AWS re:Invent

NET402

Transit Gateway and Transit VPCs
Reference Architectures for Many VPCs

Nick Matthews
Principal Solutions Architect
AWS

@nickpowpow

AWS re:Invent

aws

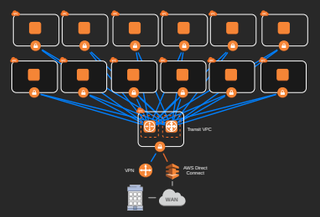
What to expect

How it works

Transit VPC

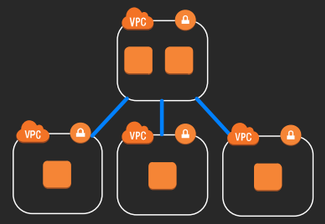
Transit Gateway New

Build out a reference architecture:

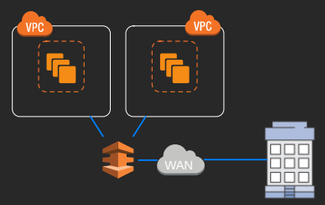


Account Strategy

Segmentation Model

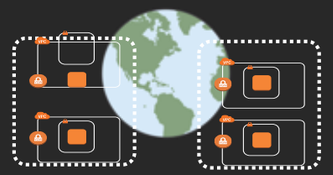


Shared Services



Connectivity

Network Services



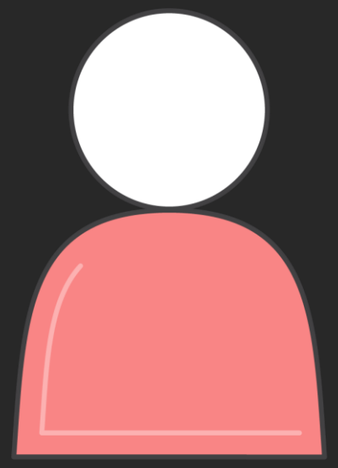
Multi-Region Options

Challenges with many VPCs

VPC management differences

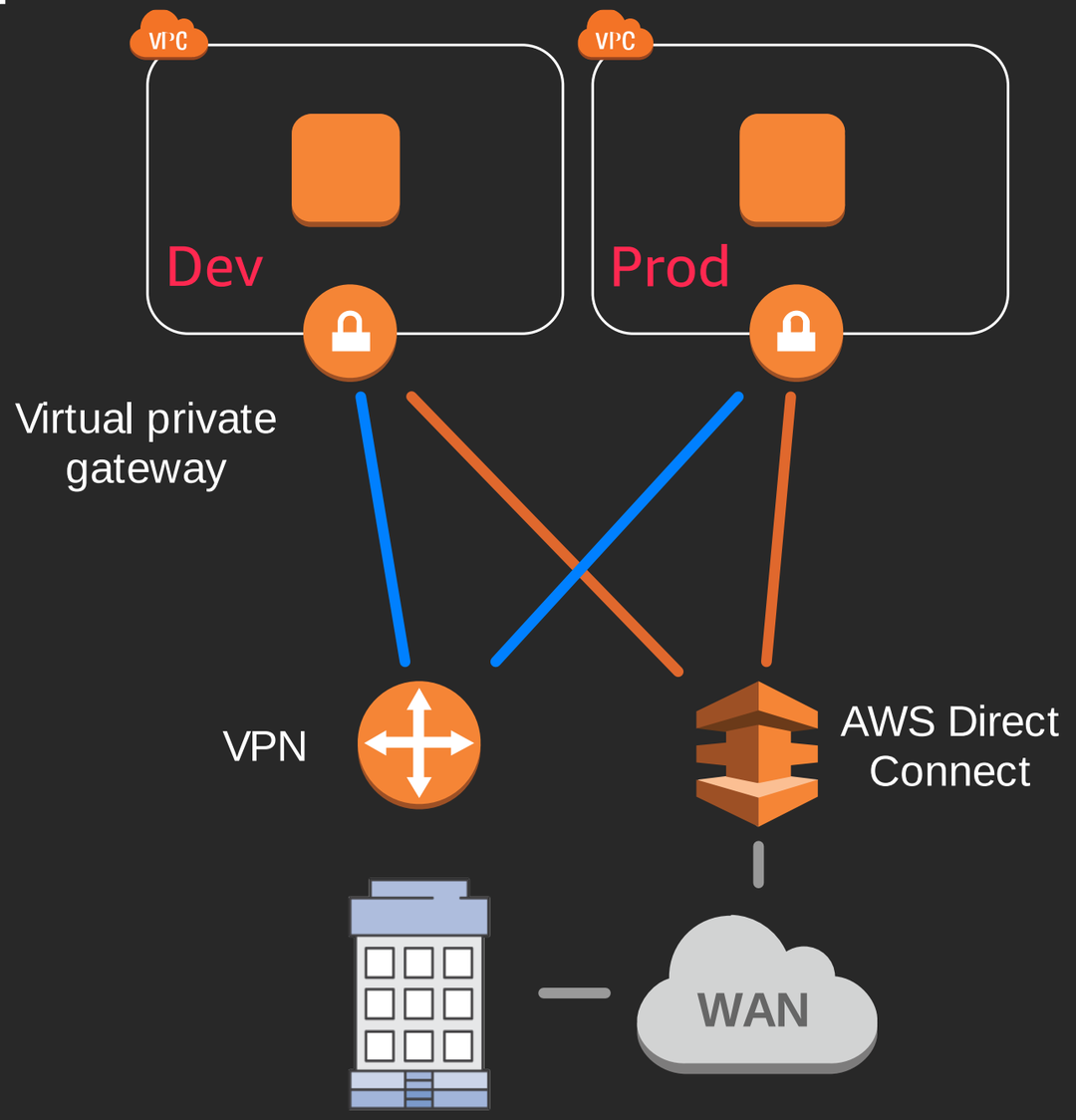
Ease of creation

Access models

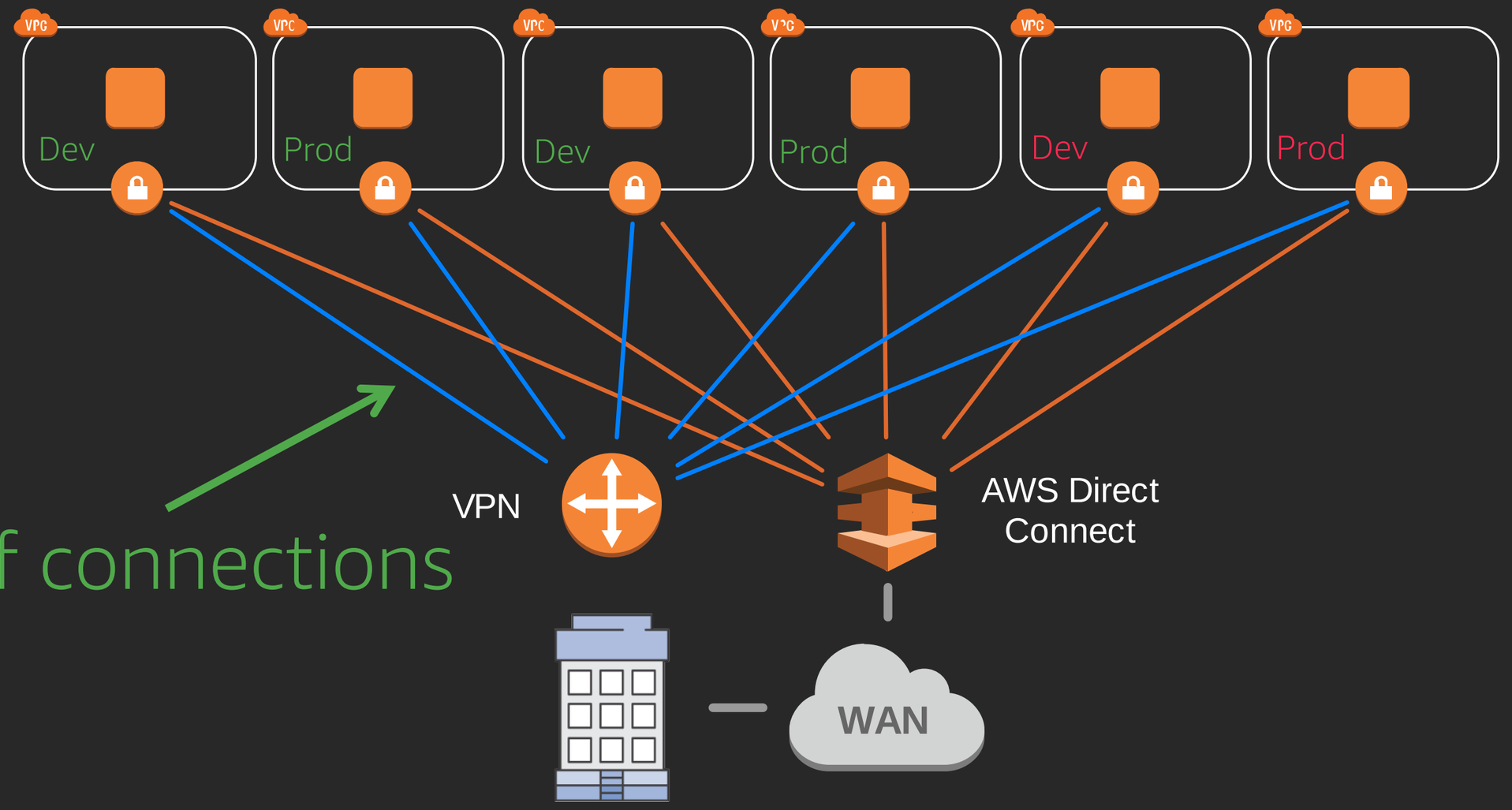


Diverse ownership

Our starting point

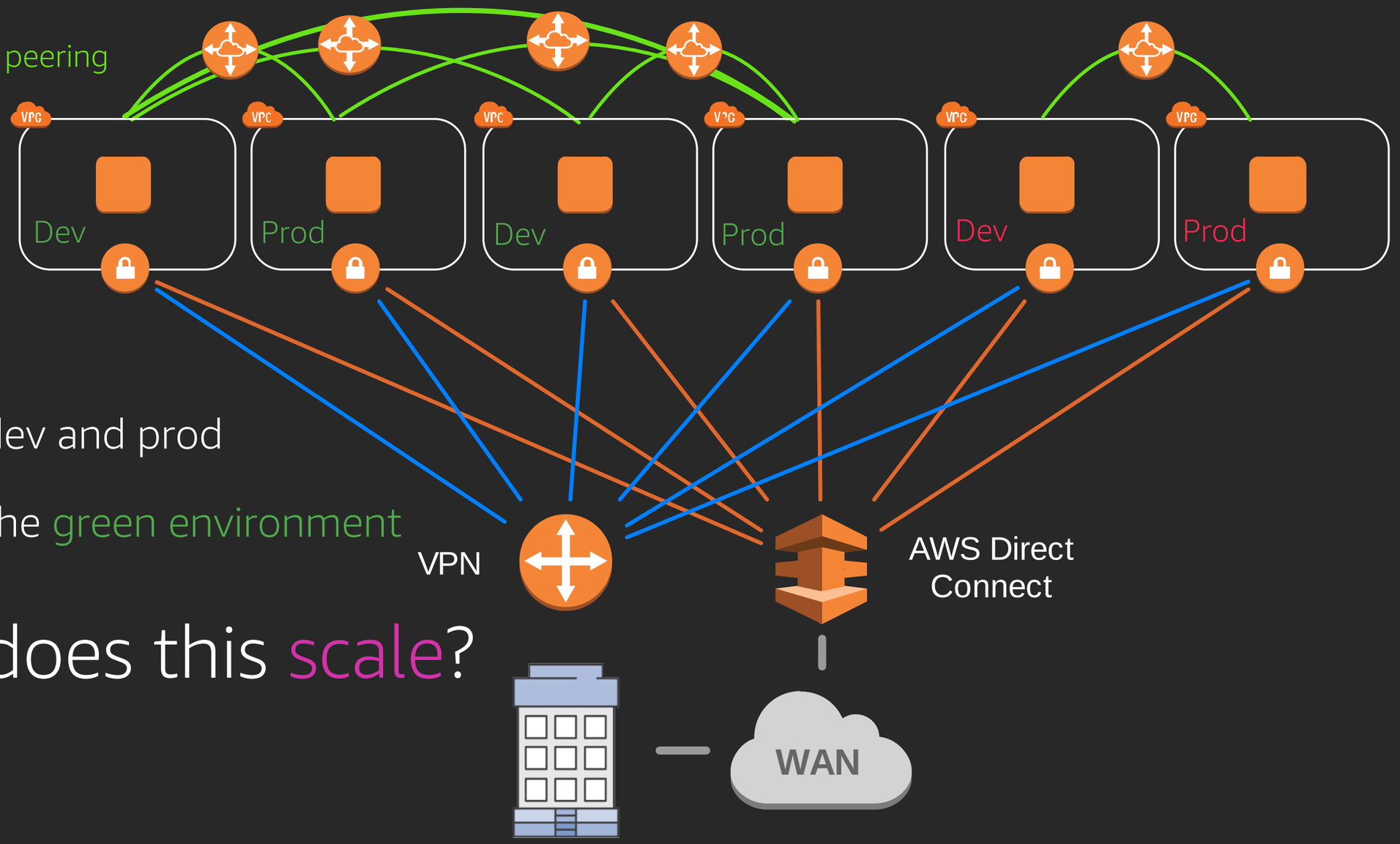


Challenge: Adding more VPCs



Lots of connections

Challenge: Peering VPCs



Let's:

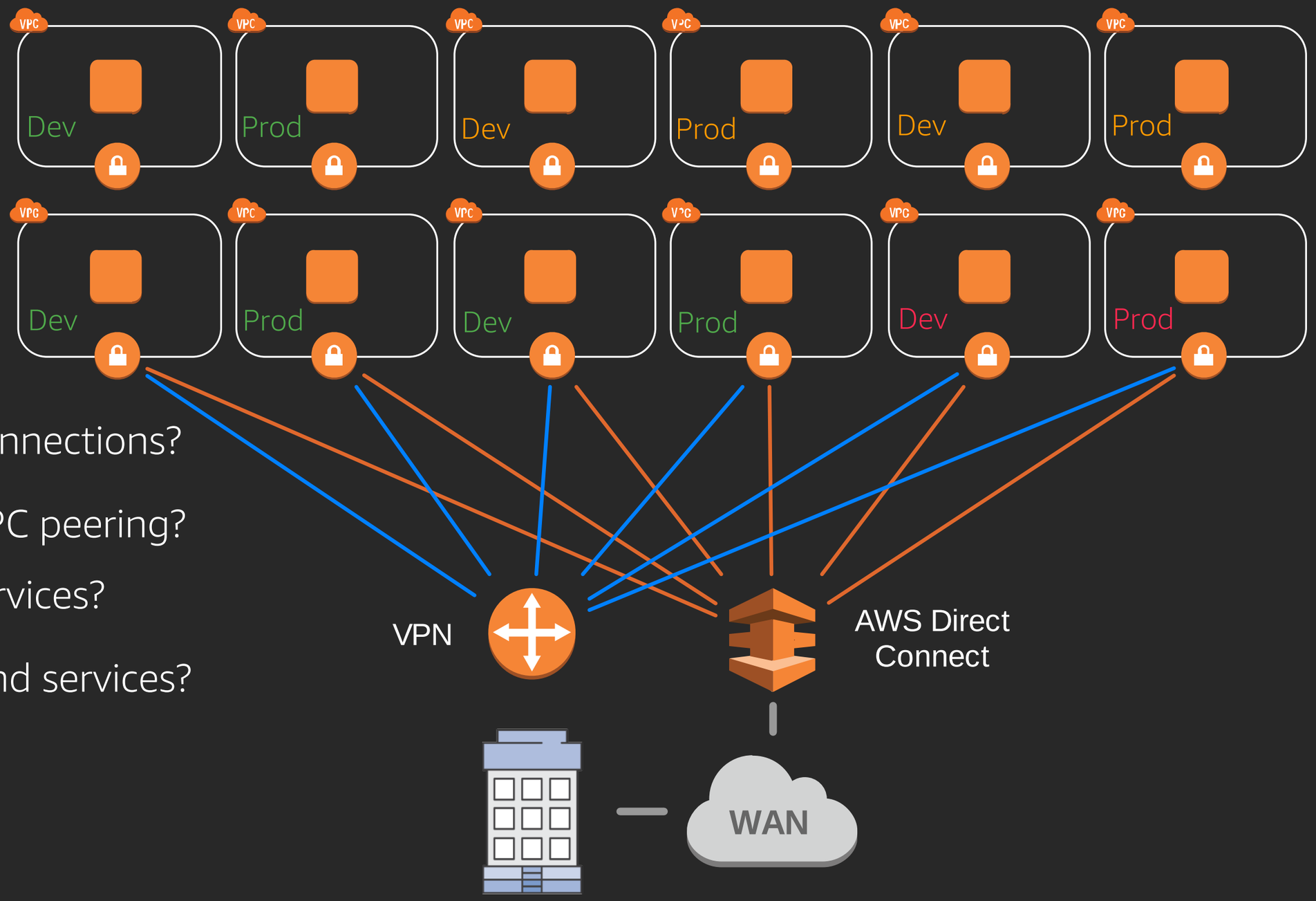
Connect dev and prod

Connect the green environment

How does this scale?

AWS re:Invent

aws



Scaling connections?

Scaling VPC peering?

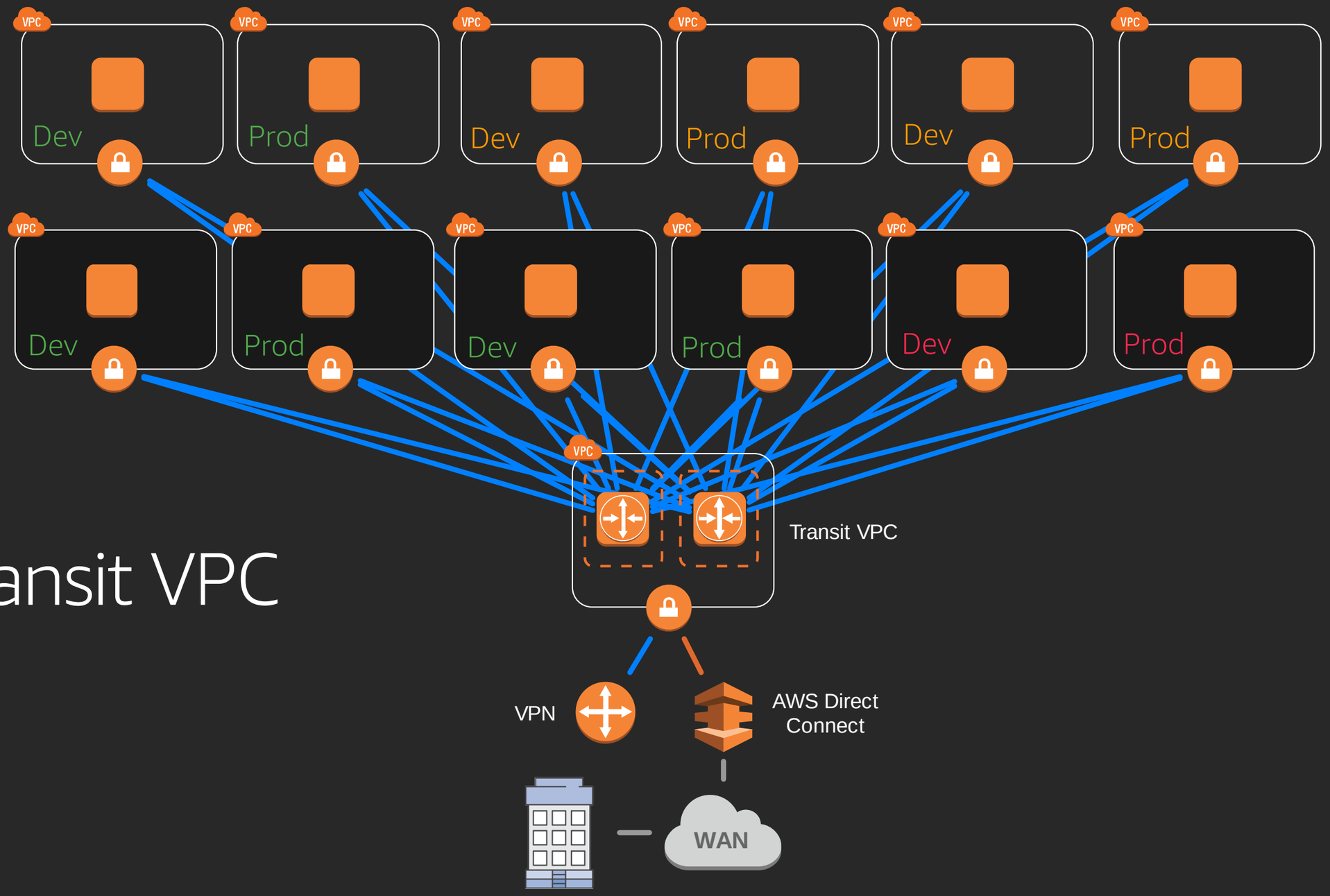
Shared services?

Firewall and services?

