AWS re:Invent

NET402

Networking wizards: Ask me anything (AMA)

Nick Matthews

Principal Solutions Architect
Amazon Web Services

@nickpowpow

Matt Lehwess

Principal Solutions Architect
Amazon Web Services

AWS re:Invent

AWS Transit Gateway

Regional service

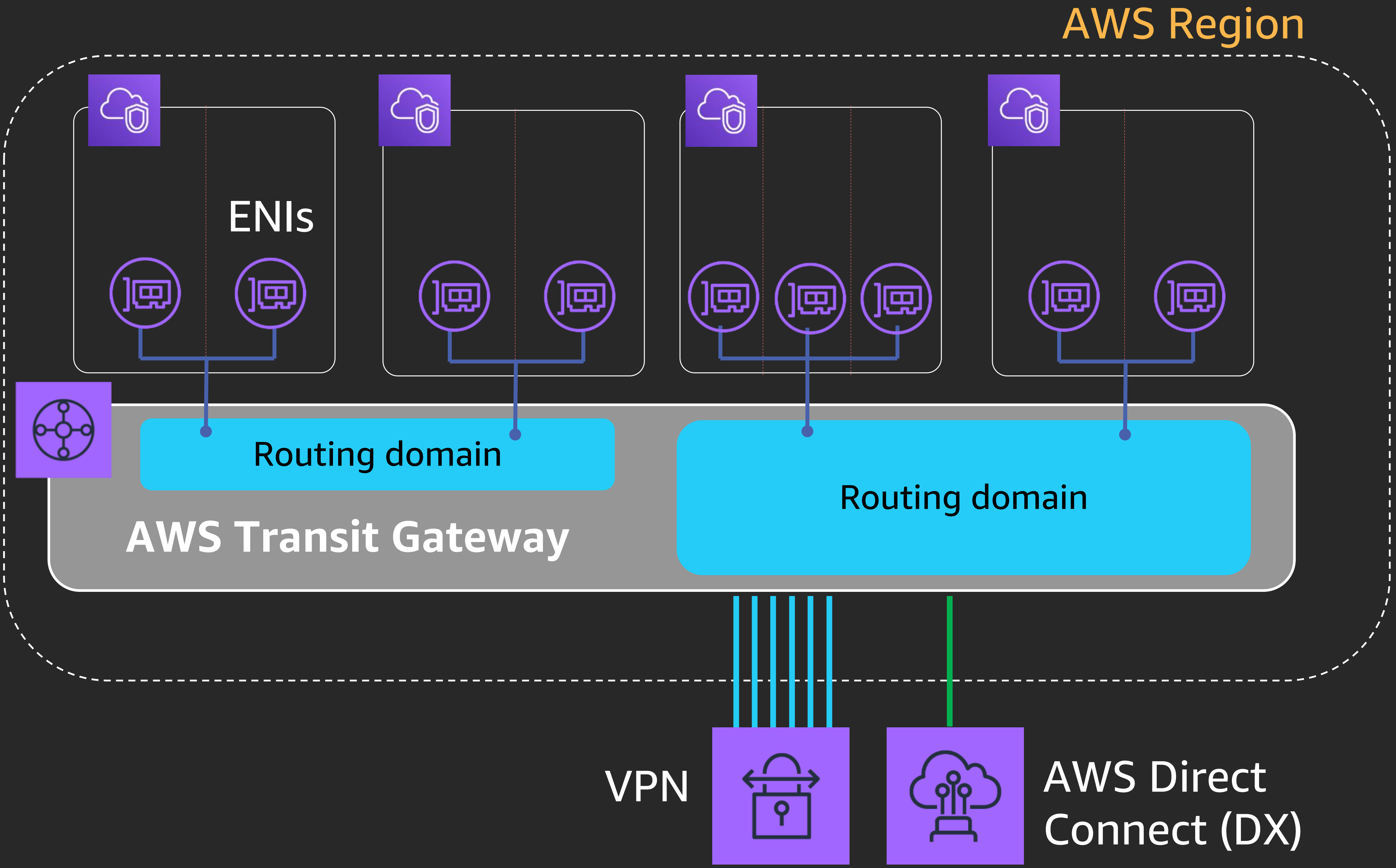
- Centralize VPN and AWS Direct Connect

Scalable

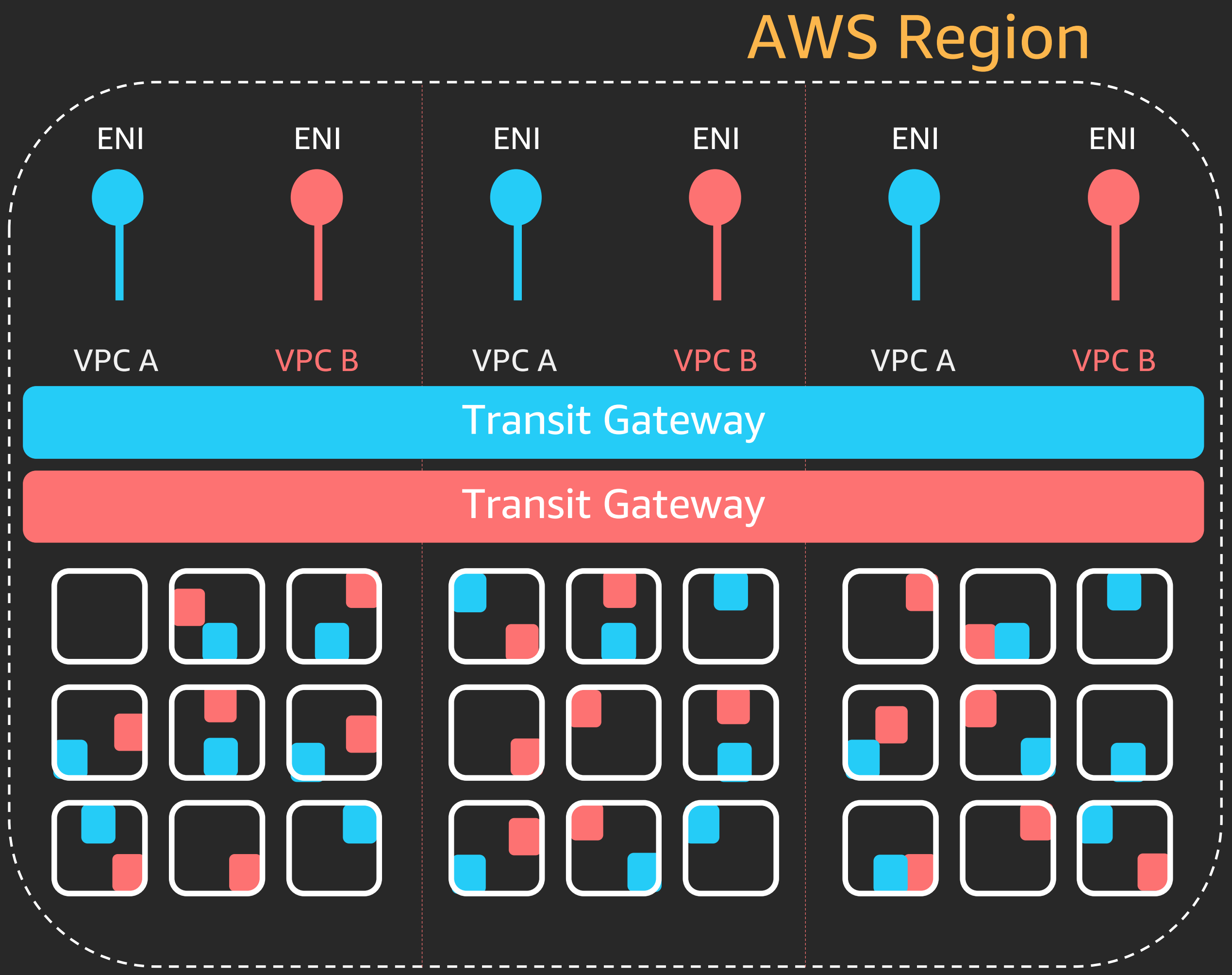
- Thousands of VPCs **across accounts**
- Spread traffic over many VPN connections

