AWS re:Invent

NET406

AWS Transit Gateway reference architectures for many VPCs

Nick Matthews

Principal Solutions Architect

Amazon Web Services

@nickpowpow

AWS re:Invent

aws

What to expect

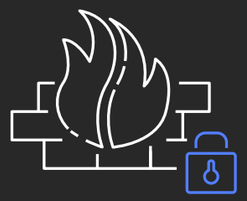
How Transit Gateway works

Let’s build an architecture:

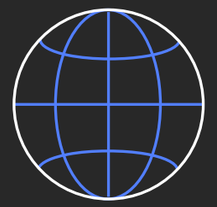
Account Strategy

Segmentation

Connectivity



Network services

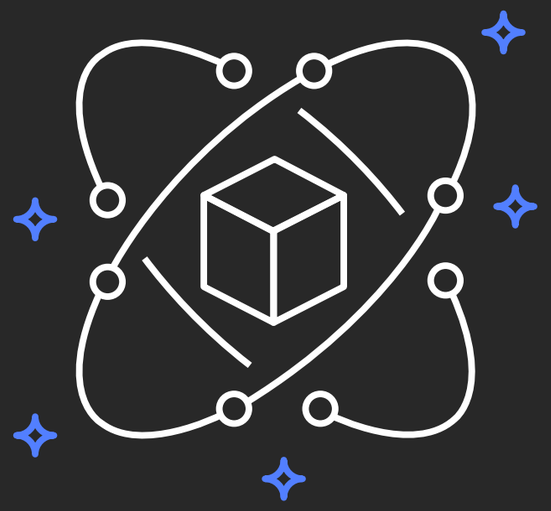


Multi-Region

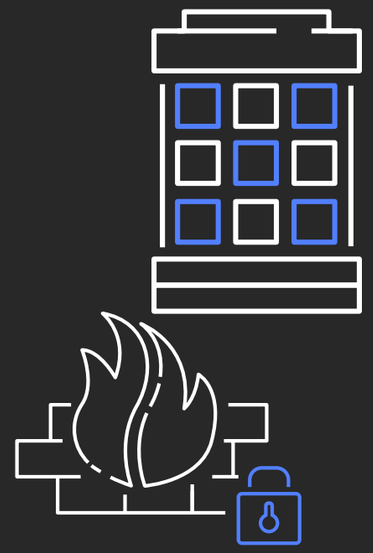
Cost

Challenges with many VPCs

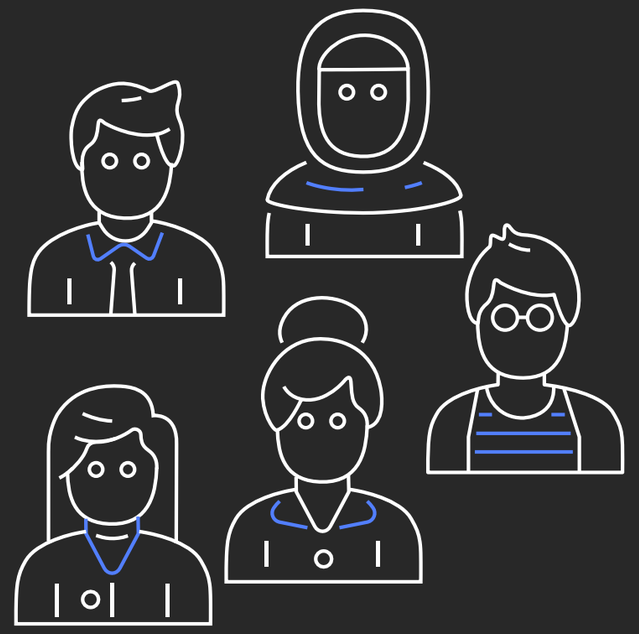
VPC management differences



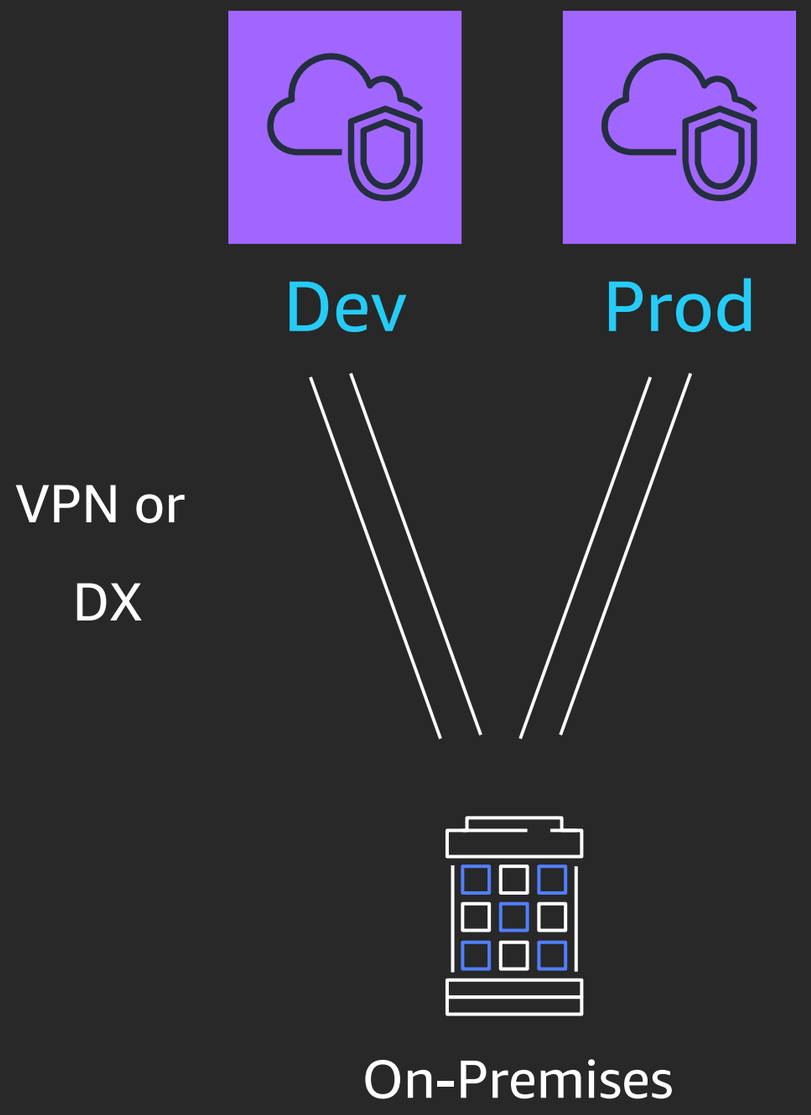
Ease of creation

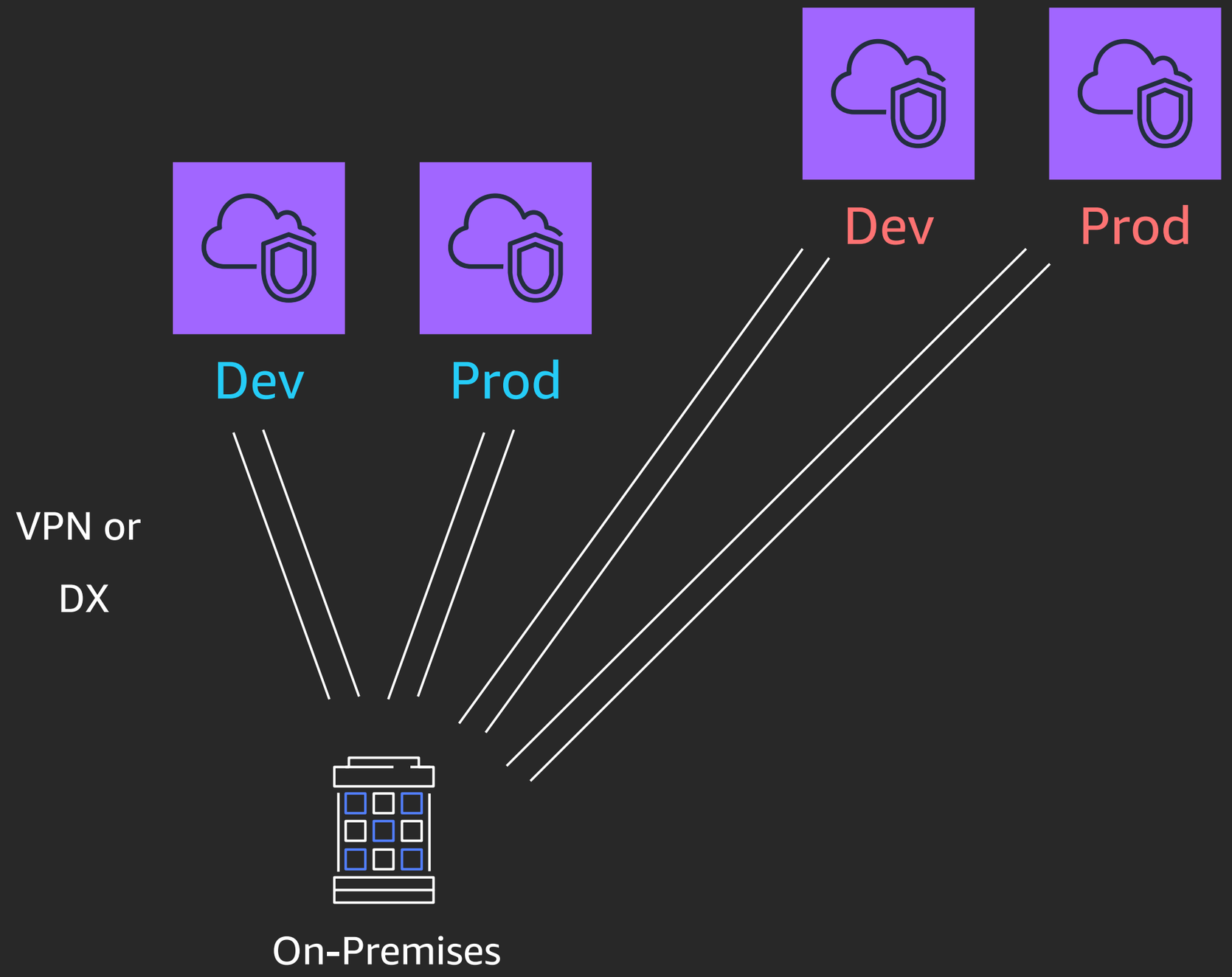


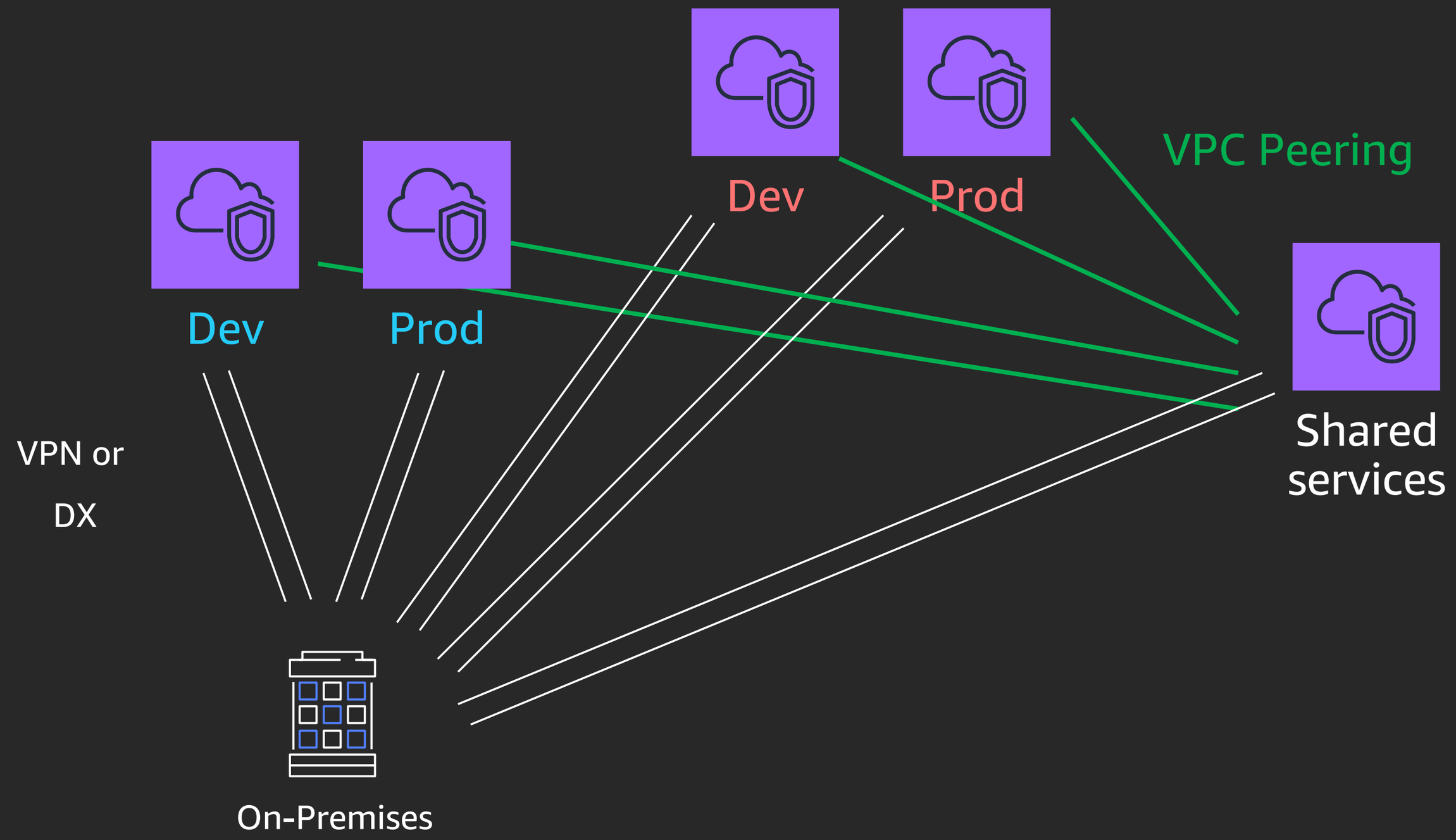
Access models



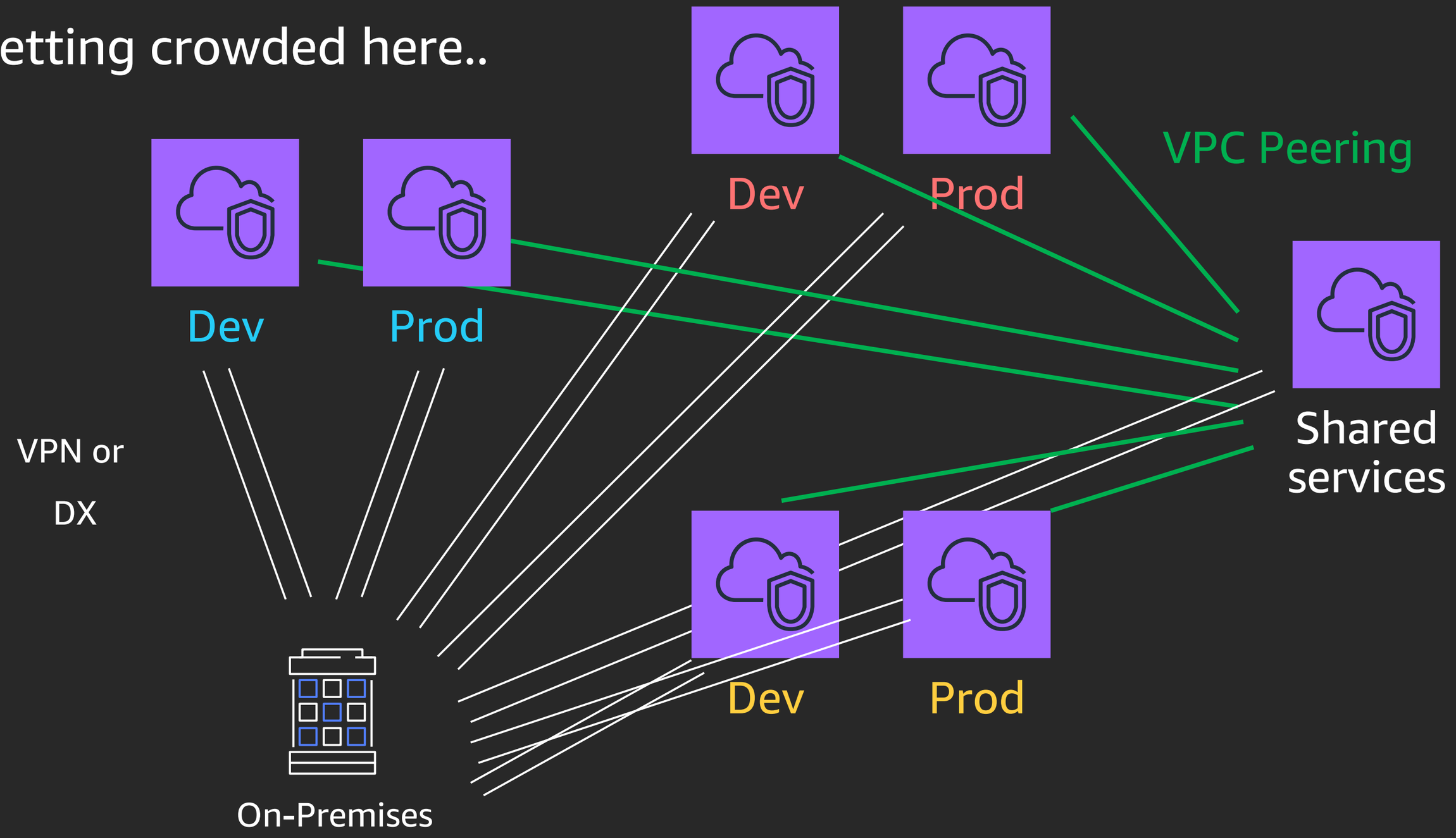
Diverse ownership

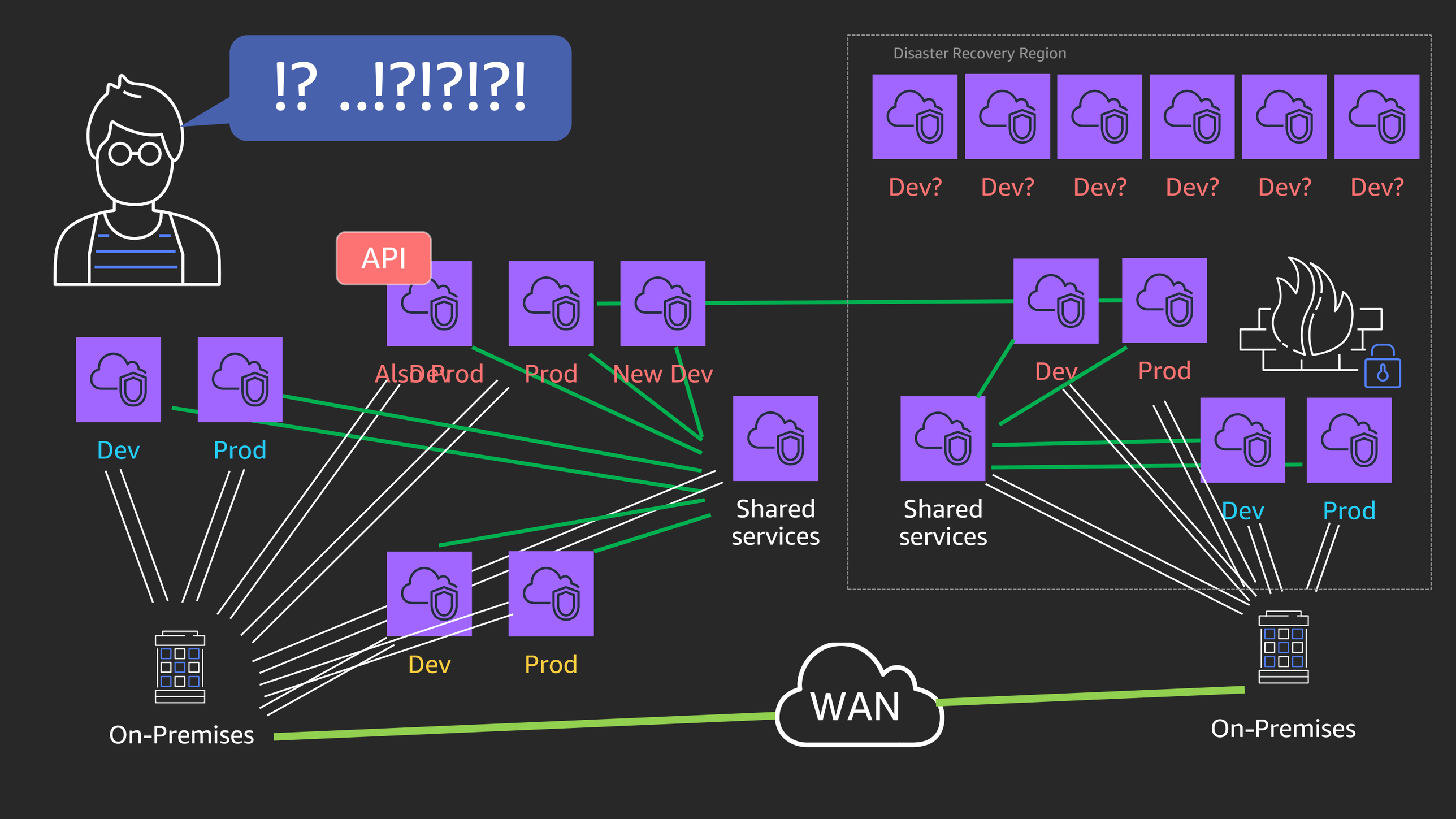

Dev
Prod
VPN or
DX
On-Premises

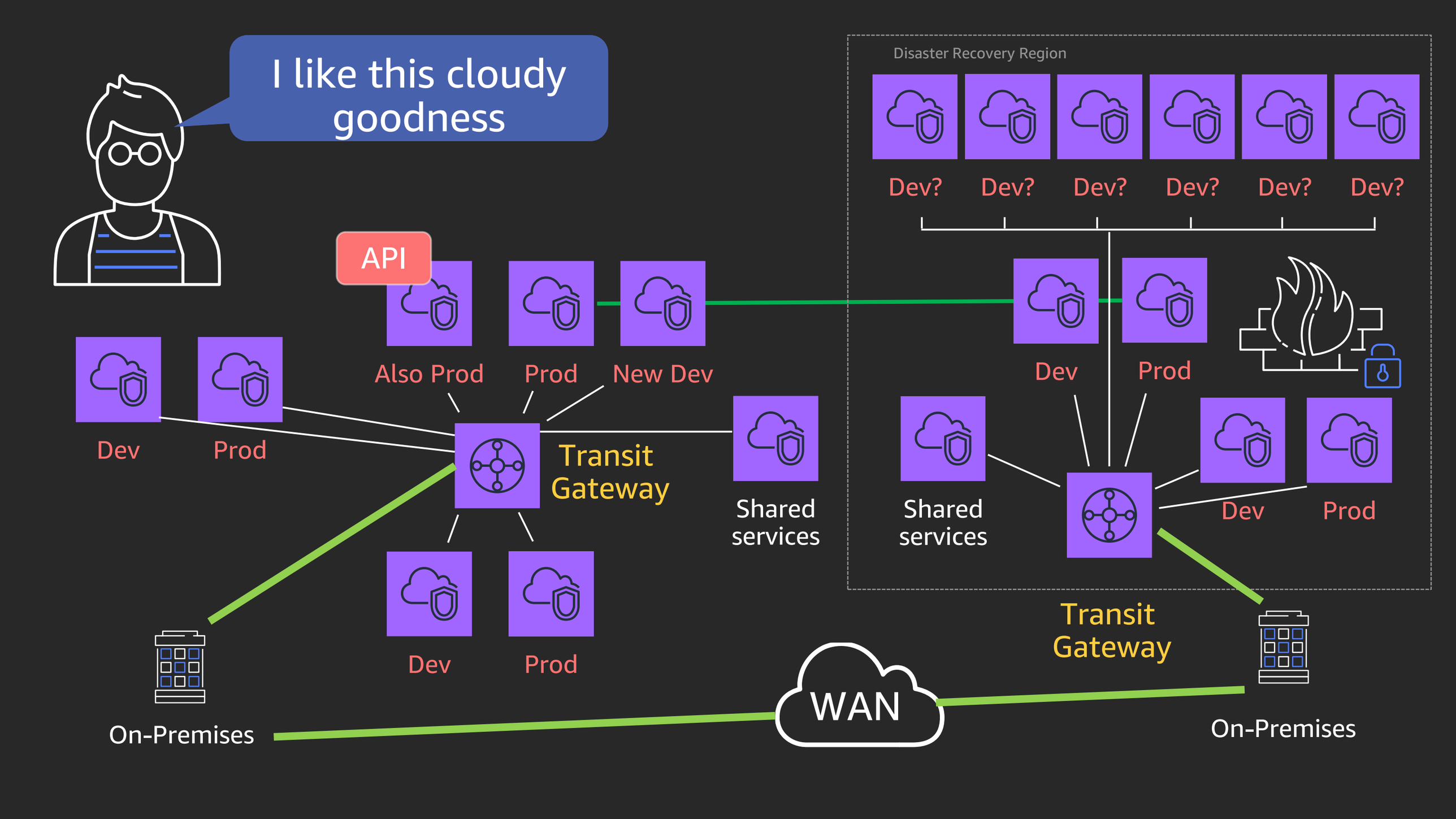

Dev
Prod
Dev
Prod