AWS re:Invent

aws

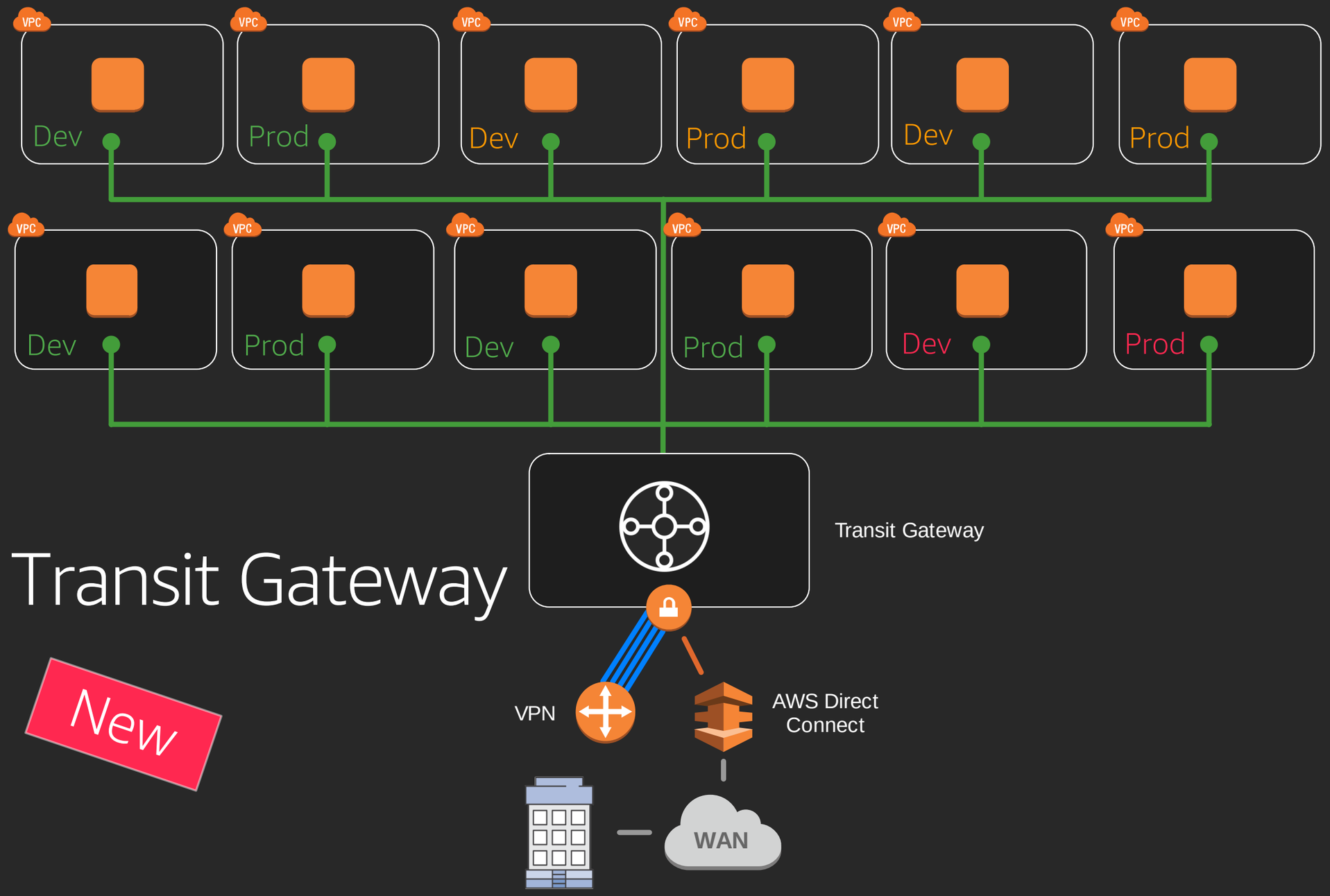
Transit VPC



AWS re:Invent

aws

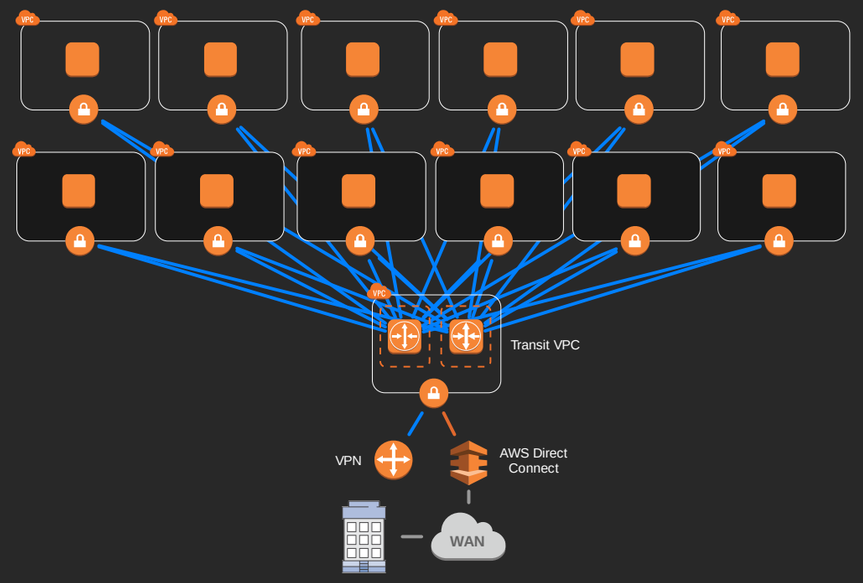
AWS Transit Gateway



New

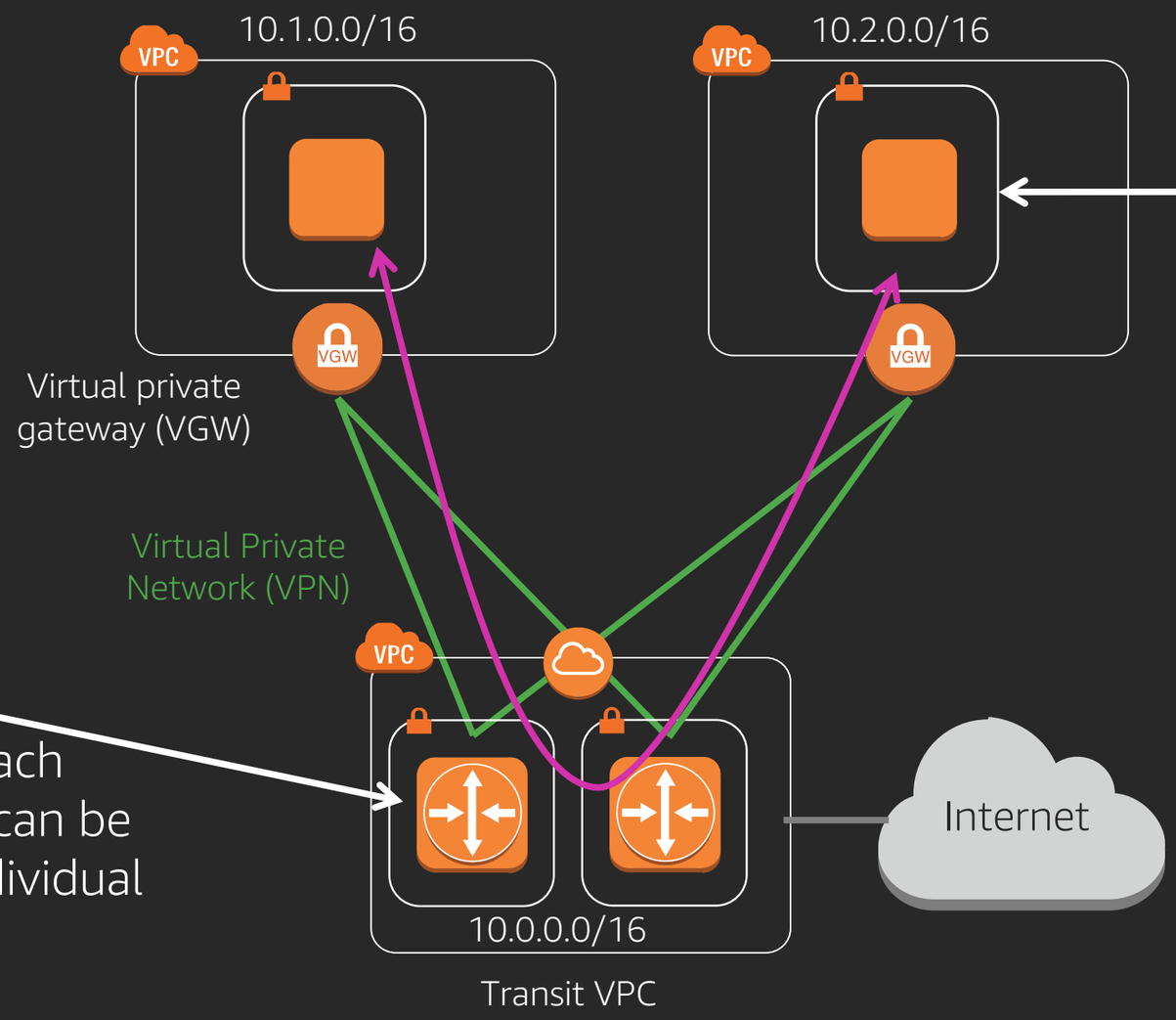
AWS re:Invent

aws



Transit VPC Mechanics

Transit VPC: Routing



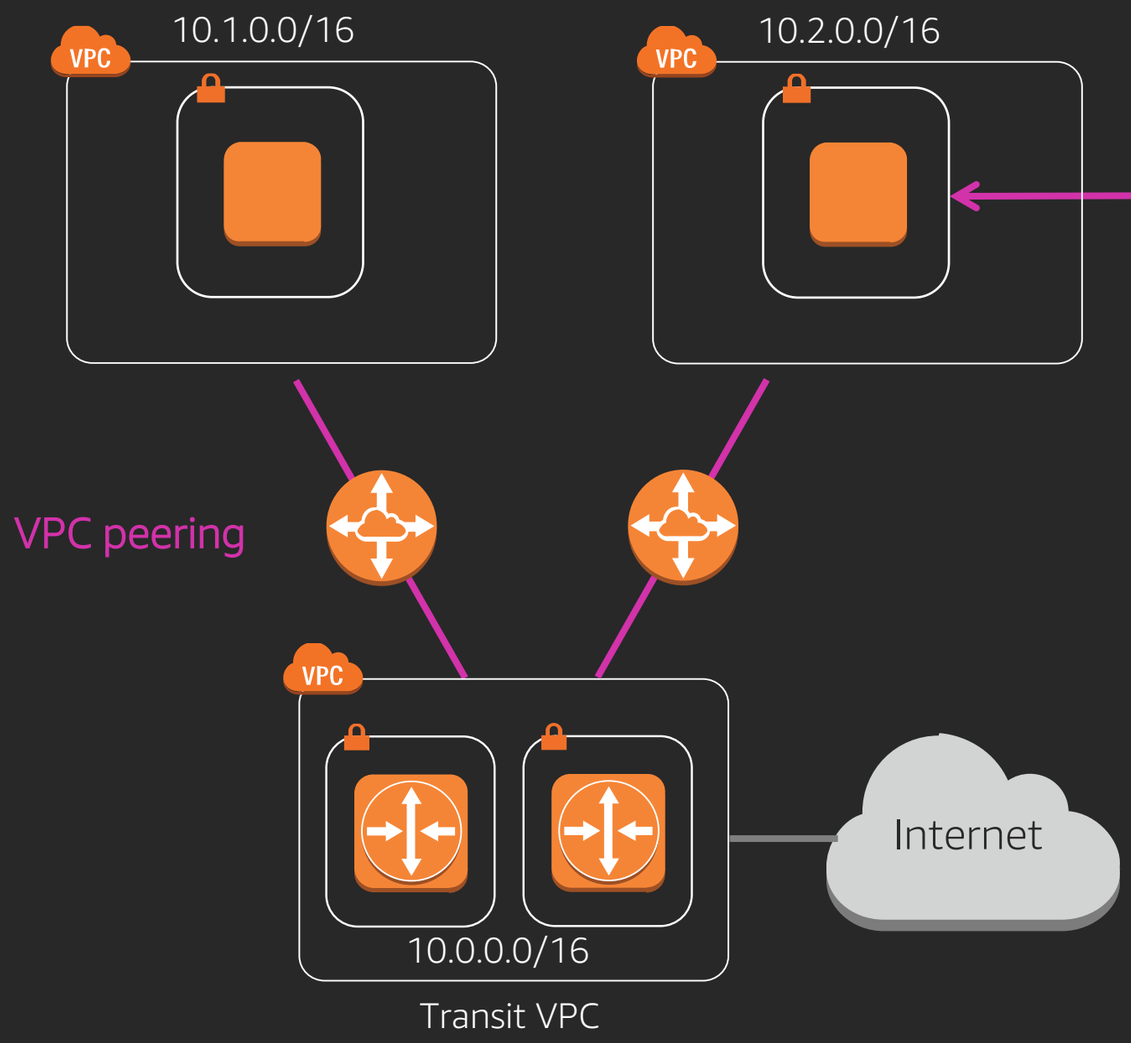
Route table	
Destination	Target
10.2.0.0/16	Local
10.1.0.0/16	VGW

The VPN Instances advertise routes to each VGW with BGP. This can be a default route or individual routes.

AWS re:Invent

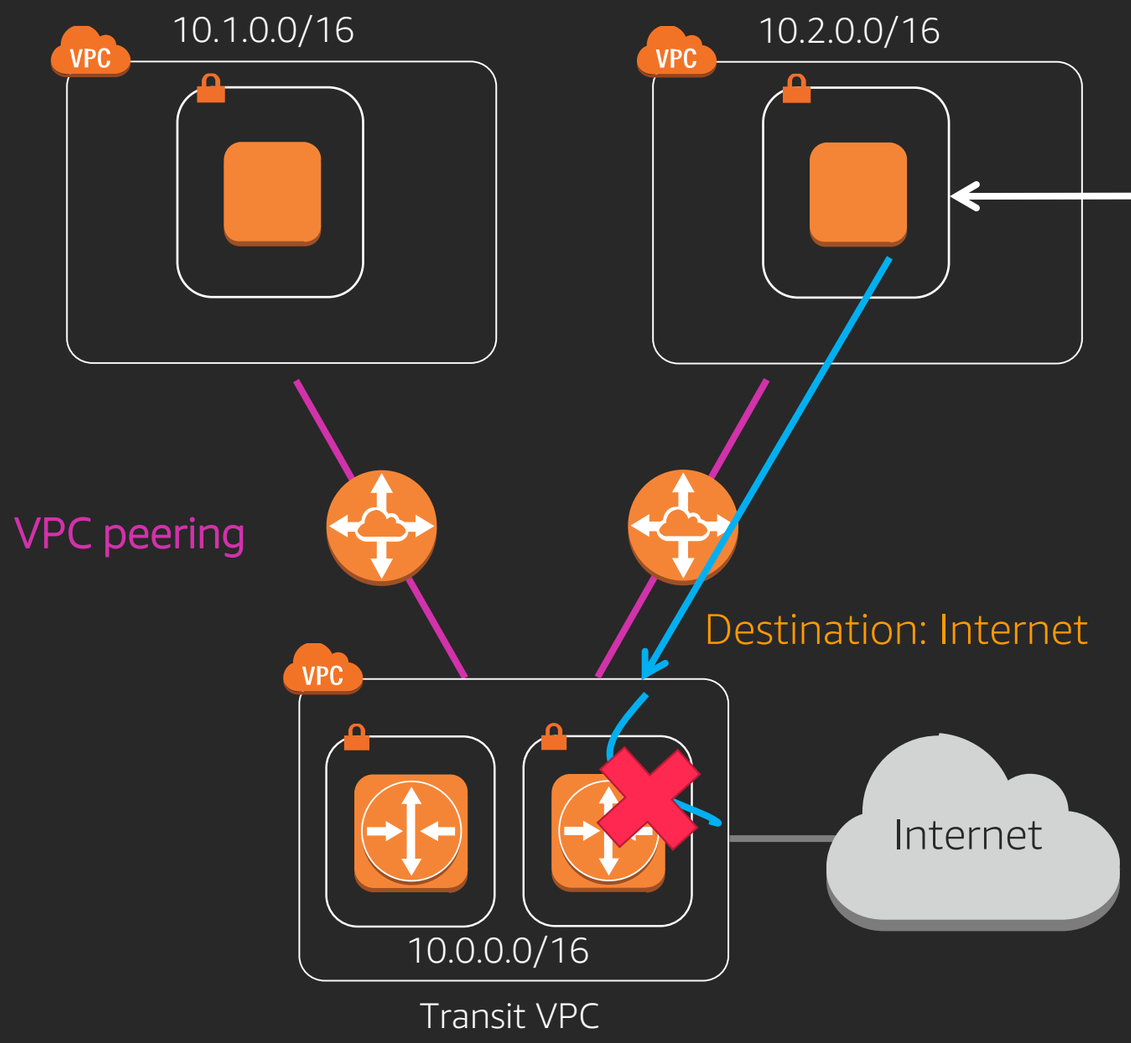
aws

Why doesn't peering work?



Route table	
Destination	Target
10.2.0.0/16	Local
0.0.0.0/0	PCX

Why doesn't peering work?

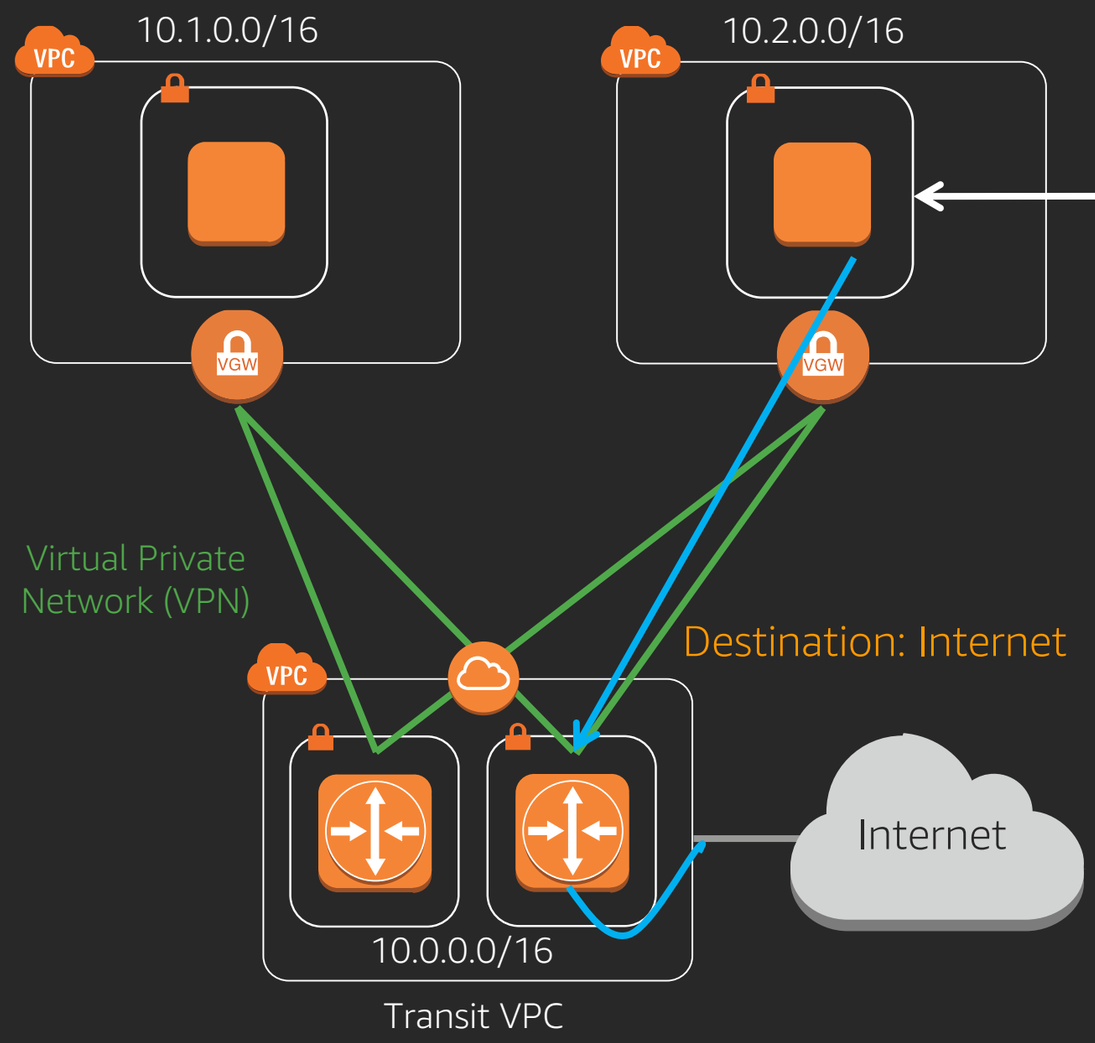


Route table	
Destination	Target
10.2.0.0/16	Local
0.0.0.0/0	PCX

Transitive routing

Traffic must either originate or terminate on a network interface in the VPC

Why does VPN work?



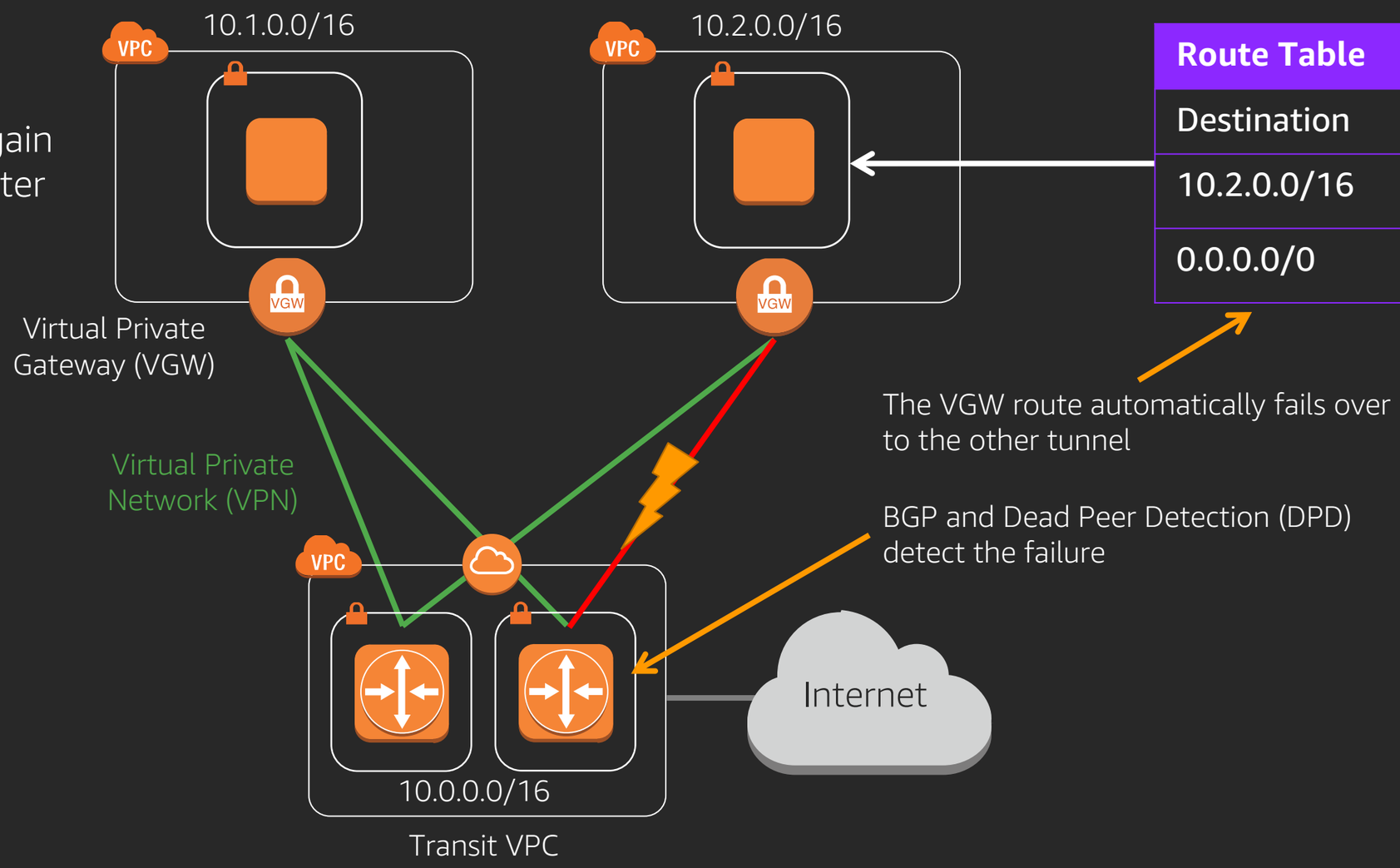
Route table	
Destination	Target
10.2.0.0/16	Local
0.0.0.0/0	VGW

Transitive routing

Traffic must either originate or terminate on a network interface in the VPC

Transit VPC: Availability

Spoiler: We'll use this again with Transit Gateway later



Route Table	
Destination	Target
10.2.0.0/16	Local
0.0.0.0/0	VGW

The VGW route automatically fails over to the other tunnel

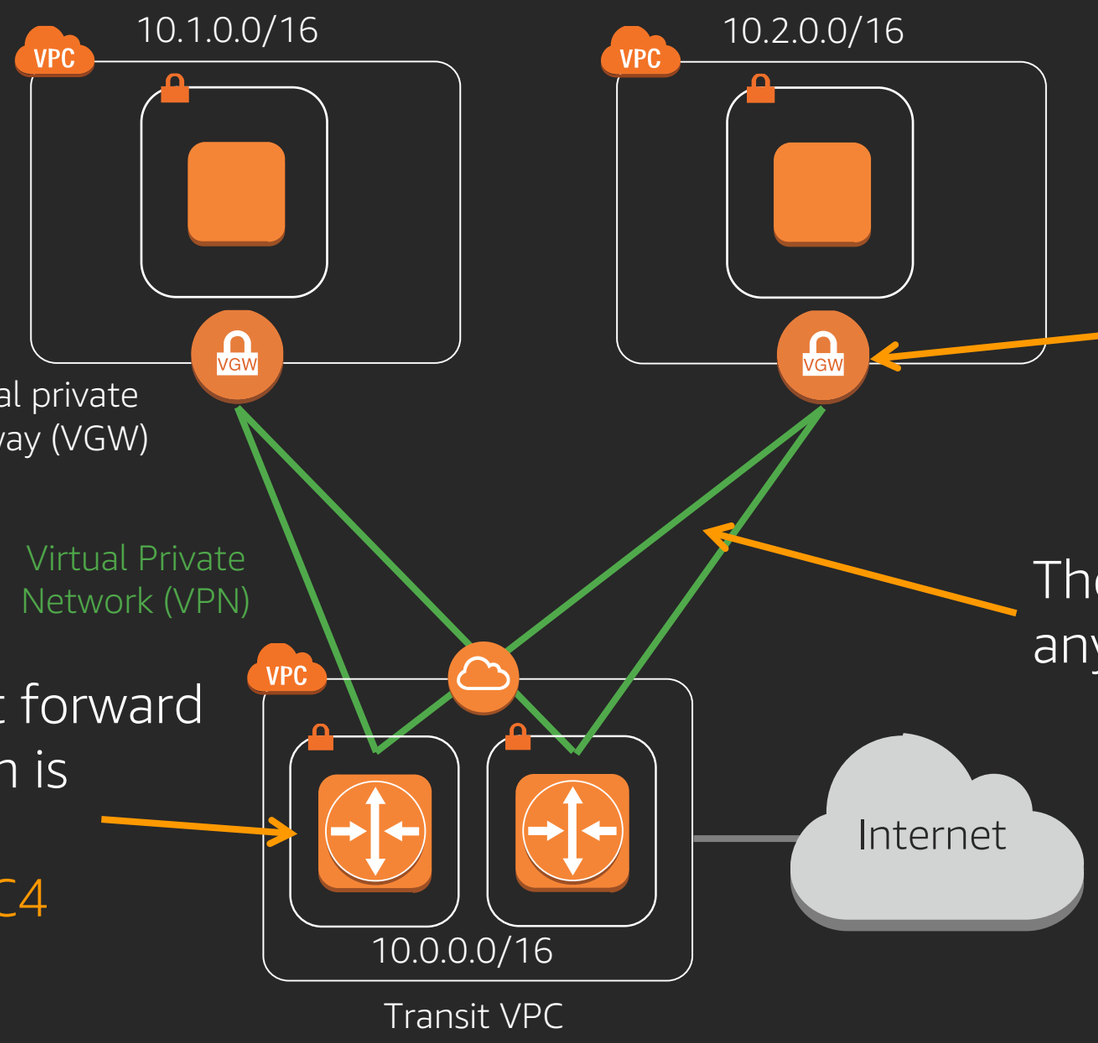
BGP and Dead Peer Detection (DPD) detect the failure

AWS re:Invent

aws

Transit VPC: Performance

Spoiler: We'll need to know this for Transit Gateway also



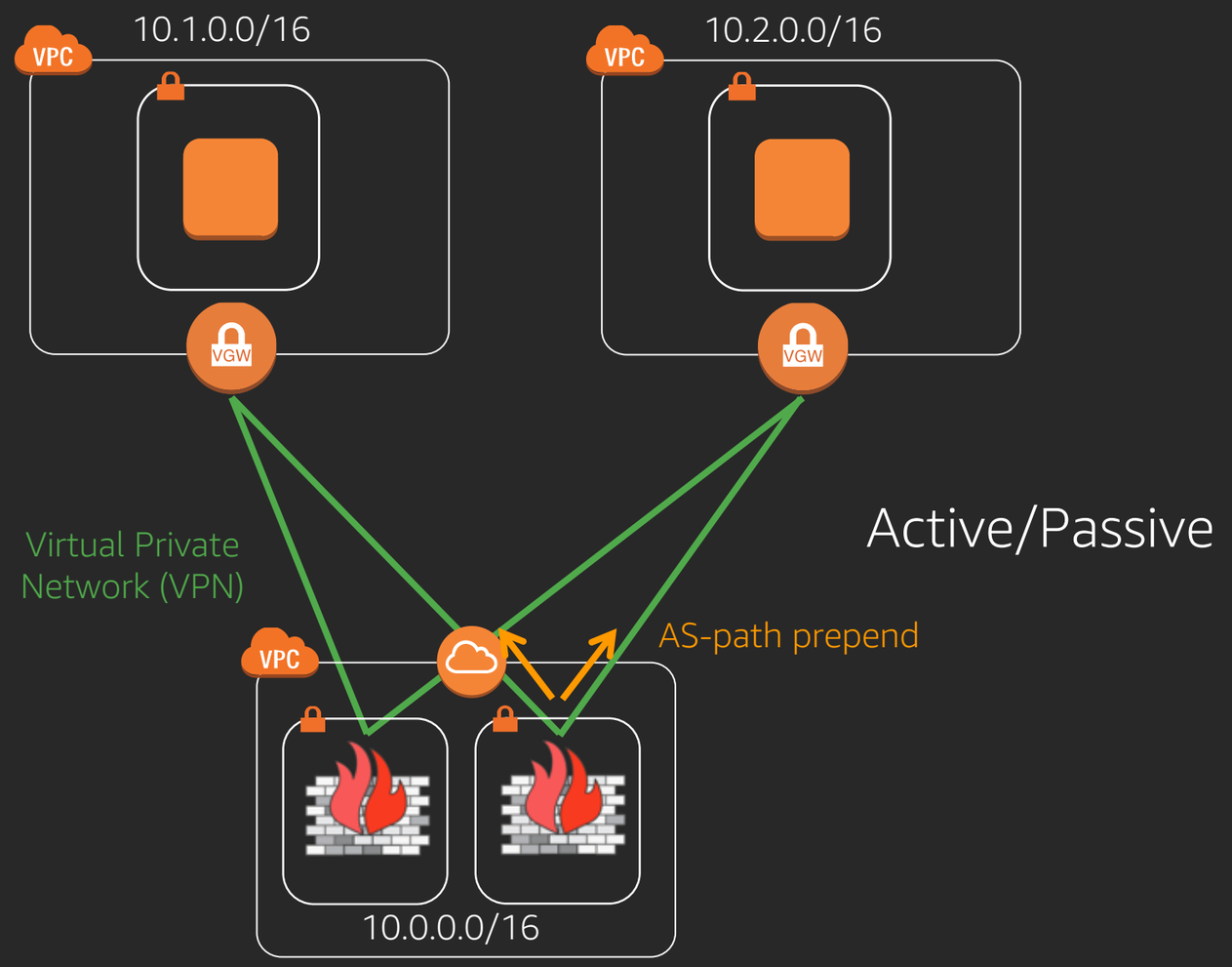
The VGW will only choose a single tunnel for outbound traffic (1.25 gbps)

The VGW accepts packets on any tunnel or connection

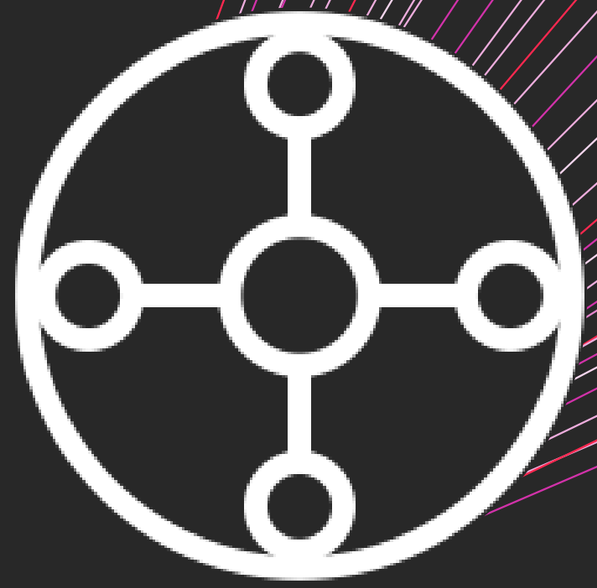
The VPN instance must forward all traffic, the maximum is based on instance size. ~1-3 gbps on M4 and C4 families.

AWS re:Invent

Transit VPC: Security Services



What is the AWS Transit Gateway?