Flexible routing

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



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

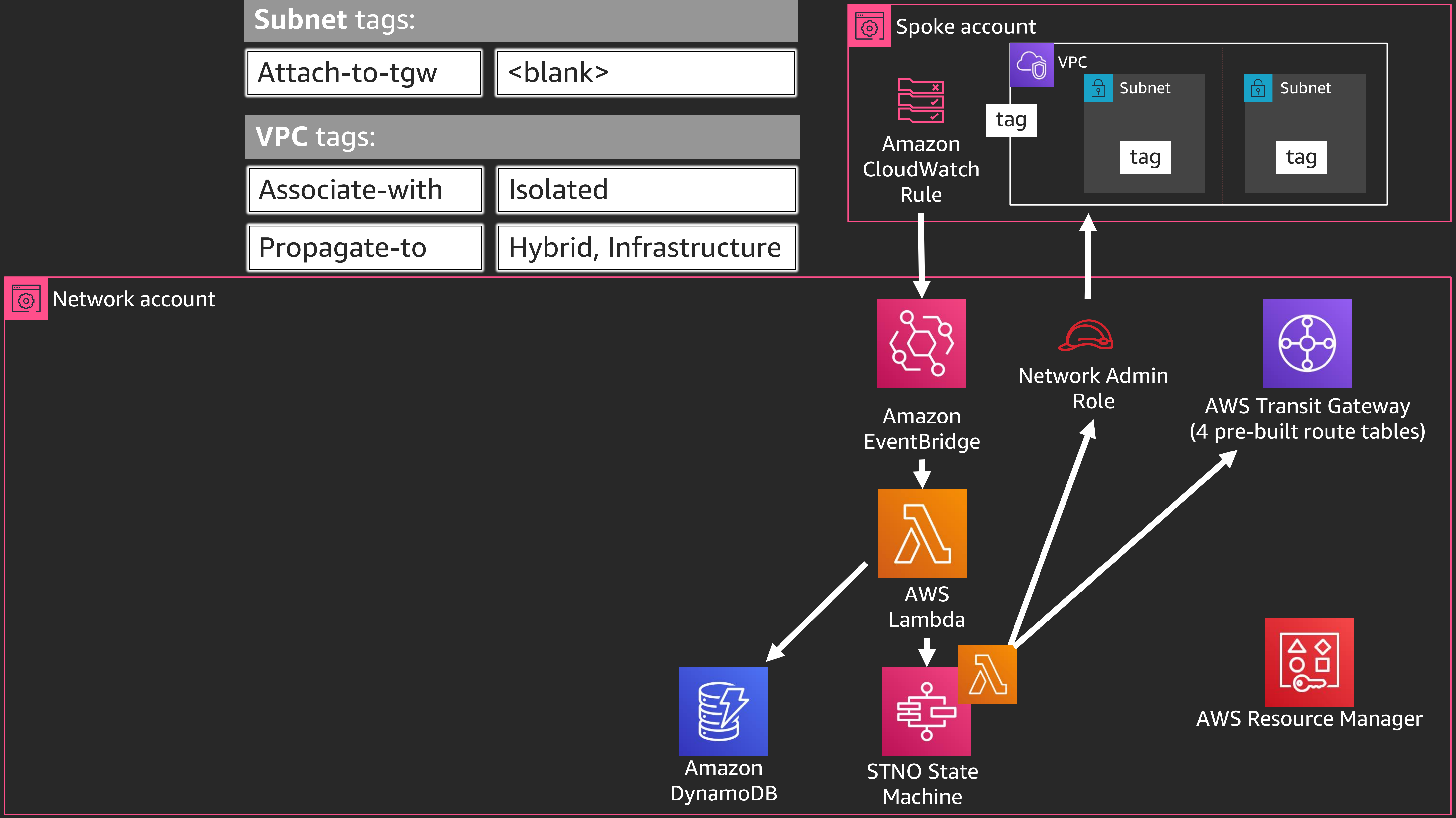
- Horizontally-scalable state management
- Terabits of multi-tenant capacity
- Supports NLB, NAT Gateway, Amazon EFS and now Transit Gateway

Subnet tags:

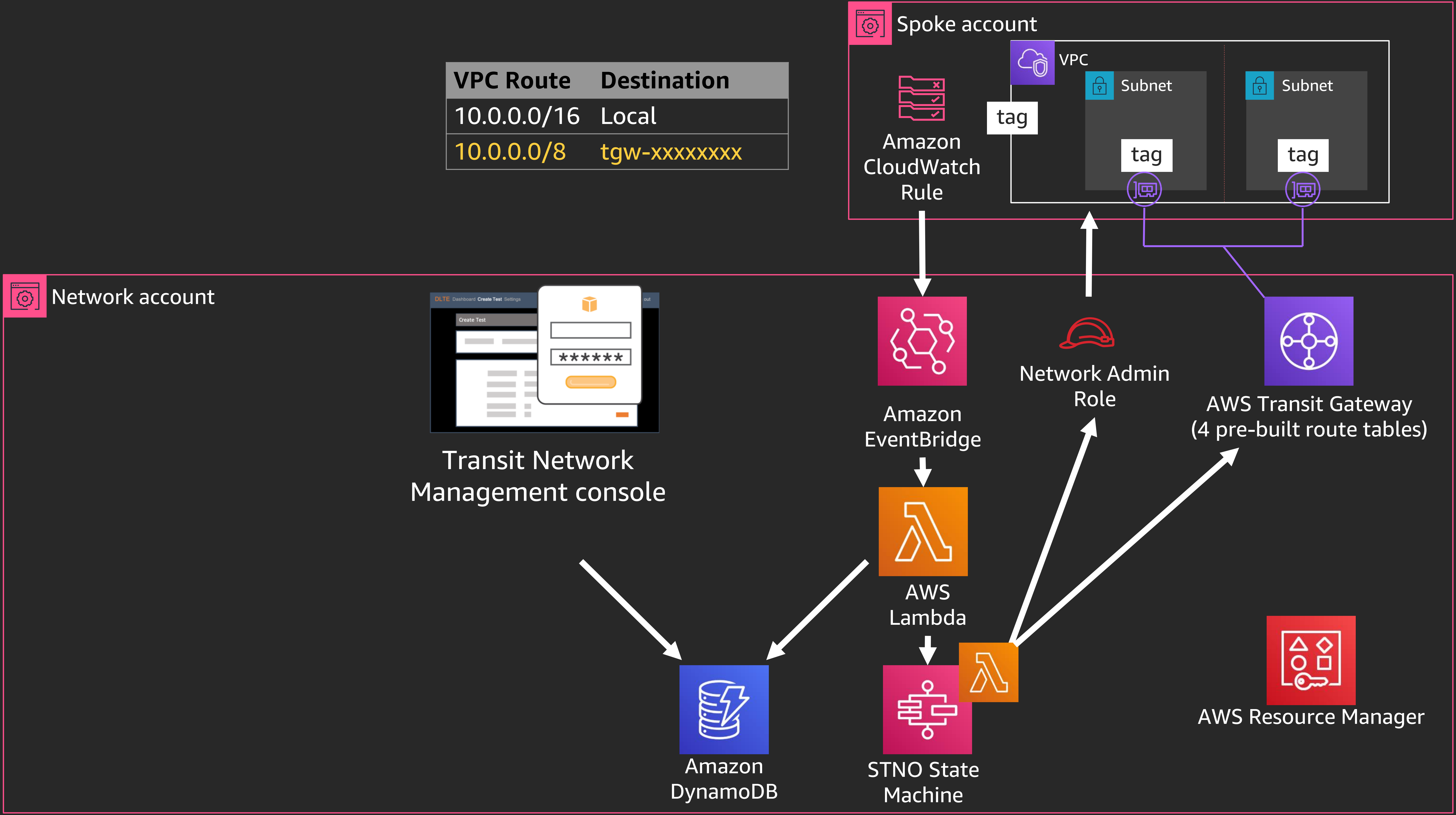
Attach-to-tgw	<blank>

VPC tags:

Associate-with	Isolated
Propagate-to	Hybrid, Infrastructure



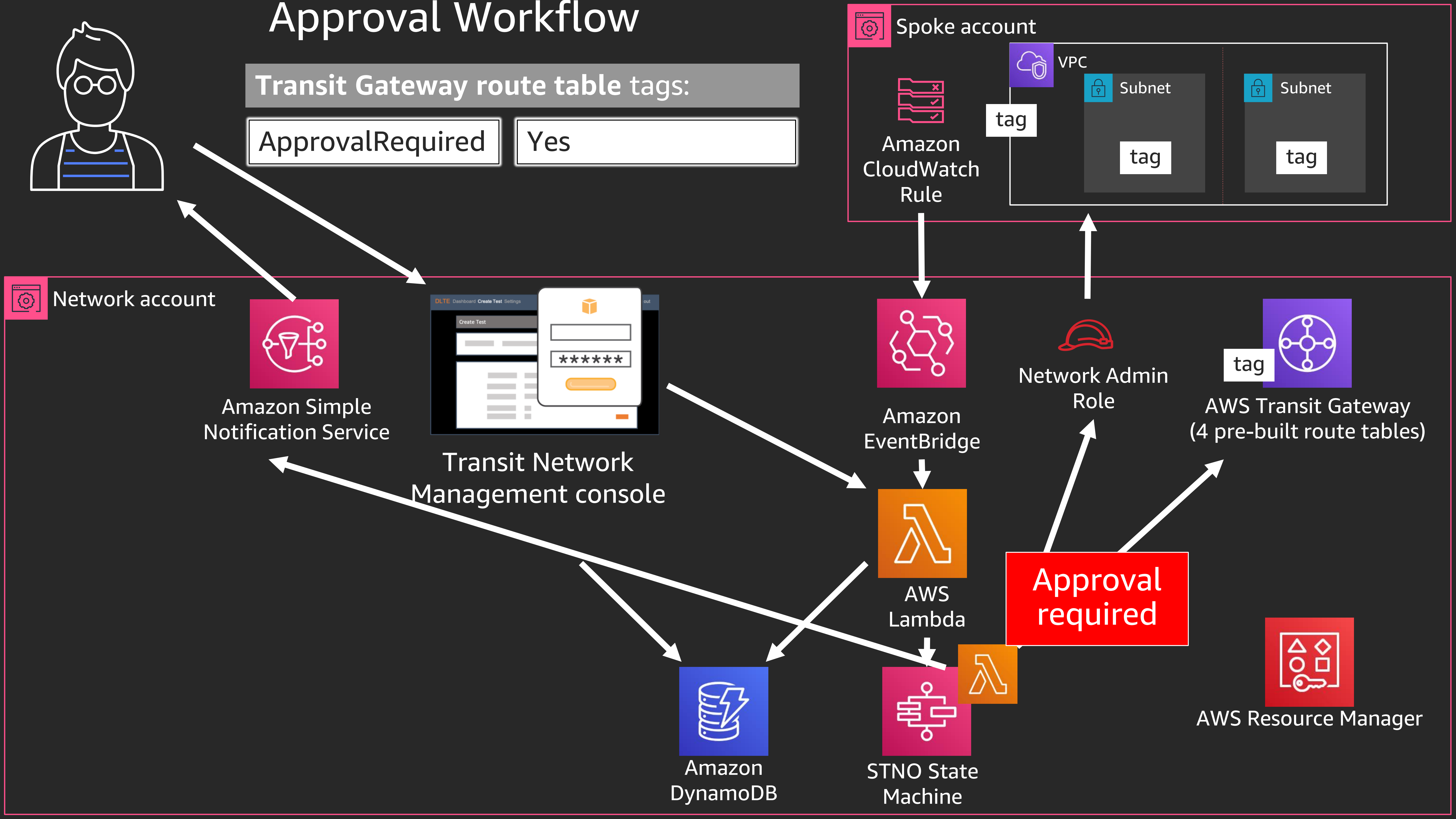
VPC Route	Destination
10.0.0.0/16	Local
10.0.0.0/8	tgw-xxxxxxxx