VPN or
DX
On-Premises


Dev
Prod
Dev
Prod
VPC Peering
Shared services
VPN or
DX
On-Premises

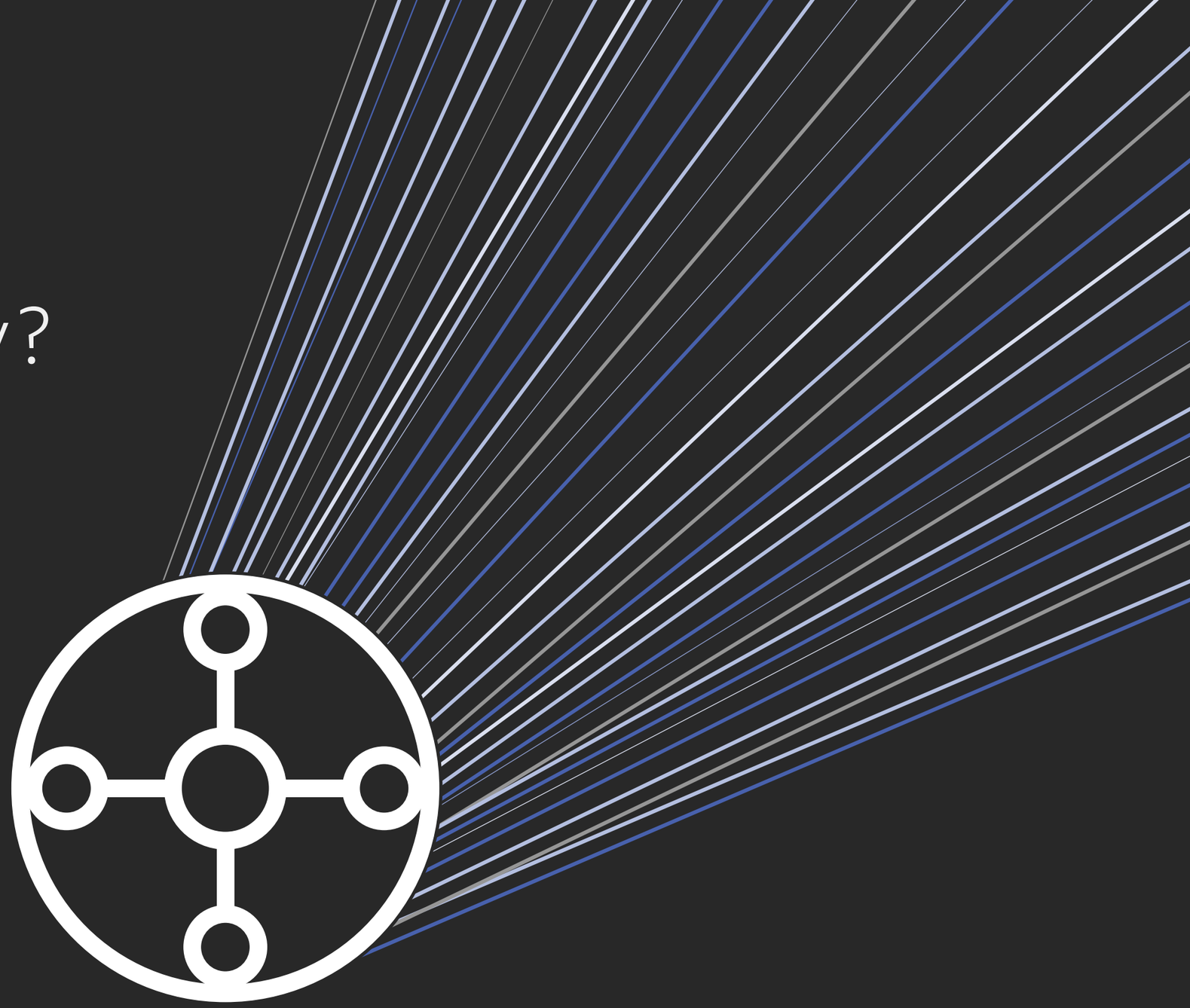
Getting crowded here..




!? ..!?!?!?!
Disaster Recovery Region
Dev?
Dev?
Dev?
Dev?
Dev?
Dev?
API
AlsoDev
Prod
Prod
New Dev
Dev
Prod
Dev
Prod
Shared services
Shared services
Dev
Prod
Dev
Prod
Dev
Prod
On-Premises
WAN
On-Premises


I like this cloudy goodness
API
Also Prod
Prod
New Dev
Dev
Prod
Transit Gateway
Shared services
Dev
Prod
On-Premises
Disaster Recovery Region
Dev?
Dev?
Dev?
Dev?
Dev?
Dev?
Dev
Prod
Shared services
Dev
Prod
Transit Gateway
WAN
On-Premises

What is AWS Transit Gateway?



AWS Transit Gateway

Regional service

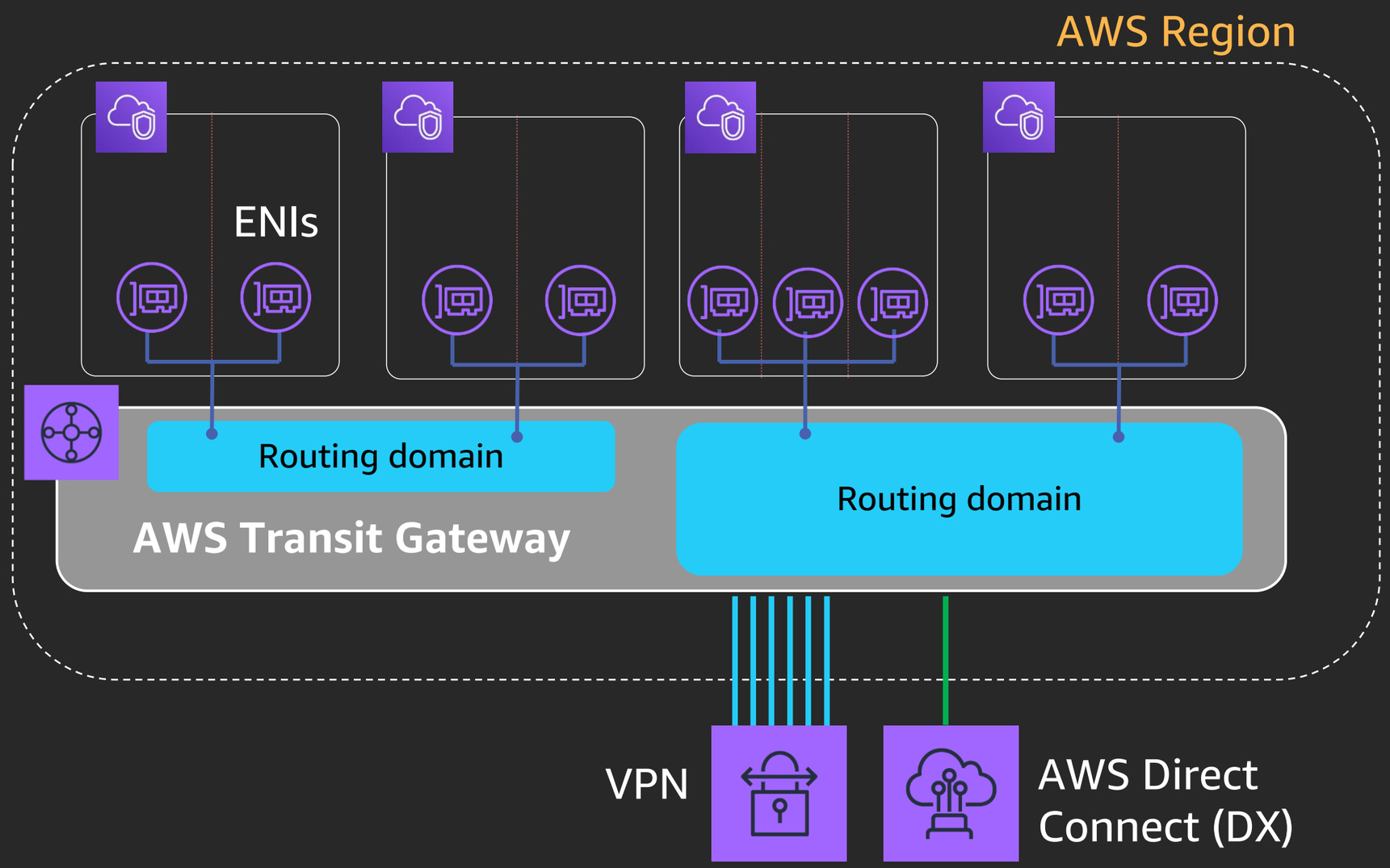
- Centralize VPN and DX

Scalable

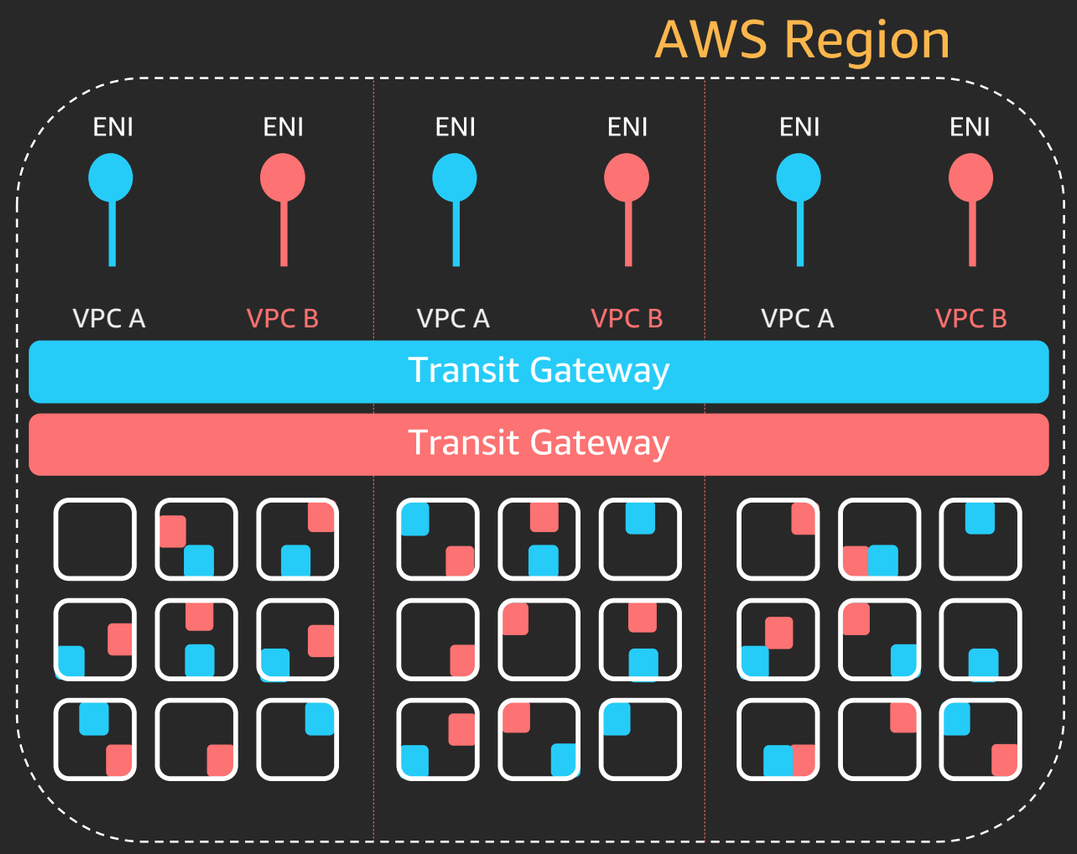
- Thousands of VPCs **across accounts**
- Spread traffic over many VPN connections

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

Transit Gateway example time!