AWS re:Invent

aws

Introducing: Transit Gateway

New

Regional router

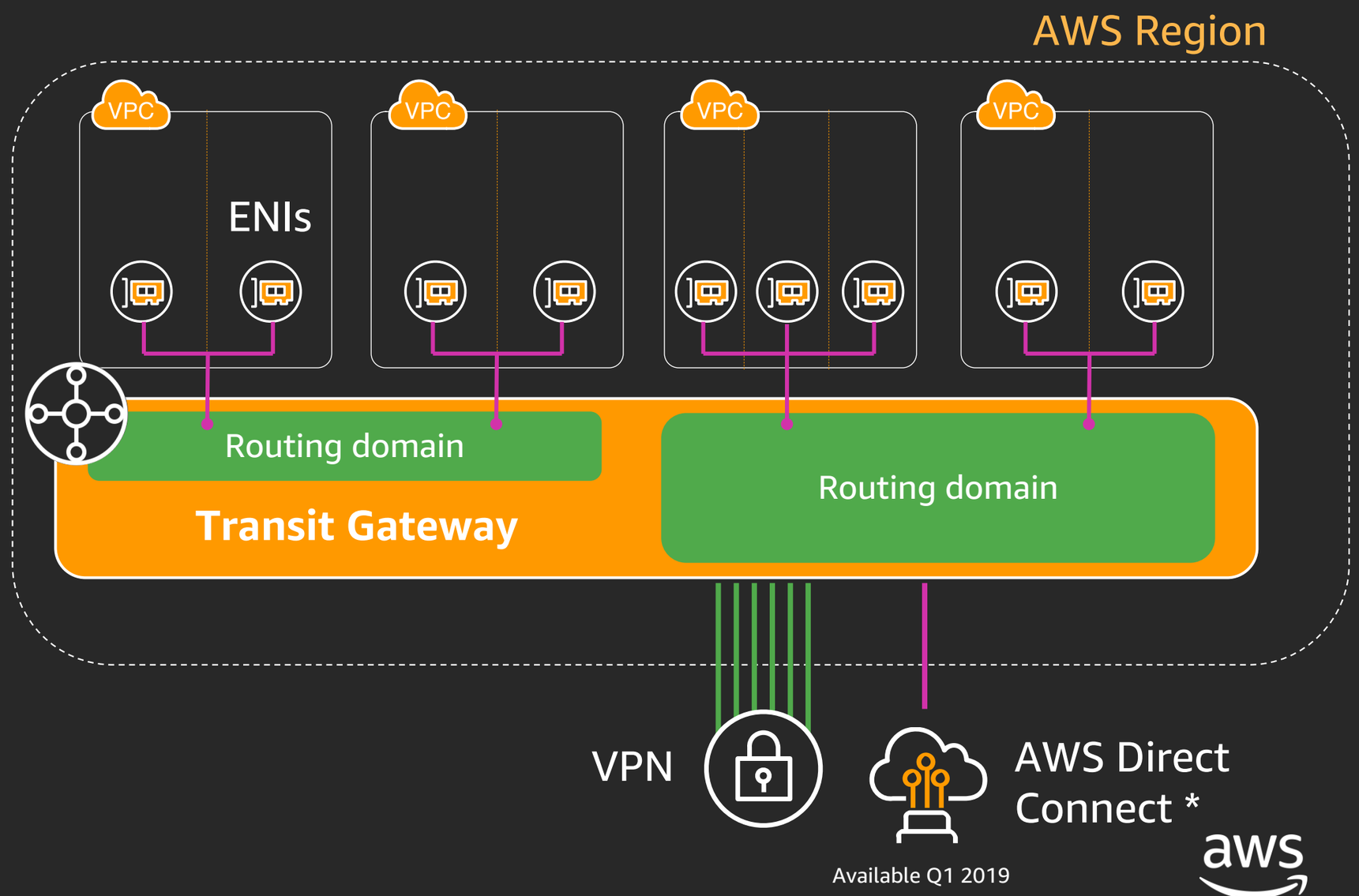
- Centralize VPN and AWS Direct Connect

Scalable

- Thousands of VPCs **across accounts**
- Spread traffic over many VPN connections

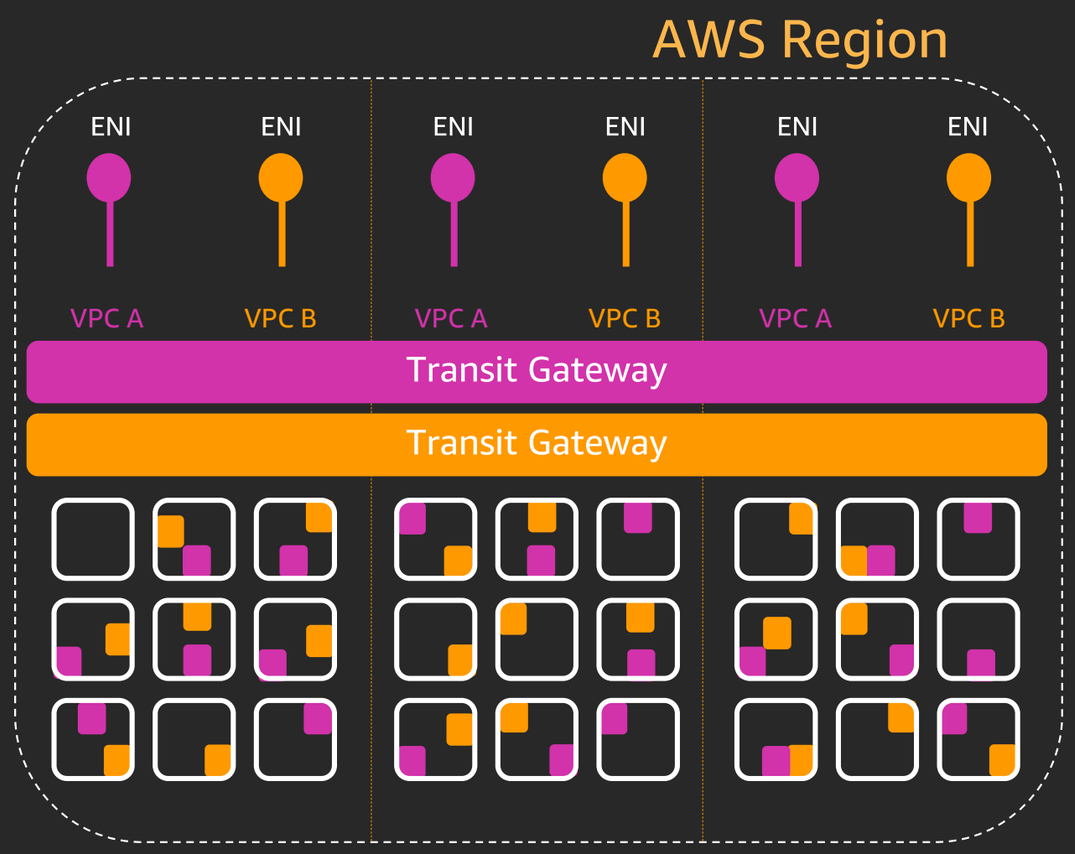
Flexible routing

- Network interfaces in subnets
- Control segmentation and sharing with routing



AWS re:Invent

AWS HyperPlane and AWS Transit Gateway



Attachments

- One network interface per Availability Zone
- Highly available per Availability Zone
- Network capacity shards
- Tens of microseconds of latency

AWS HyperPlane

- Horizontally-scalable state management
- Terabits of multi-tenant capacity
- Supports NLB, NAT Gateway, Amazon EFS and now Transit Gateway

AWS re:Invent

aws

Transit Gateway example time!

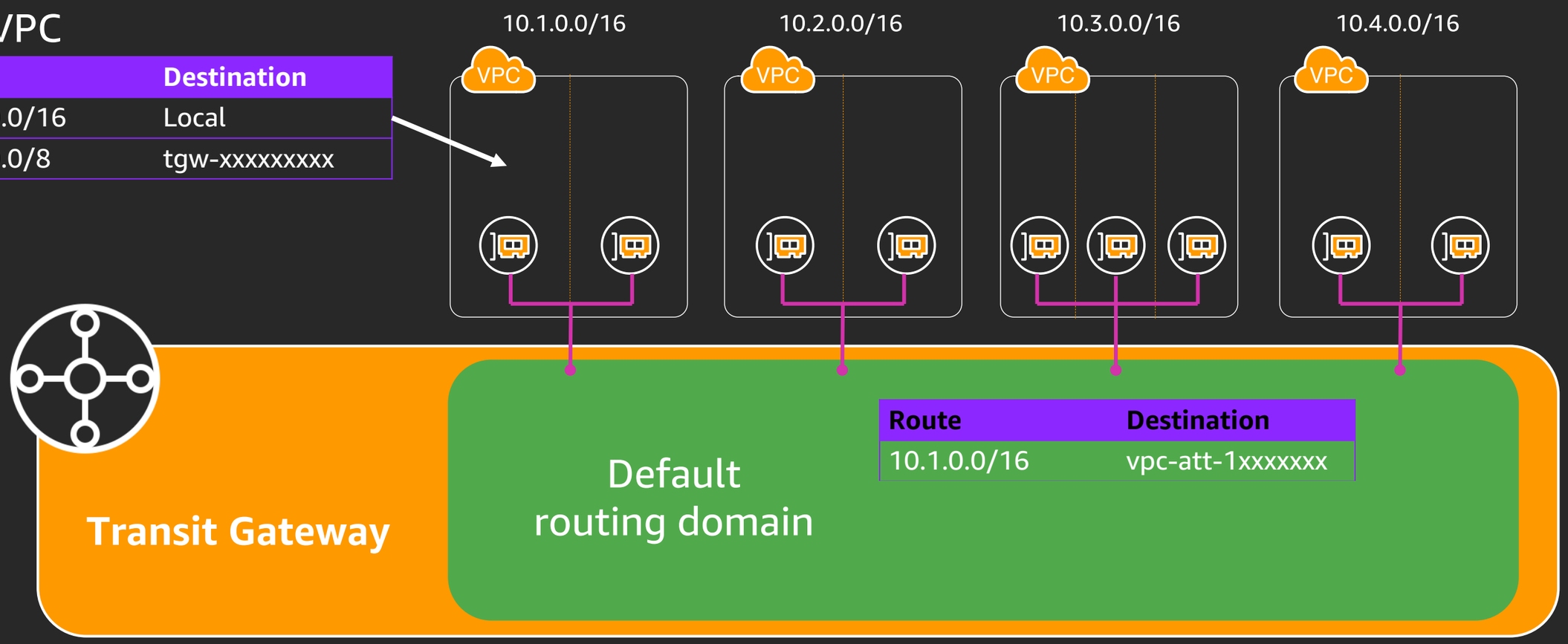
Flat: Every VPC should talk to every VPC!

Isolated: Don’t let anything talk! Send everything back over VPN!

Flat: Transit Gateway route domains (route tables)

Per VPC

Route	Destination
10.1.0.0/16	Local
10.0.0.0/8	tgw-xxxxxxxxx

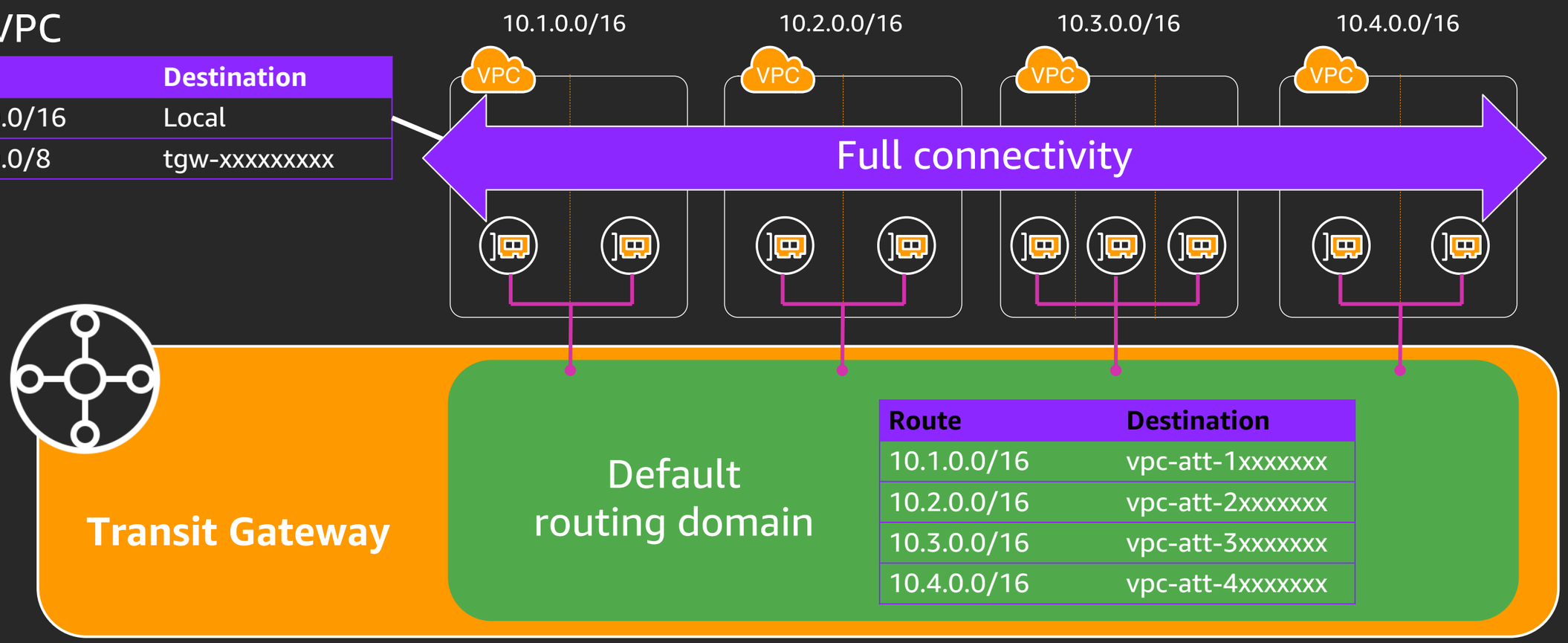


Route	Destination
10.1.0.0/16	vpc-att-1xxxxxxx

Flat: Transit Gateway route domains (route tables)

Per VPC

Route	Destination
10.1.0.0/16	Local
10.0.0.0/8	tgw-xxxxxxxxx



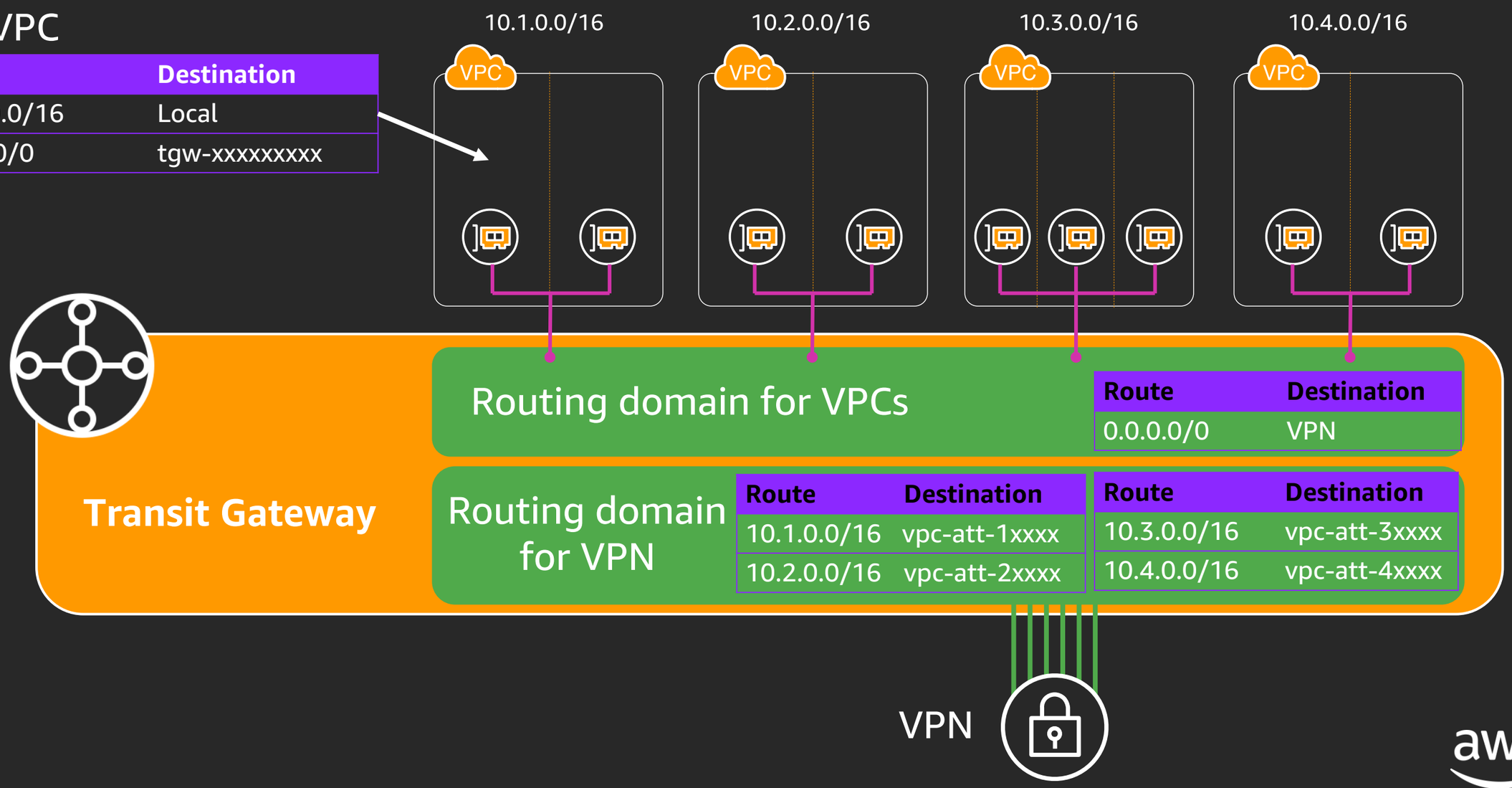
Route	Destination
10.1.0.0/16	vpc-att-1xxxxxxx
10.2.0.0/16	vpc-att-2xxxxxxx
10.3.0.0/16	vpc-att-3xxxxxxx
10.4.0.0/16	vpc-att-4xxxxxxx

Wording warning: In this presentation a route domain is a route table of a Transit Gateway

Isolated: Transit Gateway route domains

Per VPC

Route	Destination
10.1.0.0/16	Local
0.0.0.0/0	tgw-xxxxxxxxx



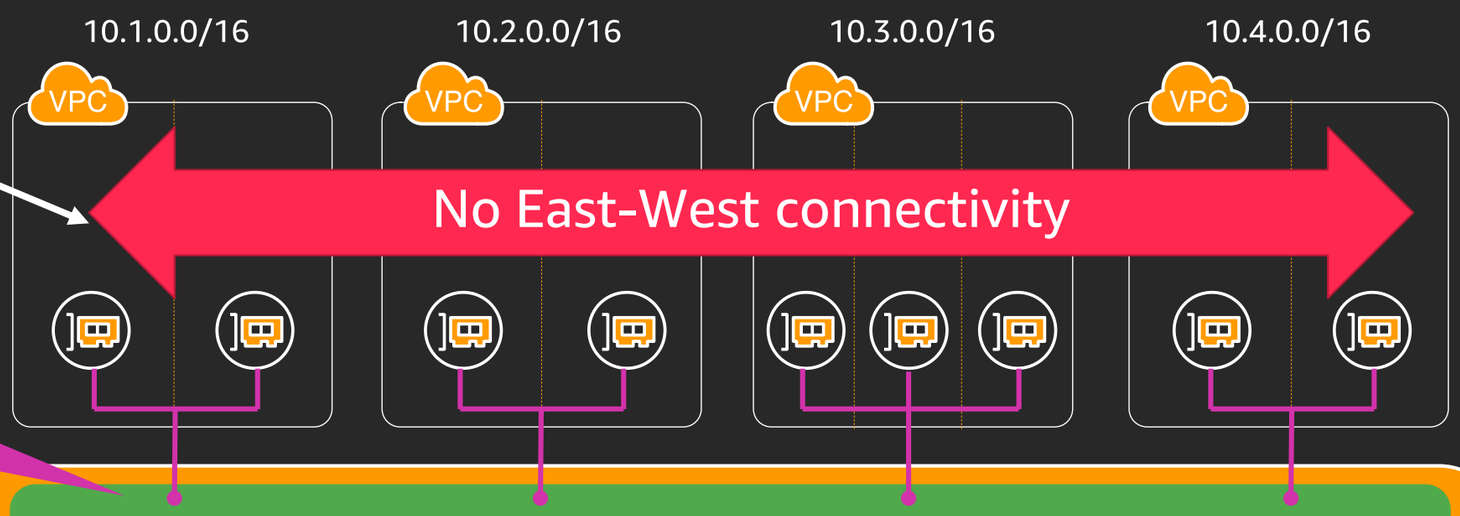
AWS re:Invent

aws

Isolated: Transit Gateway route domains

Per VPC

Route	Destination
10.1.0.0/16	Local
0.0.0.0/0	tgw-xxxxxxxxx



Attach VPCs to route domains to determine where it can **go**

Transit Gateway

Routing domain for VPCs

Route	Destination
0.0.0.0/0	VPN

Routing domain for VPN

Route	Destination
10.1.0.0/16	vpc-att-1xxxx
10.2.0.0/16	vpc-att-2xxxx

Route	Destination
10.3.0.0/16	vpc-att-3xxxx
10.4.0.0/16	vpc-att-4xxxx

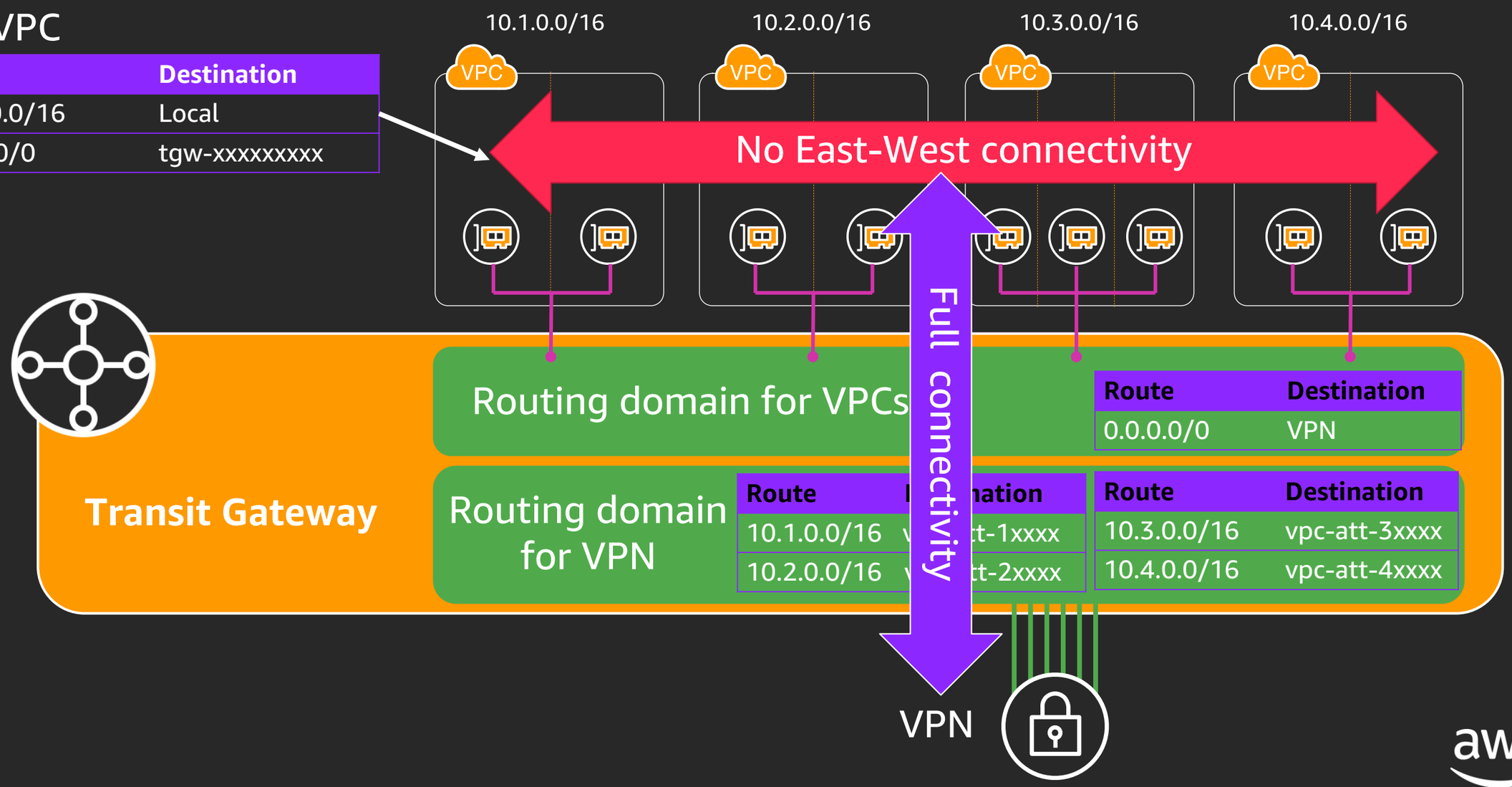
Propagate routes to places that **can reach** the attachment

VPN

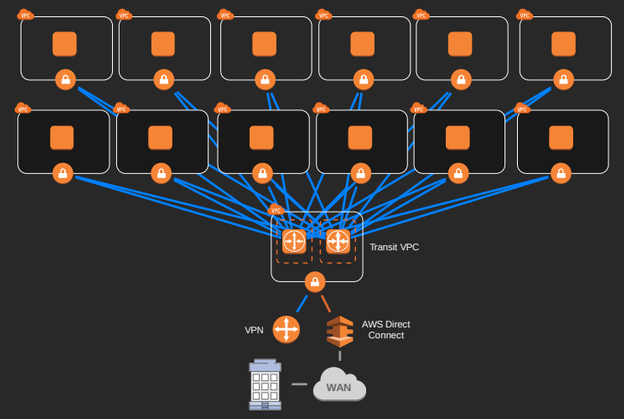
Isolated: Transit Gateway route domains

Per VPC

Route	Destination
10.1.0.0/16	Local
0.0.0.0/0	tgw-xxxxxxxxx



Quick comparison: Transit Gateway and Transit VPC



Transit VPC

- Customer managed instances
- Uses VPN and virtual private gateways
- Hard to scale and manage
- Difficult to segment

Transit Gateway

- AWS native service
- Uses elastic network interfaces
- Scales horizontally
- Flexible segmentation

Transit Gateway details

Find on YouTube

NET 331: NEW LAUNCH: Introduction to Transit Gateway

AWS re:Invent

aws

Are there any reasons to use a Transit VPC?

- You currently use one, and it works for you
  - Migration to Transit Gateway
- Additional visibility and monitoring
- Automated VPC networking using tagging
- You want to use additional services:
  - Security features
  - SD-WAN
  - NAT
  - Proprietary features

We will cover how adding Transit Gateway makes these easier

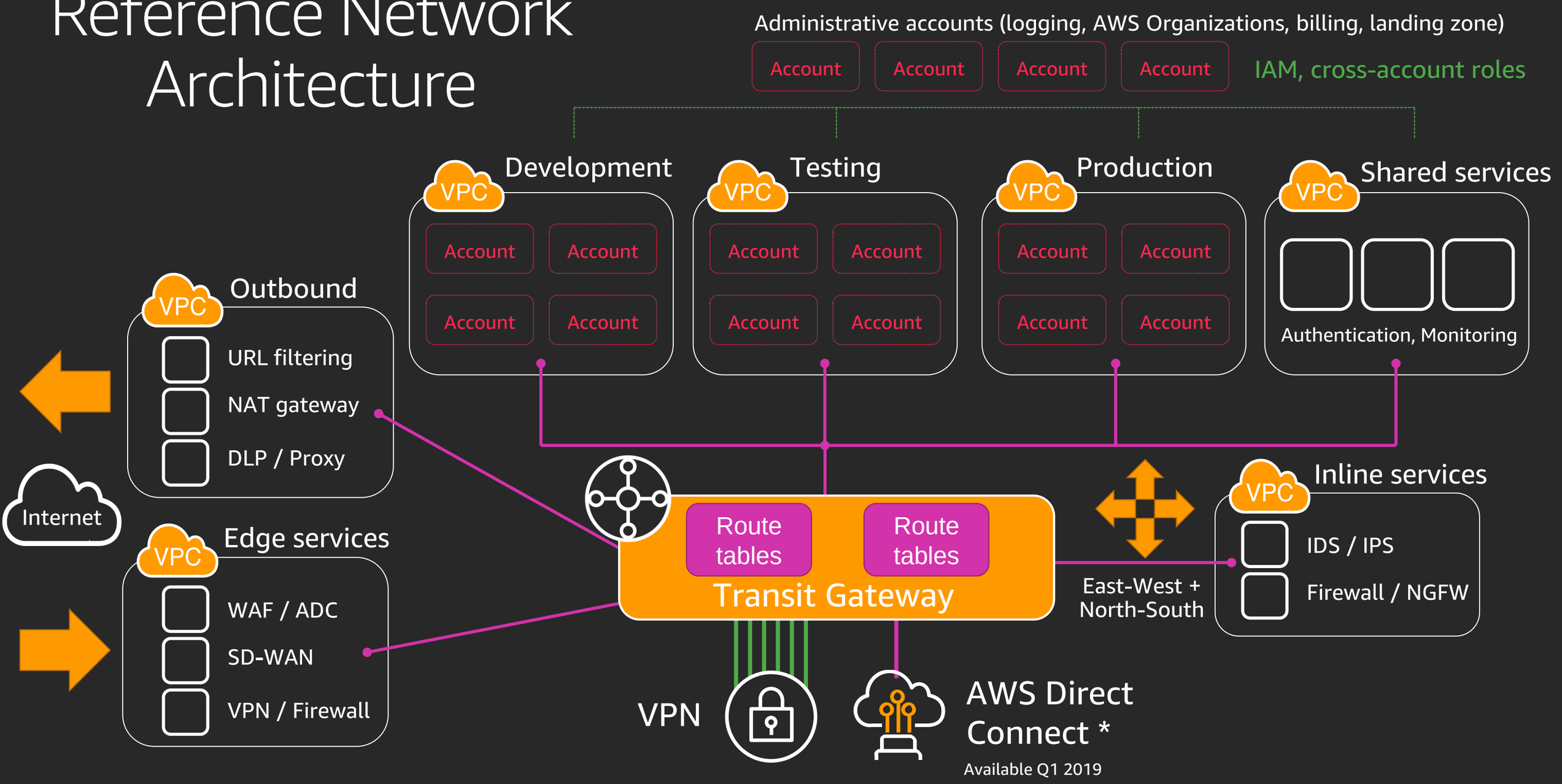
AWS re:Invent

We're only adding things

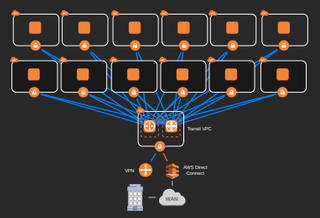
You can use all existing options with Transit Gateway:

- VPC peering
- AWS Direct Connect
- Elastic Load Balancing
- AWS PrivateLink
- AWS CloudWatch metrics
- AWS CloudFormation
- Transit VPC

Reference Network Architecture

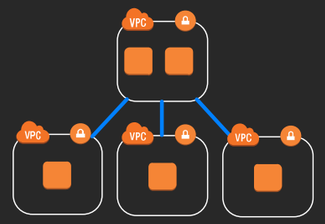


Architecture walk through

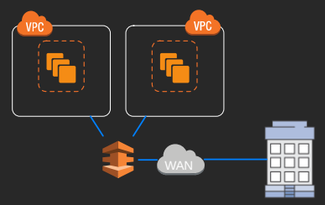


Account strategy

Segmentation model

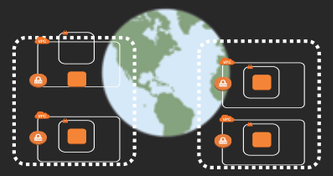


Shared services



Connectivity

Network services



Multi-region options

Account strategy

Account and VPC segmentation

Larger VPCs or accounts

AWS Identity and Access Management
Strict security groups and routing
Identifying resources with tags

Policy and IAM

Smaller VPCs or accounts

Automation of infrastructure
AWS Direct Connect and VPN standards
Subnet and routing standards

Infrastructure and Networking

Why not both?