Approval Workflow

Transit Gateway route table tags:

ApprovalRequired	Yes

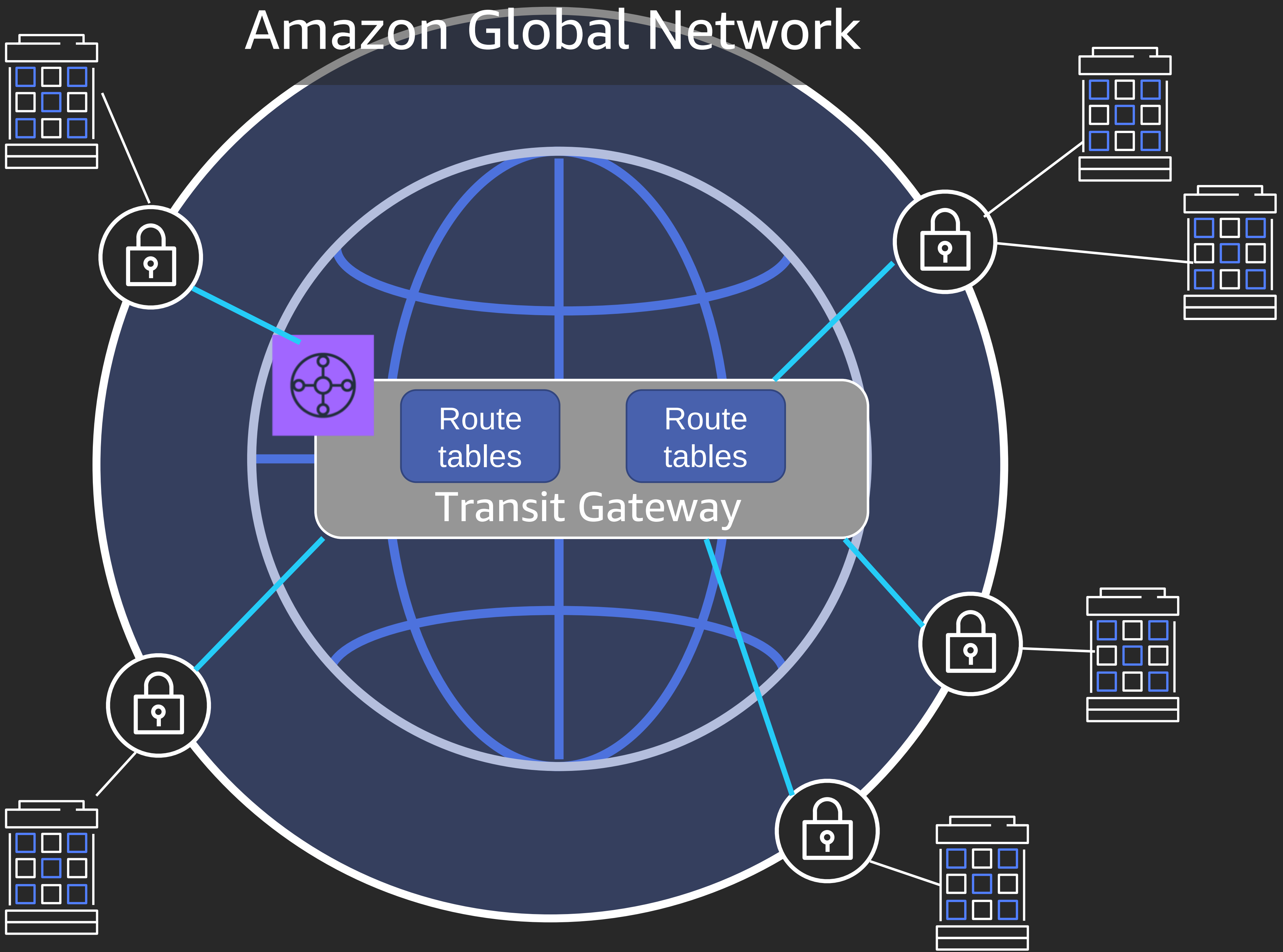


Try STNO

http://tiny.cc/aws-stno

Accelerated VPN

New



Leverage Amazon's Global Network

- Combine Amazon Global Accelerator with VPN
- Lower latency
- Ideal for branch connectivity