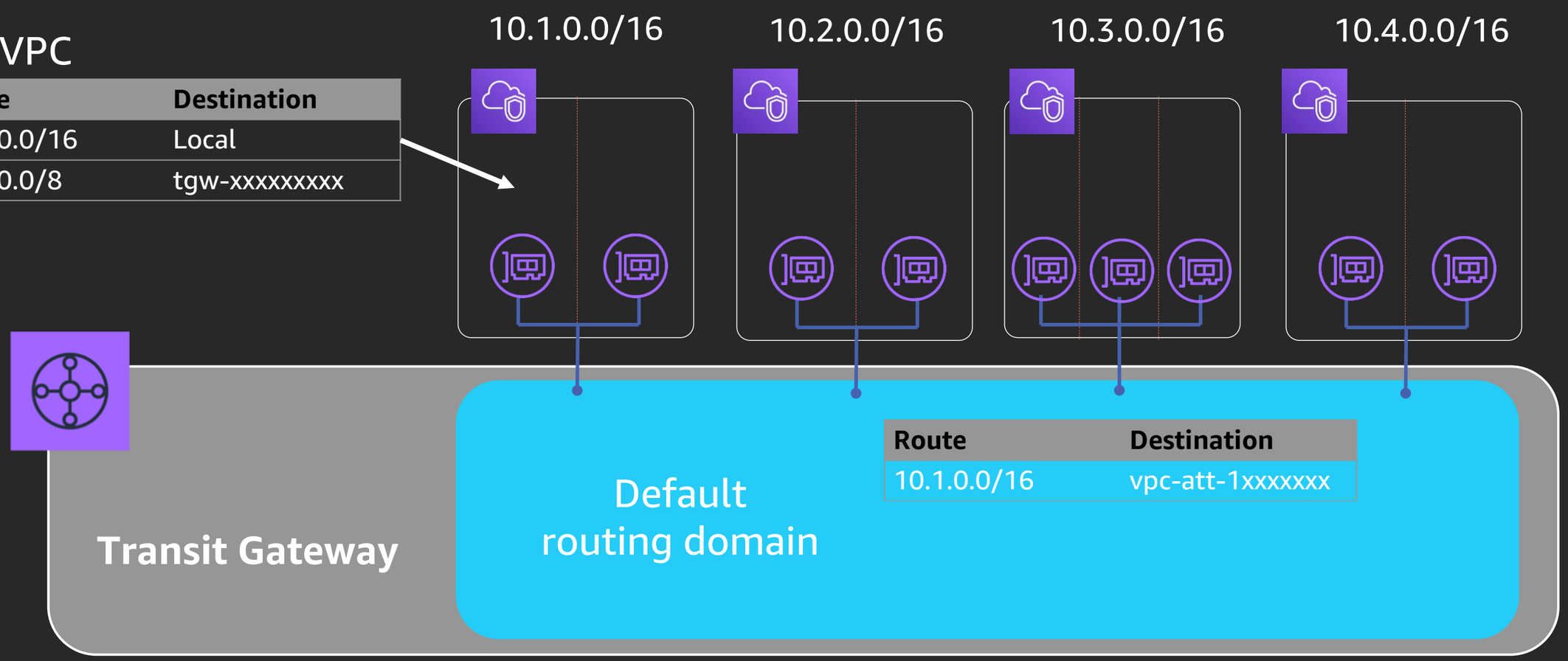
Flat: Every VPC should talk to every VPC!

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

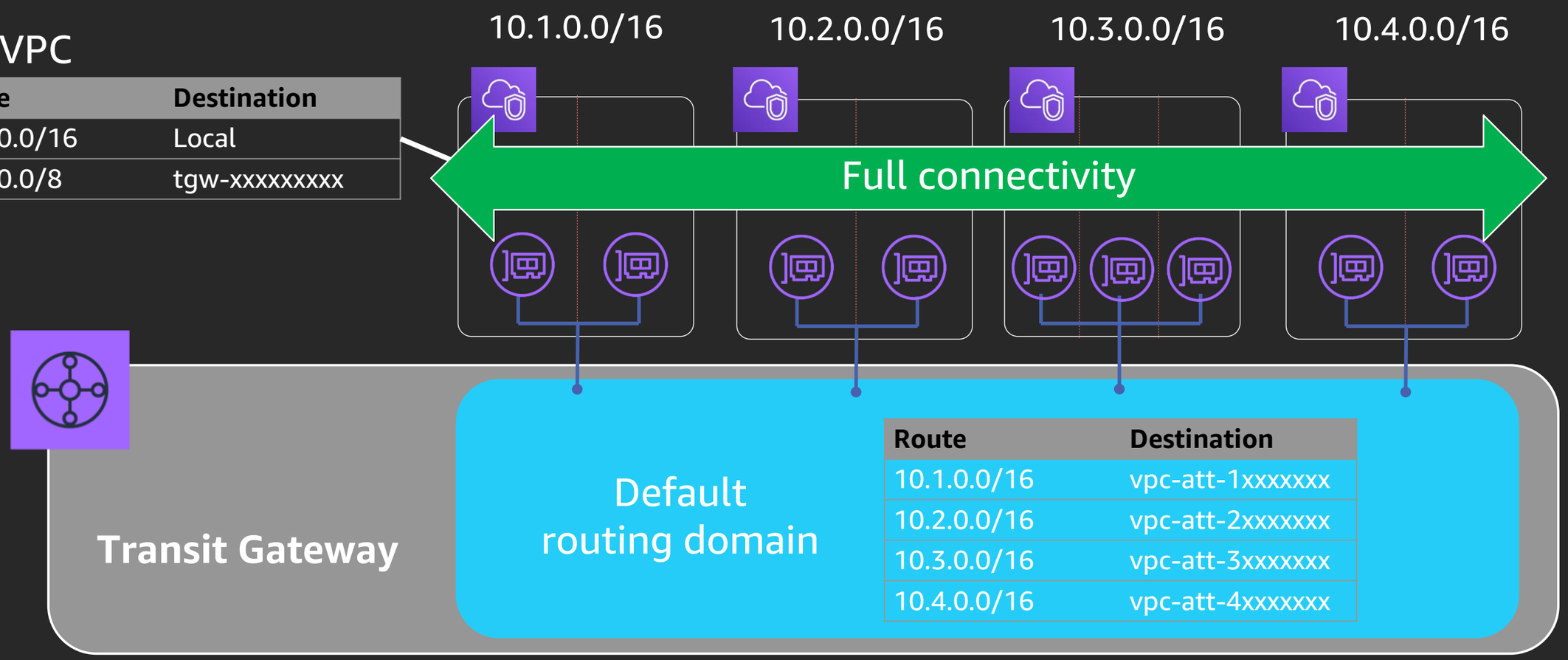


Route	Destination
10.1.0.0/16	vpc-att-1xxxxxxx

Flat: Transit Gateway route domains (route tables)

Per VPC

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



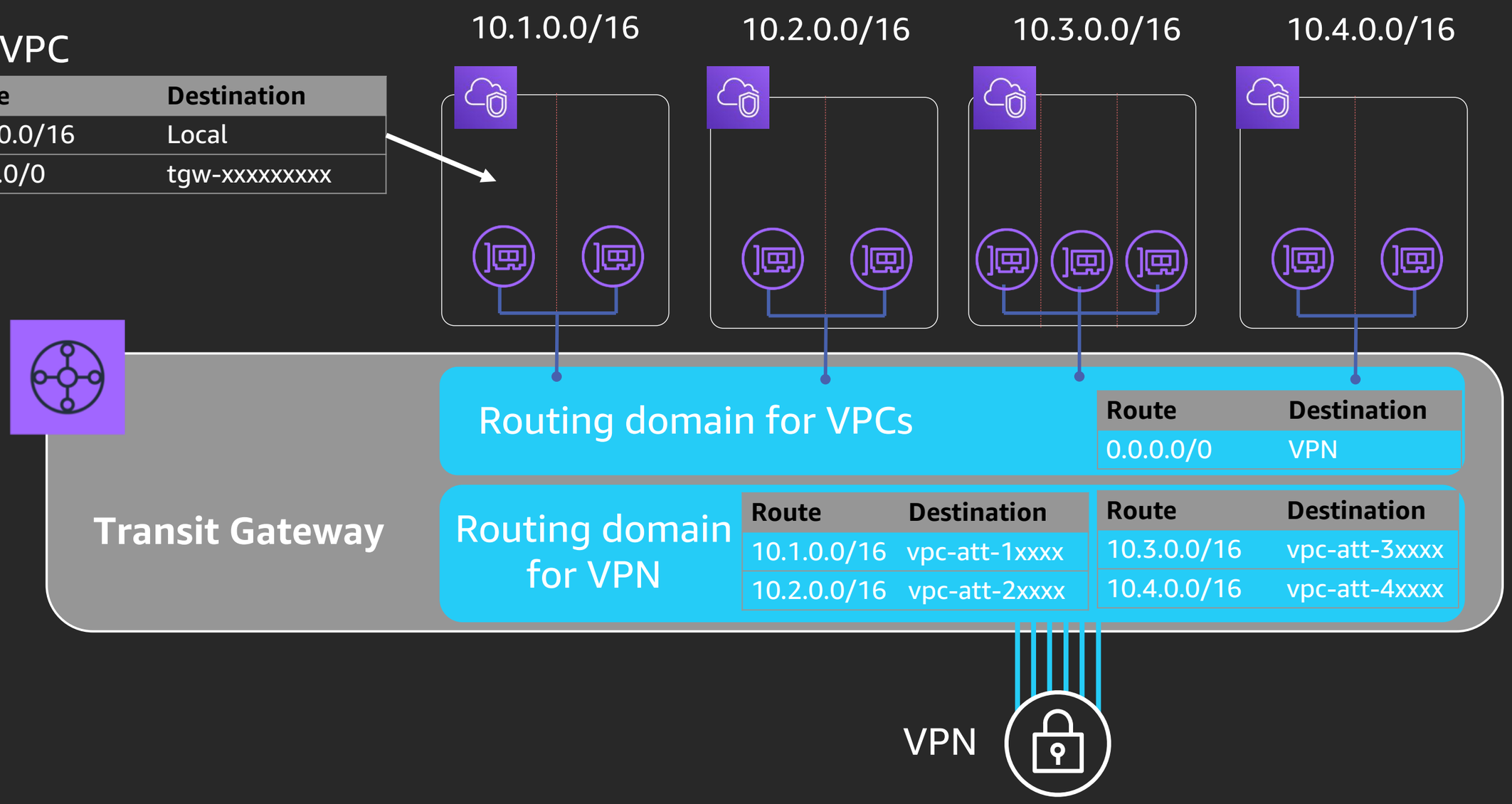
Route	Destination
10.1.0.0/16	vpc-att-1xxxxxxx
10.2.0.0/16	vpc-att-2xxxxxxx
10.3.0.0/16	vpc-att-3xxxxxxx
10.4.0.0/16	vpc-att-4xxxxxxx

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

Isolated: Transit Gateway route domains

Per VPC

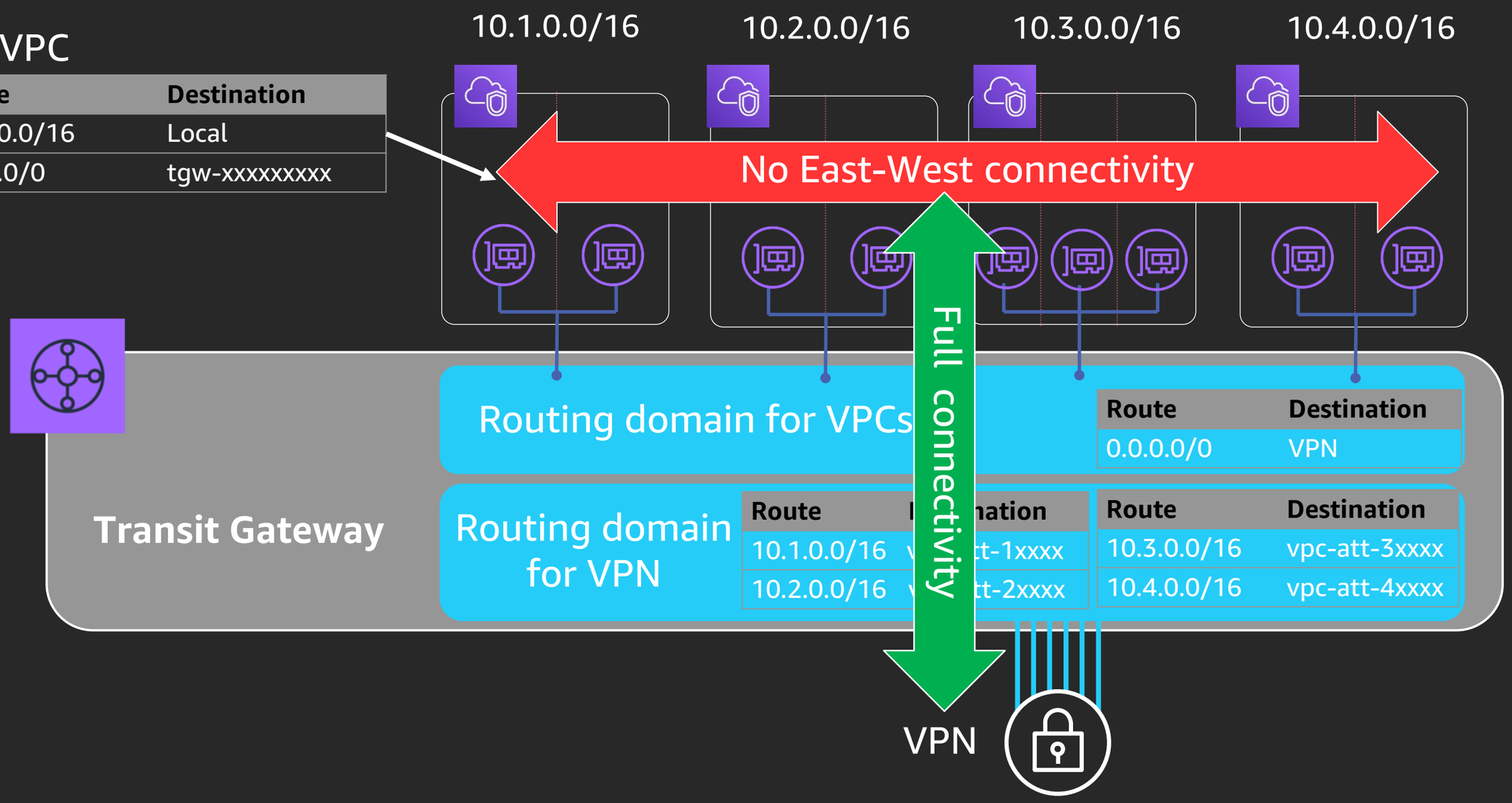
Route	Destination
10.1.0.0/16	Local
0.0.0.0/0	tgw-xxxxxxxxx