Provide granular account control
with centralized infrastructure

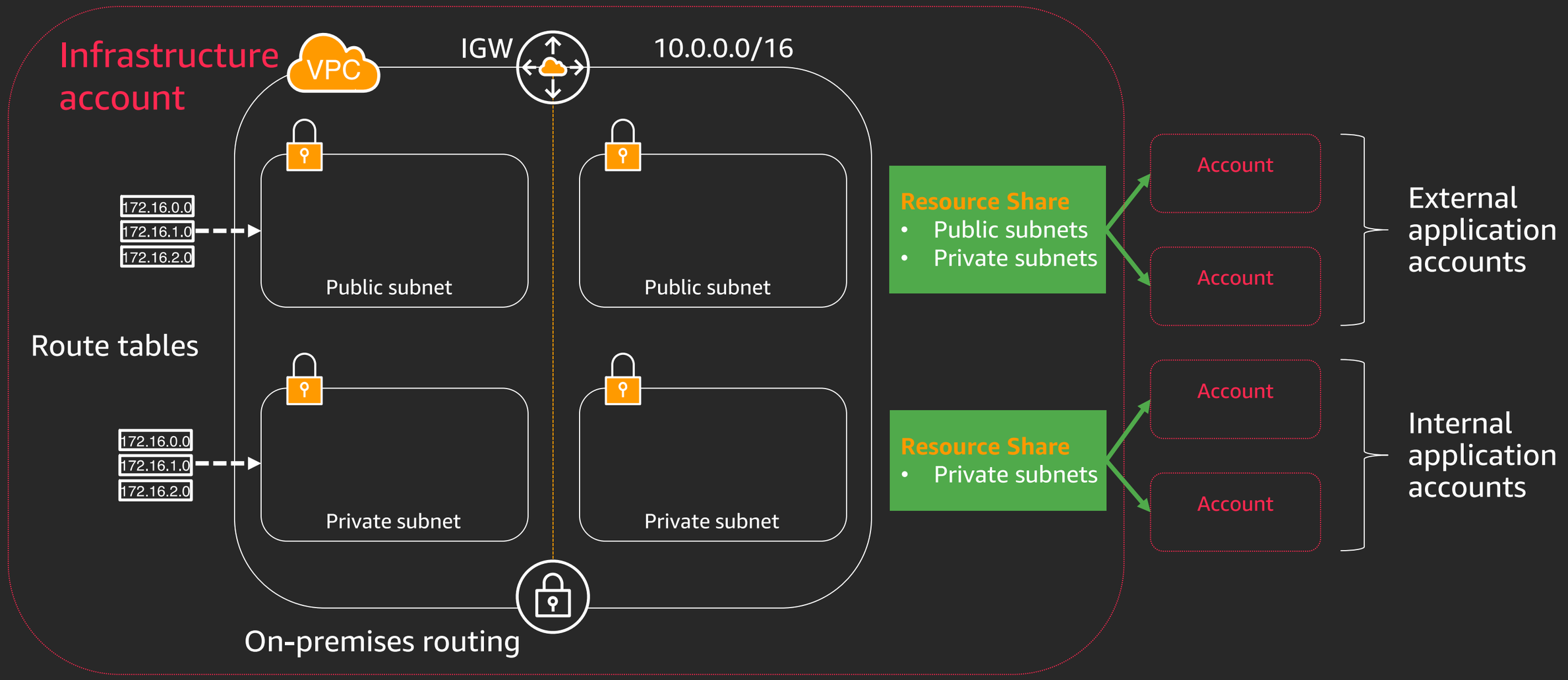
AWS re:Invent

aws

VPC Sharing and Resource Access Manager

New

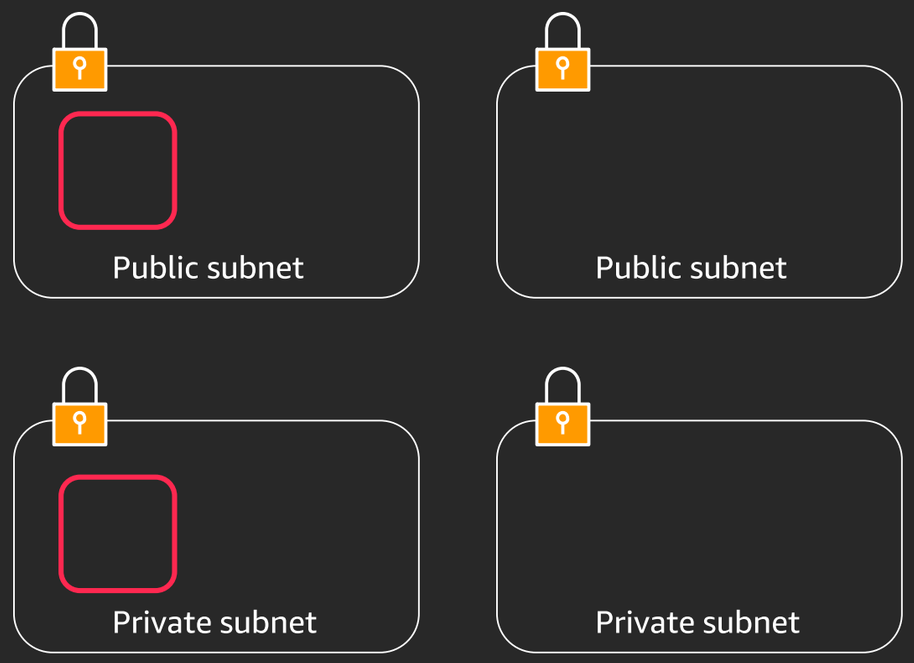
Share subnets between accounts in an AWS Organization

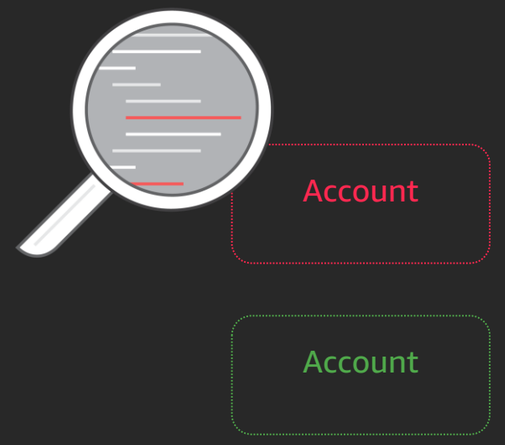


VPC Sharing and Resource Access Manager

New

Account owners only see subnets and their resources





External accounts

Internal accounts

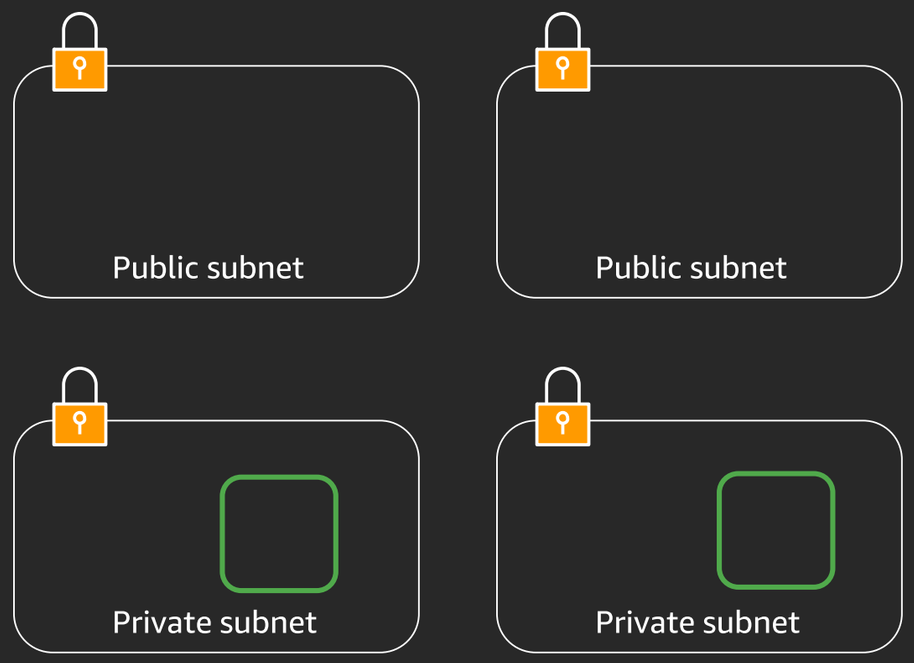
Accounts see their resources and have control:

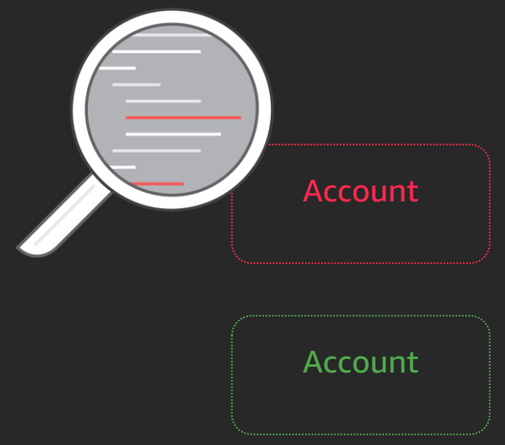
- Security groups
- Data
- Instance details
- Account configuration

VPC Sharing and Resource Access Manager

New

Account owners only see subnets and their resources





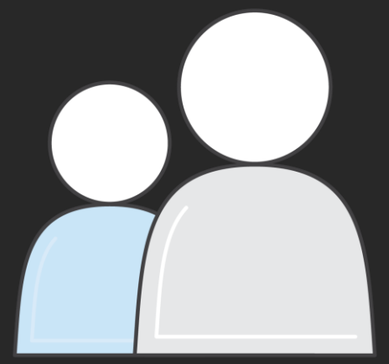
External accounts

Internal accounts

Accounts see their resources and have control:

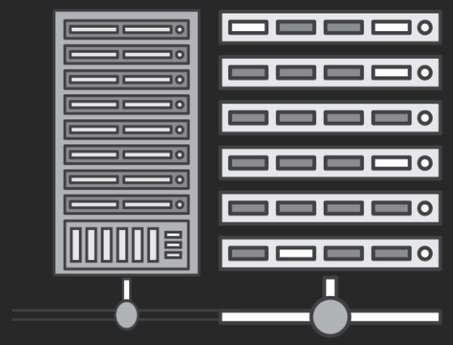
- Security groups
- Data
- Instance details
- Account configuration

VPC Sharing benefits



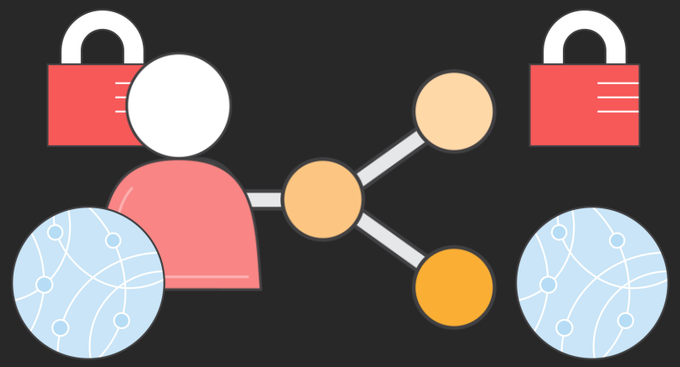
Separation of duties

- Infrastructure strictly controls routing, IP addresses, and VPC structure
- Developers own their resources, accounts, and security groups



Less unused resources

- Higher density subnets, add up to 5 additional CIDRs
- More efficient use of VPN and AWS Direct Connect



Decouple accounts and networks

- Account protection and billing without additional infrastructure
- Many accounts with fewer networks
- Avoid VPC peering charges

Other account considerations

One size does not need to fit all

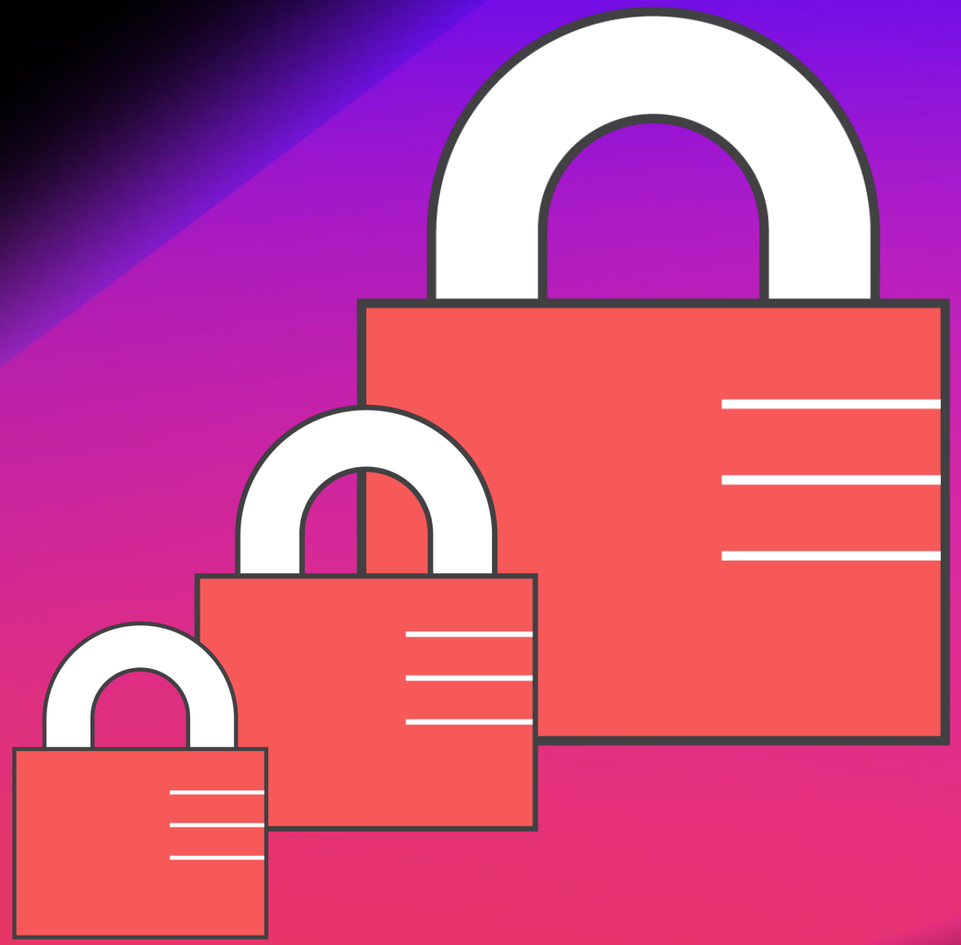
- Example: production may use separate VPCs, development can use a shared VPC
- **AWS Transit Gateway can handle large amounts of VPCs if needed**

VPC Sharing works within an AWS Organization

VPC Sharing doesn’t restrict resource utilization

- NAT gateways, VPN, subnet address space, and security groups have shared limits
- VPC Sharing doesn’t change any VPC limits, only account limits
- Give highly scalable services like AWS Lambda dedicated IP space

Segmentation



Segmentation: Decision inputs

Relationship between accounts, VPCs, and tenants?

- Do accounts and tenants trust each other?
- Is the current network segmentation intentional or a side effect?

Who owns security and networking?

- Each team or a centralized team?

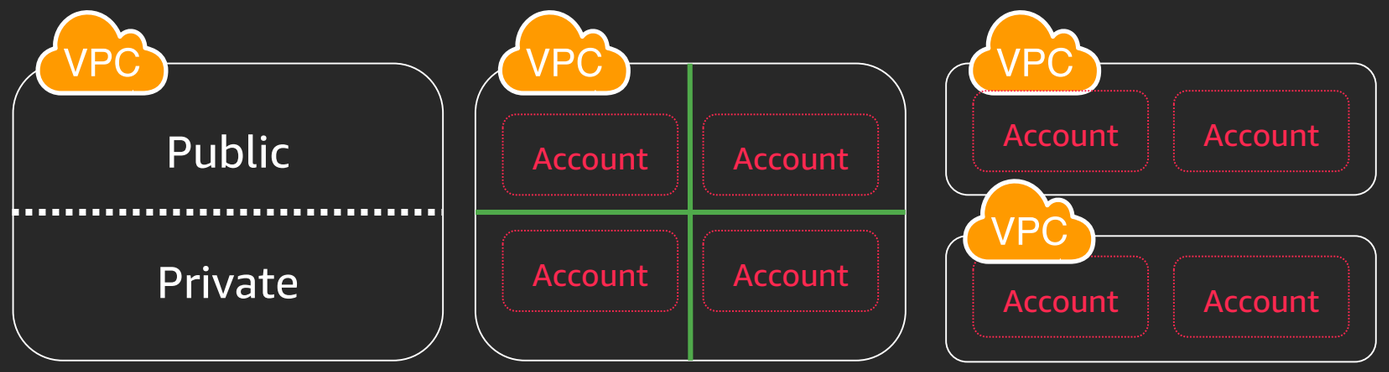
Compliance and governance requirements?

- Scope can be reduced at an account or a VPC level

Segmentation options: Layers

Inside the account
- IAM users and roles
- Security groups

At the VPC
- Route tables
- Network ACLs
- Separate VPCs



ACLs

Tenant configuration

Tenant and infrastructure Shared Security line

Infrastructure configuration

Baseline security

IAM: Control actions and privileges inside the account between users and role

Security groups: Whitelist ports, protocols, and other security groups for network access

Network security

Route tables: Route table policy defines what VPC resources can access on the network

Network ACLs: Fence off access between specific subnets, ports, or destinations.

Separate VPCs: Full separation from other tenants.

AWS re:Invent

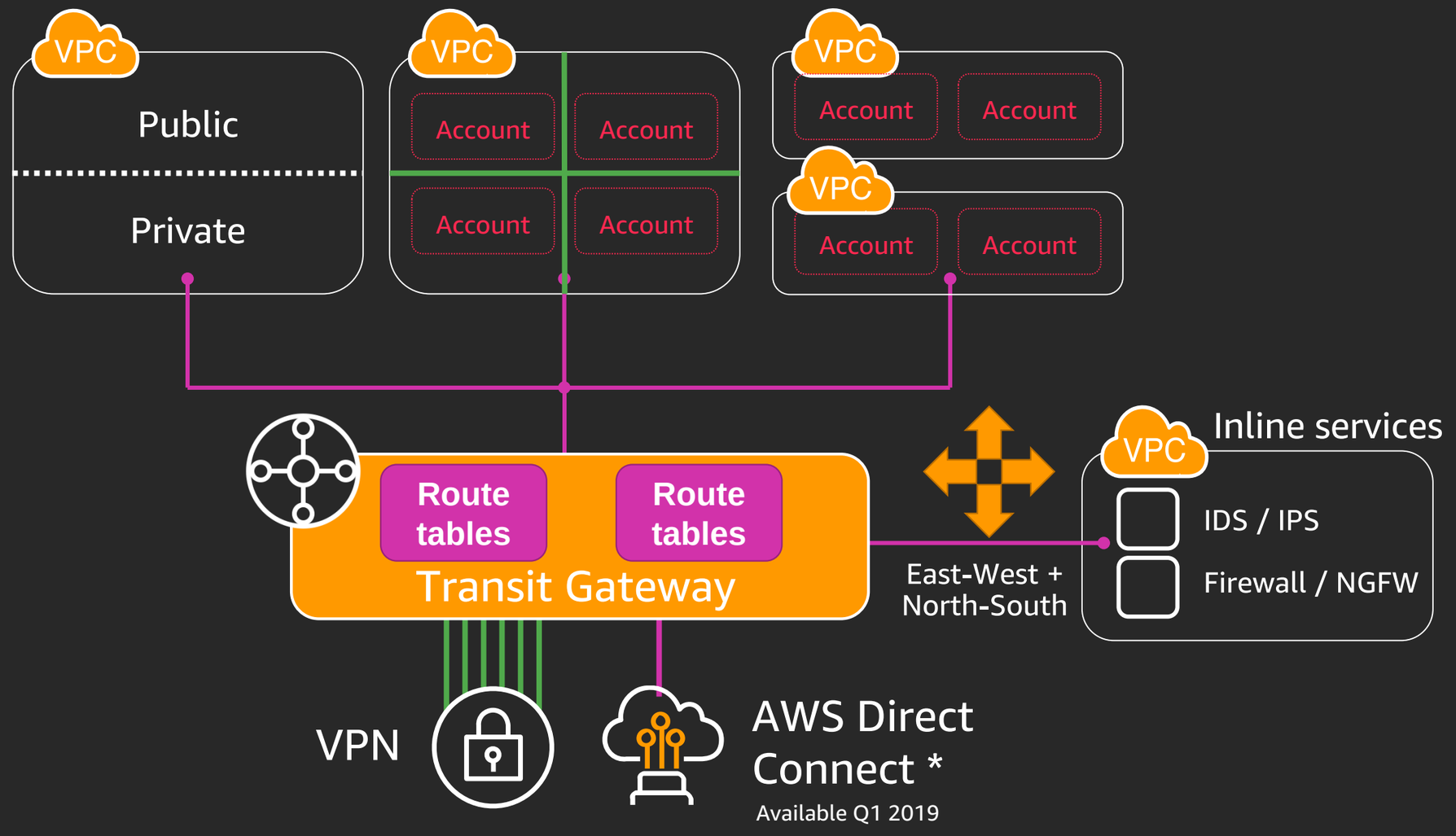
Segmentation options: Layers

Inside the account
- IAM users and roles
- Security groups

At the VPC
- Route tables
- Network ACLs
- Separate VPCs

Transit Gateway
- Route tables

Security services
- Firewalls
- Proxies
- Intrusion Detection / Prevention



AWS re:Invent

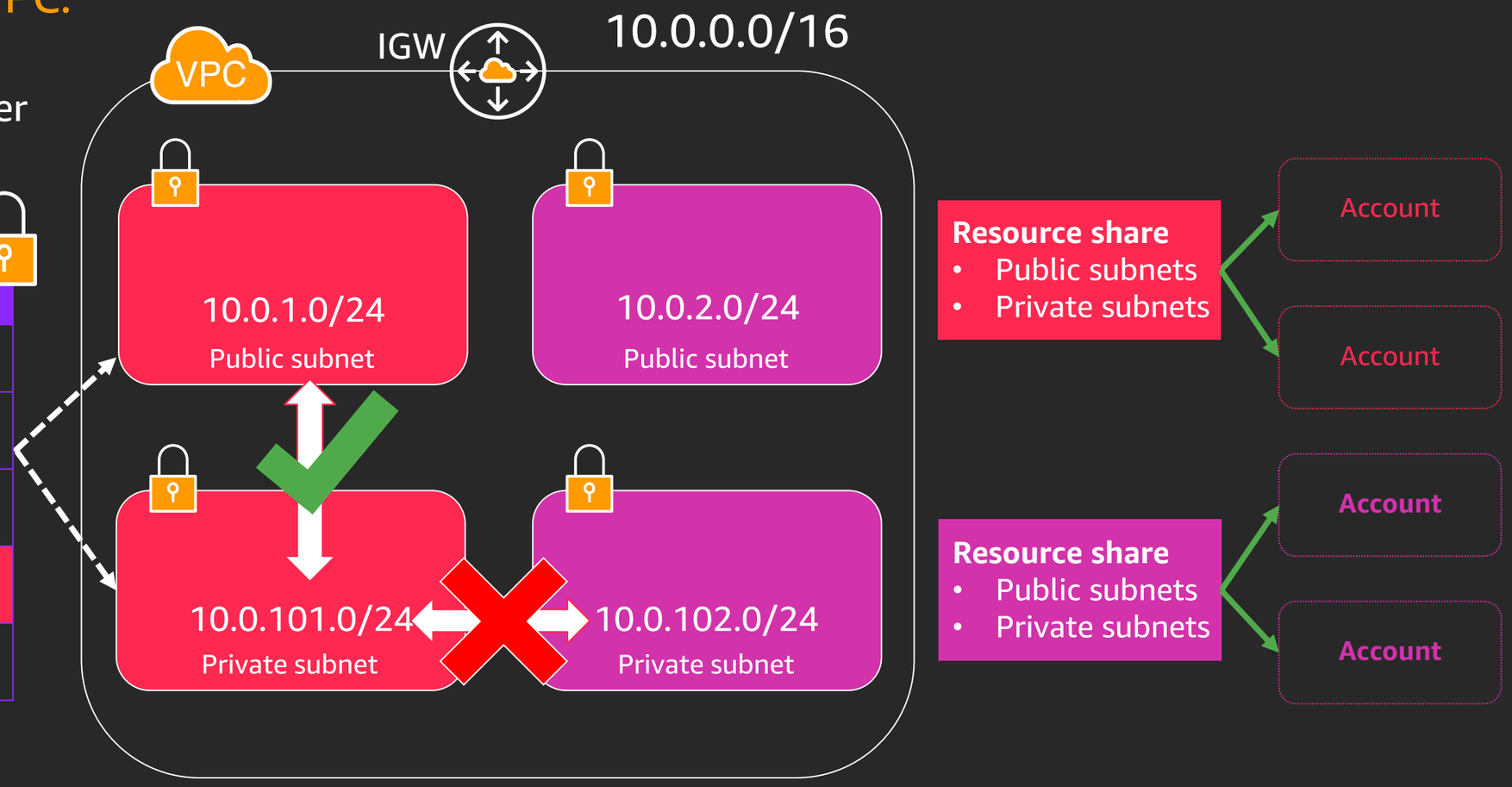
Segmentation in a Shared VPC with network ACLs

Mimic behavior of a single VPC:

- Allow traffic between subnets
- Deny inbound traffic from other tenants

Inbound network ACL

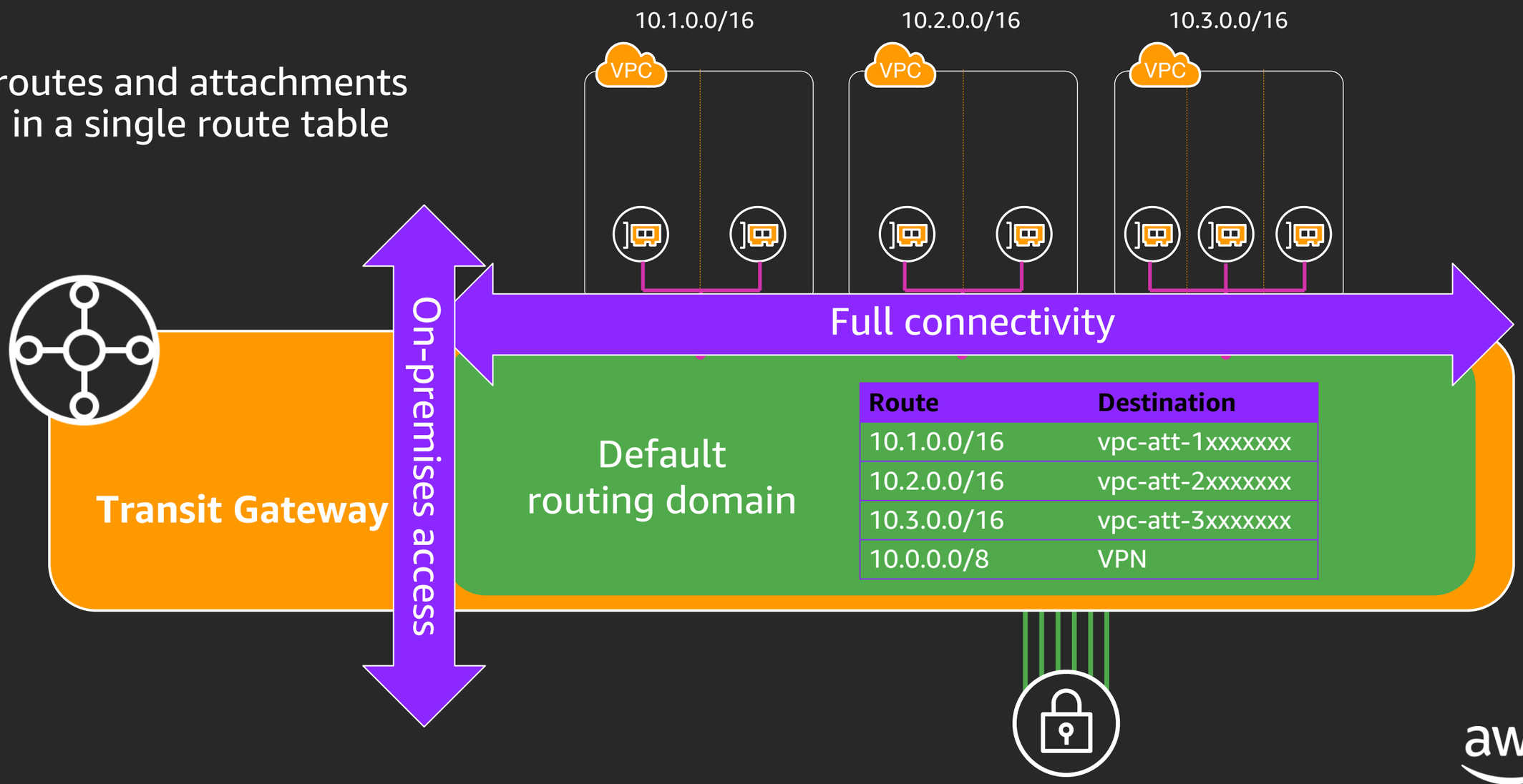
#	Source	Action
100	10.0.1.0/24	ALLOW
101	10.0.101.0/24	ALLOW
200	10.0.0.0/16	DENY
300	0.0.0.0/0	ALLOW



AWS re:Invent

Flat: Transit Gateway route domains

All routes and attachments are in a single route table



Route	Destination
10.1.0.0/16	vpc-att-1xxxxxxx
10.2.0.0/16	vpc-att-2xxxxxxx
10.3.0.0/16	vpc-att-3xxxxxxx
10.0.0.0/8	VPN

