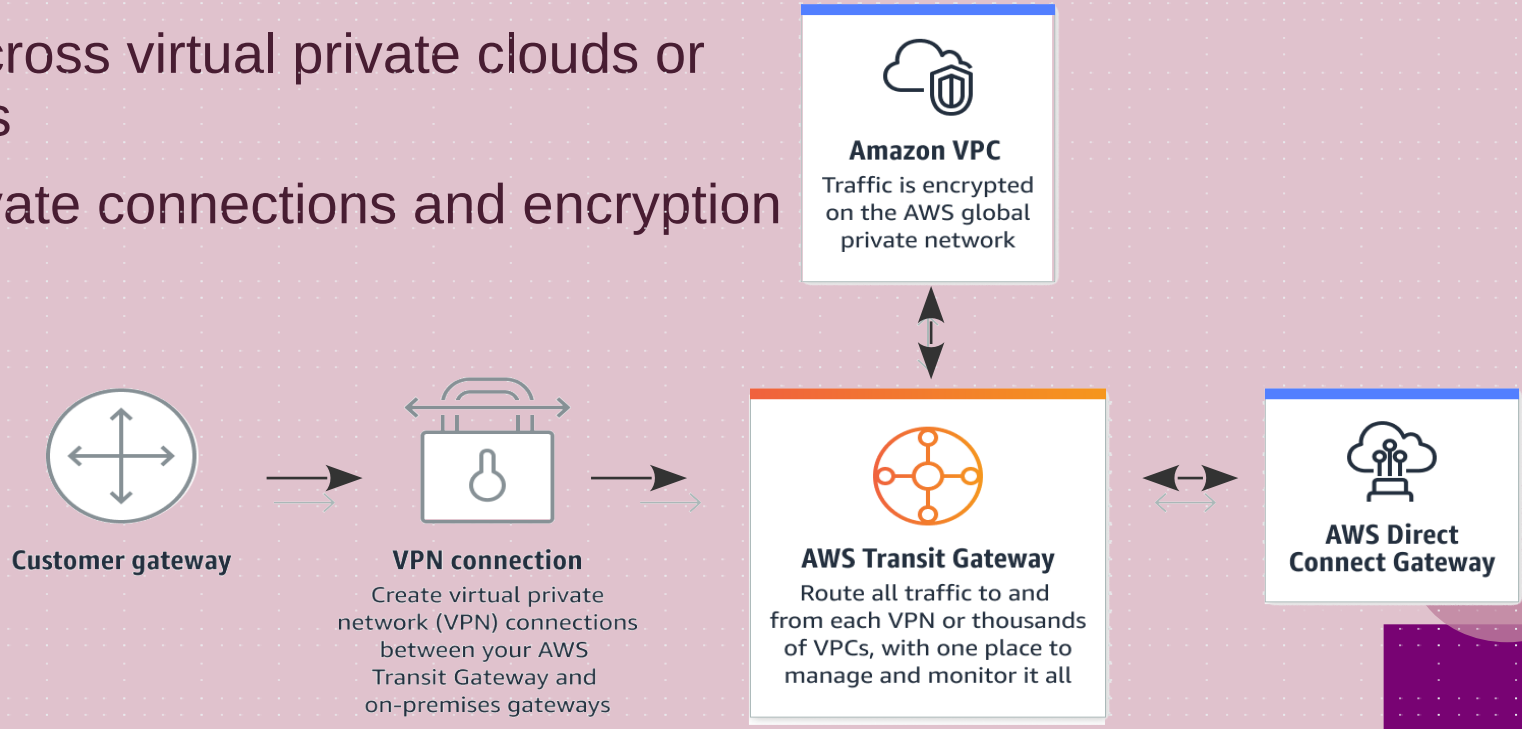
Transit Gateway

- Deliver applications around the world
- Rapidly move to global scale
- Smoothly respond to spikes in demand
- Host multicast applications on AWS