Isolated: Transit Gateway route domains

Per VPC

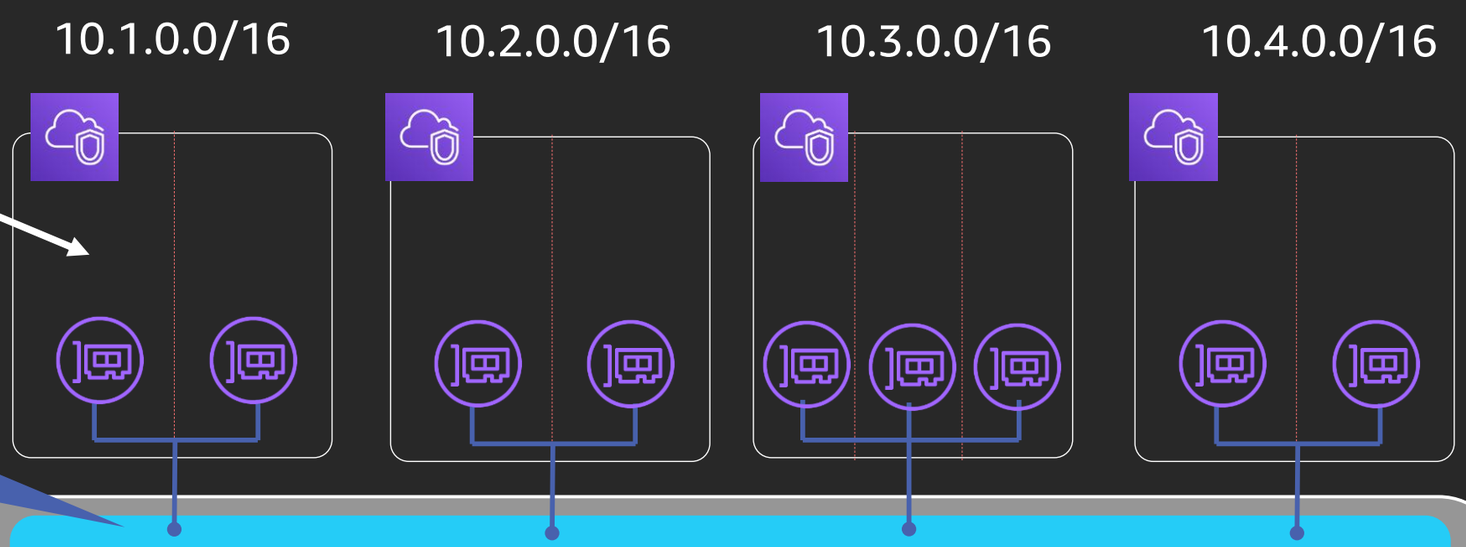
Route	Destination
10.1.0.0/16	Local
0.0.0.0/0	tgw-xxxxxxxxx



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**

You need mutual routes for connectivity

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

Routing domain for VPCs

Route	Destination
0.0.0.0/0	VPN

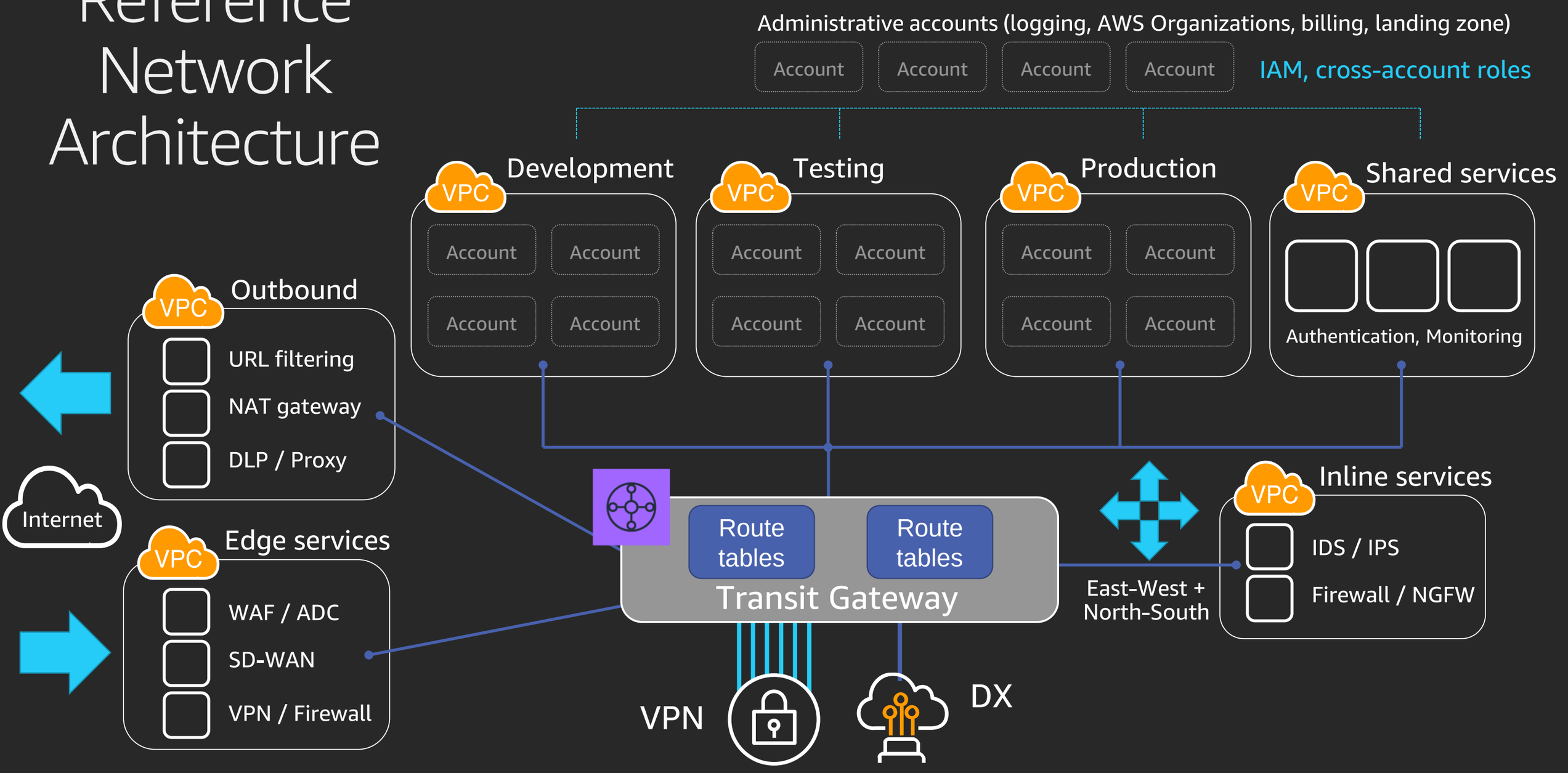
Routing domain for VPN

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

VPN

Reference Network Architecture



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
DX and VPN standards
Subnet and routing standards

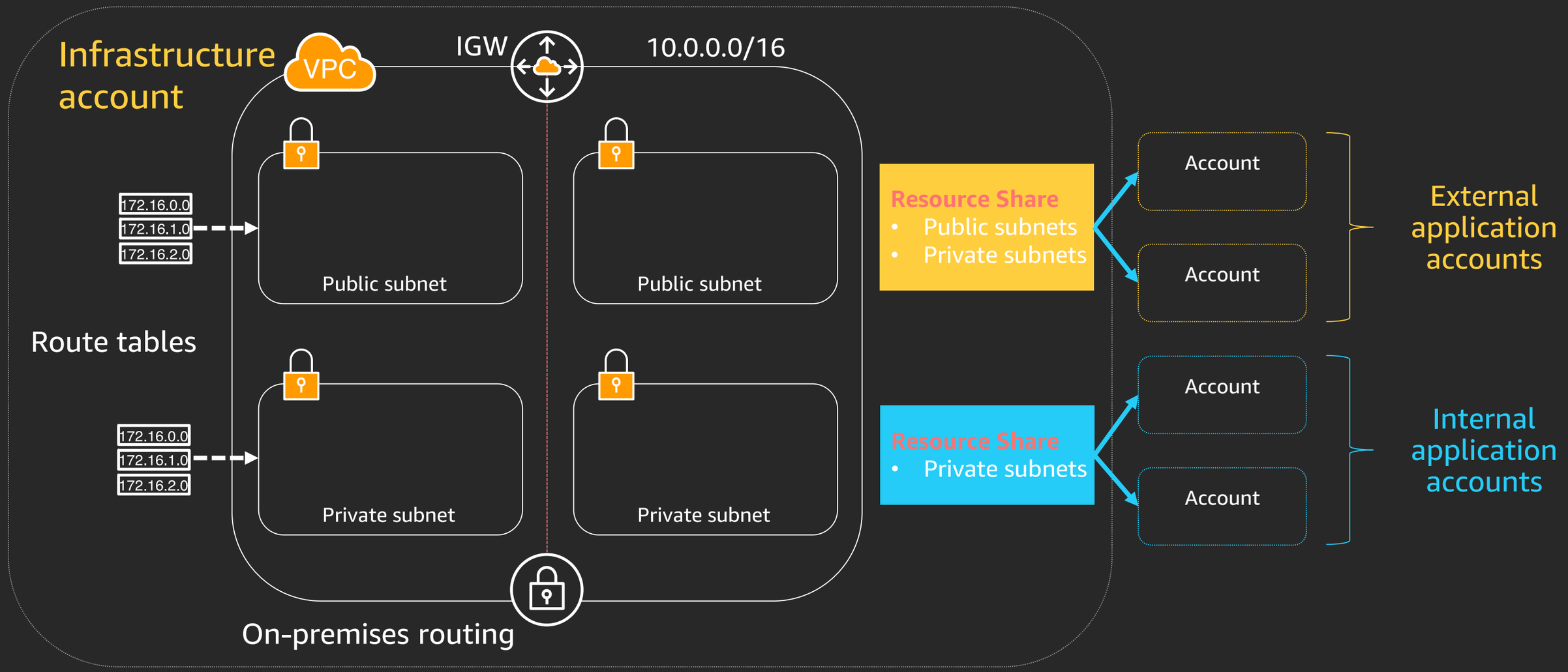
Infrastructure and Networking

Why not both?

Provide granular account control
with centralized infrastructure

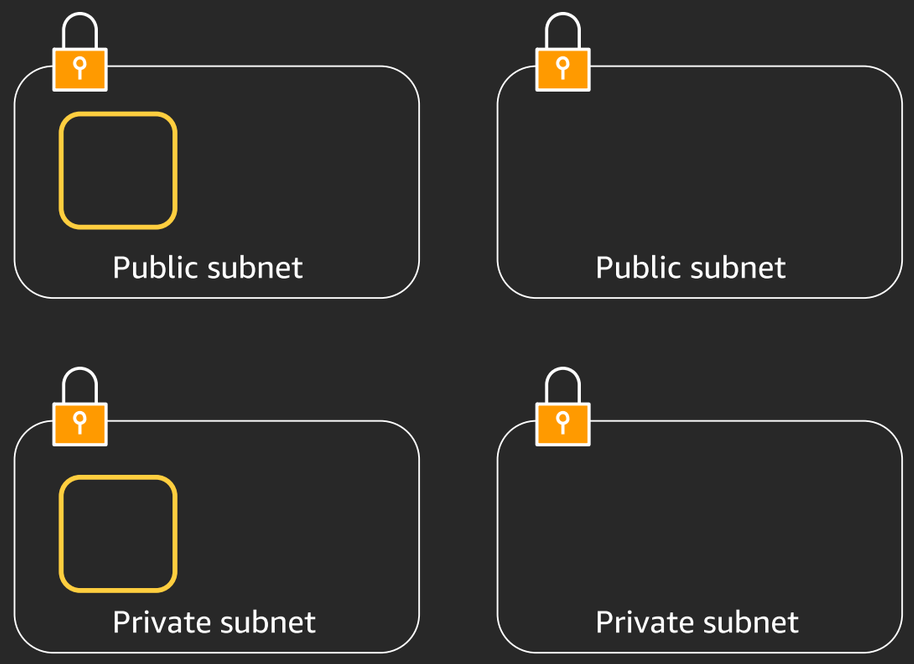
VPC Sharing and Resource Access Manager

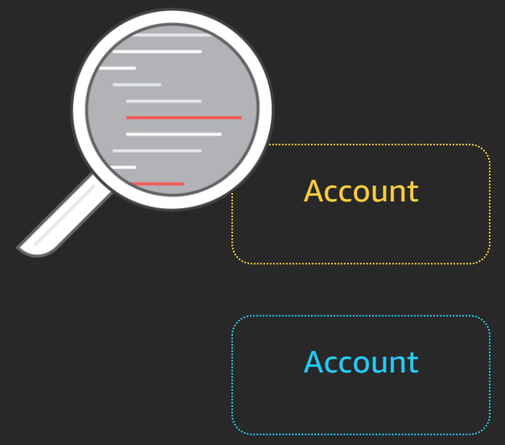
Share subnets between accounts in an AWS Organization



VPC Sharing and Resource Access Manager

Account owners only see subnets and their resources





External accounts

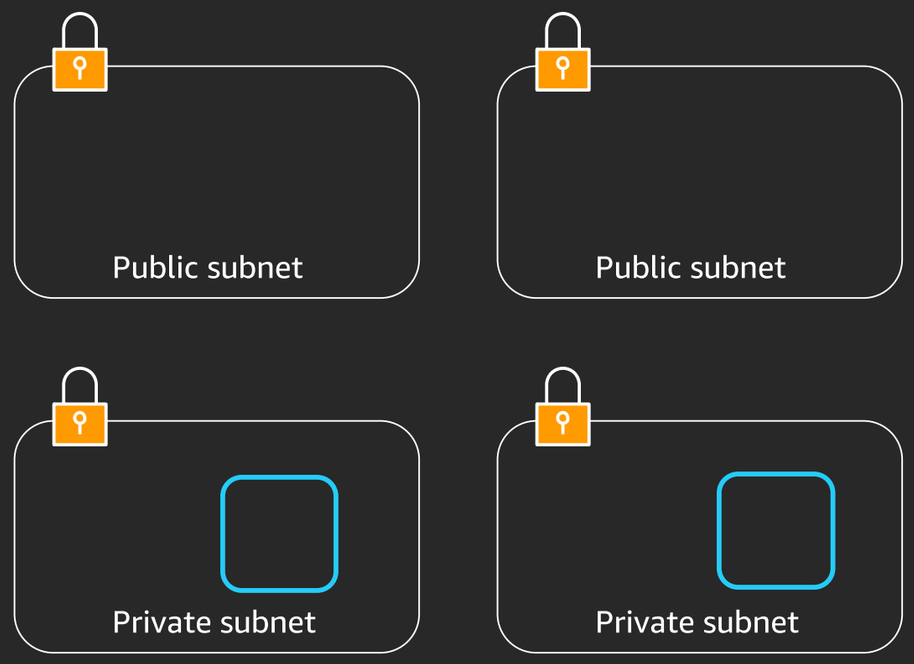
Internal accounts

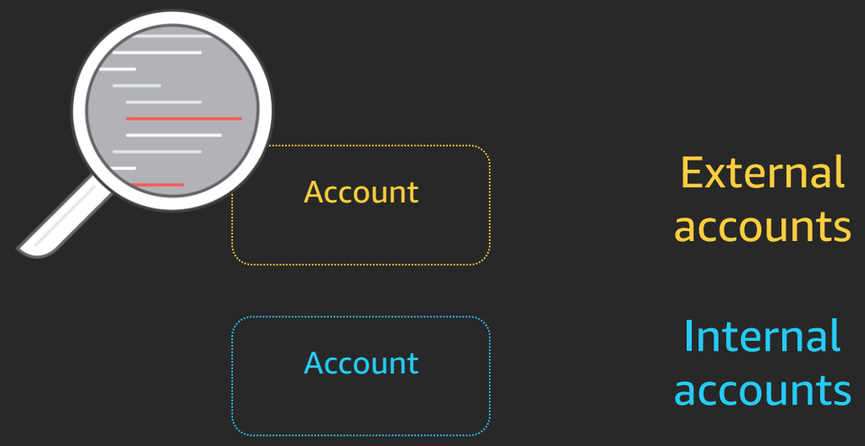
Accounts see their resources and have control:

- Security groups
- Data
- Instance details
- Account configuration

VPC Sharing and Resource Access Manager

Account owners only see subnets and their resources





Accounts see their resources and have control:

- Security groups
- Data
- Instance details
- Account configuration

Why not use VPC sharing?