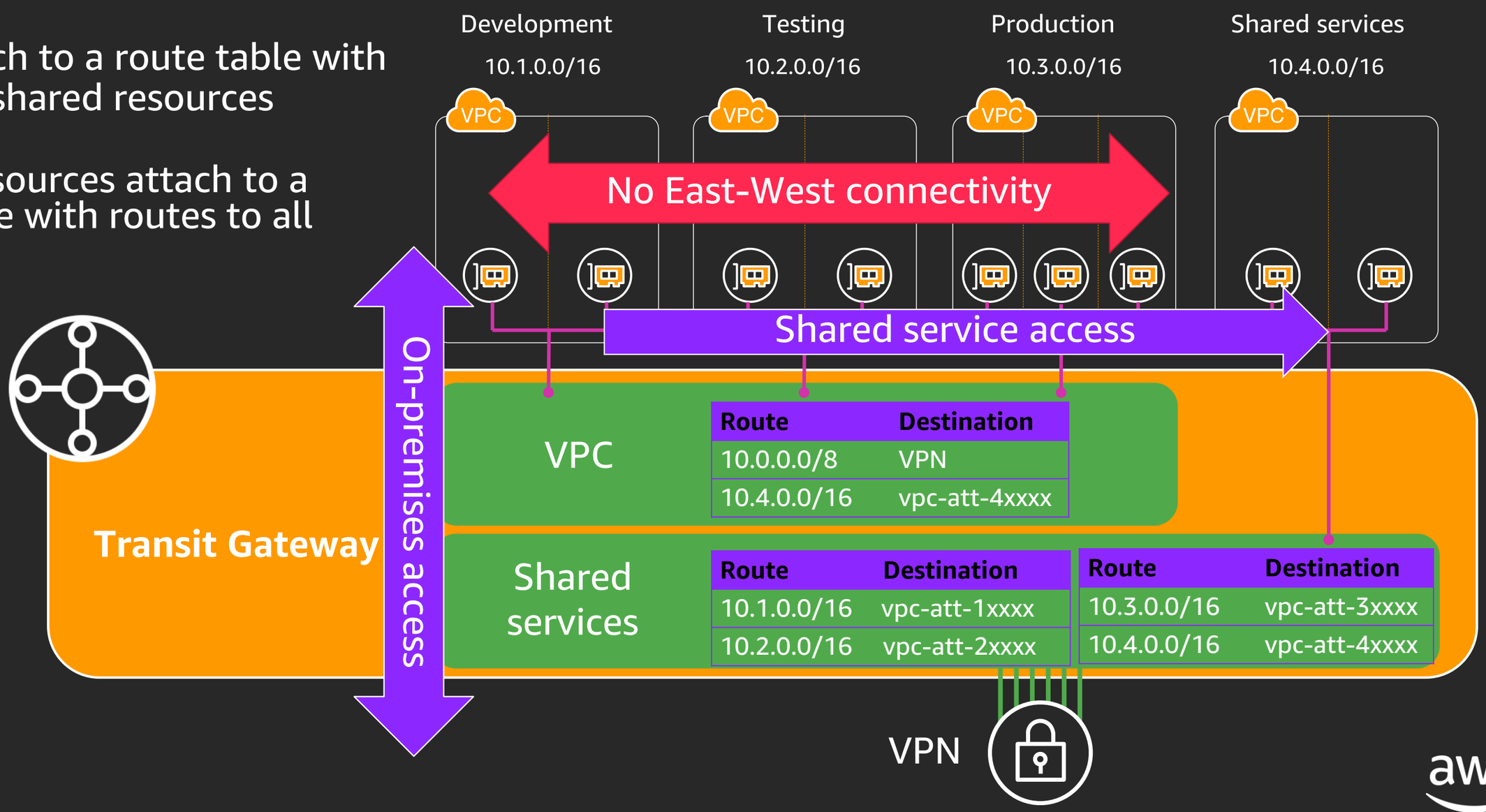
AWS re:Invent

aws

Isolated: Transit Gateway route domains

VPCs attach to a route table with routes to shared resources

Shared resources attach to a route table with routes to all resources



AWS re:Invent

aws

Segmentation considerations: Where to start

Security groups and IAM are effective and proven

- Encourage IAM and security group use and monitor security configuration

Shared VPCs

- Tenants should limit access from the internet and other tenants
- VPCs using VPC peering are likely to benefit from Shared VPCs
- Design around resource and limit contention

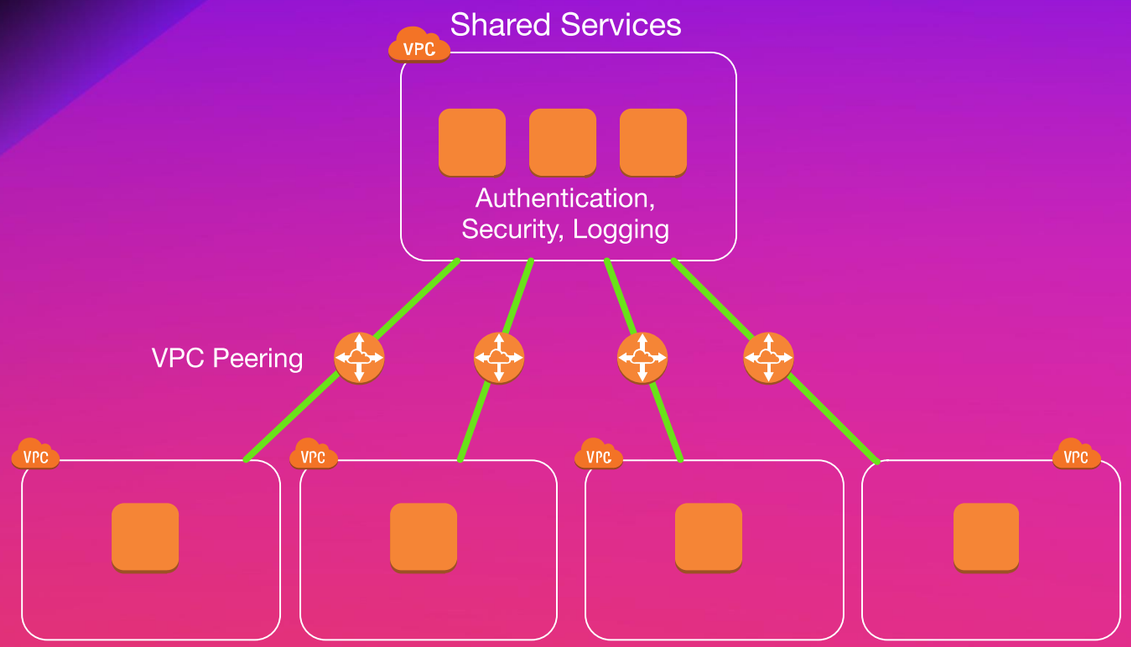
Separate VPCs

- Often the best security decision is the simplest. Separate VPCs are simple.
- Use separate VPCs for strong network segmentation and resource isolation
- Transit Gateway removes the scaling issues with many VPCs (peering, VPN, routes)

Transit Gateway route tables define multi-VPC policy

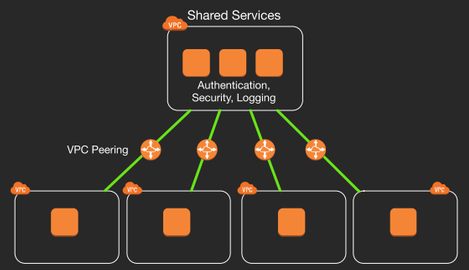
- Consider isolating environments (dev and prod) and allow access to shared resources

Shared services



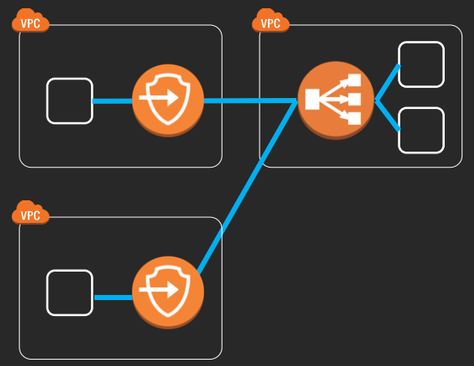
Shared services connectivity options

VPC peering



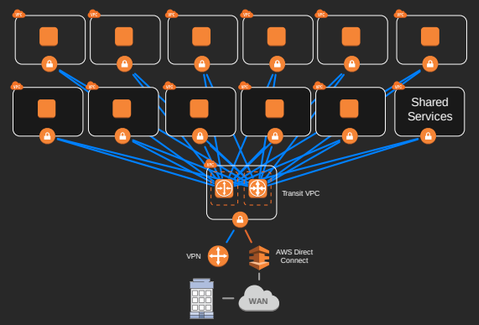
- One-to-one connectivity
- Scales to 100 VPCs
- Security groups across VPCs
- Inter-region peering

AWS PrivateLink



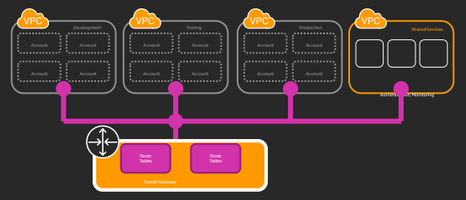
- One-to-many connectivity
- Highly scalable
- Supports overlapping CIDRs
- Uses Elastic Load Balancing
- Load balancing and hourly endpoint costs

Transit VPC



- Shared services as a spoke
- Bandwidth constrained
- Complex management
- Instance and licensing costs

AWS Transit Gateway

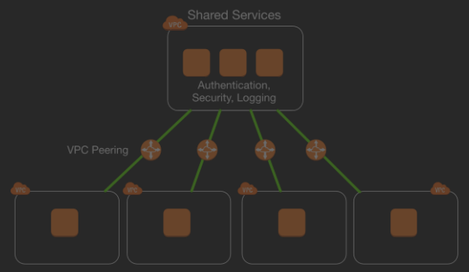


- Many-to-many or one-to-many with route tables
- Highly scalable
- Hourly per AZ endpoint costs

AWS re:Invent

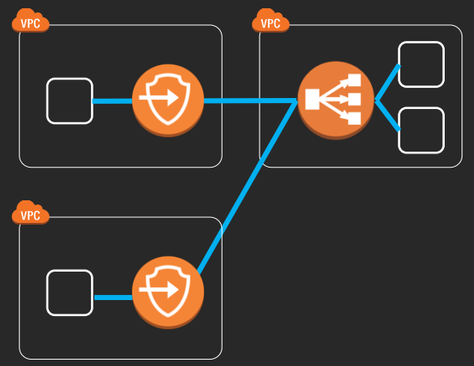
Shared services connectivity options **at scale**

VPC Peering



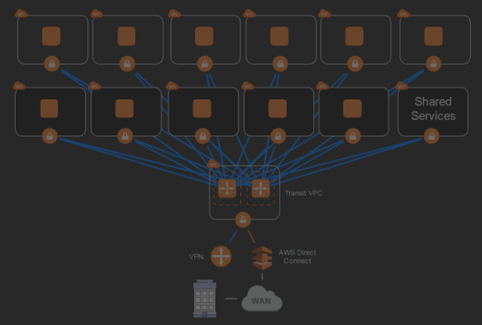
- 1-to-1 connectivity
- Scales to 100 VPCs
- Security groups across VPCs
- Inter-region peering

AWS PrivateLink



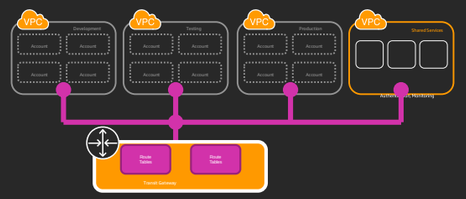
- One-to-many connectivity
- Highly scalable
- Supports overlapping CIDRs
- Uses Elastic Load Balancing
- Load balancing and hourly endpoint costs

Transit VPC



- Shared services as a spoke
- Bandwidth restricted
- Complex management
- Instance and licensing costs

AWS Transit Gateway

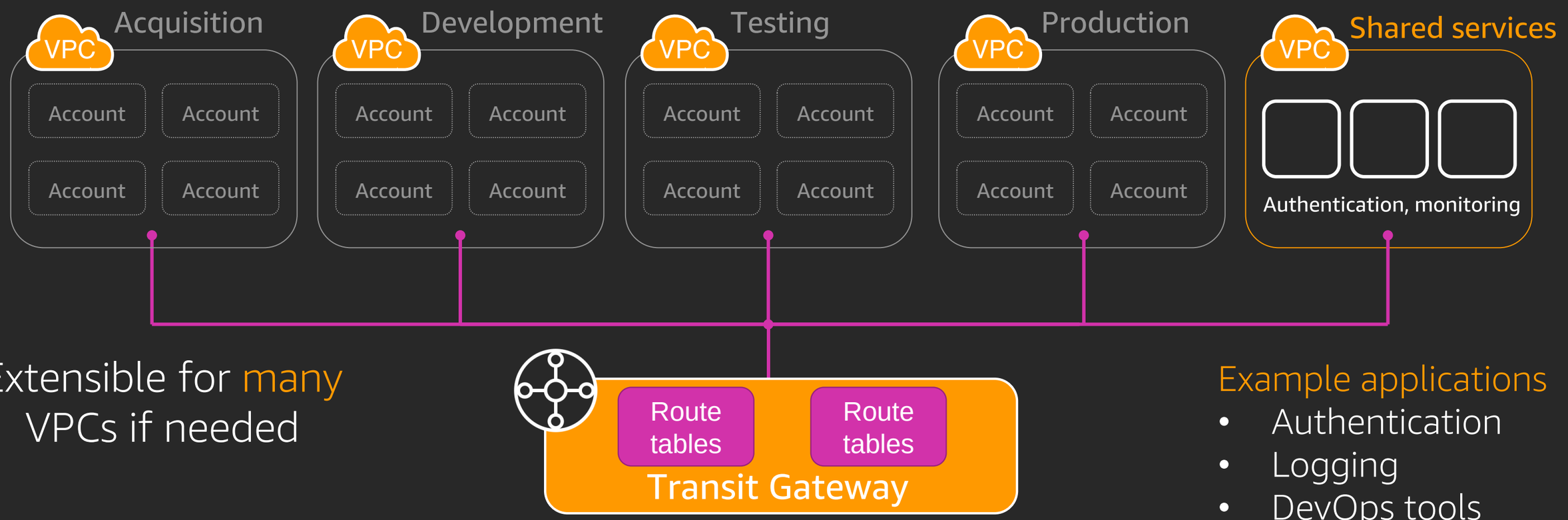


- Many-to-many or one-to-many with route tables
- Highly scalable
- Hourly per AZ endpoint costs

AWS re:Invent

aws

Shared services with Transit Gateway



Extensible for many VPCs if needed

Example applications

- Authentication
- Logging
- DevOps tools
- Security resources

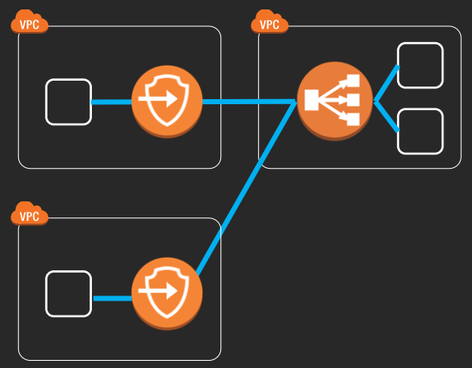
Works with flat or isolated segmentation

AWS re:Invent

aws

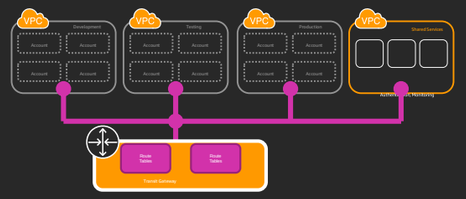
Using Transit Gateway and PrivateLink

AWS PrivateLink



- One-to-many connectivity
- Highly scalable
- Supports overlapping CIDRs
- Uses Elastic Load Balancing
- Load balancing and hourly endpoint costs

AWS Transit Gateway



- Many-to-Many or one-to-many with route tables
- Highly scalable
- Hourly per AZ endpoint costs

Scope: Application shared services

Trust model: No mutual trust

Dependencies: Load balancing and application architecture

Scale: Thousands of spoke VPCs

Scope: Network shared services to many VPCs

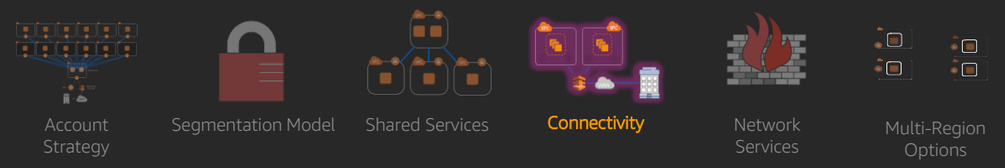
Trust model: Per VPC trust, centralized control

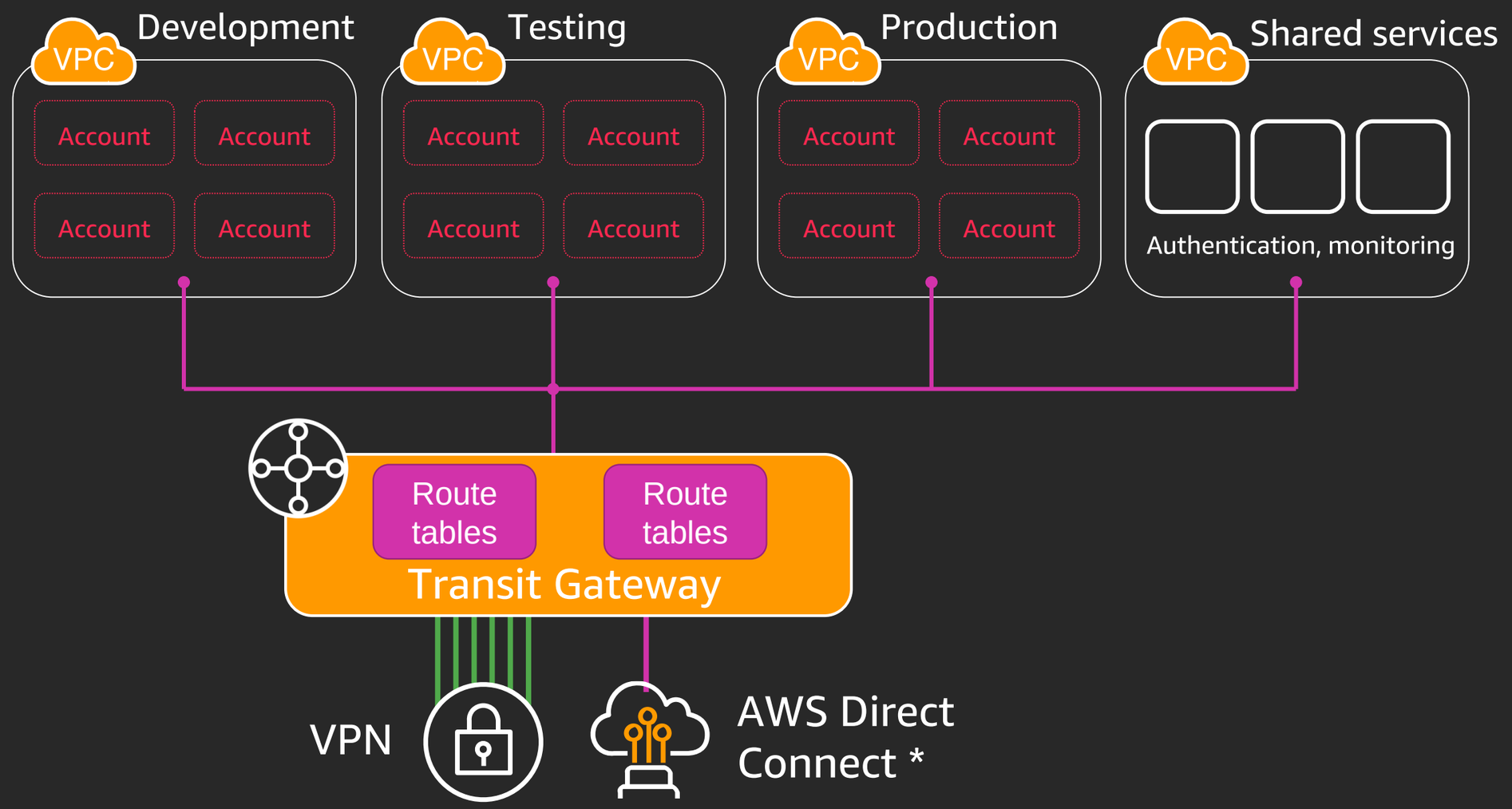
Dependencies: Centralized control of the Transit Gateway

Scale: Thousands of spoke VPCs

AWS re:Invent

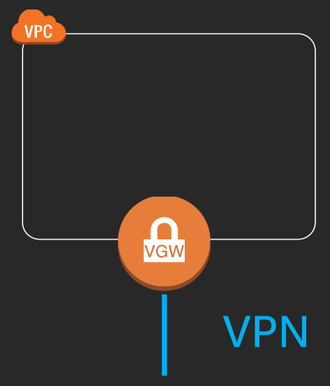
Connecting on-premises





Connecting to on-premises

Virtual Private Gateway VPN

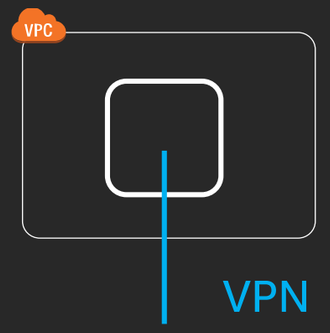


- Per VPC
- 1.25 gbps per tunnel
- Encrypted in transit

AWS Direct Connect

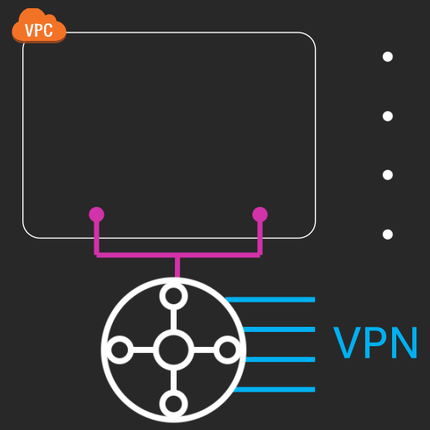
- Per VPC (50 per port)
- Multiple VPCs with Direct Connect gateway
- No bandwidth restraint

Amazon EC2 customer VPN



- Per VPC or multiple (Transit VPC)
- Bandwidths vary by instance type
- AWS Marketplace options
- Scalability is generally limited by management complexity

AWS Transit Gateway VPN

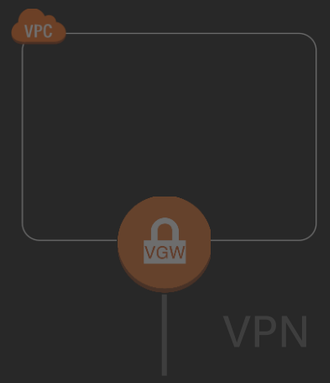


- Multiple VPCs
- Add VPN connection as needed
- 1.25 gbps per tunnel
- Roadmap: AWS Direct Connect

AWS re:Invent

Connecting to On-premises **at Scale**

Virtual Private Gateway VPN

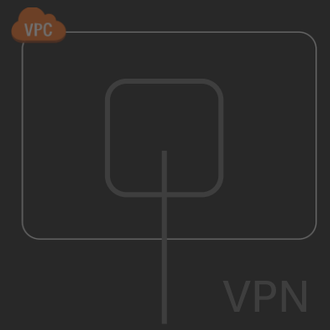


- Per VPC
- 1.25 gbps per tunnel
- Encrypted in transit

AWS Direct Connect

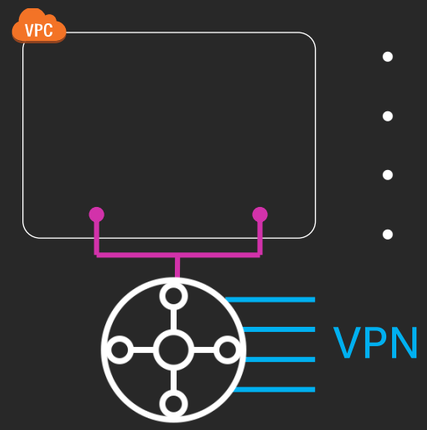
- Per VPC (50 per port)
- Multiple VPCs with Direct Connect gateway
- No bandwidth restraint

Amazon EC2 Customer VPN



- Per VPC or multiple (Transit VPC)
- Bandwidths vary by instance type
- AWS Marketplace options
- Scalability is generally limited by management complexity

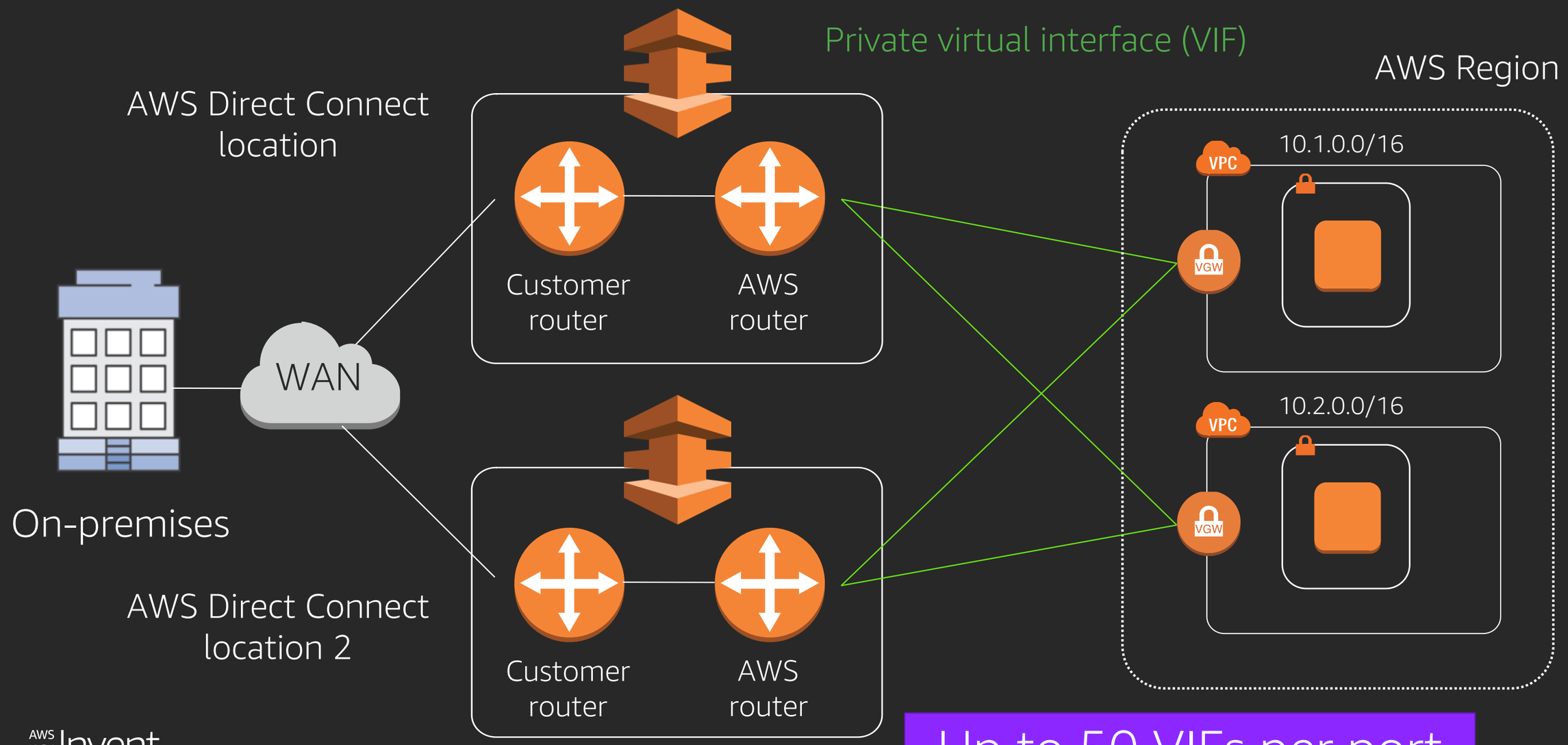
AWS Transit Gateway VPN



- Multiple VPCs
- Add VPN connection as needed
- 1.25 gbps per tunnel
- Roadmap: AWS Direct Connect