Interface method

Method one: Interface attachment

Spoke route table

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

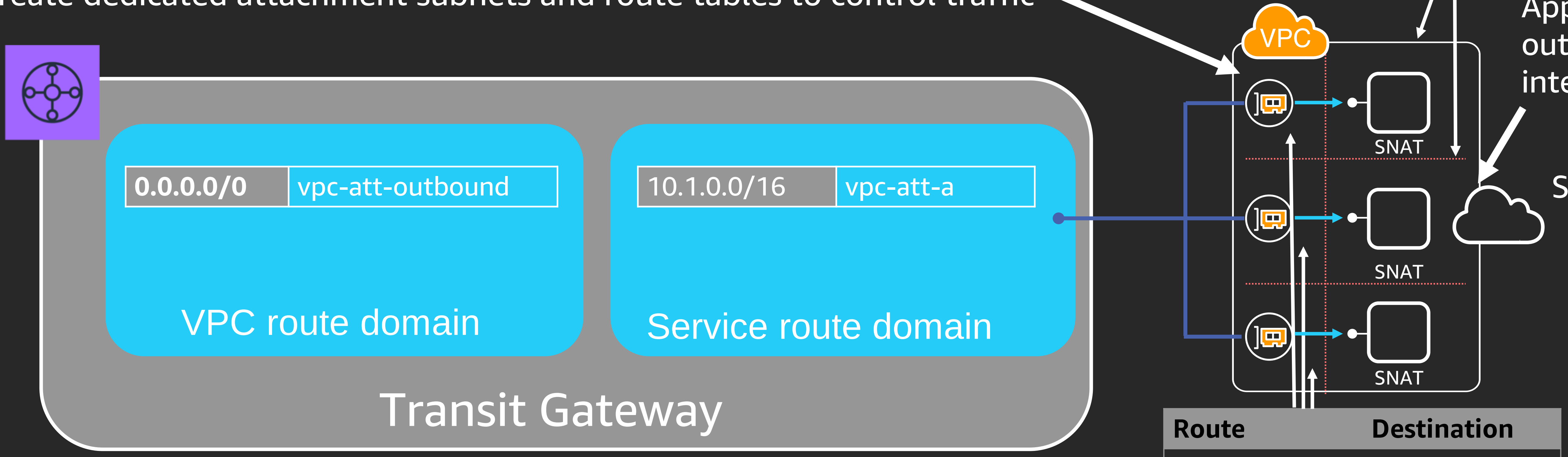
Outbound VPC route table

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

Create return route to VPCs

Control egress behavior with a ‘public’ subnet

Create dedicated attachment subnets and route tables to control traffic



Apply SNAT outbound to the internet

Service VPC

Route	Destination
0.0.0.0/0	eni-xxxxxxx

Interface method

Method one: NAT gateway

Spoke route table

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

Outbound VPC route table

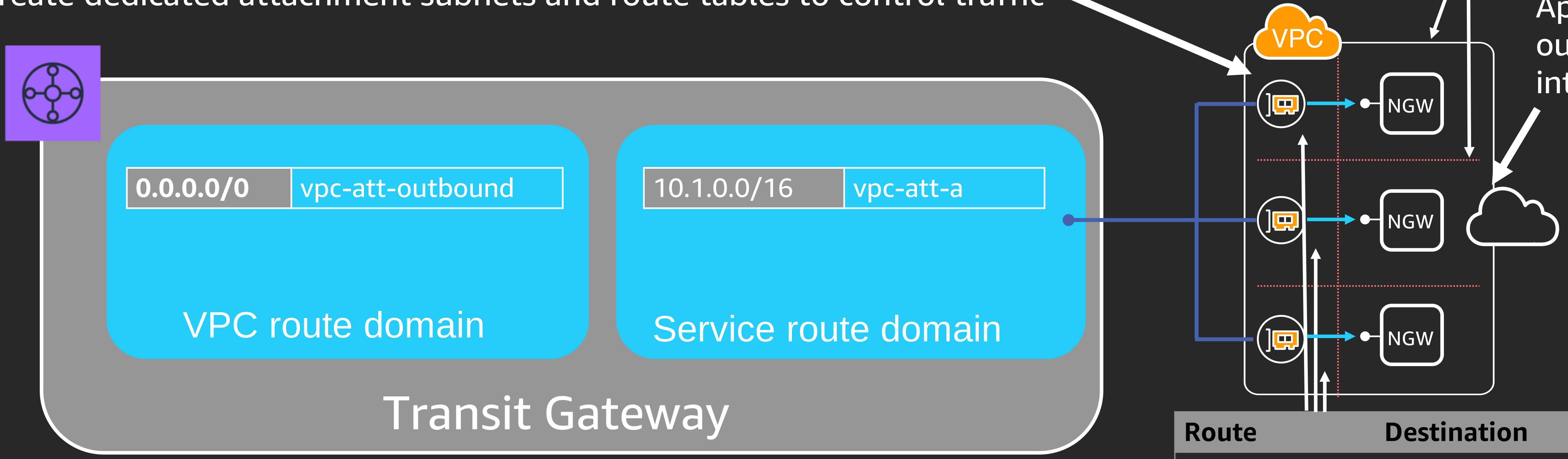
Route	Destination
100.64.0.0/16	Local
10.0.0.0/8	tgw-xxxxxxxxx
0.0.0.0/0	igw-xxxxxxxxx

Create return route to VPCs

Control egress behavior with a ‘public’ subnet

Create dedicated attachment subnets and route tables to control traffic

Apply SNAT outbound to the internet



0.0.0.0/0	vpc-att-outbound

10.1.0.0/16	vpc-att-a

Route	Destination
0.0.0.0/0	ngw-xxxxxxx

Interface method

Interface service insertion design notes

Simpler performance pattern, DIY heatlh checks

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

Instance must be able to support:

- Source NAT, or add NAT gateway

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

VPN method

Method two: VPN attachment

Spoke route table

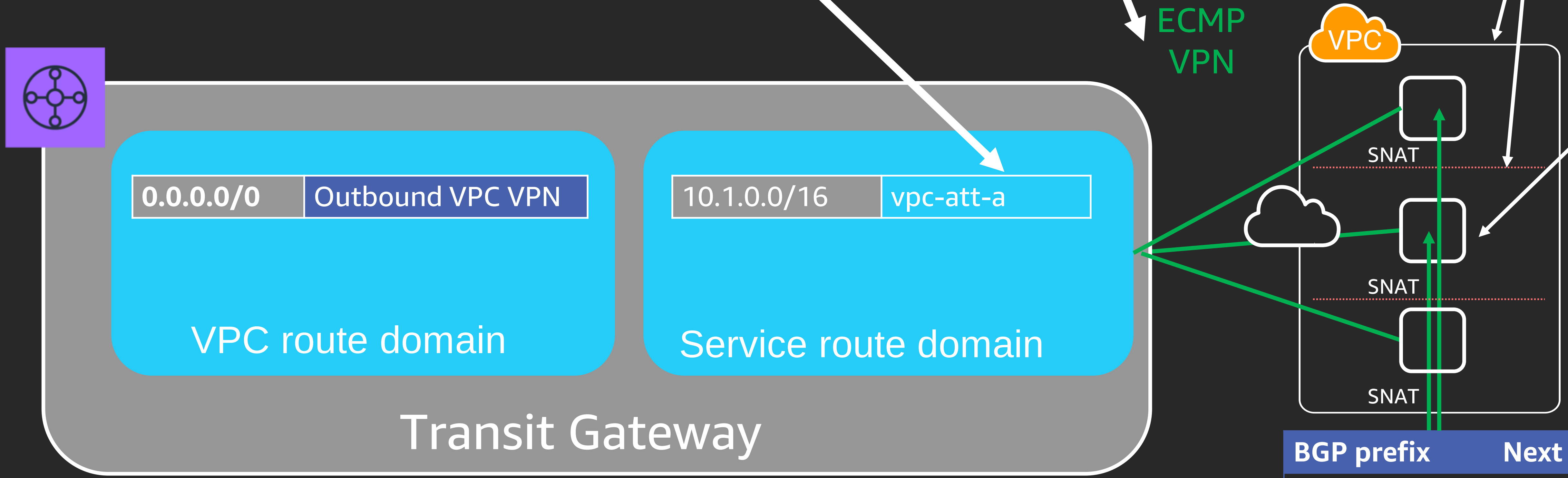
Route	Destination
10.1.0.0/16	Local
0.0.0.0/0	tgw-xxxxxxxxx

Outbound VPC route table

Route	Destination
100.64.0.0/16	Local
0.0.0.0/0	igw-xxxxxxxxx

Load balance traffic across many VPN tunnels

VPC routes will be advertised over BGP



0.0.0.0/0	Outbound VPC VPN

10.1.0.0/16	vpc-att-a

Apply SNAT outbound to the internet

Service VPC

BGP prefix	Next hop
0.0.0.0/0	Local IP

BGP advertisement

VPN method

VPN service insertion design notes

Horizontally scalable service pattern, more overhead

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

Instance must be able to support:

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

Performance

- IPsec overhead
- Compatible with auto-scaling architectures
- No cumulative bandwidth limit, each tunnel ~1.25 gbps

High availability

- BGP and VPN Dead Peer Detection handle failover
- No API calls required for fault tolerance

Stateful services

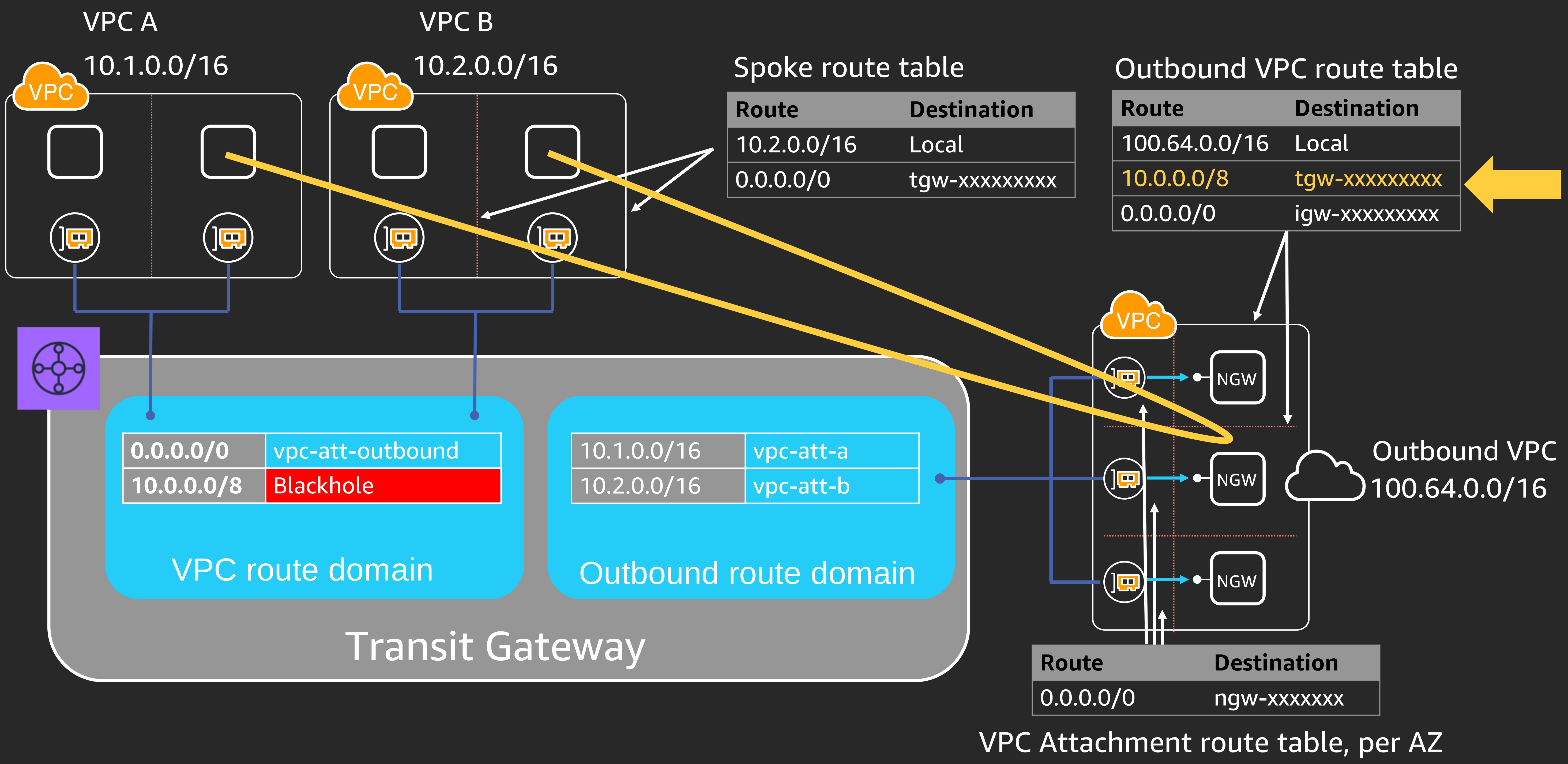
- Use Source NAT to guarantee the return flow to the same instance