Advantages of separate VPCs

- Separate VPCs reduce blast radius and VPC limits
- Compliance for applications in individual VPCs
- De-merges and spinoffs are easy

If distributed teams manage their own environments

Caveats

New

- **New:** Participants can now use NLB
- Participants can’t create Amazon FSx or AWS CloudHSM Classic endpoints
- VPC owner cannot eject running participant’s resources

Session on shared VPCs

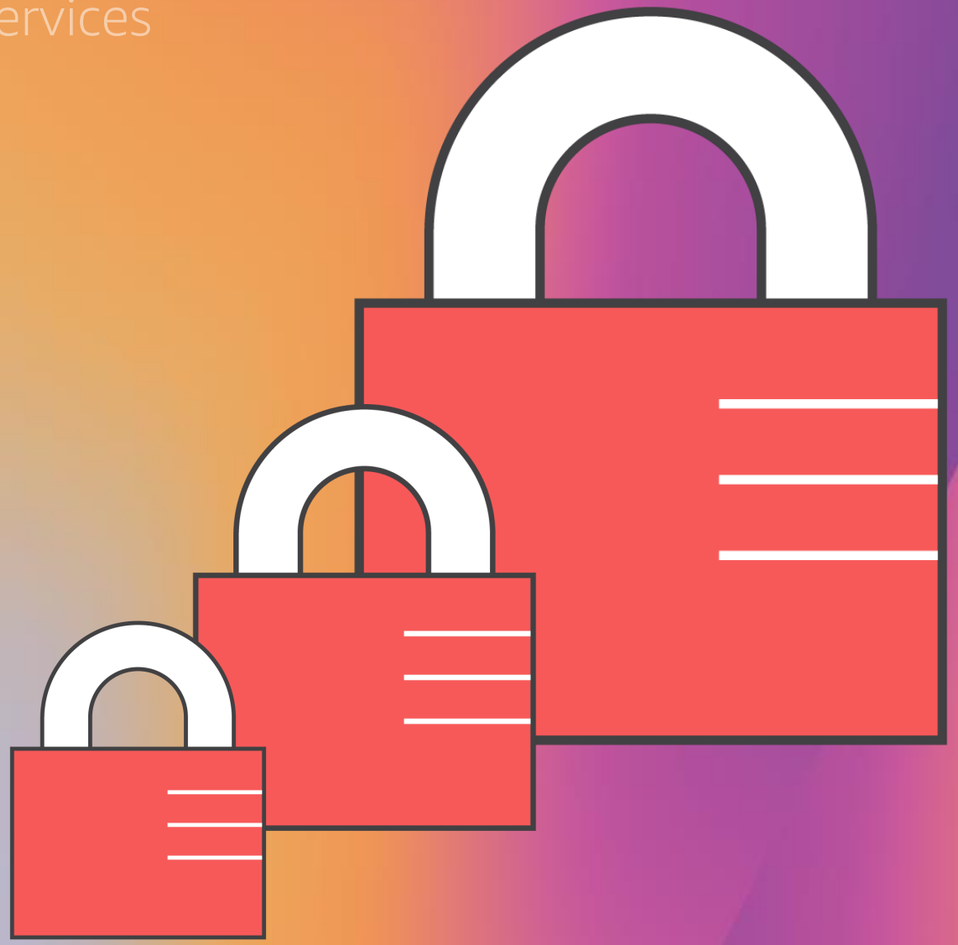
NET322-R

Shared VPC: Simplify your AWS Cloud scale network with VPC sharing

Wednesday, Dec 4, 3:15 PM - 4:15 PM

Thursday, Dec 5, 1:00 PM - 2:00 PM

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?

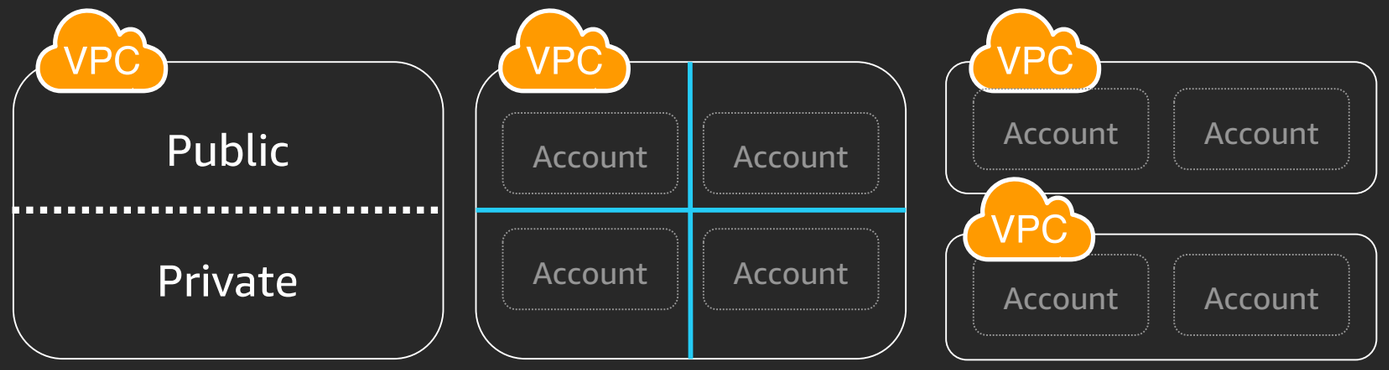
Compliance and governance requirements?

- 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.

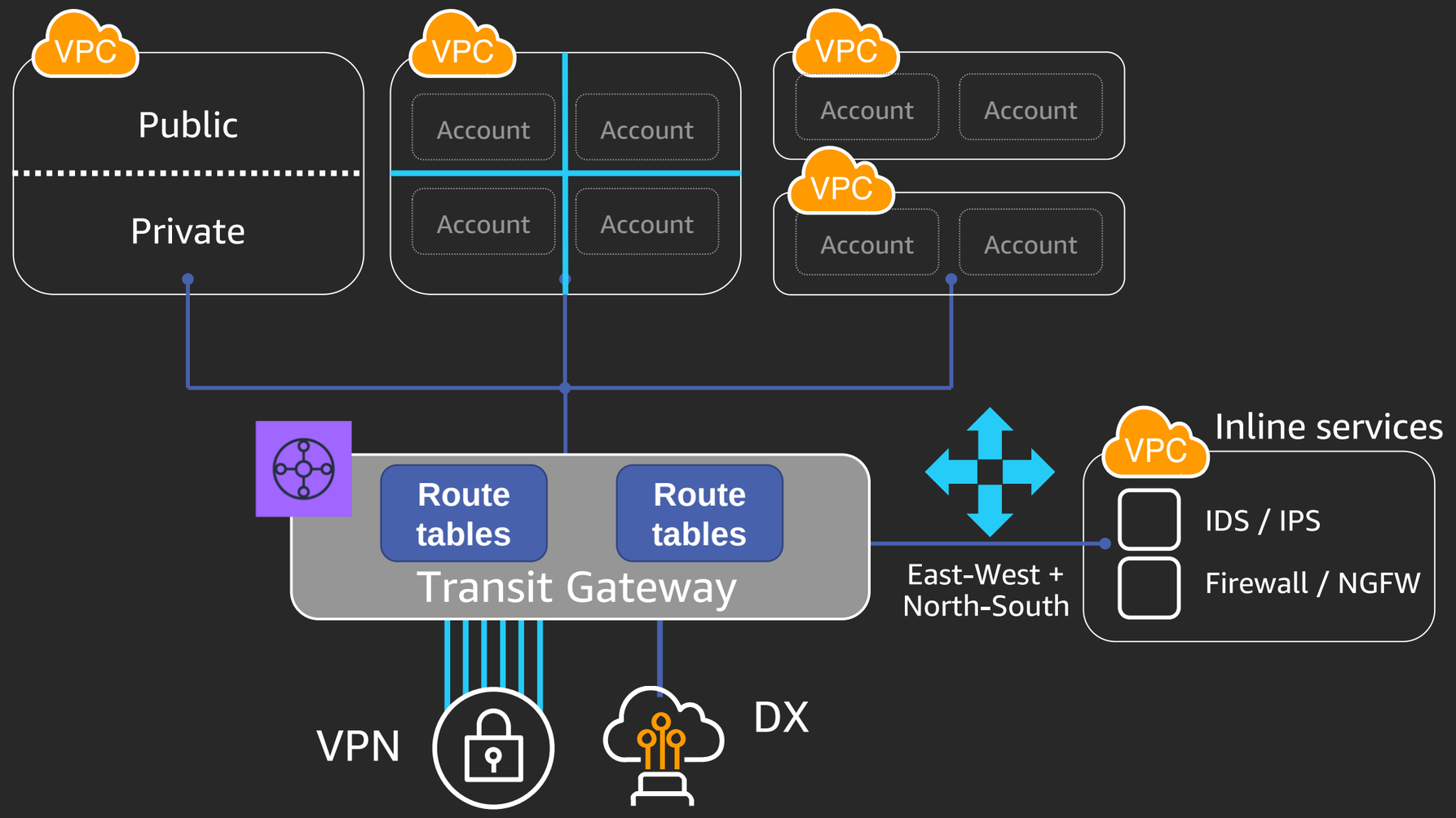
Segmentation options: Layers

Inside the account
- IAM users and roles
- Security groups

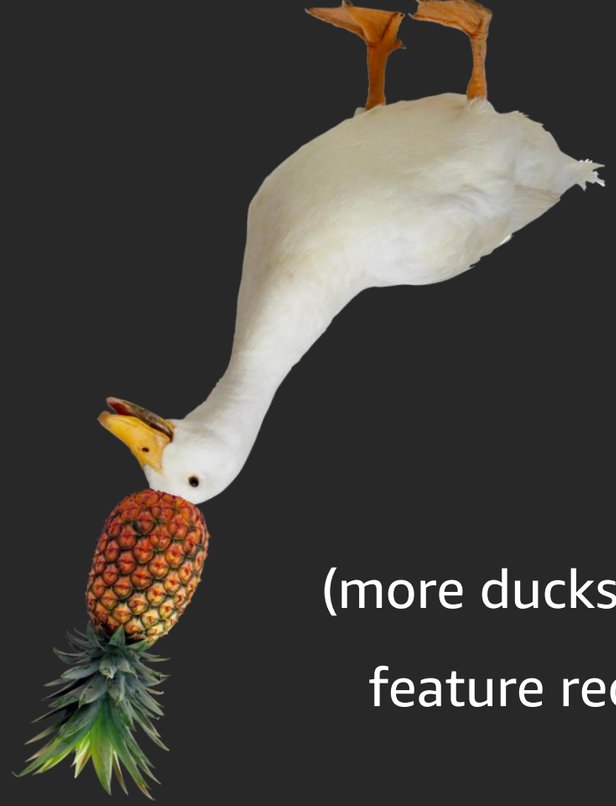
At the VPC
- Route tables
- Network ACLs
- Separate VPCs

Transit Gateway
- Route tables

Security services
- Firewalls
- Proxies
- Intrusion Detection / Prevention



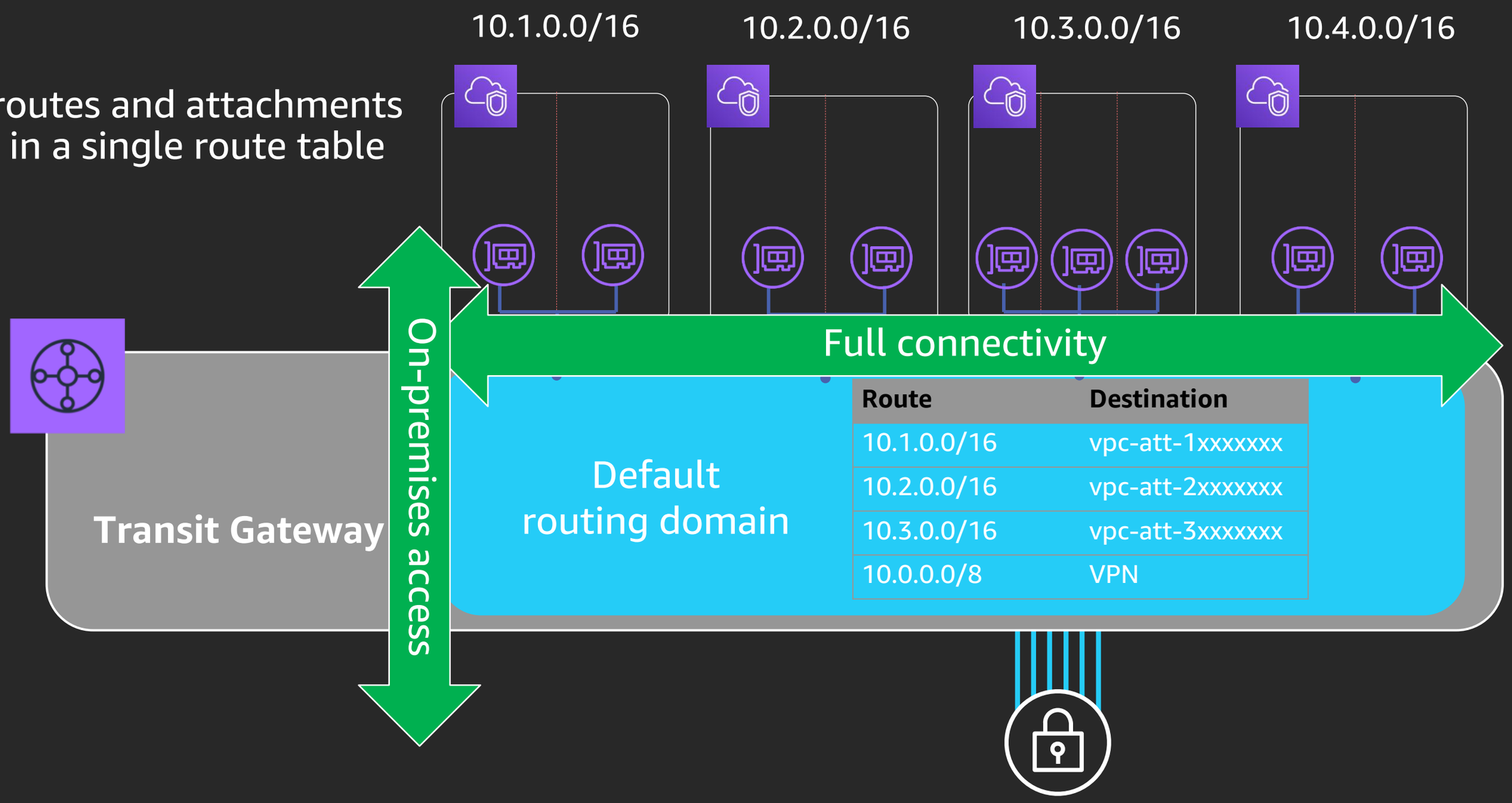
Let’s do some examples



(more ducks were a feature request)

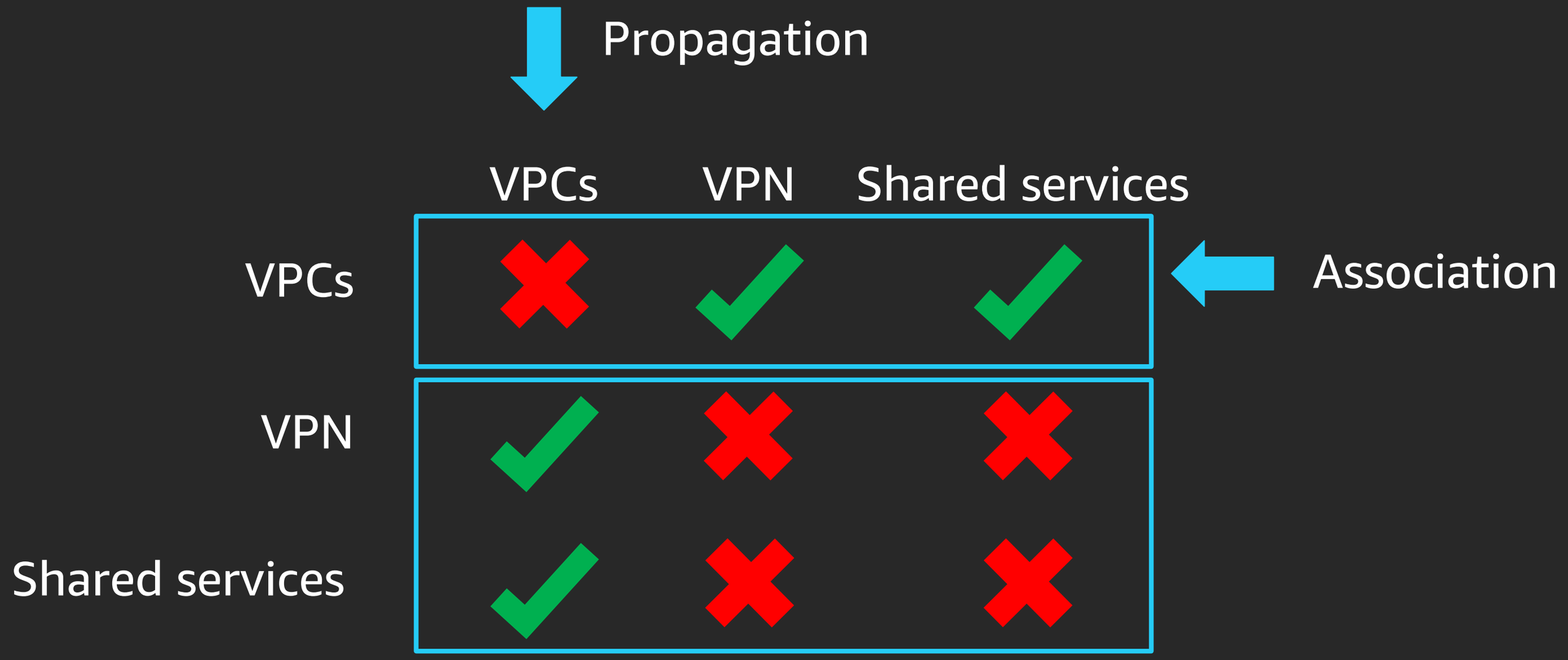
Flat: Transit Gateway route domains

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

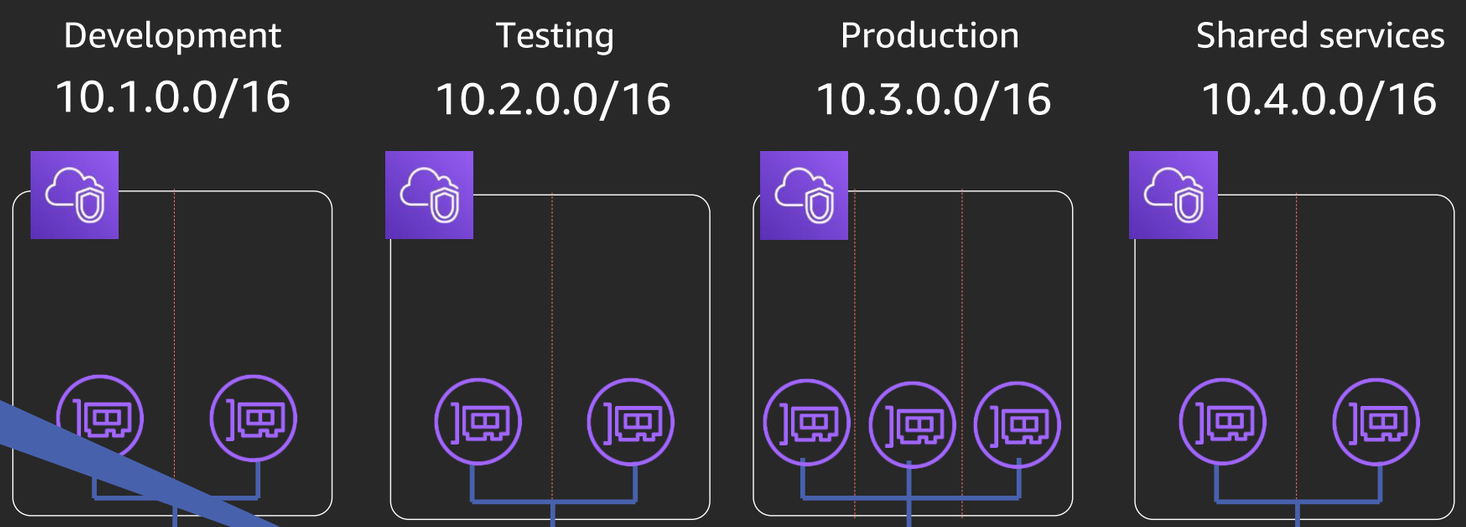
Building a routing policy: Isolated VPCs and shared services