AWS re:Invent

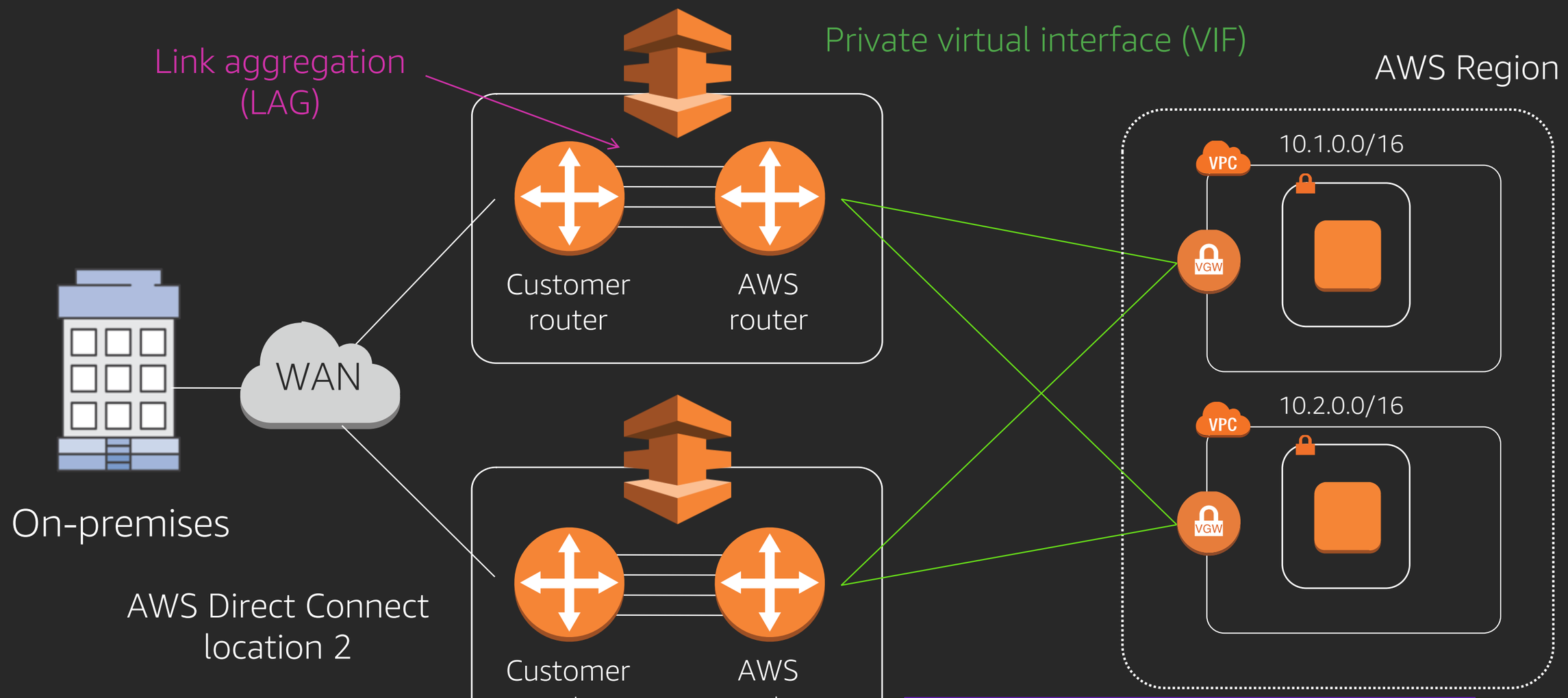
AWS Direct Connect to Many VPCs



Up to 50 VIFs per port

AWS re:Invent

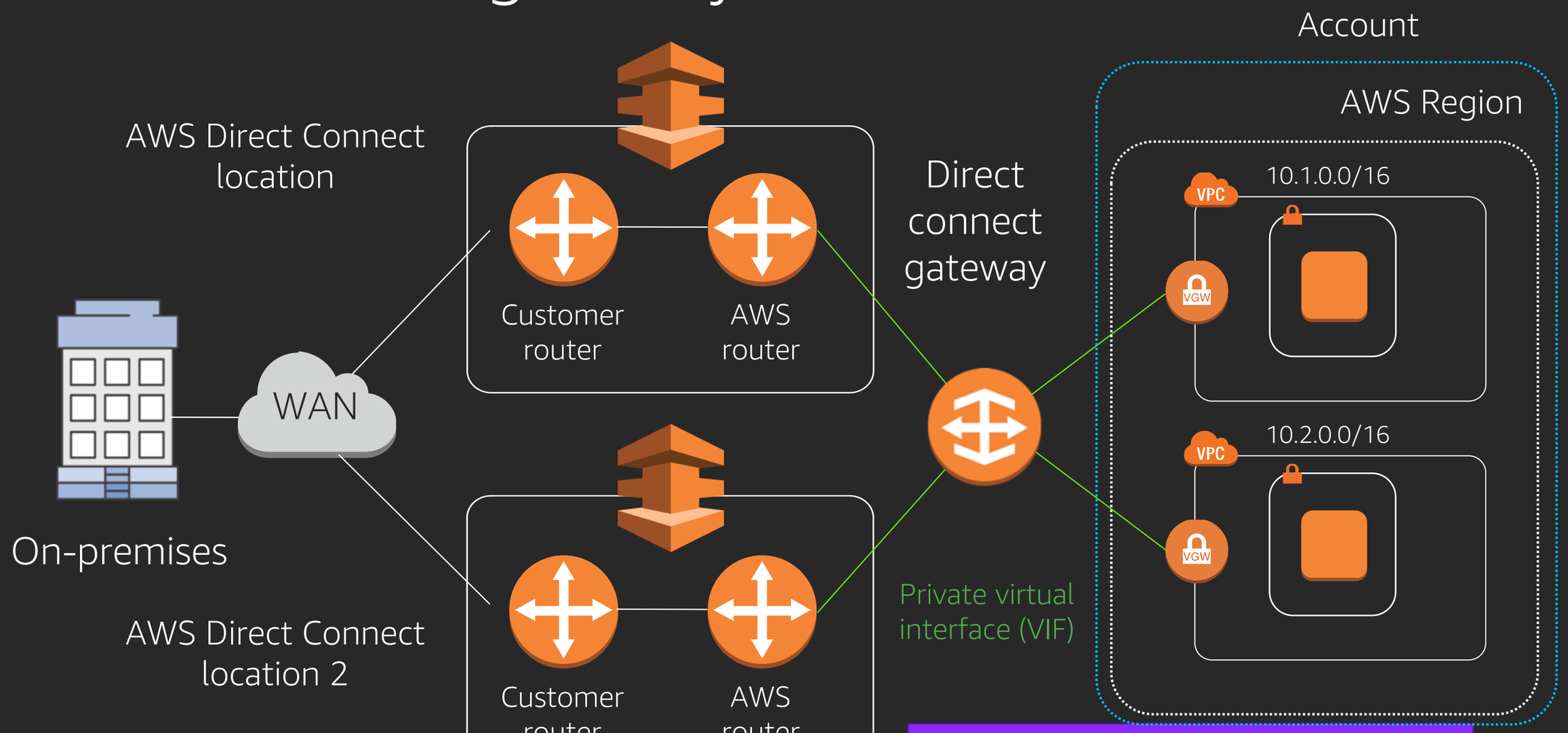
AWS Direct Connect: Link Aggregation



Up to 4 ports in a LAG, each with 50 VIFs

AWS re:Invent

Direct Connect gateway

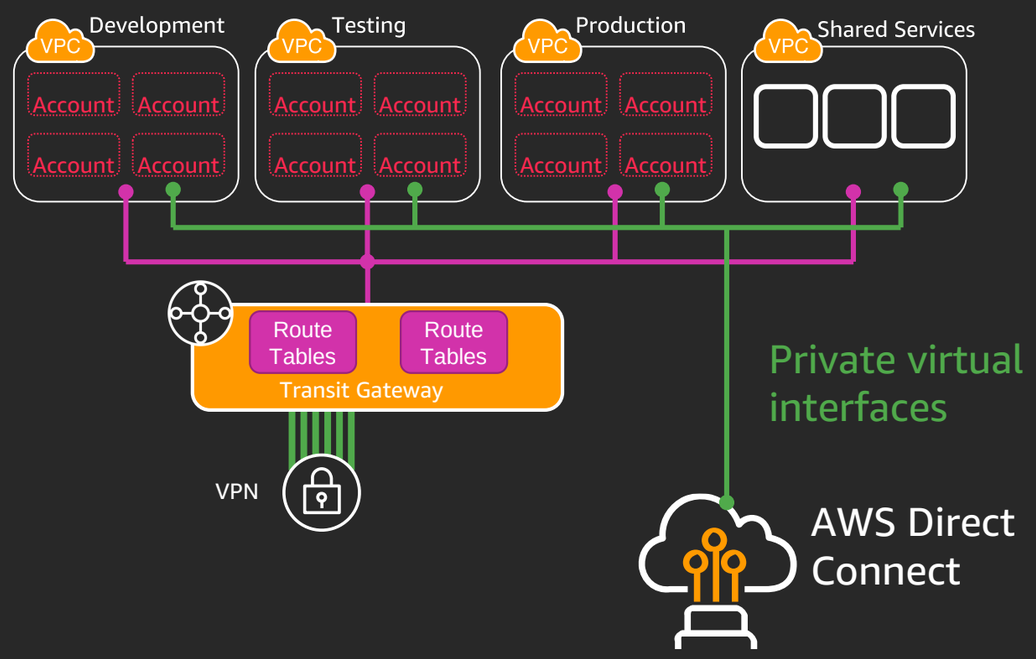


Up to 10 VGWs per direct connect gateway

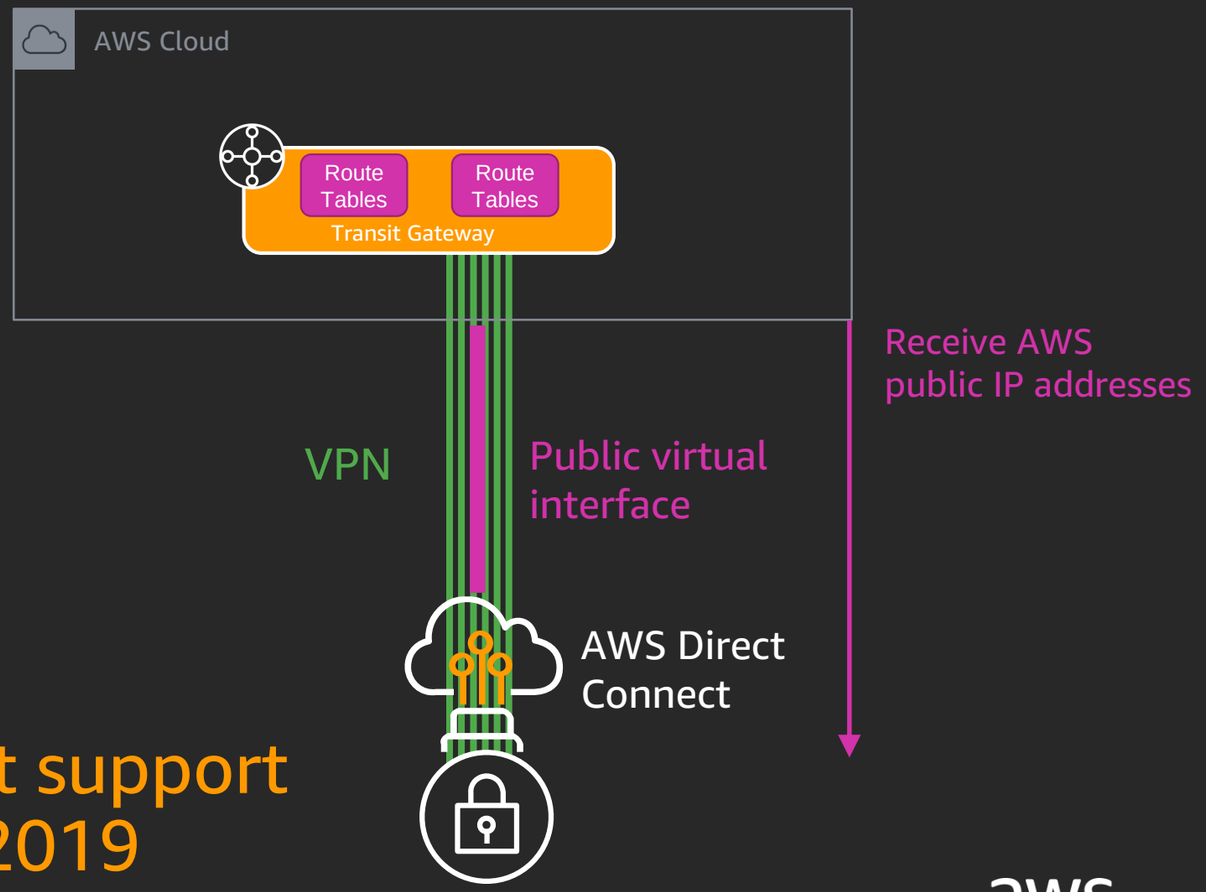
AWS re:Invent

AWS Direct Connect and Transit Gateway

Use Direct Connect in parallel



Use VPN over a Direct Connect public virtual interface (VIF)



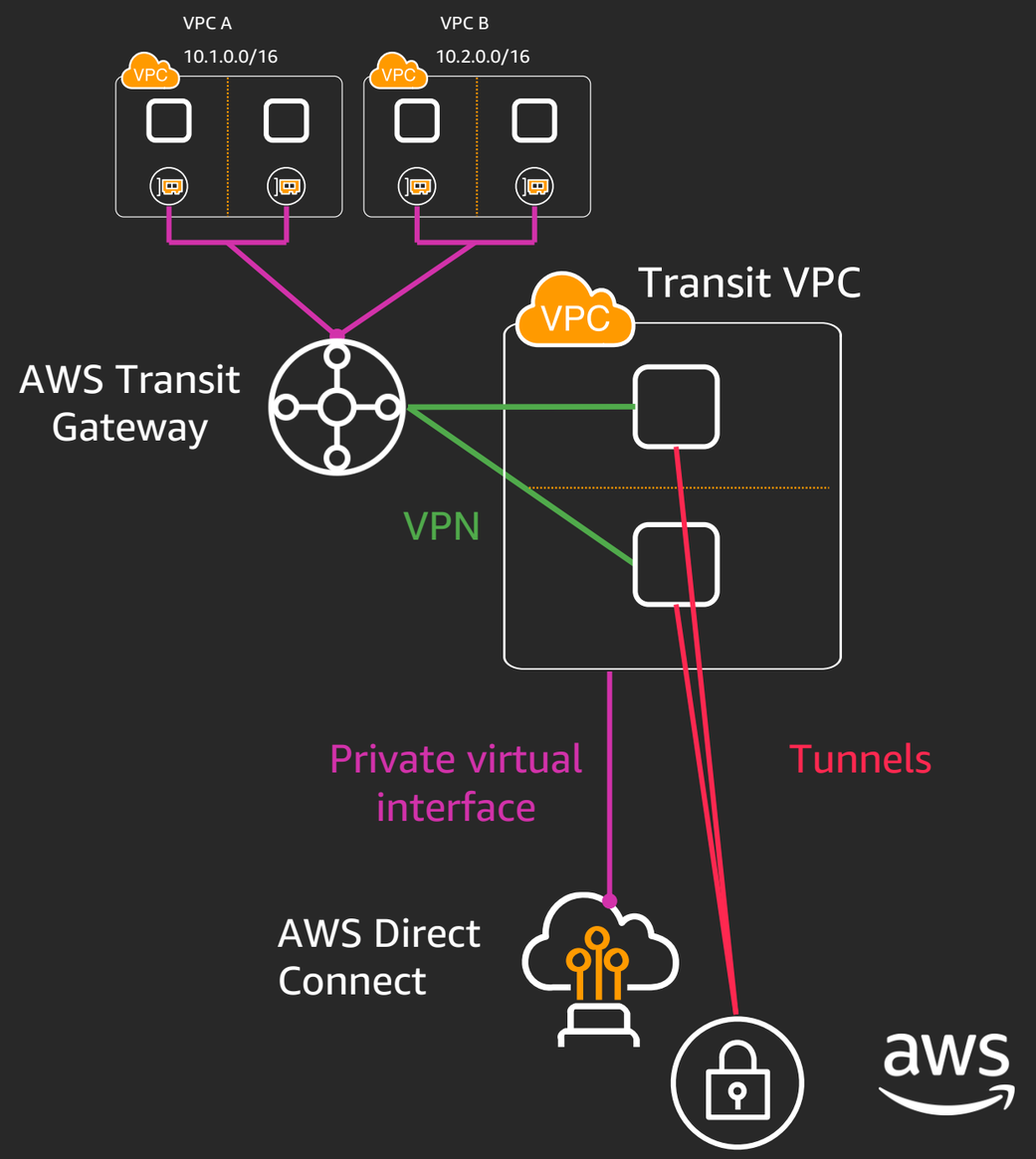
Native Direct Connect support planned for Q1 2019

AWS re:Invent

AWS Direct Connect and Transit Gateway

Use an edge services VPC in front of a private virtual interface

- More detail in the network services section
- Also how used to migrate or extend existing Transit VPCs
- Helpful for single-VIF (<1 Gbps) Direct Connect
- Can be used for North-South inspection use-cases



AWS re:Invent

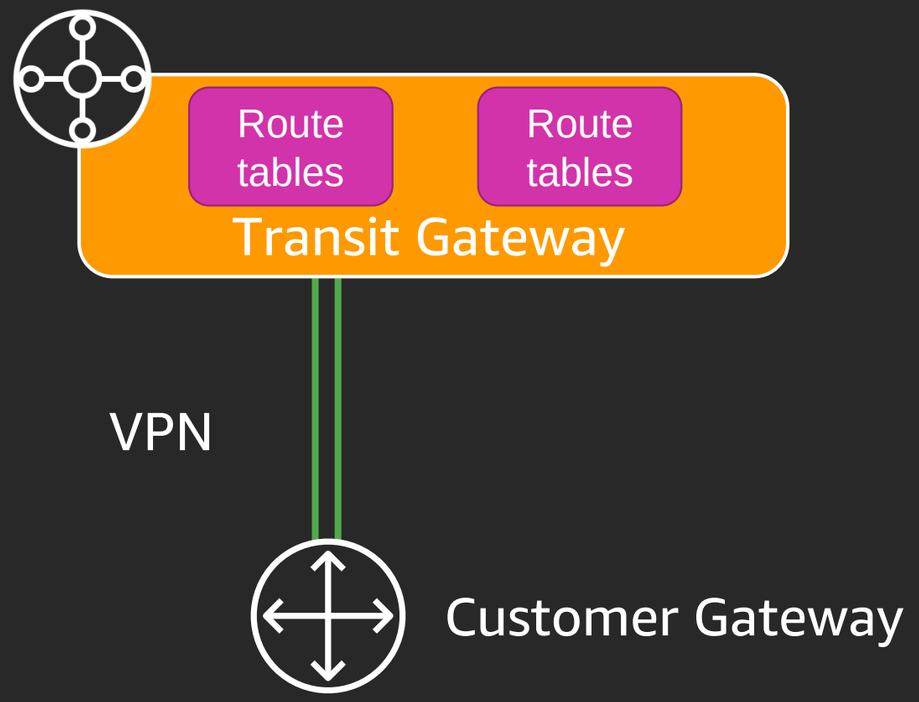
VPN With Transit Gateway

Consolidate VPN at the Transit Gateway (TGW)

- VPN acts similar to the Virtual Private Gateway (VGW)
  - Bandwidth, configuration, APIs, cost, and experience
  - VPN is attached to a TGW instead of a VGW
  - Same 1.25 gbps bandwidth per tunnel applies

Encryption to the edge of many VPCs

  - Traffic is encrypted until it's inside the VPC
  - Does not natively encrypt traffic between VPCs
    - Inter-region VPC peering does



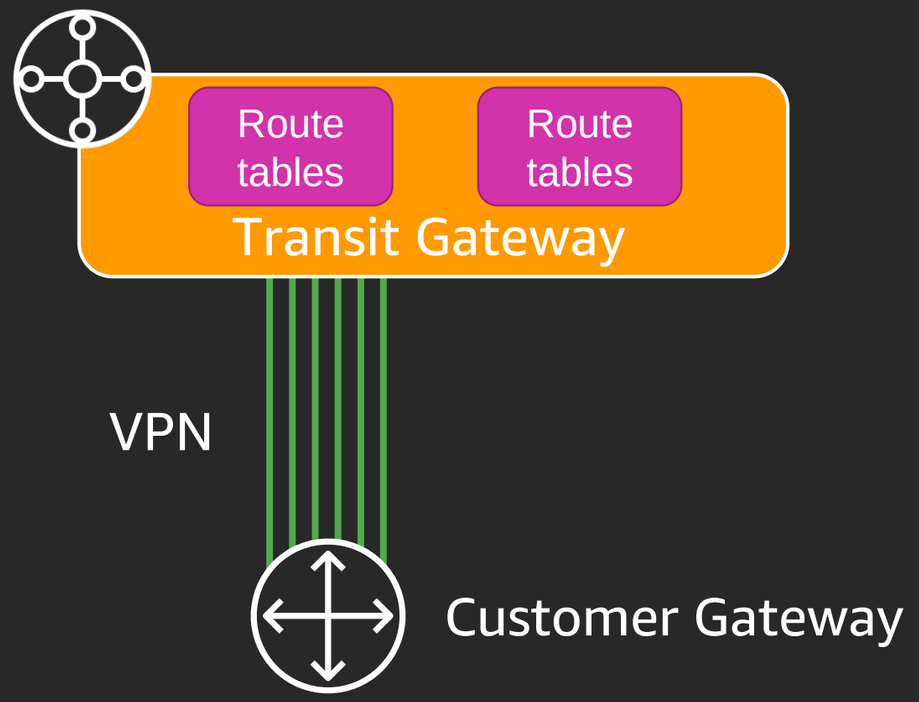
VPN with Transit Gateway: Add more bandwidth

Support for spreading traffic across many tunnels

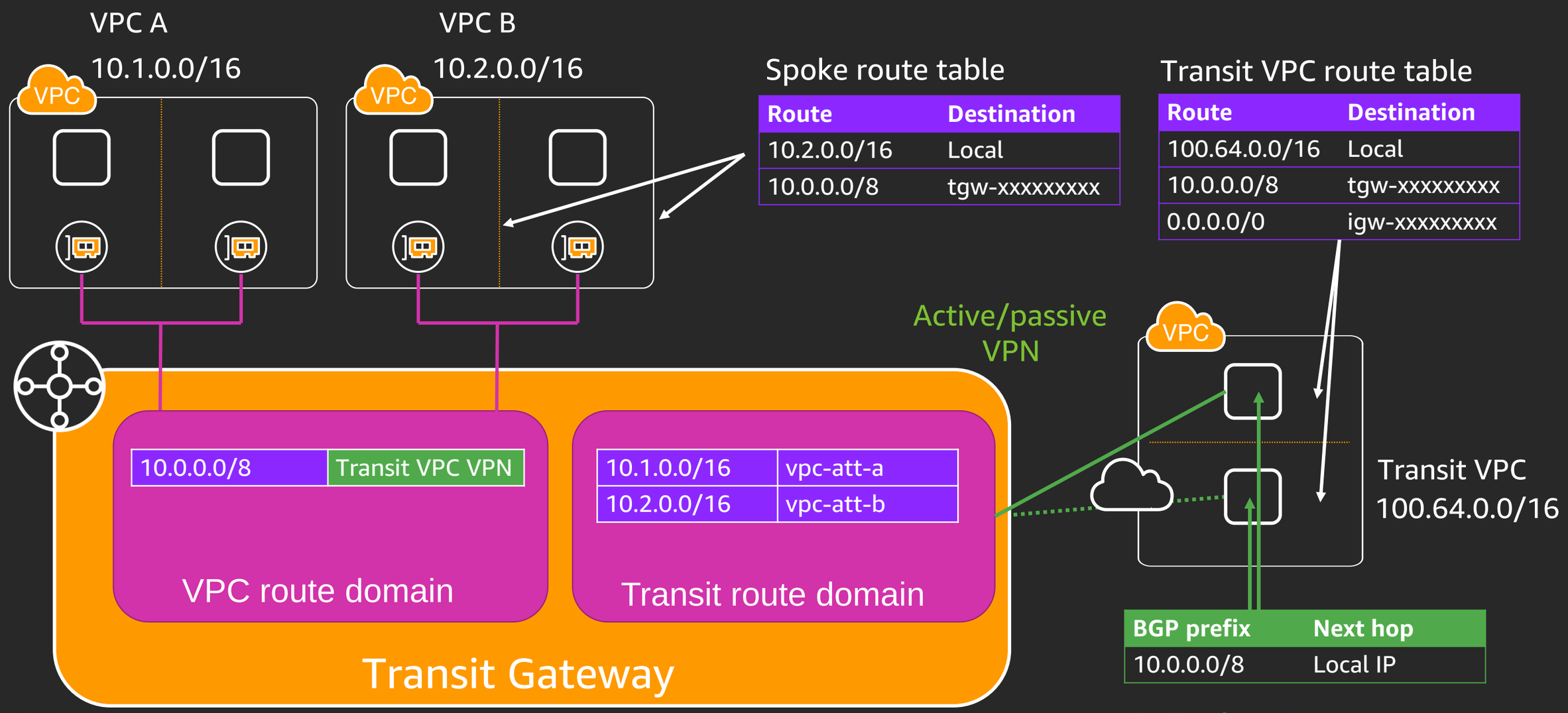
- Equal Cost Multi-Path (ECMP) support with BGP multi-path
- Tested up to 50 Gbps of traffic
- Split traffic into smaller flows, multi-part uploads, etc.

Check your on-premises configuration

- Multi-path BGP
- ECMP support, amount of equal paths, reverse-path forwarding/spoofing checks
- Only supported with BGP, not static routing



Transit VPC 1.1



AWS re:Invent

¿ Neat. But, why? ?

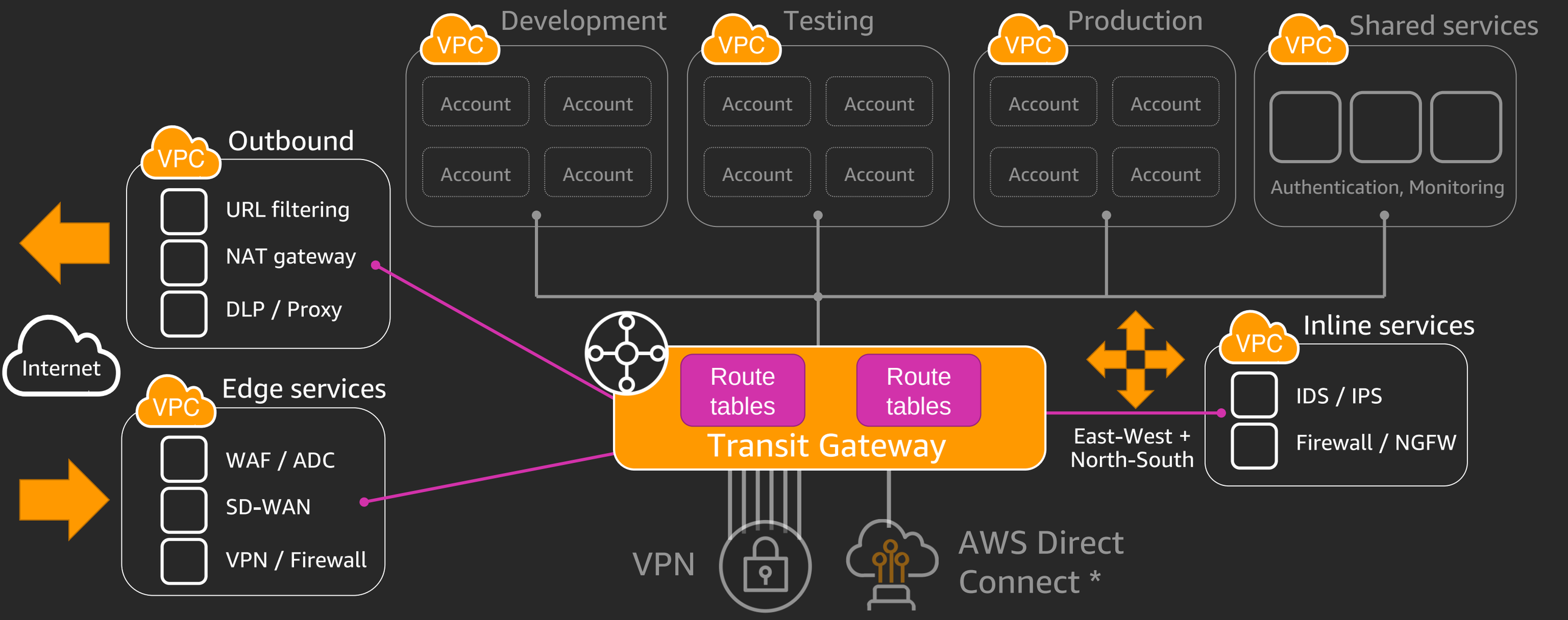
AWS re:Invent

aws

Network services

Reference Network Architecture

Optional network services



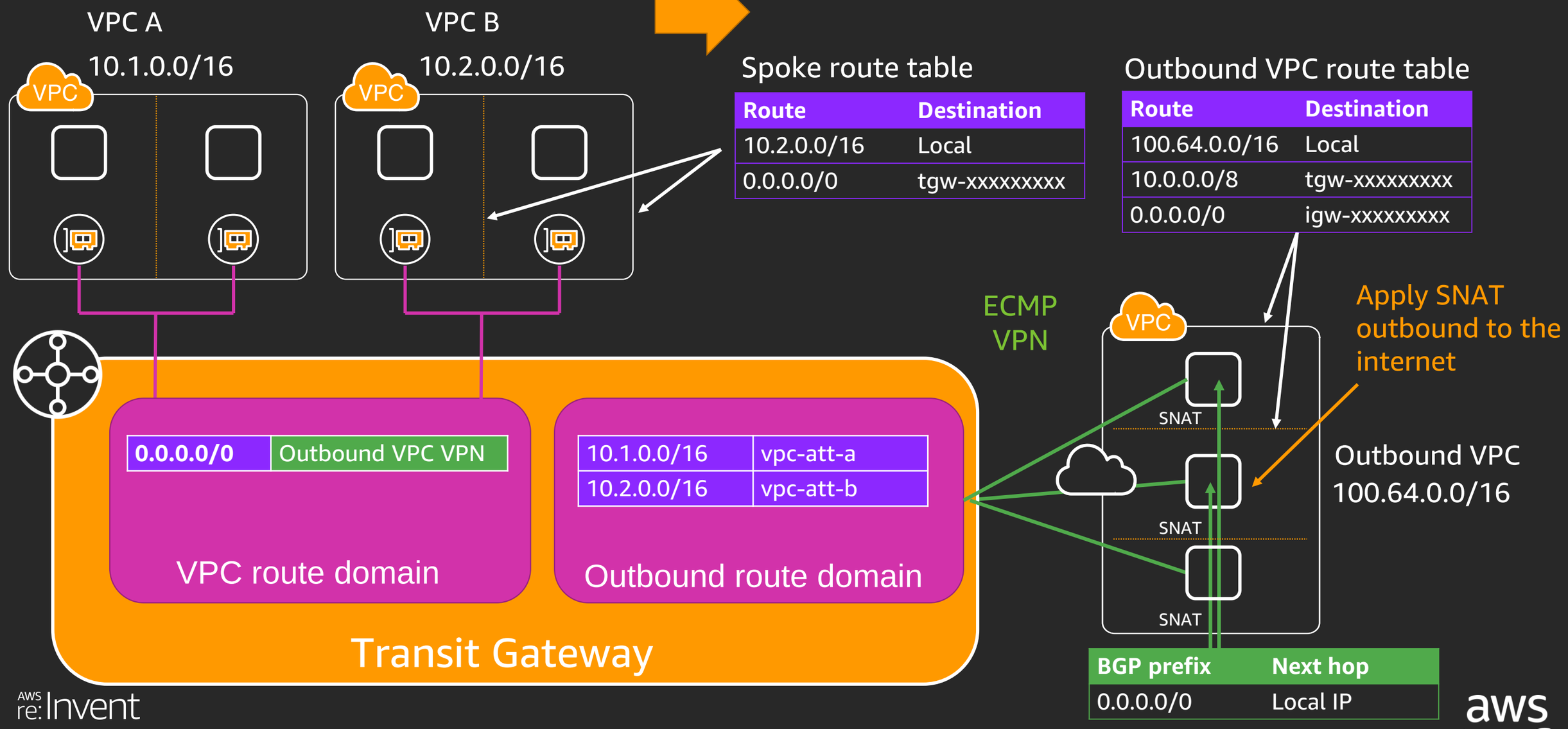
Do I need to put service each into their own VPC?