Interface method

Outbound services: Interface

Use cases:

NAT gateways, services without VPN support



Spoke route table

Route	Destination
10.2.0.0/16	Local
0.0.0.0/0	tgw-xxxxxxxxx

Outbound 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

VPC route domain

0.0.0.0/0	vpc-att-outbound
10.0.0.0/8	Blackhole

Outbound route domain

10.1.0.0/16	vpc-att-a
10.2.0.0/16	vpc-att-b

VPC Attachment route table, per AZ

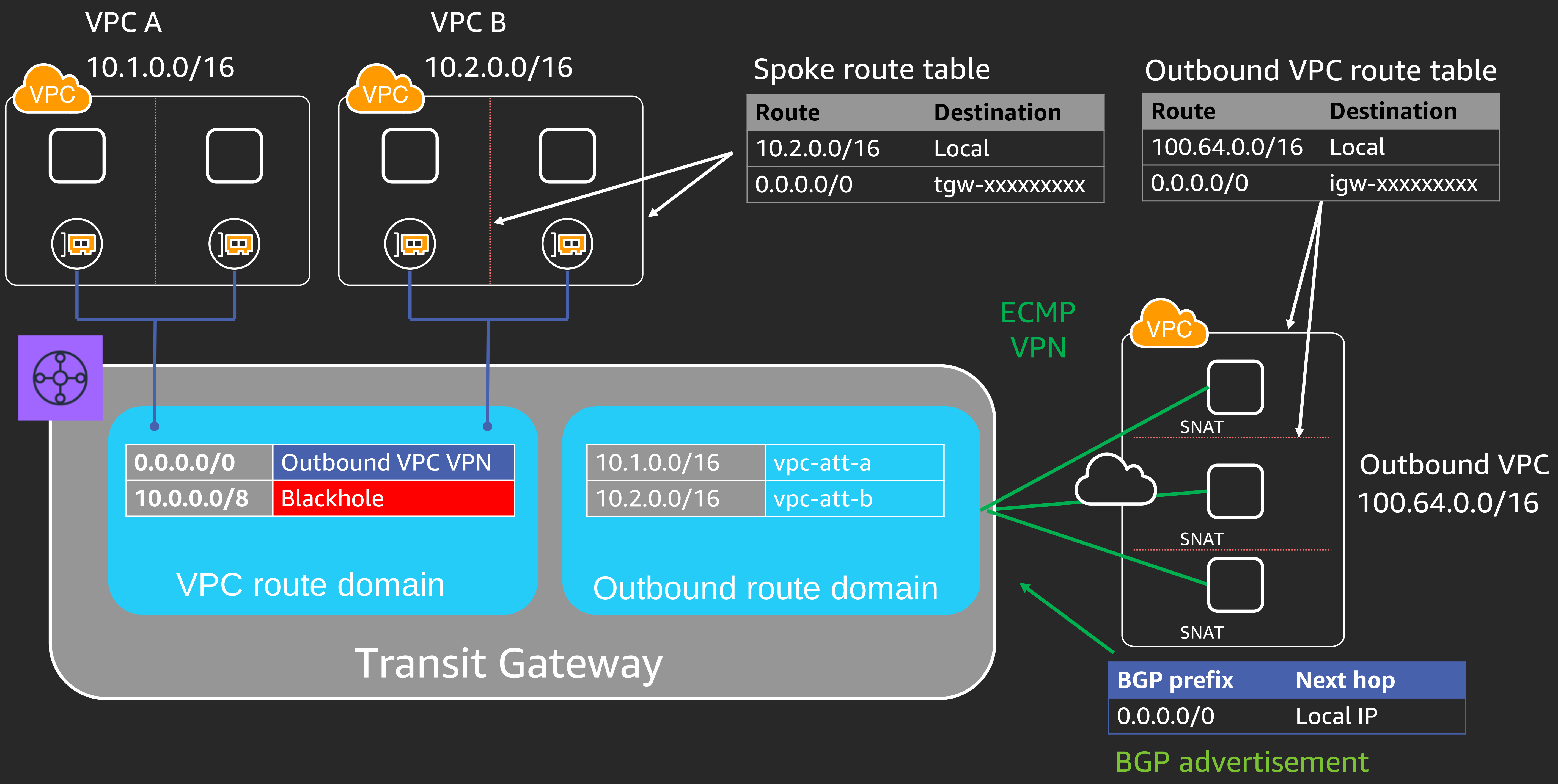
Route	Destination
0.0.0.0/0	ngw-xxxxxxx

VPN method

Outbound services: VPN

Use cases:

URL filtering, firewalls, IPS, web proxy services



Spoke route table

Route	Destination
10.2.0.0/16	Local
0.0.0.0/0	tgw-xxxxxxxxx

Outbound VPC route table

Route	Destination
100.64.0.0/16	Local
0.0.0.0/0	igw-xxxxxxxxx

VPC route domain

0.0.0.0/0	Outbound VPC VPN
10.0.0.0/8	Blackhole

Outbound route domain

10.1.0.0/16	vpc-att-a
10.2.0.0/16	vpc-att-b

BGP prefix	Next hop
0.0.0.0/0	Local IP

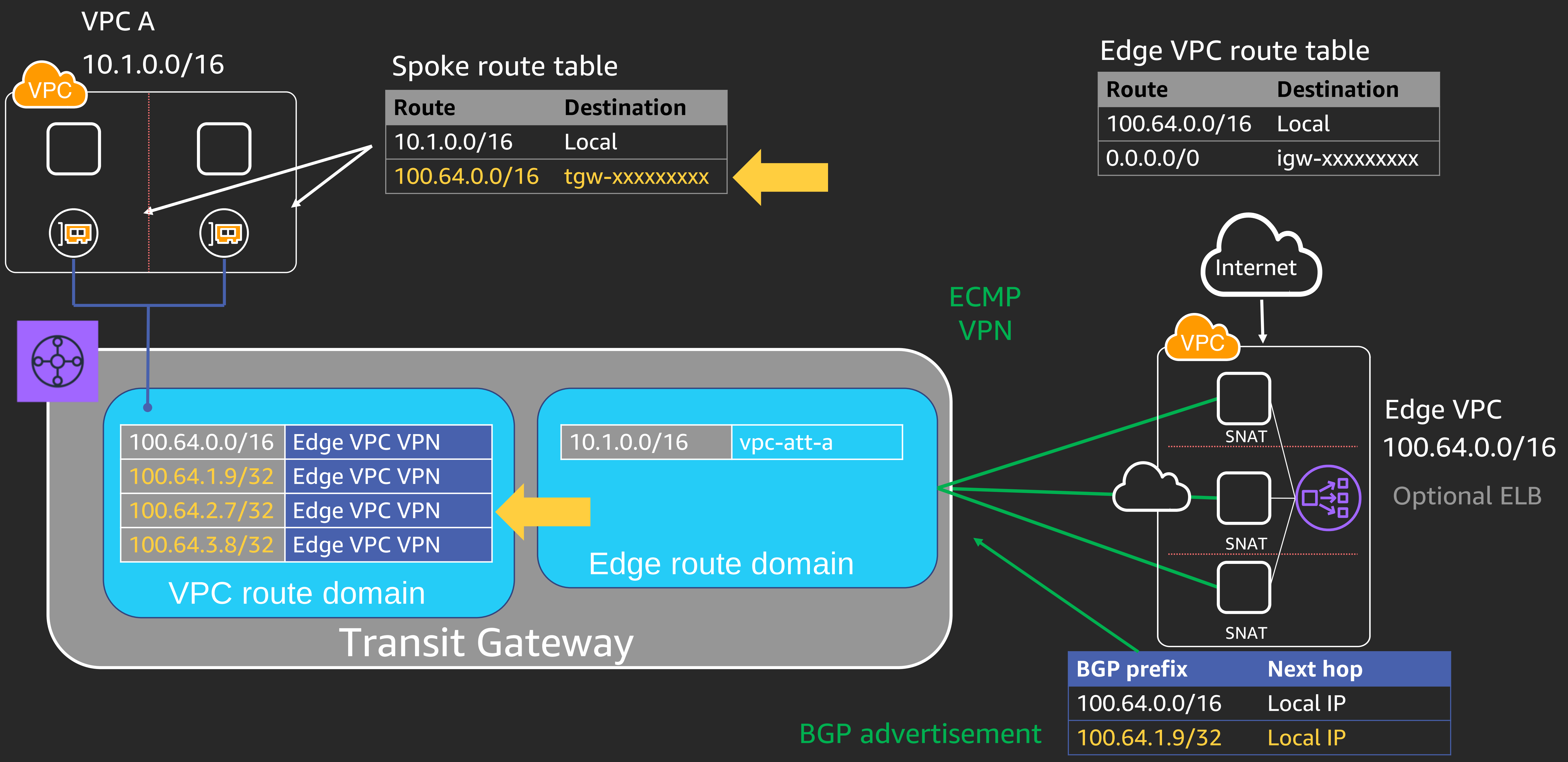
BGP advertisement

Ingress services

VPN method

Use cases:

WAF, inspection, Load balancing



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
0.0.0.0/0	igw-xxxxxxxxx

VPC route domain

100.64.0.0/16	Edge VPC VPN
100.64.1.9/32	Edge VPC VPN
100.64.2.7/32	Edge VPC VPN
100.64.3.8/32	Edge VPC VPN

Edge route domain

10.1.0.0/16	vpc-att-a

Transit Gateway

Edge VPC
100.64.0.0/16

Optional ELB

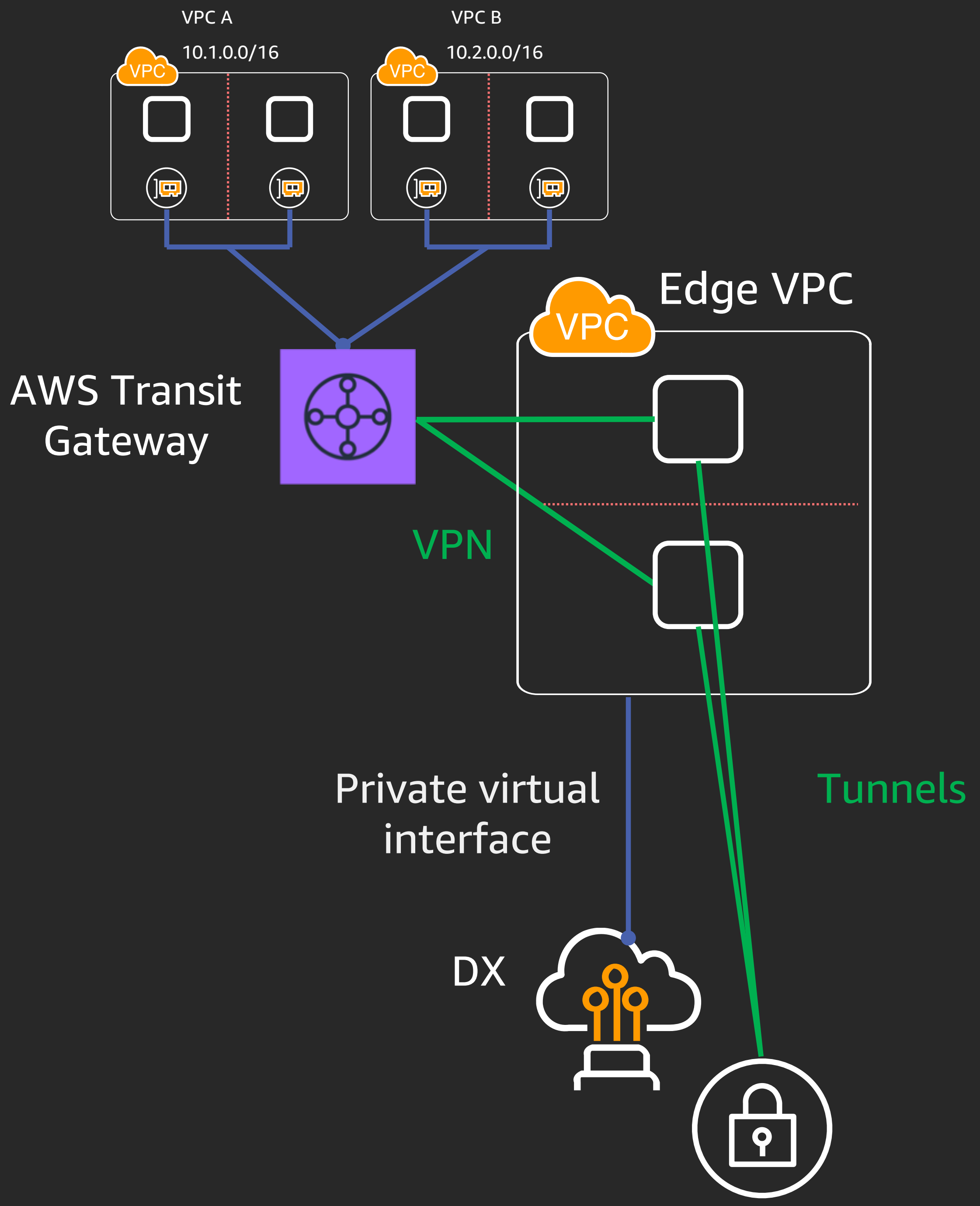
BGP advertisement

BGP prefix	Next hop
100.64.0.0/16	Local IP
100.64.1.9/32	Local IP

Edge services: SDWAN, VPN, Firewalls

Use an edge services VPC in front of Transit Gateway

- Encryption over DX or the internet
- Scalable VPN access for third-party VPN, SDWAN
- Also how used to migrate or extend existing Transit VPCs
- Helpful for hosted VIF (<1 Gbps) DX
- Ingress firewall inspection use case

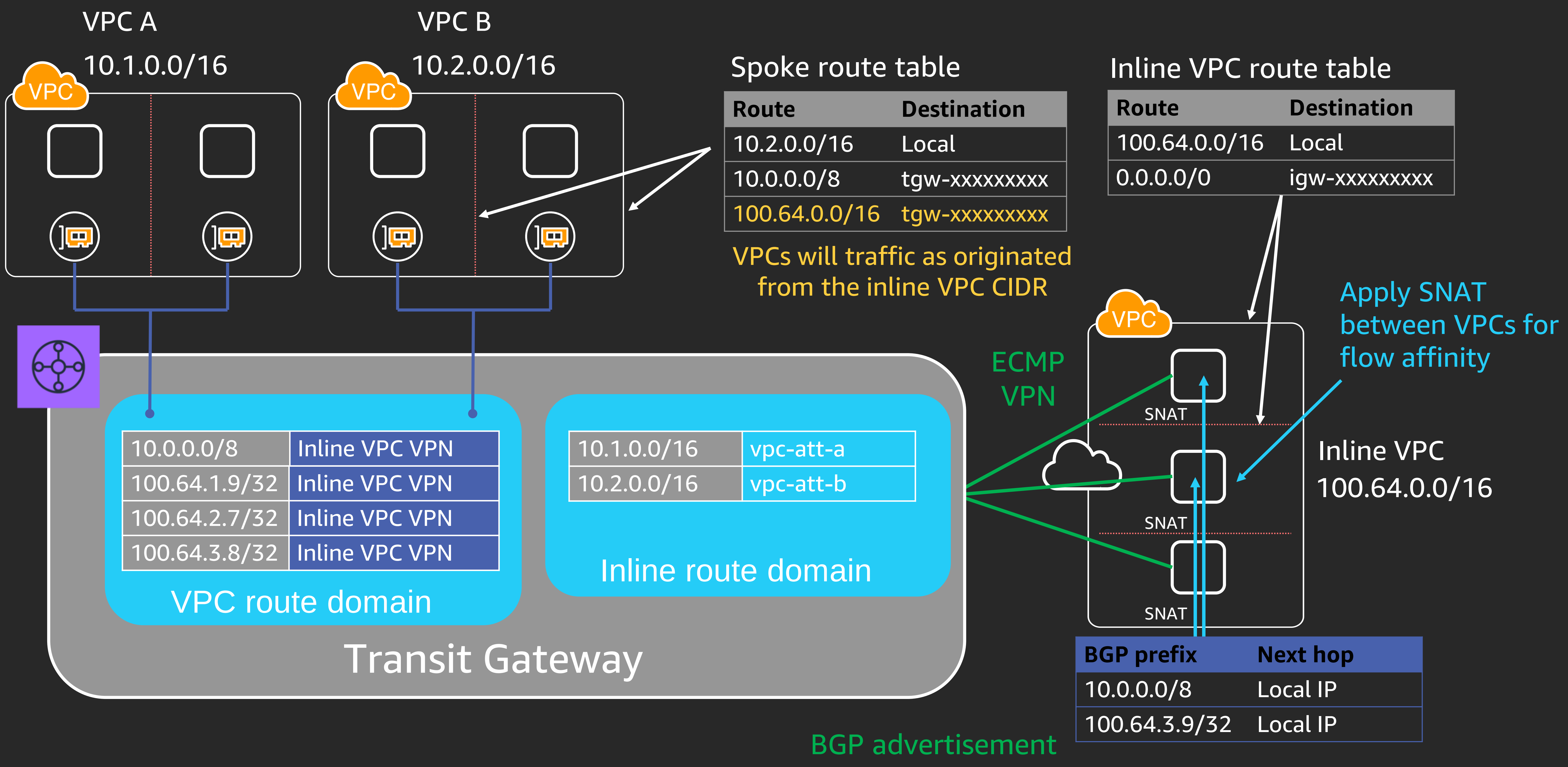


Inline service: VPN

VPN method

Use cases:

Intrusion detection/prevention (IDS/IPS), firewalls



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
0.0.0.0/0	igw-xxxxxxxxx

VPC route domain

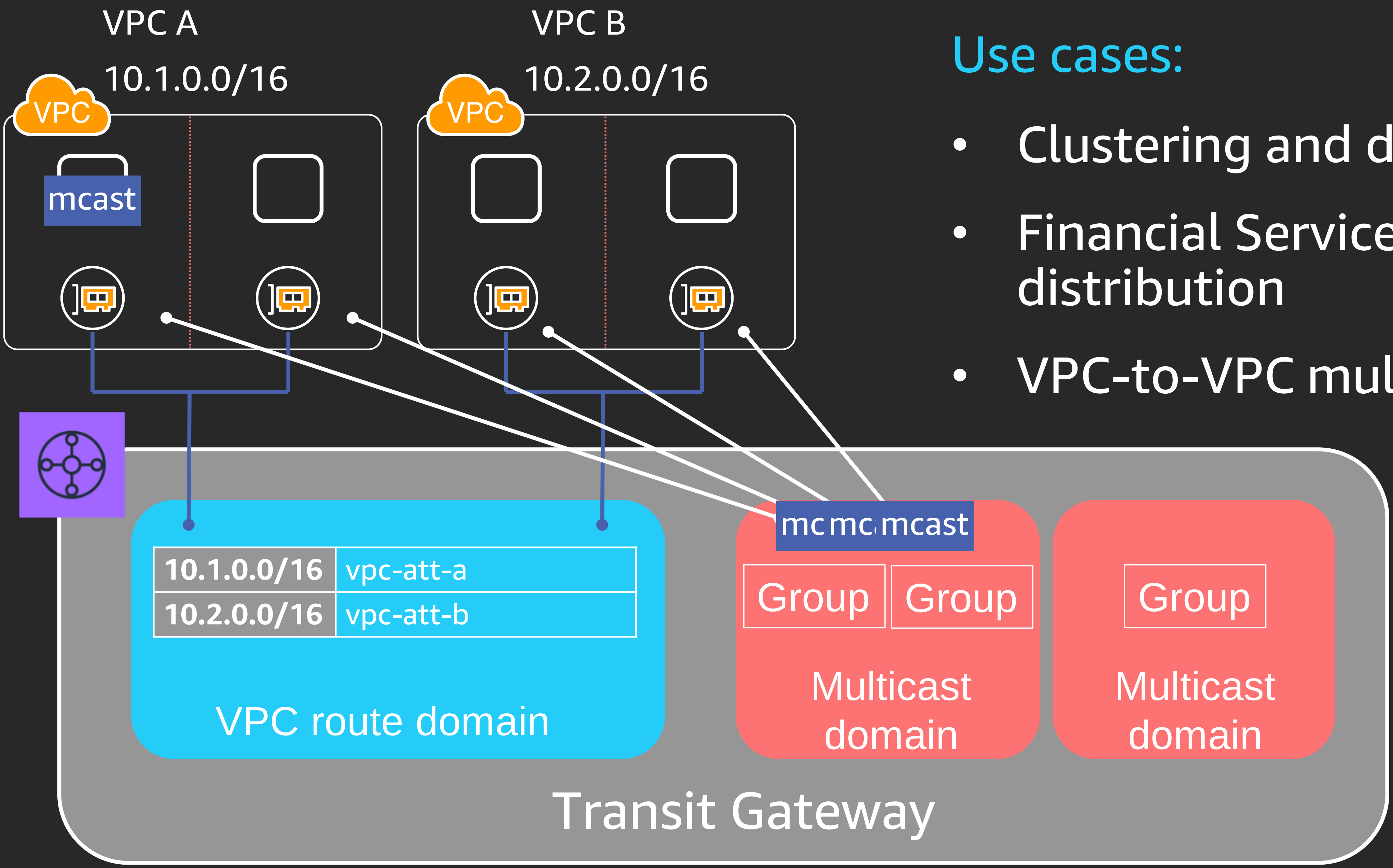
10.0.0.0/8	Inline VPC VPN
100.64.1.9/32	Inline VPC VPN
100.64.2.7/32	Inline VPC VPN
100.64.3.8/32	Inline VPC VPN

Inline route domain

10.1.0.0/16	vpc-att-a
10.2.0.0/16	vpc-att-b

BGP prefix	Next hop
10.0.0.0/8	Local IP
100.64.3.9/32	Local IP

Multicast on AWS Transit Gateway



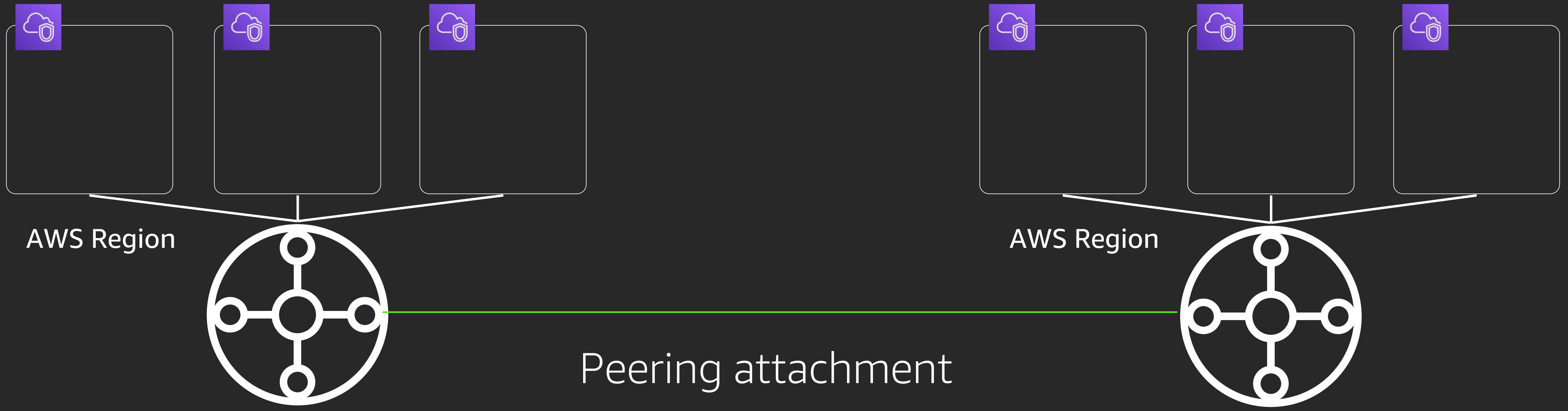
10.1.0.0/16	vpc-att-a
10.2.0.0/16	vpc-att-b

Use cases:

- Clustering and databases
- Financial Services and Media distribution
- VPC-to-VPC multicast

Cross-region Transit Gateway peering

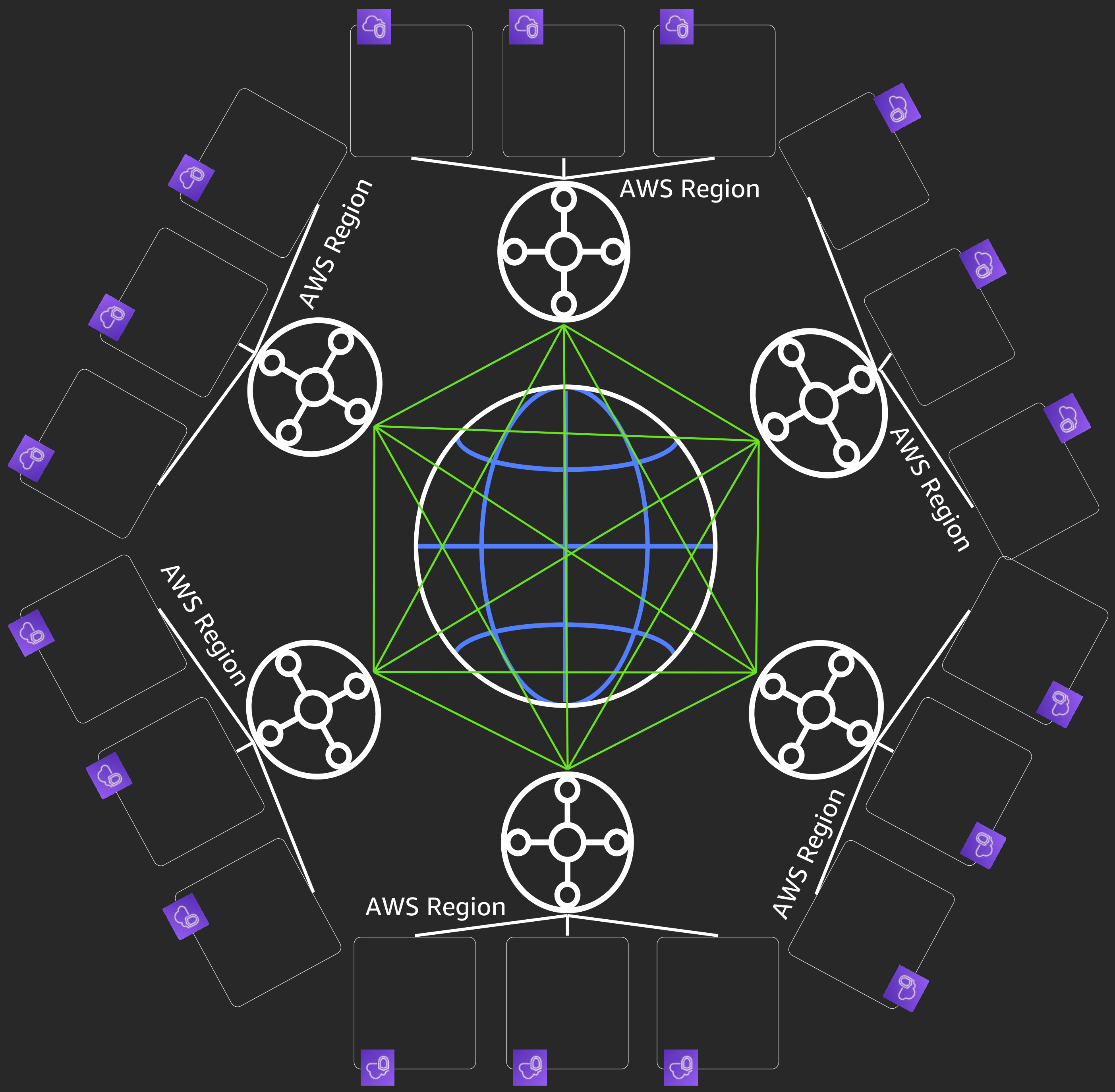
New



- Static peering between regions (US only at launch)
- New attachment type
- Uses encrypted VPC peering across the AWS backbone
- No peering within the same Region