Route table setup

VPN and Shared services propagation

VPC propagation



way

VPC

Route	Destination
10.0.0.0/8	VPN
10.4.0.0/16	vpc-att-4xxxx

VPN + shared services

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

VPN

	VPCs	VPN	Shared services
VPCs	✗	✓	✓
VPN	✓	✗	✗
Shared services	✓	✗	✗

Cheat Sheet

Route table setup

VPN and Shared services propagation

VPC propagation

I want to manage Shared services from on-premises

Development
10.1.0.0/16

vices
/16

way

VPC

Route	Destination
10.0.0.0/8	VPN
10.4.0.0/16	vpc-att-4xxxx

VPN + shared services

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

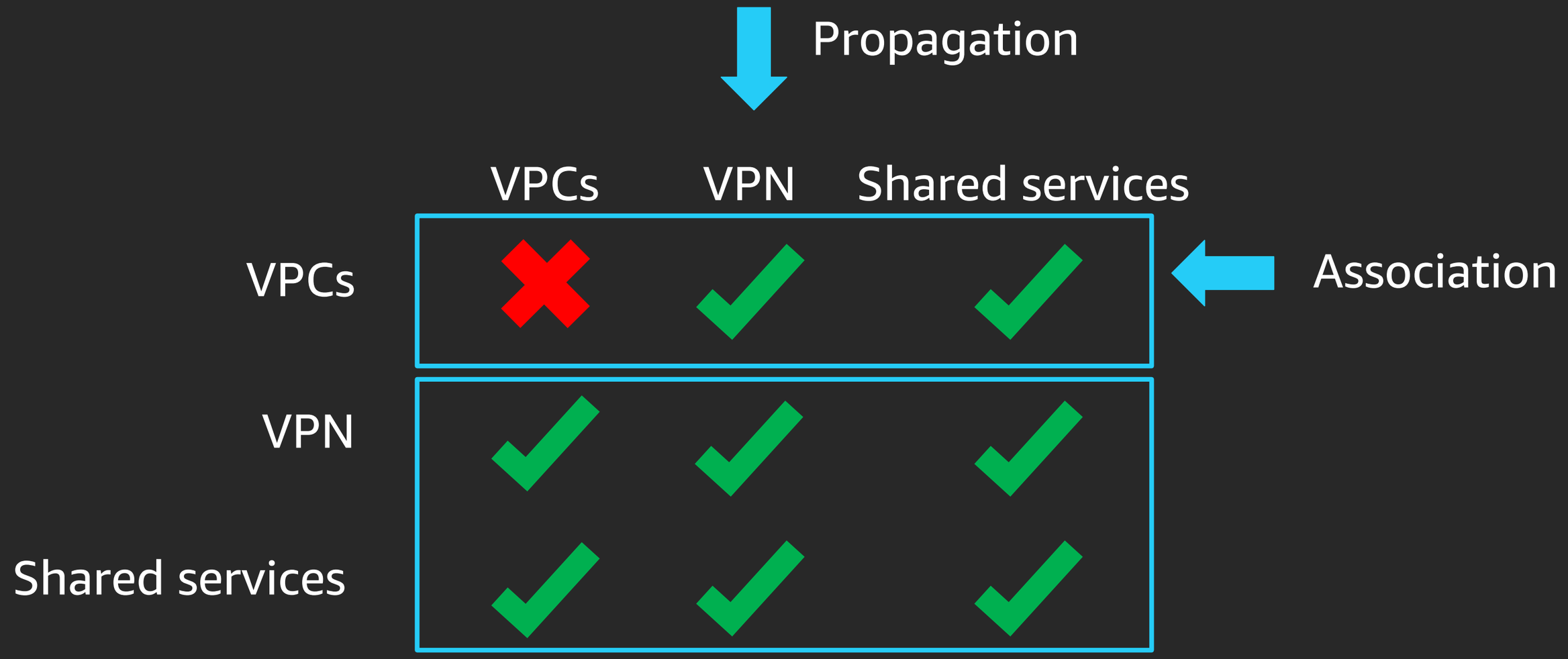
Route	Destination
10.3.0.0/16	vpc-att-3xxxx

VPN

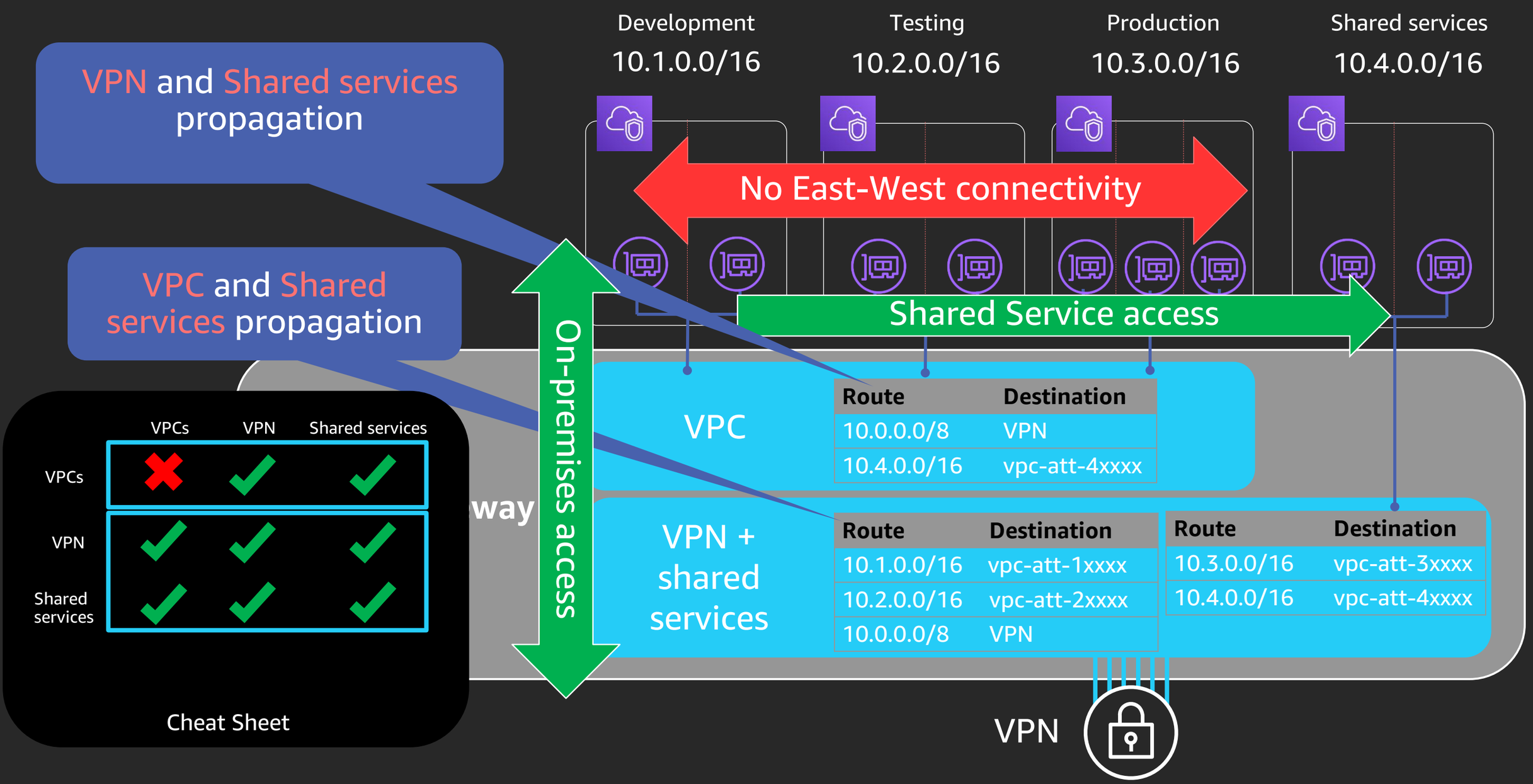
	VPCs	VPN	Shared services
VPCs	✗	✓	✓
VPN	✓	✗	✗
Shared services	✓	✗	✗

Cheat Sheet

Routing policy: Flat shared services with on-premises



Route table setup



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

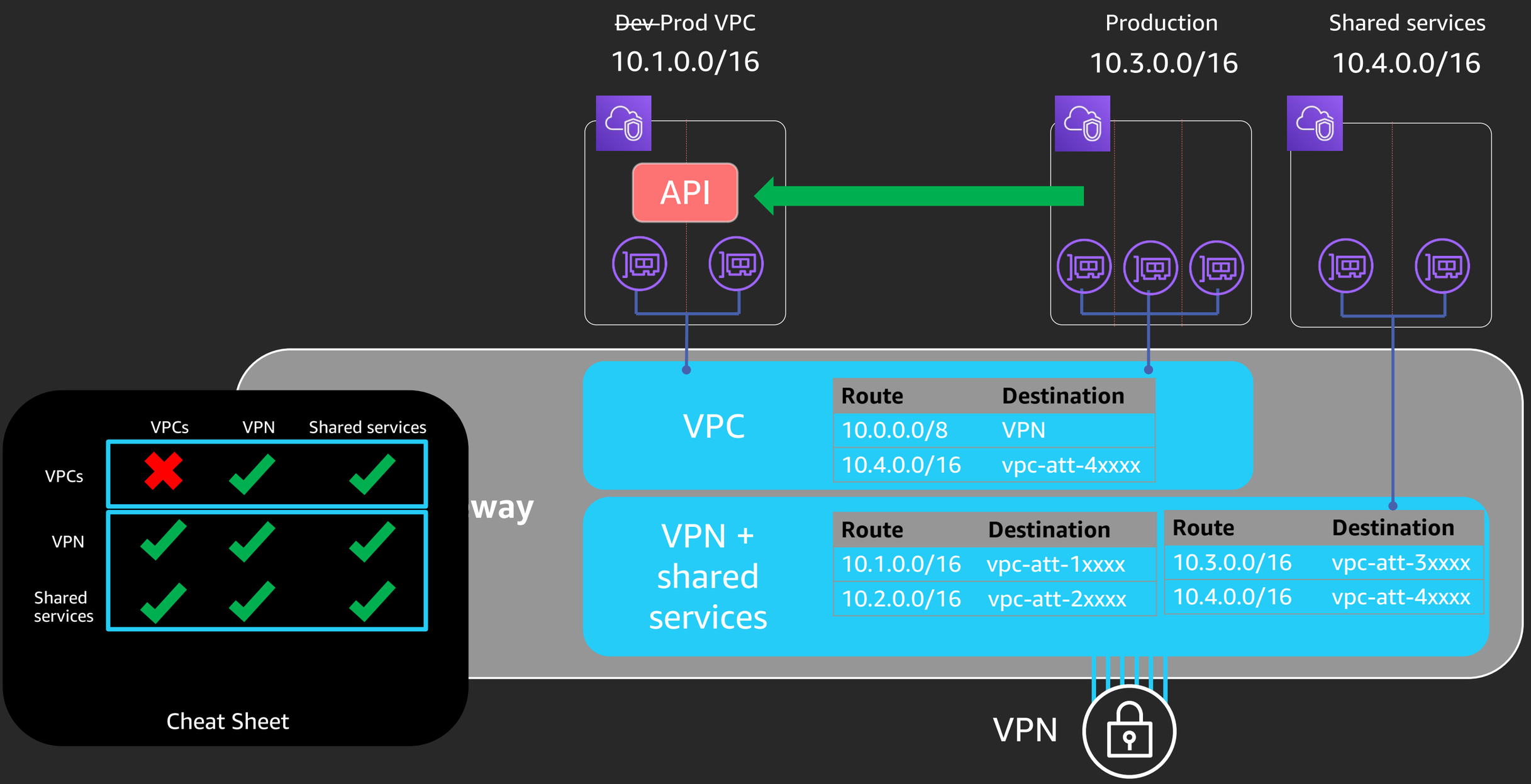
- Enforce controls between tenants (security groups, NACLs) and the internet
- Peered VPCs are likely to benefit from Shared VPCs

Separate VPCs

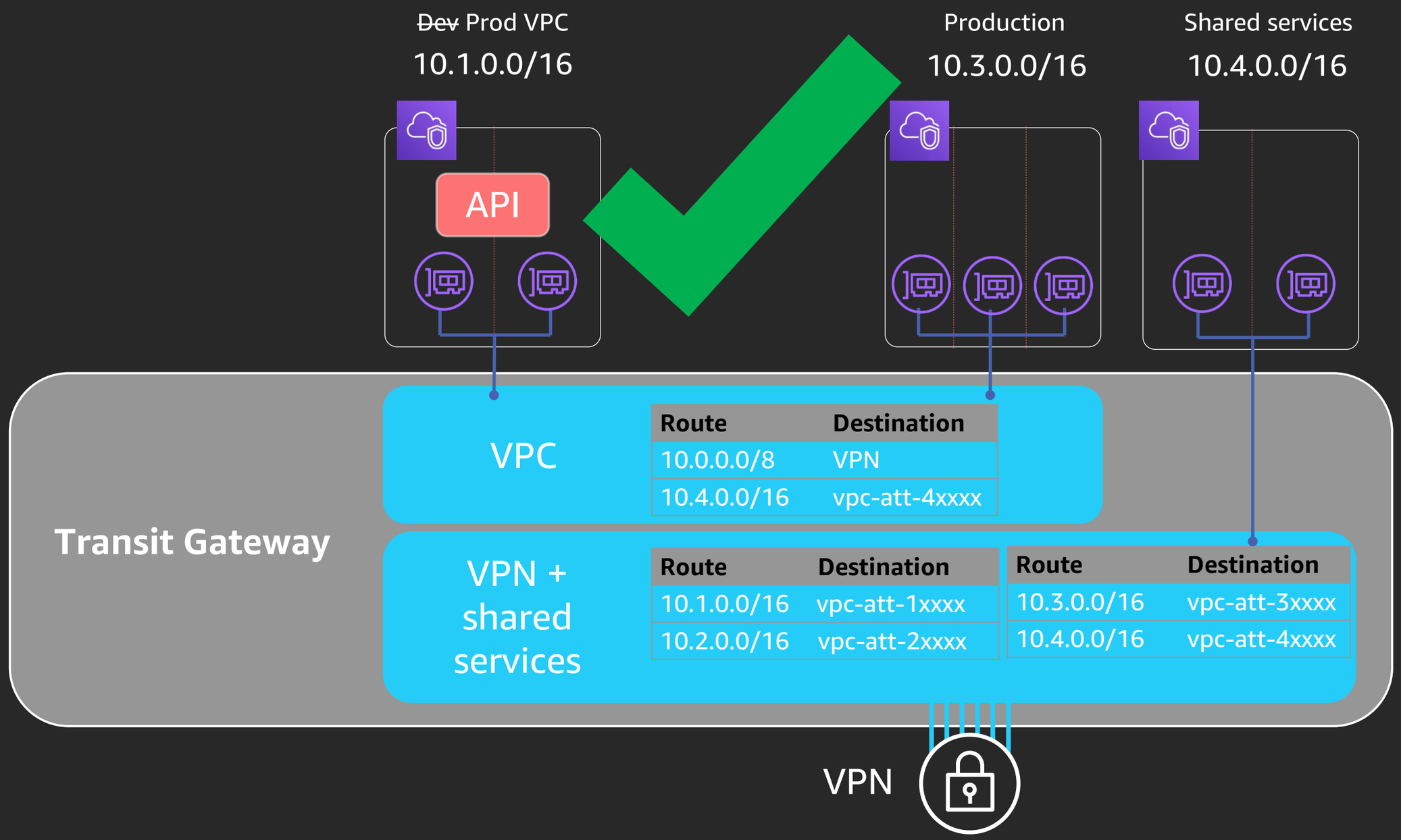
- Often the best security decision is the simplest
- Strong network segmentation and resource isolation
- Transit Gateway removes the scaling issues with many VPCs (peering, VPN, routes)