No, but let’s understand the routing separately first.

Outbound services VPC

Use cases:

URL filtering, NAT gateway, data-loss prevention (DLP), web proxy services



AWS re:Invent

aws

VPN service insertion design notes

Instance must be able to support:

- VPN to the Transit Gateway
- BGP to the Transit Gateway (ECMP requirement)
- Source NAT to the internet

Performance

- IPsec overhead
- Compatible with auto-scaling architectures
- No cumulative bandwidth limit

High availability

- BGP and VPN Dead Peer Detection handle failover
- No API calls required for fault tolerance
- Optionally place instances in Amazon EC2 automatic recovery

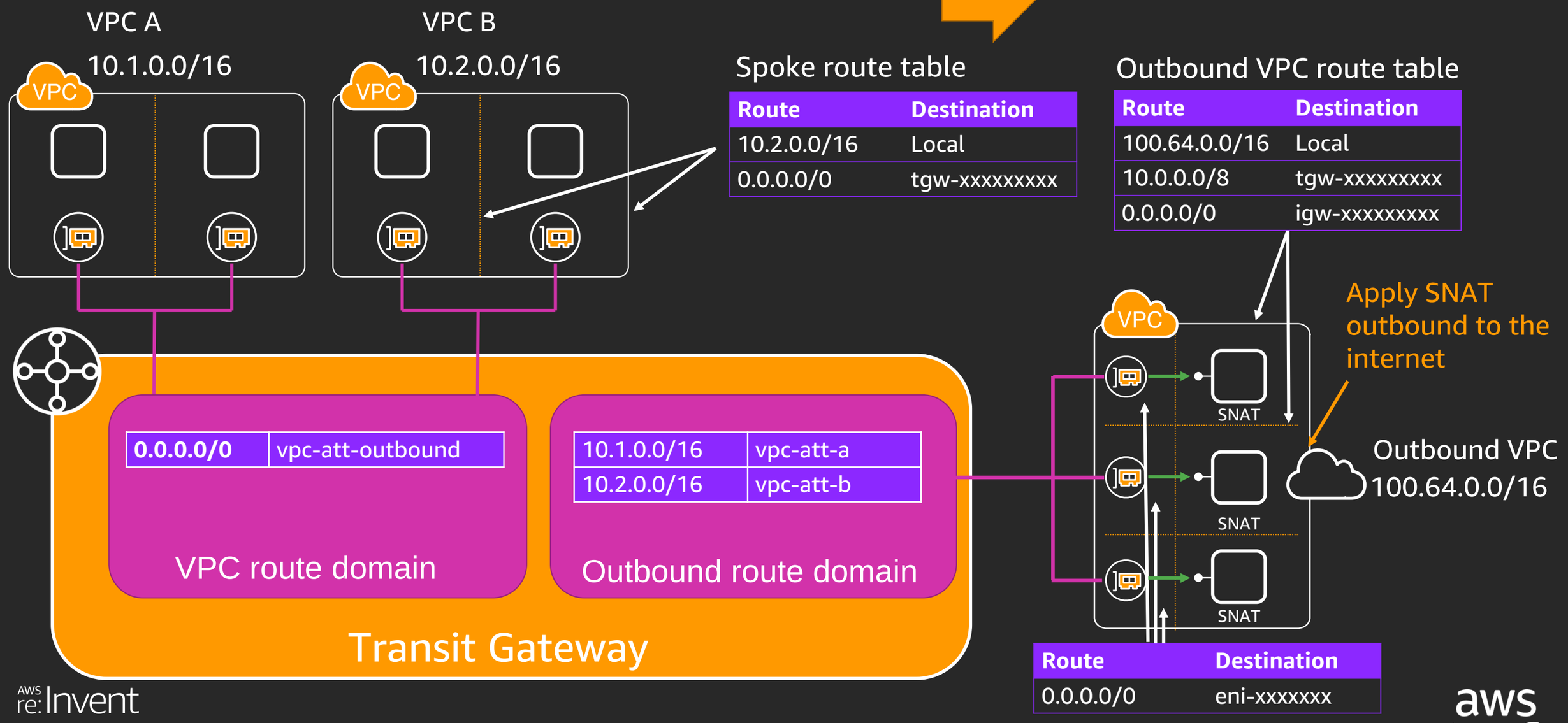
Stateful services

- Use Source NAT to guarantee the return flow to the same instance

Horizontally scalable service pattern

Preferred method if the service supports BGP, VPN and NAT.

Outbound services VPC: Interface



Interface service insertion design notes

Instance must be able to support:

- Source NAT to the internet

Performance

- No overhead (8500 MTU)
- Limited to one Transit Gateway attachment per Availability Zone, so one route table
- Traffic is forwarded within the same Availability Zone if possible
  - Likely that traffic isn't evenly distributed across instances

High availability

- There are no built-in health checks for the VPC routes, requires monitoring and management
- Optionally place instances in Amazon EC2 automatic recovery

Stateful services

- Use Source NAT to guarantee the return flow to the same instance

Simpler performance pattern

Stay within the performance of a single service instance (worst-case scenario) and configure your own high availability checks.

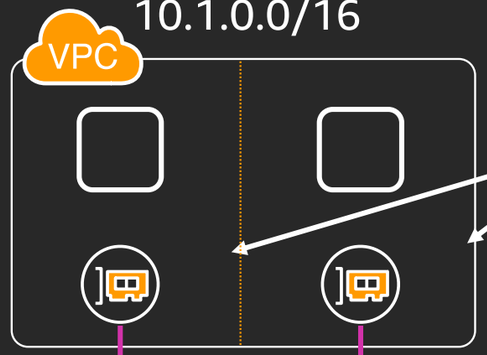
AWS re:Invent

aws

Edge services VPC: Ingress

BGP prefix	Next hop
100.64.0.0/16	Local IP

VPC A
10.1.0.0/16

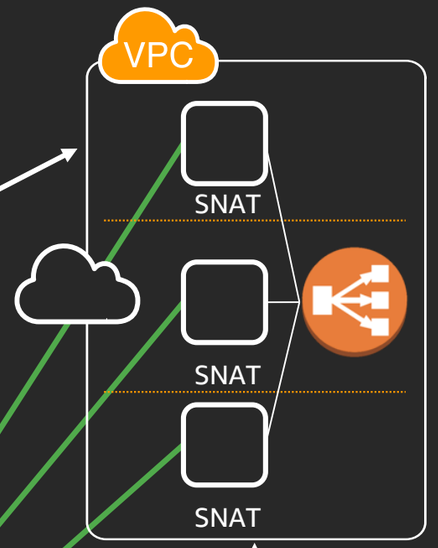


Spoke route table

Route	Destination
10.1.0.0/16	Local
100.64.0.0/16	tgw-xxxxxxxxx

Edge VPC route table

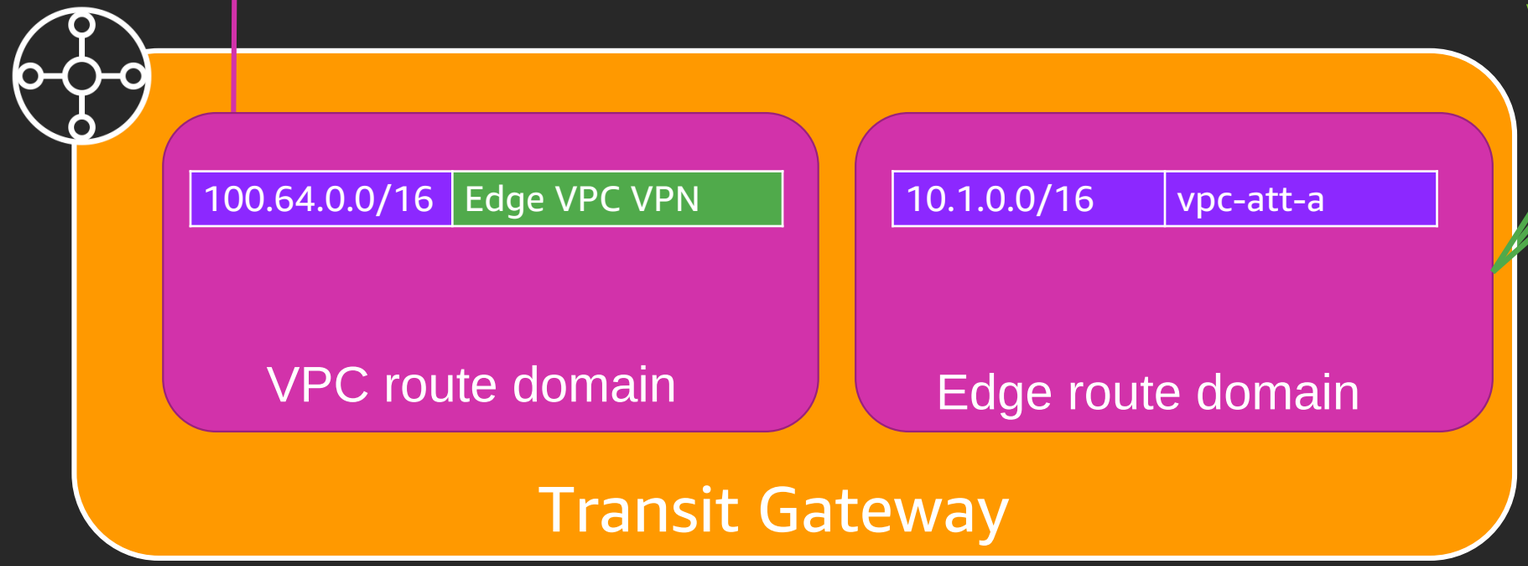
Route	Destination
100.64.0.0/16	Local
10.0.0.0/8	tgw-xxxxxxxxx
0.0.0.0/0	igw-xxxxxxxxx



Edge VPC
100.64.0.0/16

Optional ELB

ECMP
VPN



Use cases:

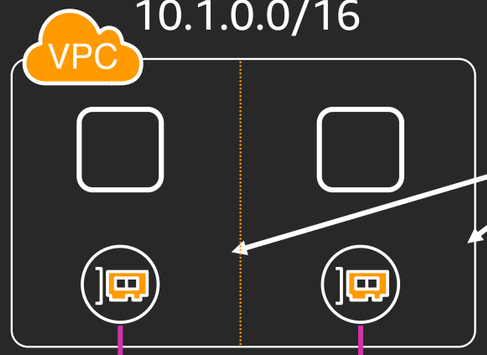
WAF, inspection, Load balancing

Edge services VPC: SD-WAN

BGP prefix	Next hop
Many prefixes	Local IP

VPC A

10.1.0.0/16



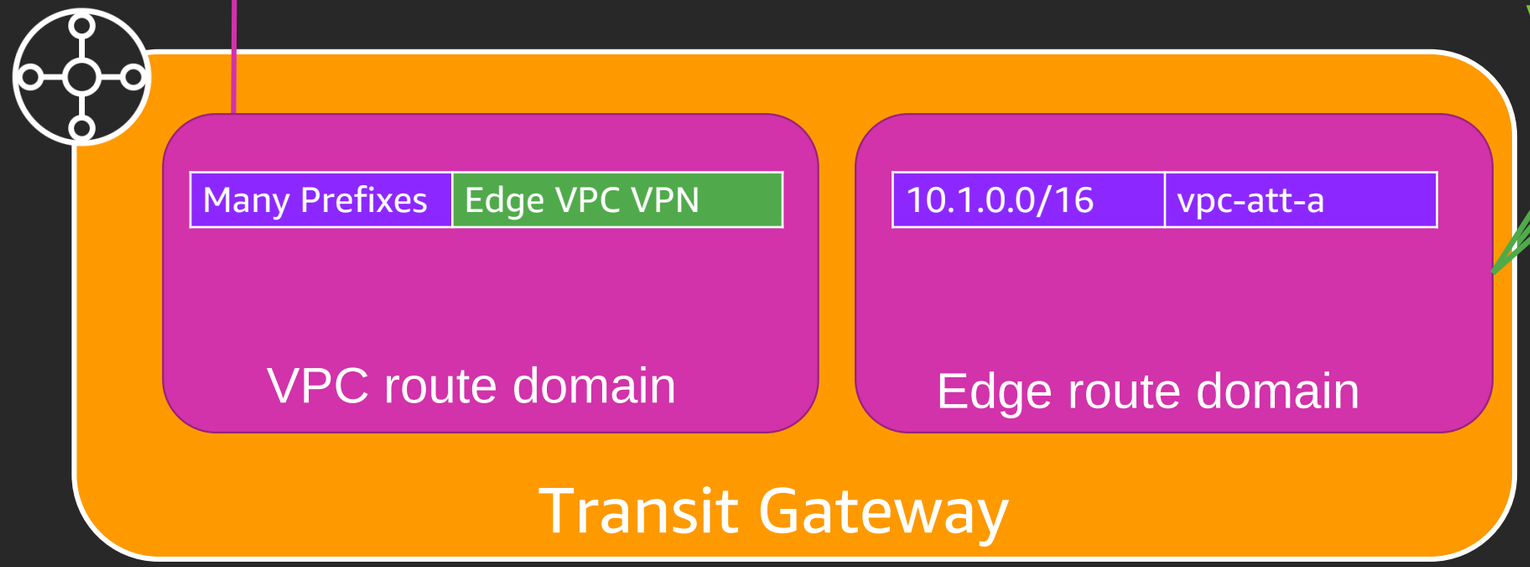
Spoke route table

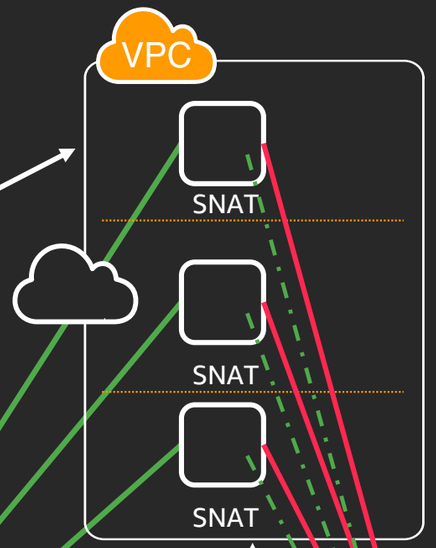
Route	Destination
10.1.0.0/16	Local
0.0.0.0/0	tgw-xxxxxxxxx

Can be a summary or default route in each VPC

Edge VPC route table

Route	Destination
100.64.0.0/16	Local
10.0.0.0/8	tgw-xxxxxxxxx
0.0.0.0/0	igw-xxxxxxxxx





Edge VPC

100.64.0.0/16

Only stateful services require NAT

ECMP

VPN

Tunnels and BGP

Internet

Data Center, Branches, Clients, etc.

Use cases:

SD-WAN, Routing, Third-party client VPN, AWS Direct Connect over a Private VIF

AWS re:Invent

aws

Reminder:

Existing network services or DMZs may be convenient, but they may also be the problem.

Remember to evaluate operational processes, alternatives, and automation

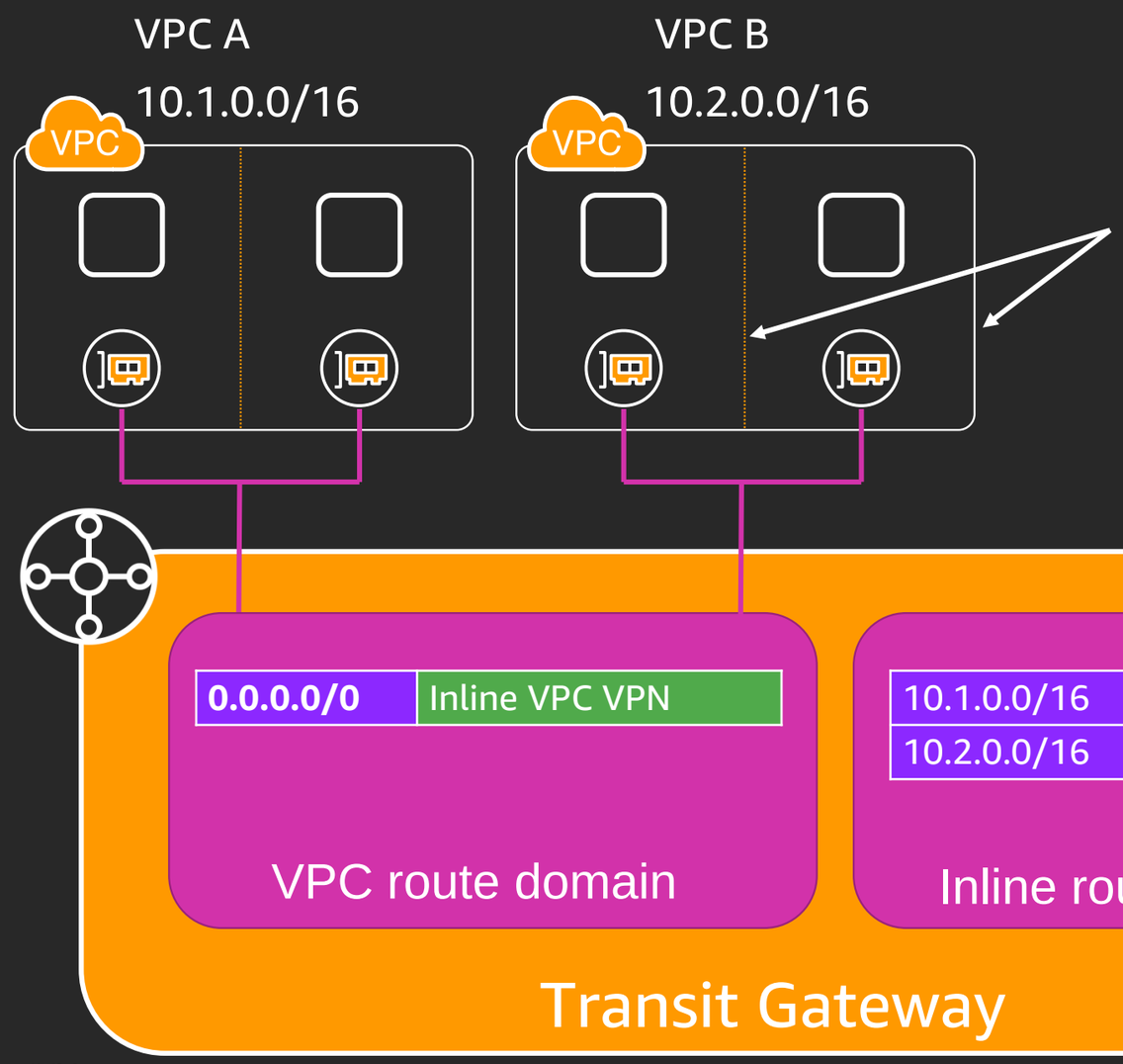
AWS re:Invent

aws

VPC to VPC service insertion

Use cases:

Intrusion detection/prevention (IDS/IPS), firewalls, NextGen Firewalls (NGFW), Unified Threat Management (UTM)



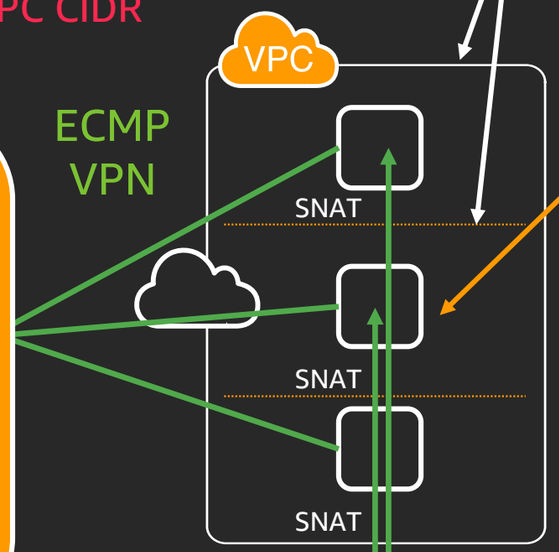
Spoke route table

Route	Destination
10.2.0.0/16	Local
10.0.0.0/8	tgw-xxxxxxxxx
100.64.0.0/16	tgw-xxxxxxxxx

VPCs will traffic as originated from the inline VPC CIDR

Inline VPC route table

Route	Destination
100.64.0.0/16	Local
10.0.0.0/8	tgw-xxxxxxxxx
0.0.0.0/0	igw-xxxxxxxxx



ECMP VPN

Apply SNAT between VPCs for flow affinity

Inline VPC 100.64.0.0/16

BGP prefix	Next hop
0.0.0.0/0	Local IP

BGP advertisement

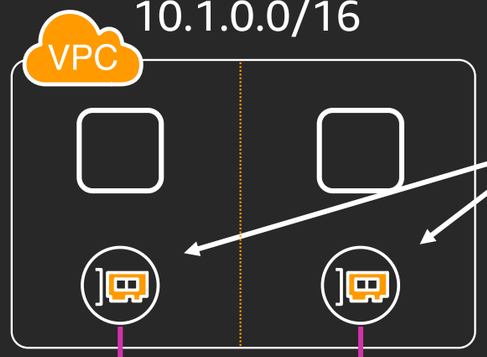
AWS re:Invent

aws

VPC to on-premises service insertion

VPC A

10.1.0.0/16



Spoke route table

Route	Destination
10.2.0.0/16	Local
On-premises	tgw-xxxxxxxxx
100.64.0.0/16	tgw-xxxxxxxxx

VPCs will see traffic sourced from the inline VPC CIDR range due to SNAT

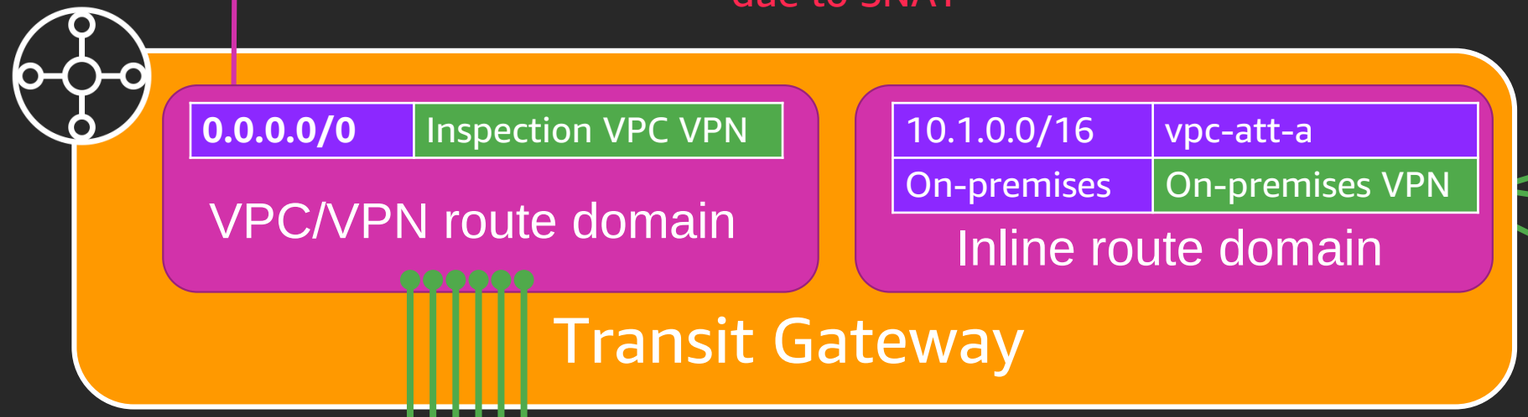
This forces VPC-to-VPC and between on-premises and VPCs through the inline VPC

Using an edge services model with VPN terminated on the firewalls may be simpler

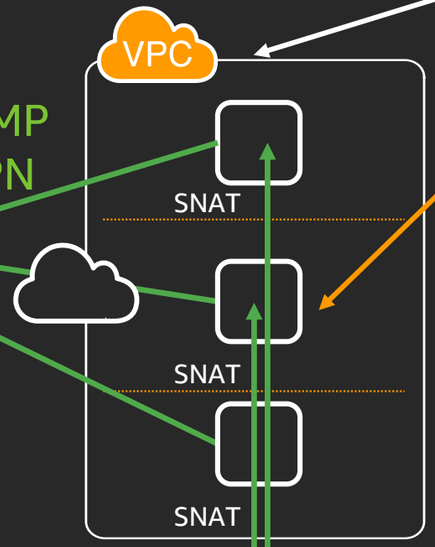
Inline VPC route table

Route	Destination
100.64.0.0/16	Local
10.0.0.0/8	tgw-xxxxxxxxx
On-premises	tgw-xxxxxxxxx
0.0.0.0/0	igw-xxxxxxxxx

Apply SNAT between VPCs for flow affinity



ECMP VPN



Inline VPC 100.64.0.0/16

On-premises

BGP prefix	Next hop
On-premises	Local IP

BGP advertisement

BGP prefix	Next hop
0.0.0.0/0	Local IP

BGP advertisement

AWS re:Invent

Transit Gateway launch partners

aviatrix

O E I M

Check Point
SOFTWARE TECHNOLOGIES LTD.