AWS Transit Gateway Cross-Region Peering

New



Full mesh network across multiple regions with static peering

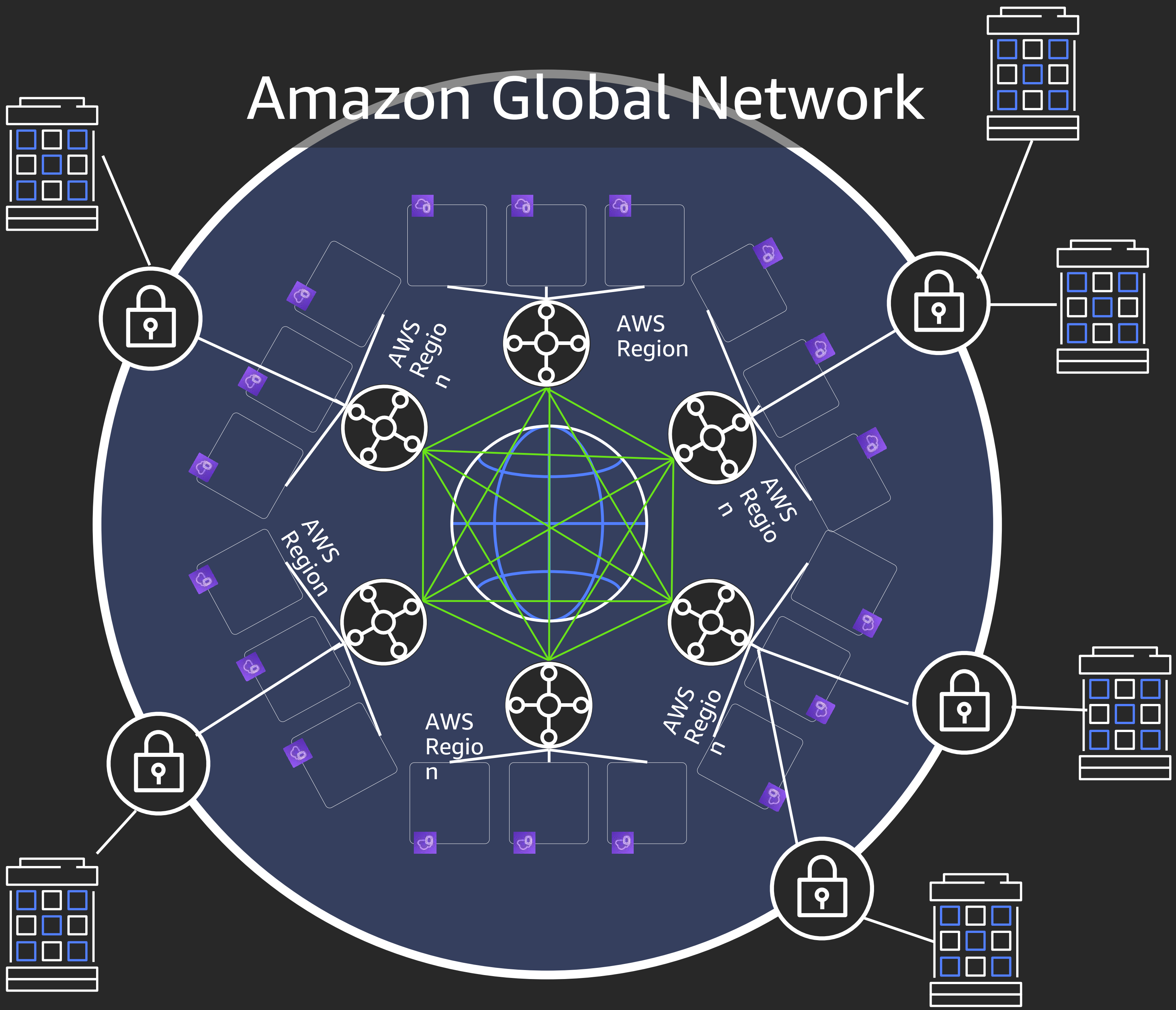
Private and performant connectivity across the AWS Global Network

All traffic across Transit Gateway Cross-Region peering is encrypted

Horizontally scalable

Global network connectivity

New



Leverage the AWS Global Network

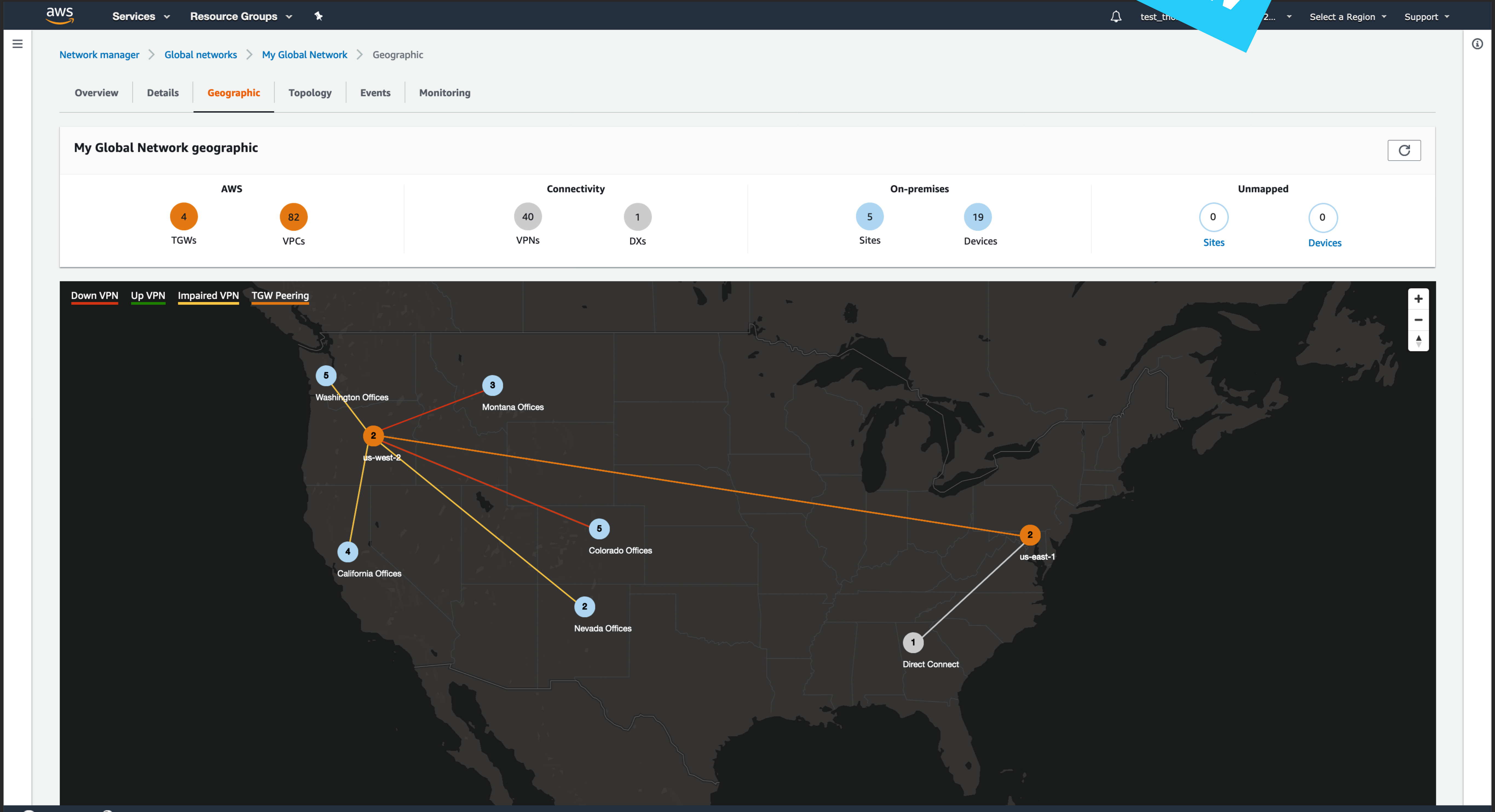
Combine AWS Global Accelerator with VPN

Lower latency, less jitter, consistent connectivity

Ideal for branch connectivity

My global network

New



My Global Network geographic

AWS		Connectivity		On-premises		Unmapped	
4	82	40	1	5	19	0	0
TGWs	VPCs	VPNs	DXs	Sites	Devices	Sites	Devices

Thank you!

Nick Matthews

🐦 @nickpowpow

Matt Lehwess

mattyloo@amazon.com

Please complete the session survey in the mobile app.

AWS re:Invent

aws