Use Transit Gateway route tables to define policy for groups of VPCs

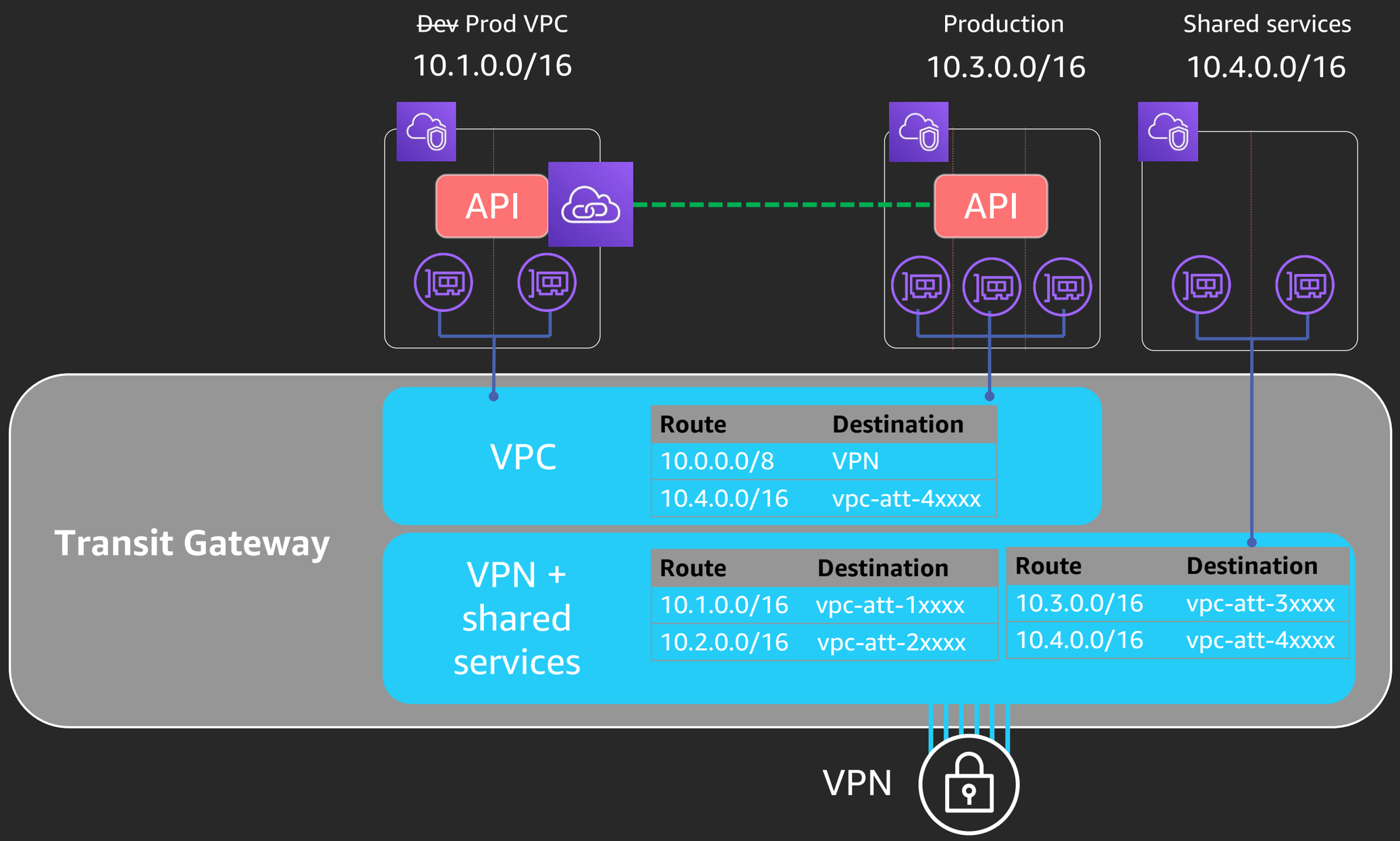
How to handle exceptions?



How to handle exceptions? Shared services

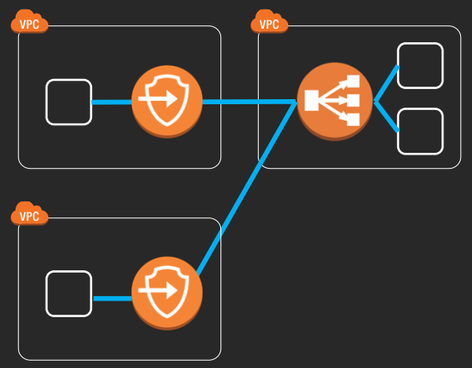


How to handle exceptions? PrivateLink



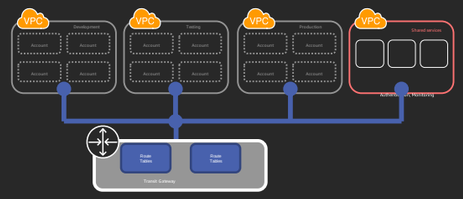
PrivateLink versus Transit Gateway

AWS PrivateLink



- One-to-many connectivity
- Supports overlapping CIDRs
- Uses Network Load Balancer

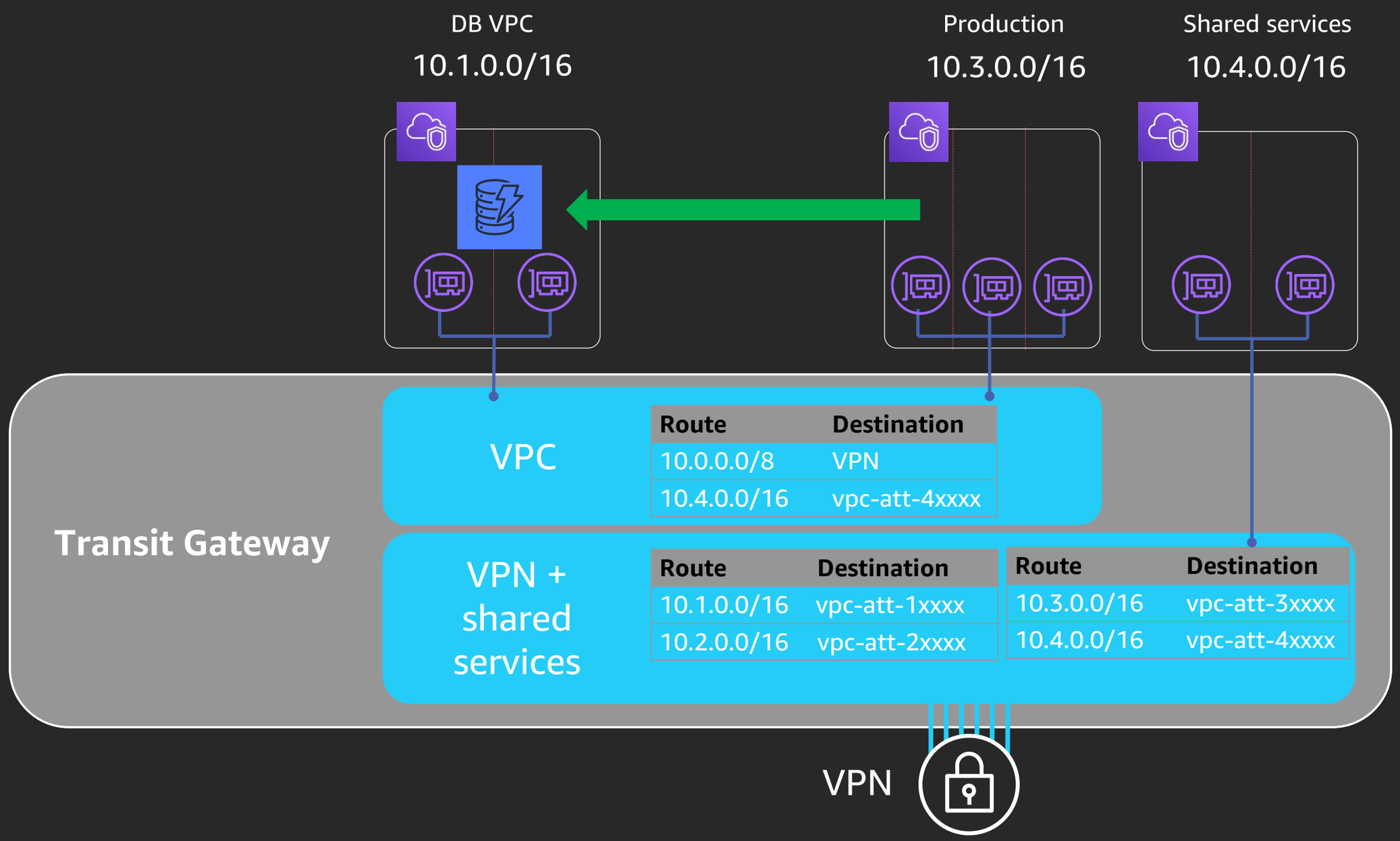
AWS Transit Gateway



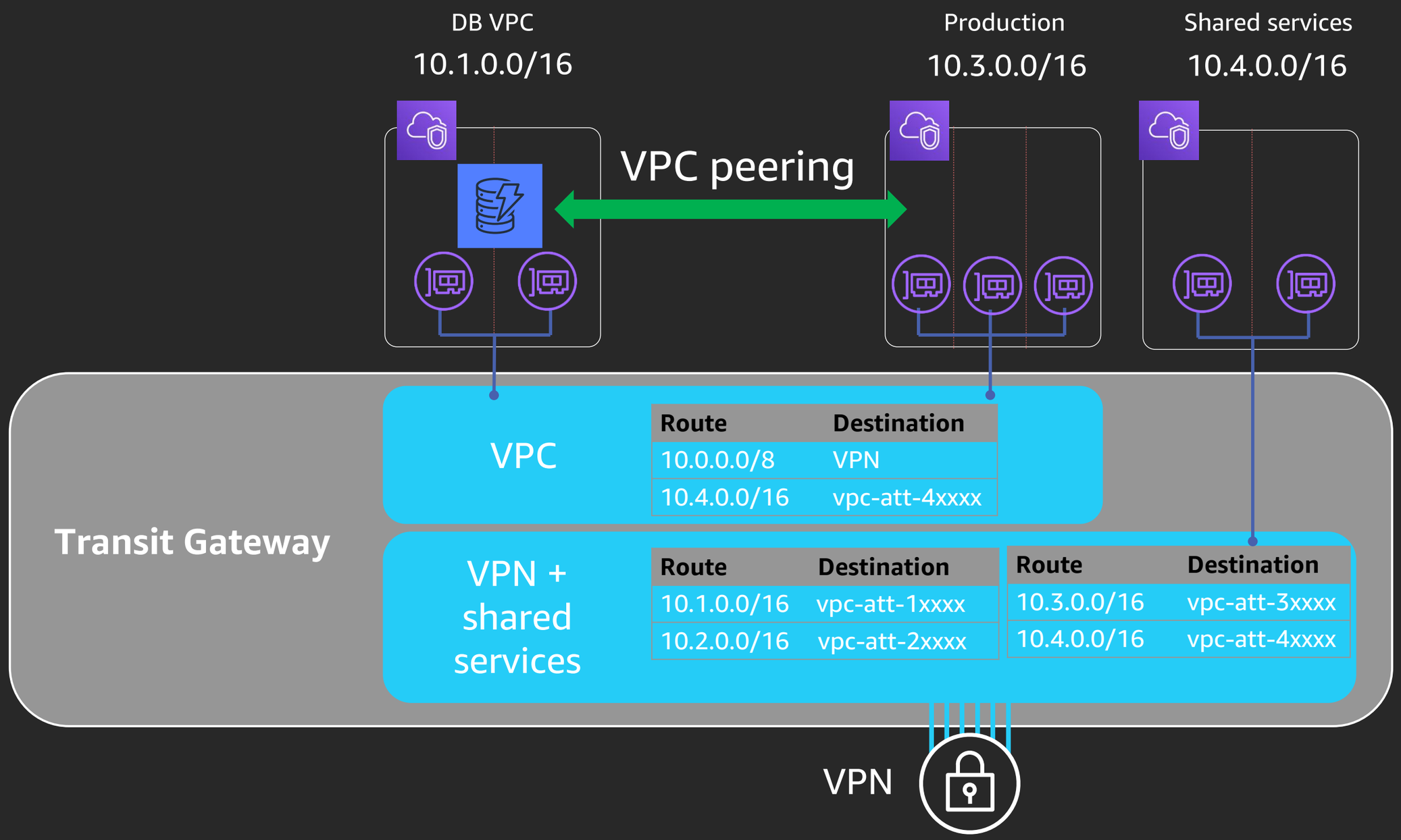
- Many-to-Many or one-to-many with route tables

Scope: Per application, for one application	Scope: Per VPC, for many VPCs
Trust model: No mutual trust	Trust model: Per VPC trust, centralized control
Dependencies: Load balancing and application architecture	Dependencies: Centralized control of the Transit Gateway
Scale: Thousands of spoke VPCs	Scale: Thousands of spoke VPCs
Cost: Load balancing hourly endpoint costs	Cost: Per attachment endpoint costs

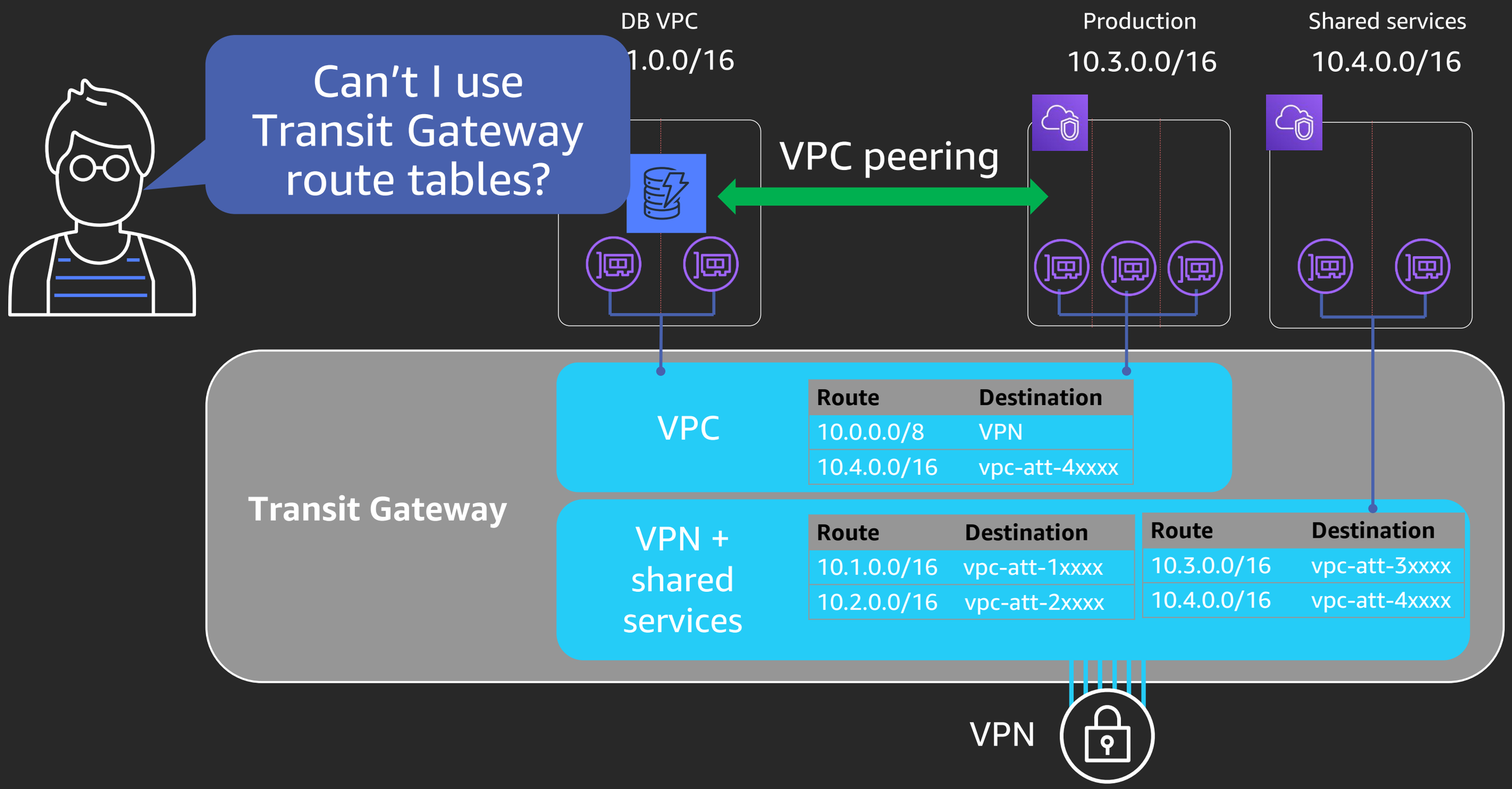
How to handle exceptions?