O E I M

CISCO
vEdge SD-WAN

O E I M

f5

O E I M

FORTINET

O E I M

paloalto NETWORKS
VM-Series

O E I M

Symantec
ProxySG

O E I M

HashiCorp
Terraform

O E I M

Outbound services

Edge services

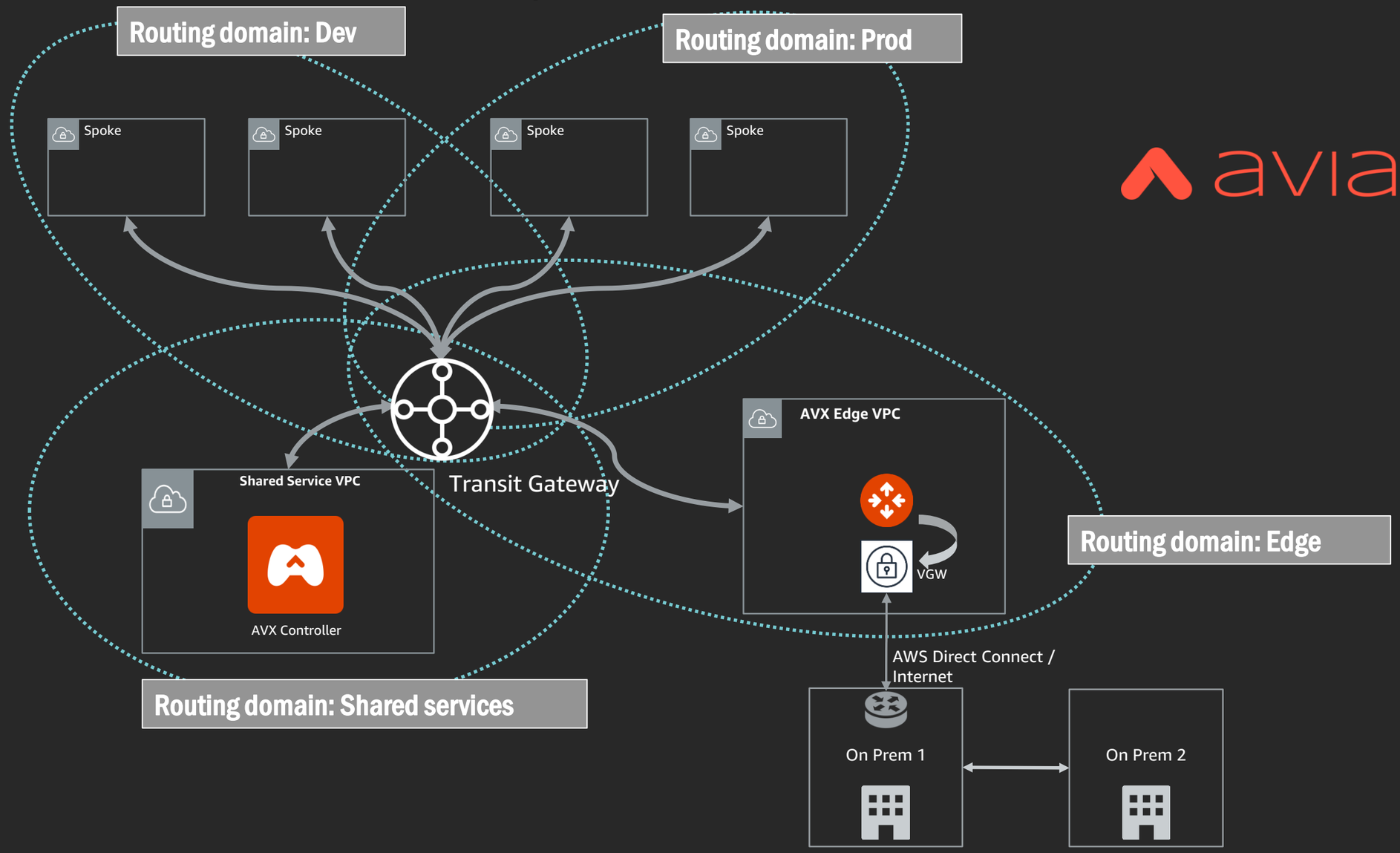
Inline services

Management

AWS re:Invent

aws

Orchestration: Dev & prod isolated transit network

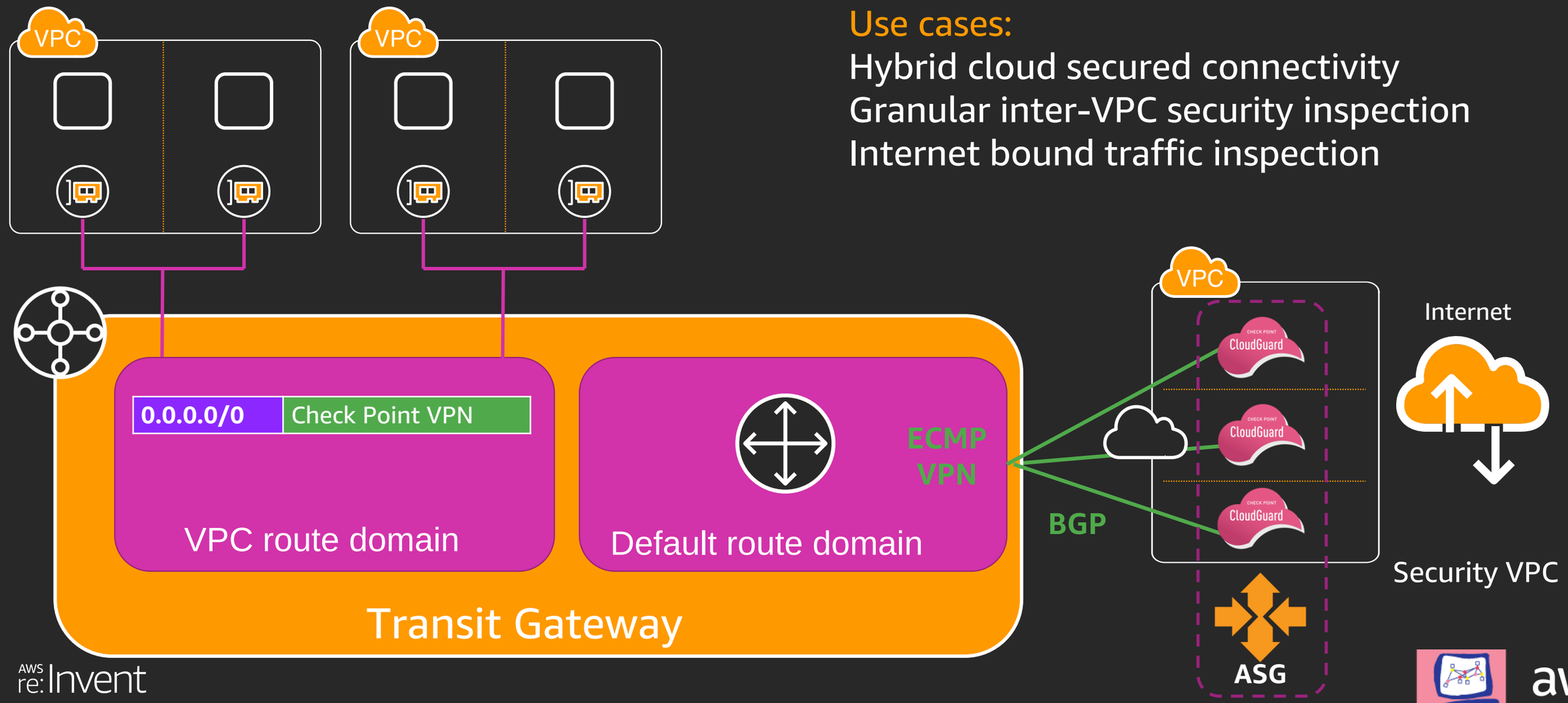


aviatrix

AWS re:Invent

aws

Check Point Auto-Scaling integration

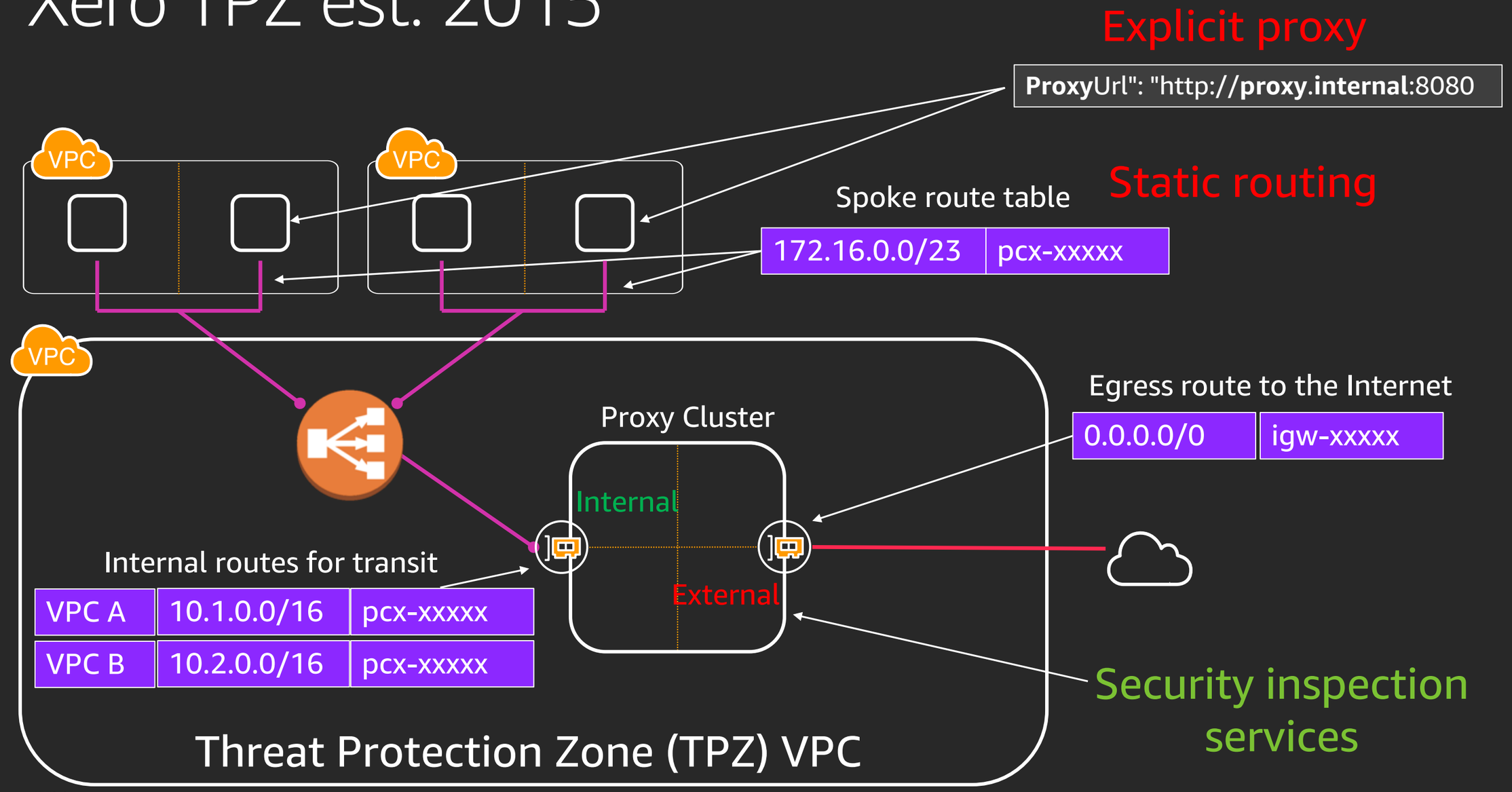


Use cases:

Hybrid cloud secured connectivity
Granular inter-VPC security inspection
Internet bound traffic inspection

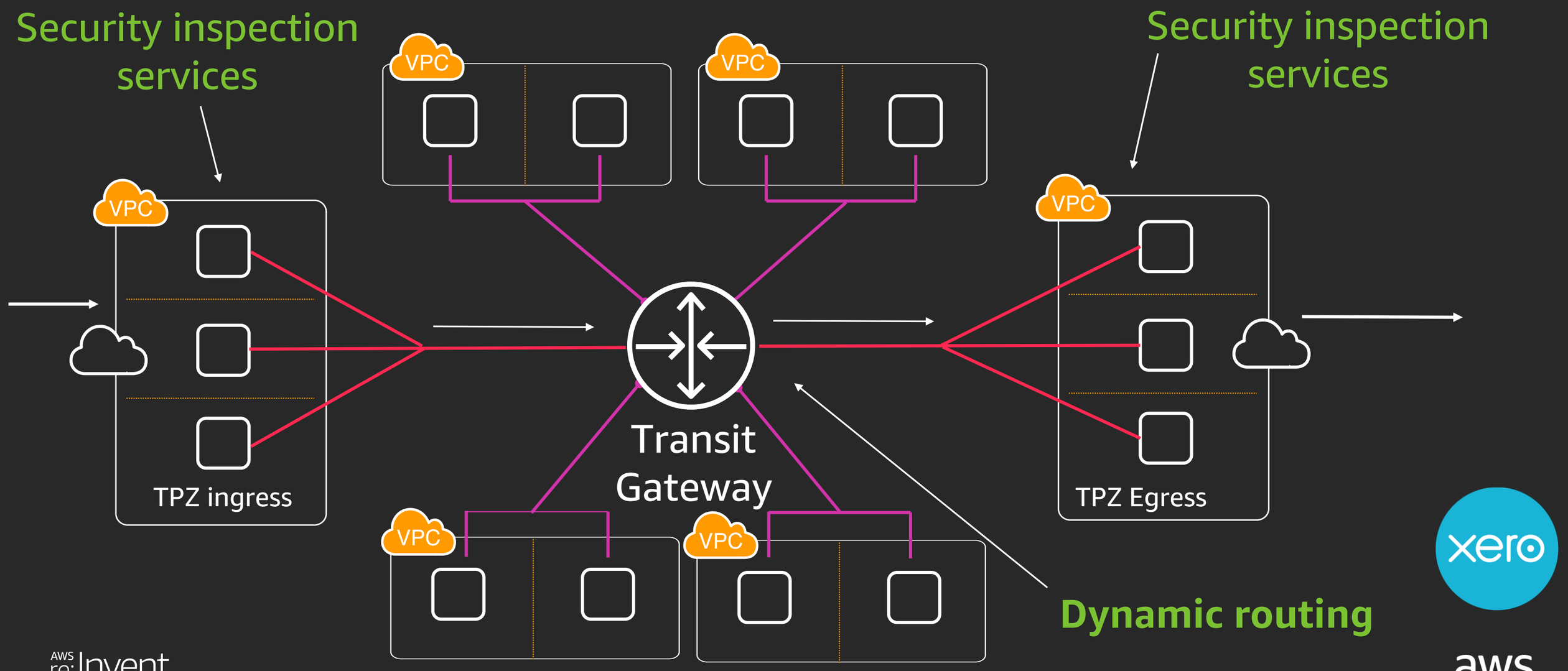
AWS re:Invent

Xero TPZ est. 2015



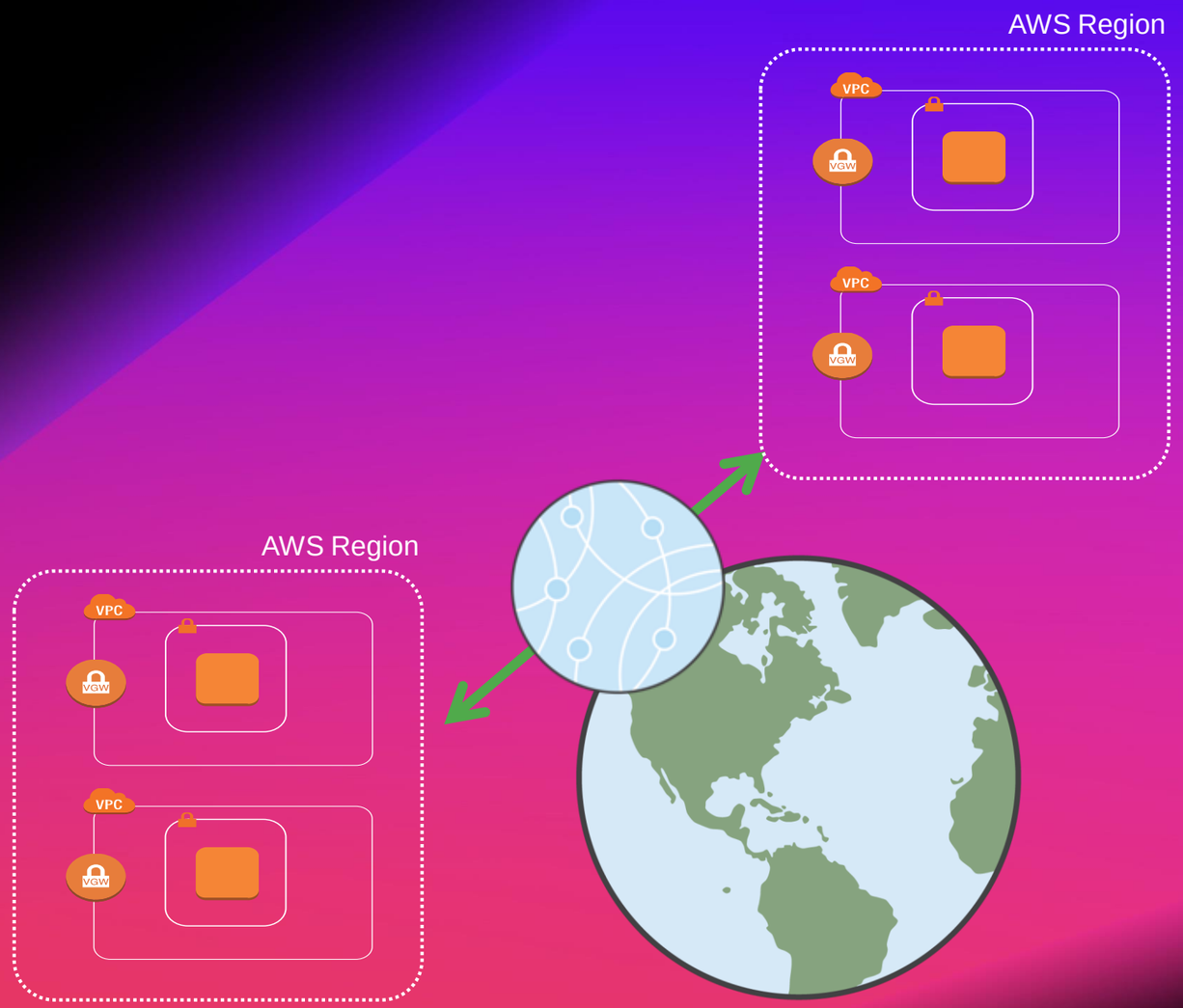
AWS re:Invent

Xero TPZ future state

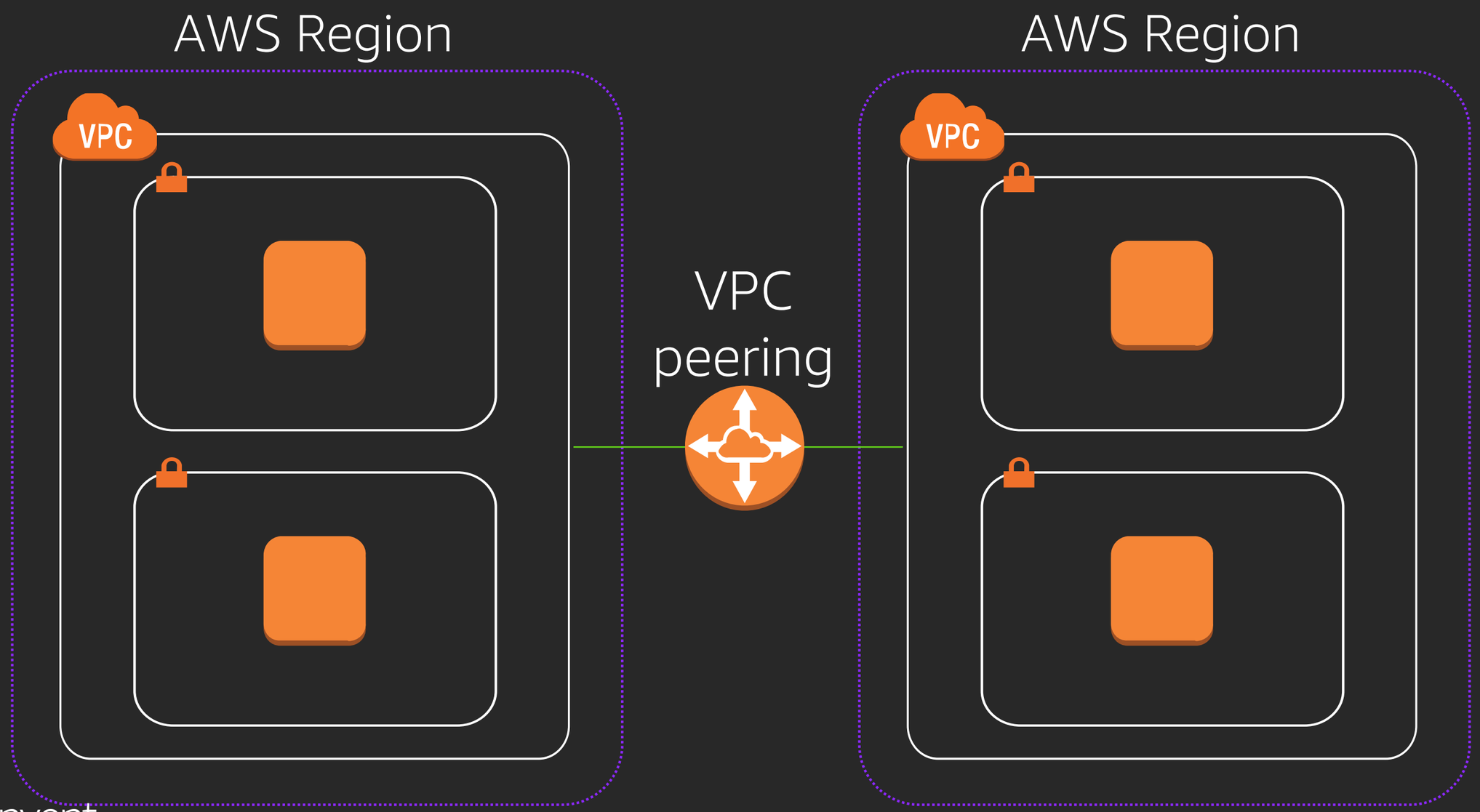


AWS re:Invent

Multiple Regions

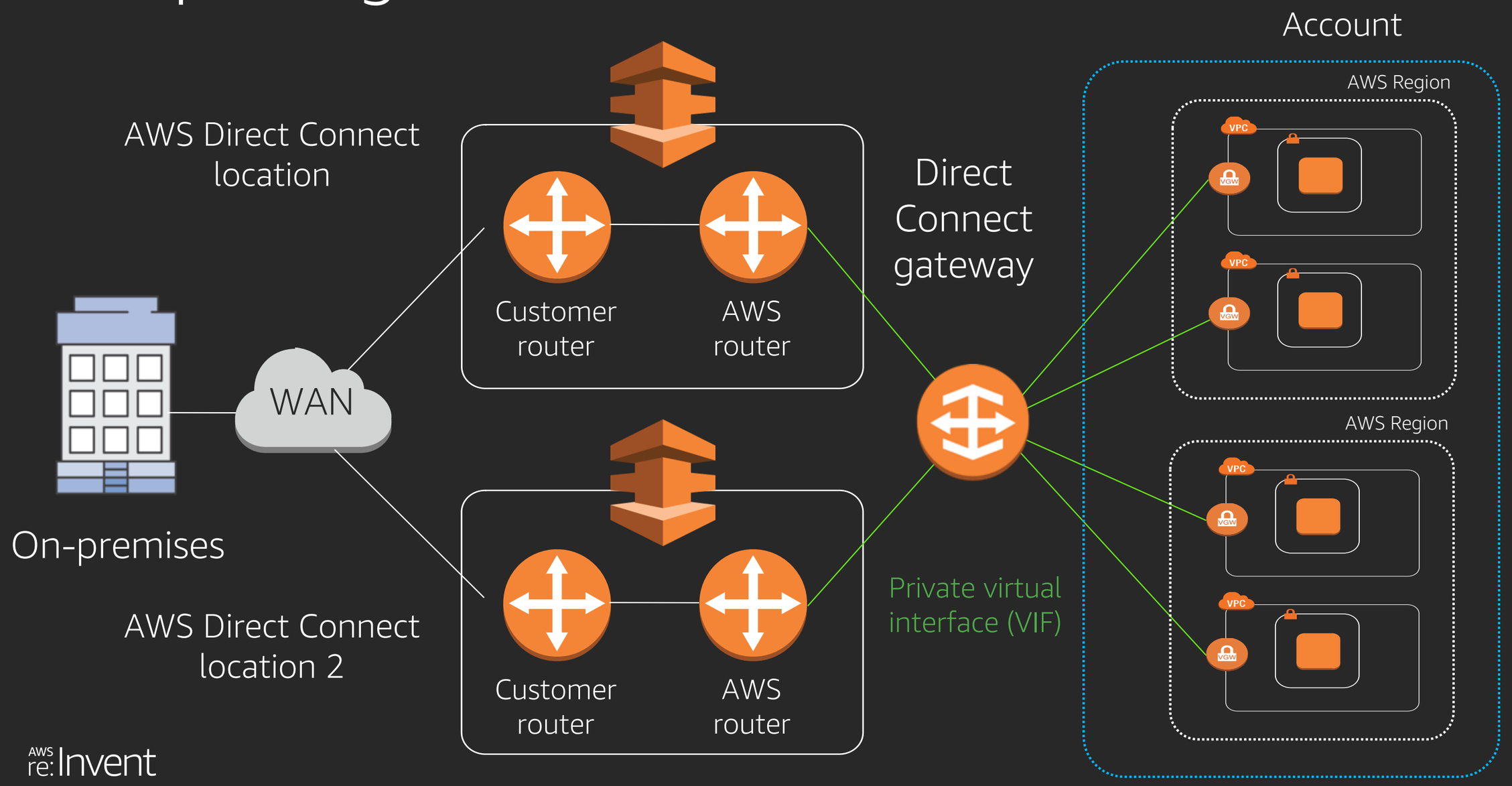


Inter-region VPC peering



AWS re:Invent

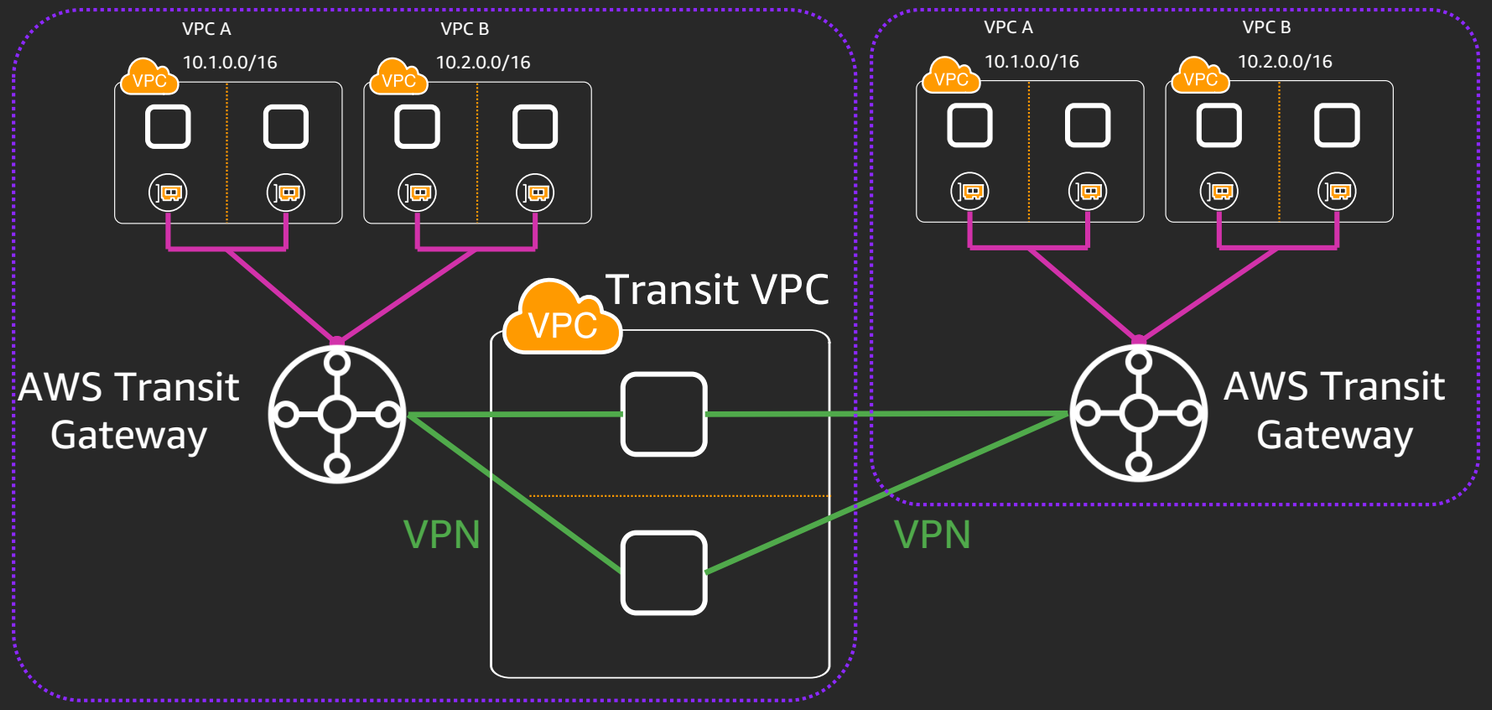
Multiple Regions



AWS re:Invent

aws

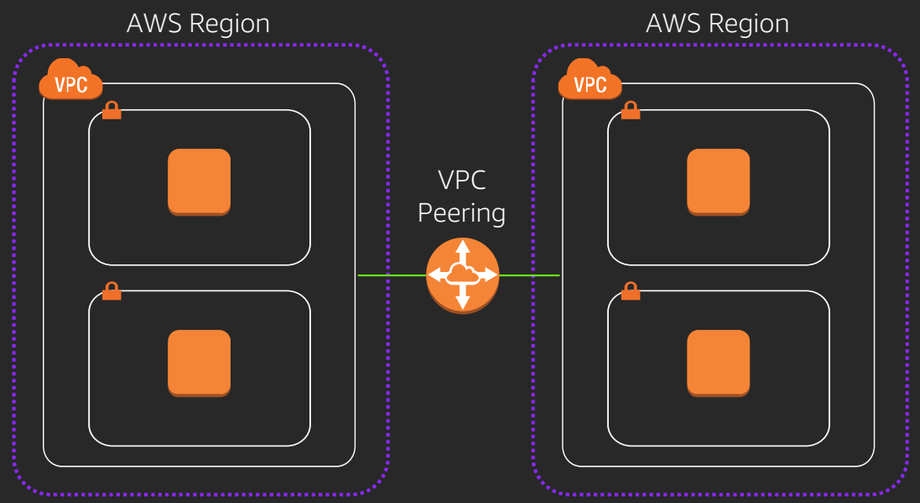
Transit Gateway in multiple Regions



Connecting Regions with VPN

Transit Gateway inter-region support coming soon!

Inter-region peering



Conclusions

Takeaways

We have tools and architectures that horizontally scale to many VPCs

There's wiggle room for your specific use cases

Use services in combination to meet scale and security requirements

AWS re:Invent

aws

Advice

- Networking changes fast, no more crystal balls
- Start simple! Stay simple. Reduce complexity to smaller scopes
- Segment and modify as needed
- Experiment and test

Thank you!

Nick Matthews
@nickpowpow

AWS re:Invent

aws

Please complete the session survey in the mobile app.