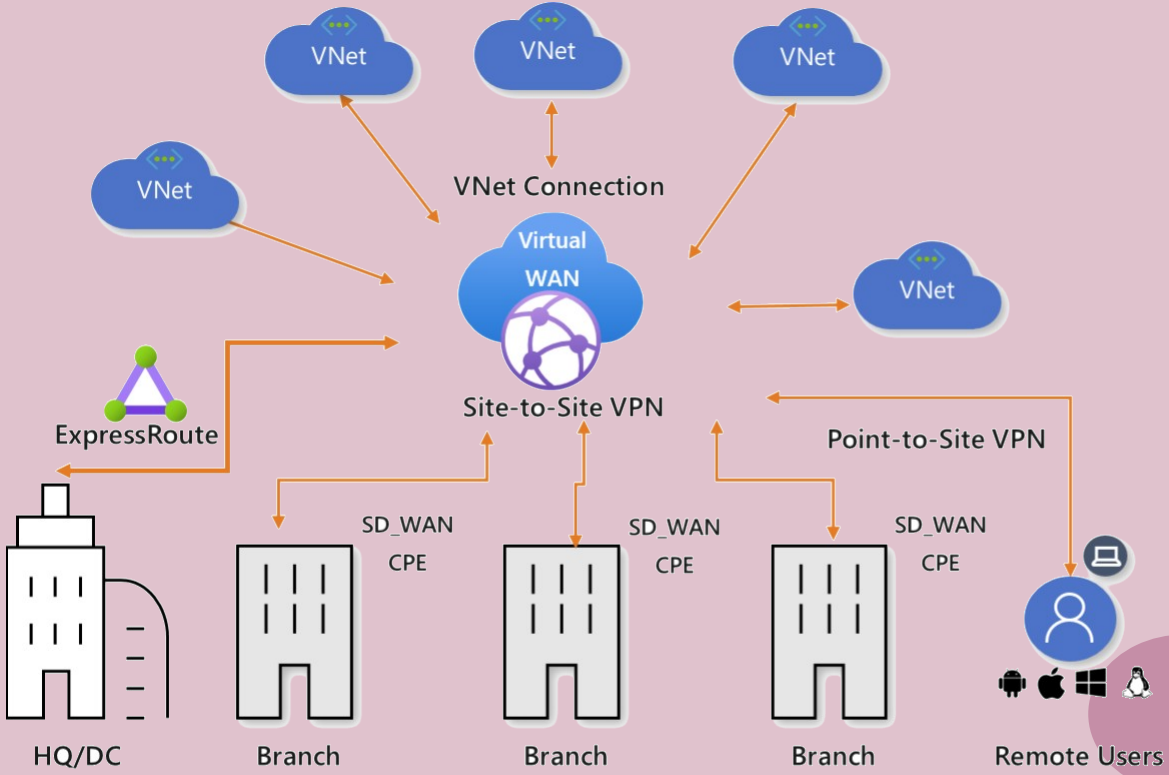
Transit Gateway

- Manage growth
- Highly scalable cloud router
- Better visibility across virtual private clouds or edge connections
- Internal AWS private connections and encryption



TG on Azure[1]

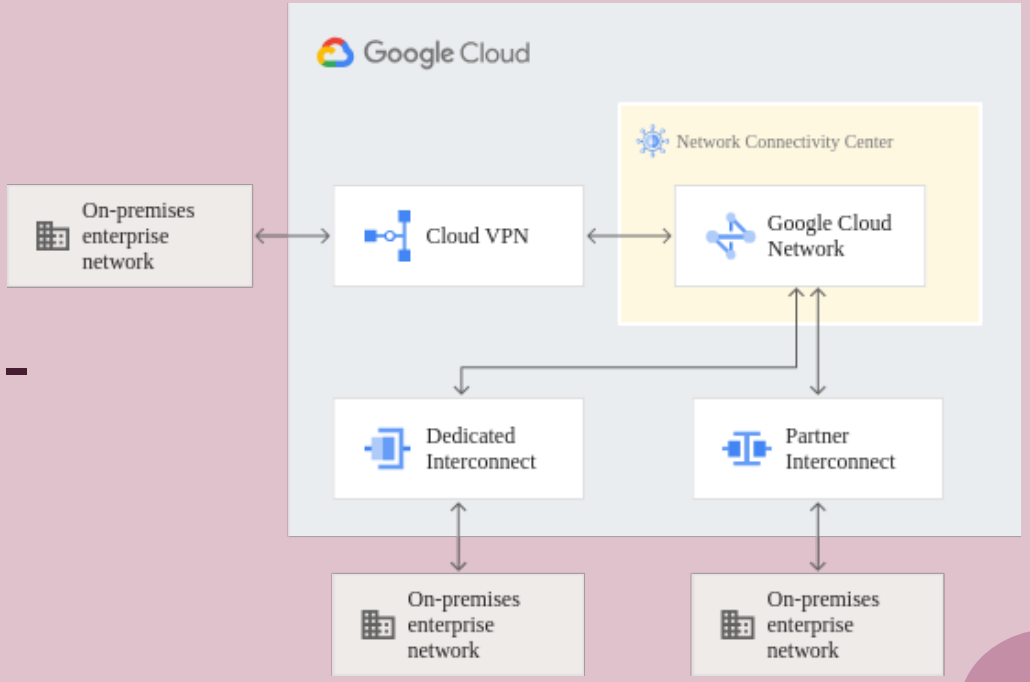
- It is called Virtual WAN
- Global Transit infrastructure
- Hub and spoke architecture



[1] Link to Microsoft

TG on GCP[1]

- It's called Network Connectivity Center
- Yet another implementation of Hub-n-Spoke topology



[1] https://googlecloudarchitect.us/transit-gateway-equivalent-in-gcp/

Lab 2

- Similar to Lab 1
- Create three different VPC:
  - Oregon
  - N. Virginia
  - N. Virginia
- Connect the three of them through a Transit Gateway
- Setup all the traffic to go to the Internet through the Transit Gateway[1]

[1] https://docs.aws.amazon.com/vpc/latest/tgw/transit-gateway-nat-igw.html