How to handle exceptions? VPC peering



How to handle exceptions? VPC peering



Exceptions: Route tables for DB and consumer VPCs

	DB VPC	Consumer VPC	VPCs	VPN	Shared services
DB VPC	✗	✓	✗	✓	✓
Consumer VPC	✓	✗	✗	✓	✓
VPCs	✗	✗	✗	✓	✓
VPN	✓	✓	✓	✓	✓
Shared services	✓	✓	✓	✓	✓

Exceptions: Route tables for DB and consumer VPCs

	DB VPC	Consumer VPC	VPCs	VPN	Shared services
DB VPC	✗	✓	✗	✓	✓
Consumer	✗	✗	✗	✓	✓
VPN	✓	✓	✓	✓	✓
Shared services	✓	✓	✓	✓	✓

Ok. Maybe that wouldn't scale for me

How to handle connectivity exceptions

- Ask them to move it to Shared services
- Use PrivateLink when possible
- Use VPC peering or shared VPCs for clusters of connected VPCs
- Build new policies for groups of VPCs

Serverless Transit Network Orchestrator (STNO)

New

Centralized and automated Transit Gateway management

- Spoke VPC route tables (default, RFC1918, custom)
- TGW attachments within the AWS Organization or pre-approved accounts

Includes a Transit Network Management console

- Approval and audit workflow for additional security

Spoke accounts only need to tag their VPC and subnets

No servers to manage using AWS Lambda and AWS Step Functions

STNO route tables

Propagation controlled with tags

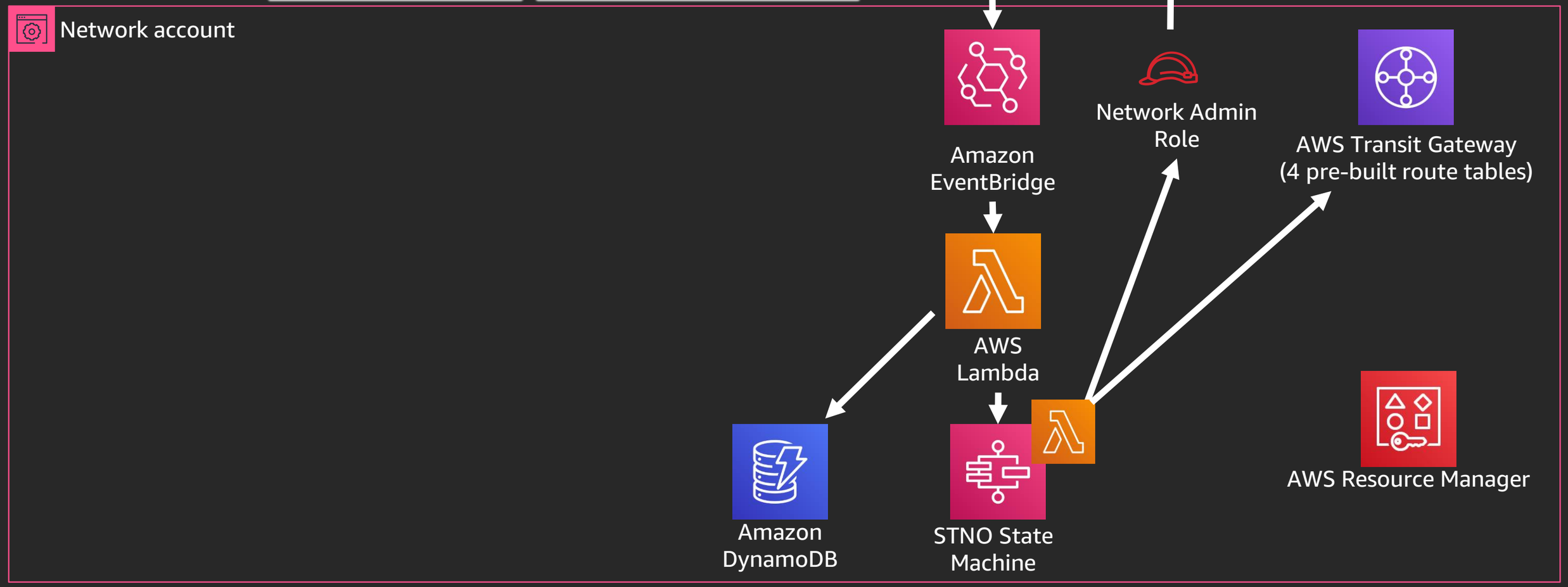
Created by default

	Flat	Isolated	Hybrid	Infrastructure
Flat	✓	✗	✓	✓
Isolated	✗	✗	✓	✓
Hybrid	✓	✓	✓	✓
Infrastructure	✓	✓	✓	✗

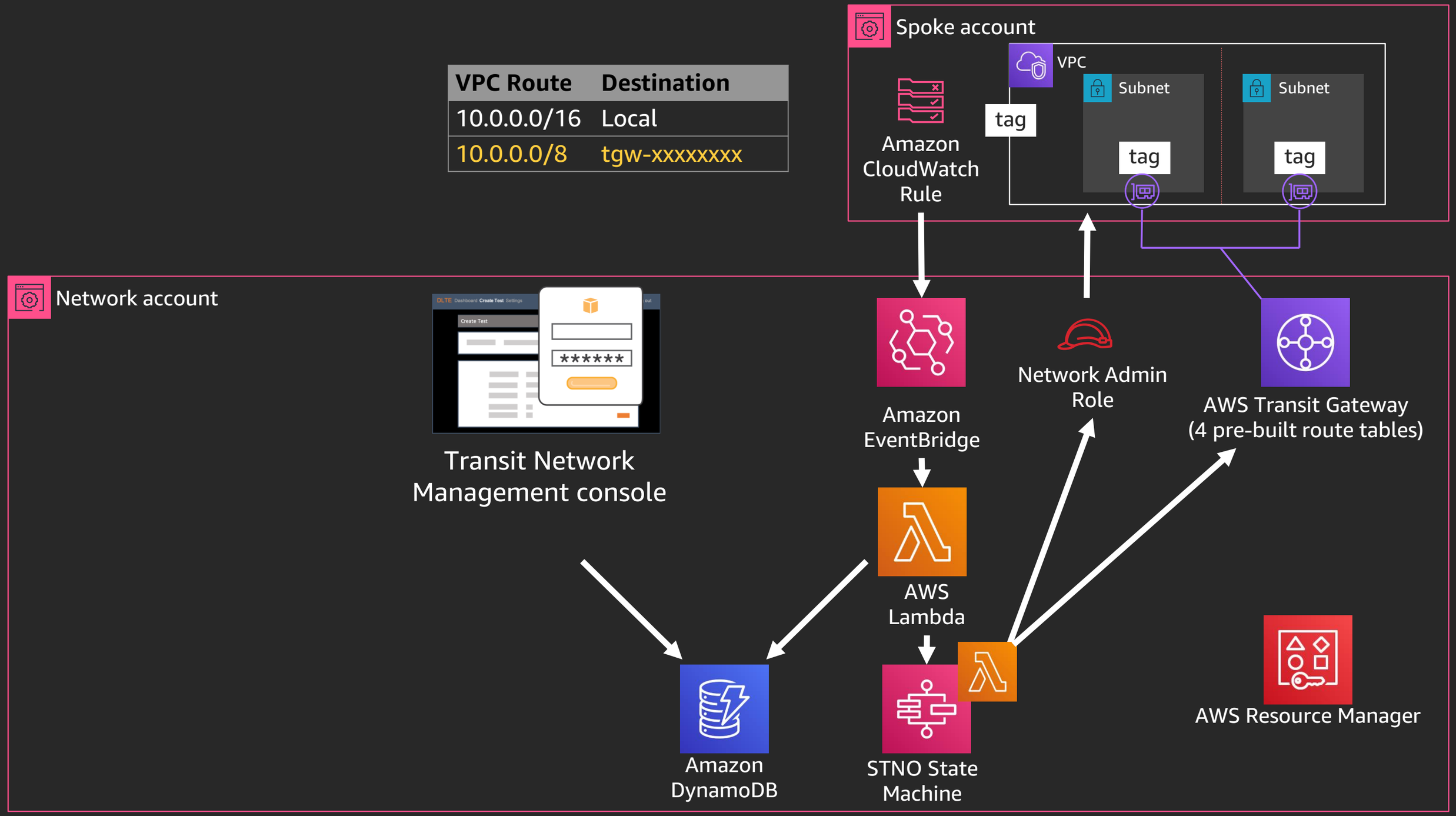
Association controlled with tags

Subnet tags:	
Attach-to-tgw	<blank>

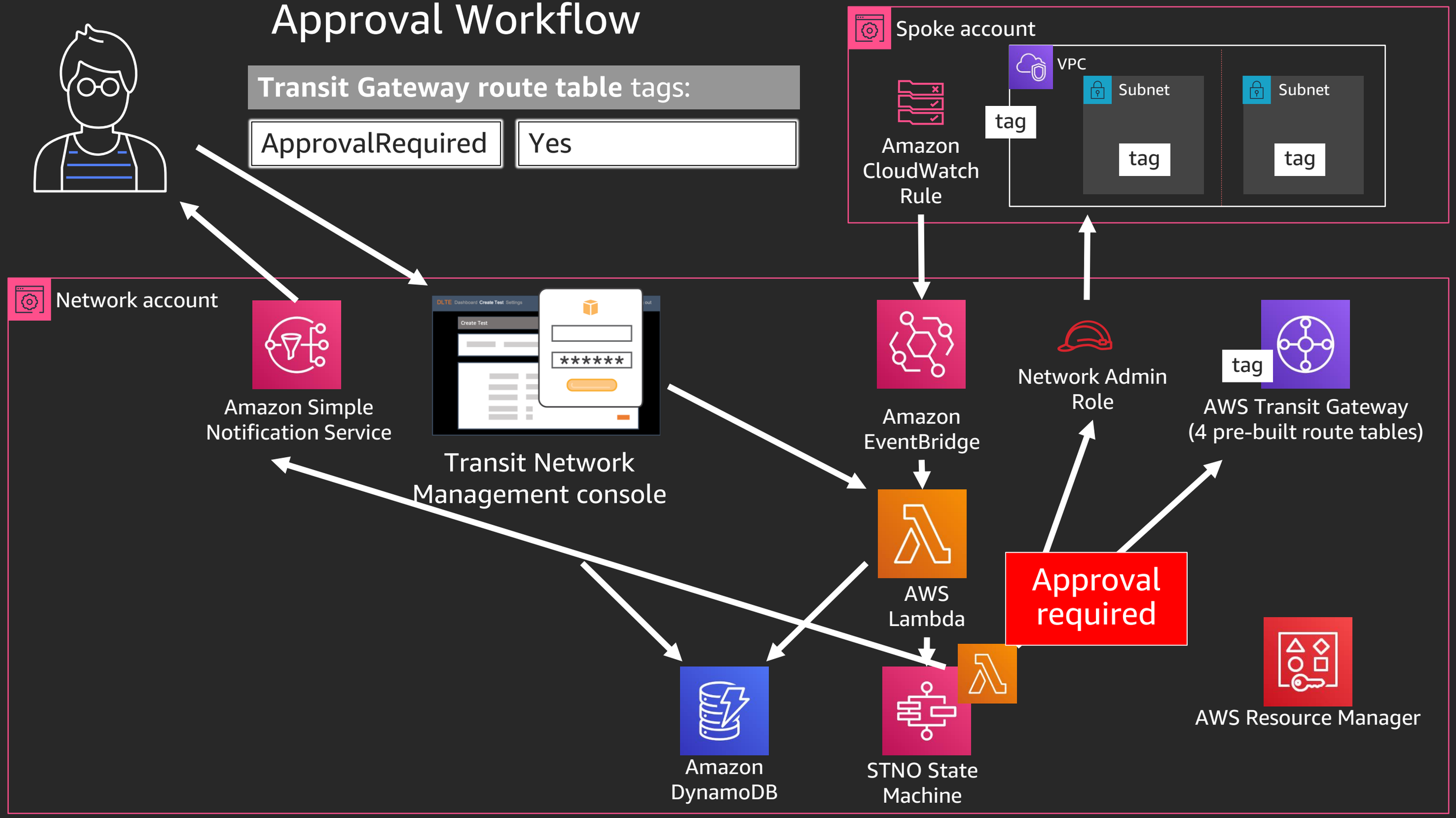
VPC tags:	
Associate-with	Isolated
Propagate-to	Hybrid, Infrastructure



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



Approval Workflow



Try STNO

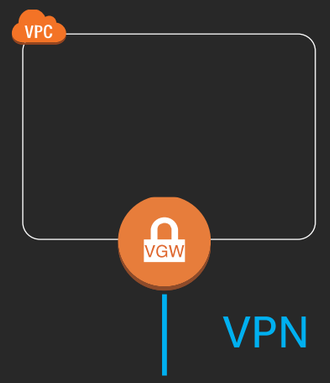
https://amzn.to/37KQMzD

(or search for AWS STNO)

On-premises connectivity

Connecting to on-premises

Virtual Private Gateway VPN

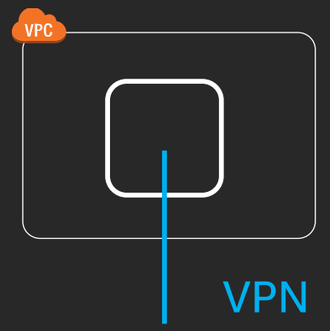


- Per VPC
- 1.25 gbps per tunnel
- Encrypted in transit

DX

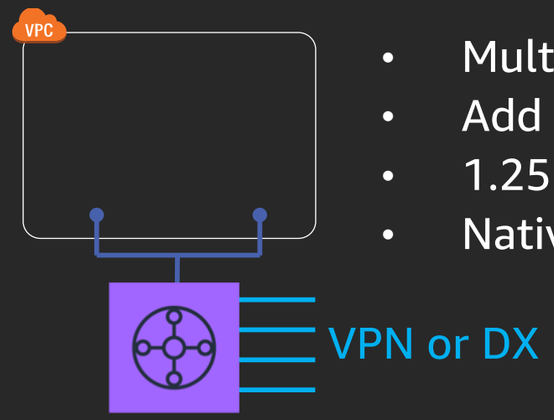
- Per VPC (50-500 per port)
- Multiple VPCs with DX gateway
- No bandwidth restraint

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

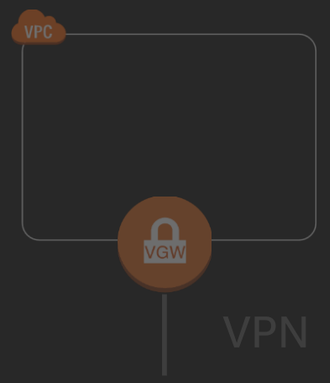
AWS Transit Gateway VPN / DX



- Multiple VPCs
- Add VPNs as needed
- 1.25 gbps per tunnel
- Native DX support

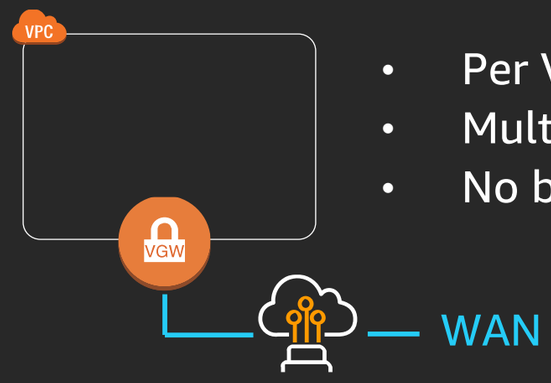
Connecting to On-premises at Scale

Virtual Private Gateway VPN



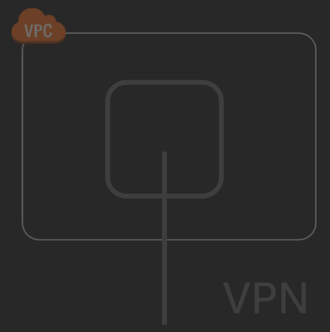
- Per VPC
- 1.25 gbps per tunnel
- Encrypted in transit

DX



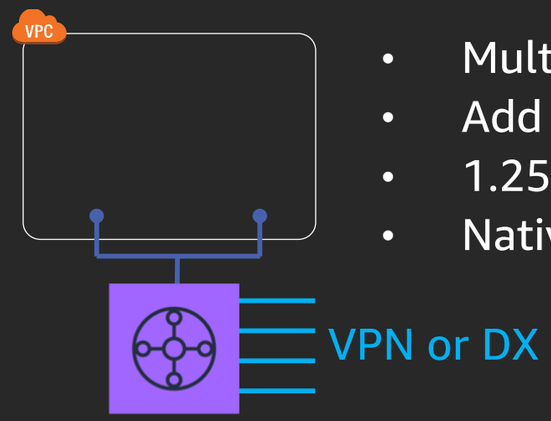
- Per VPC (50-500 per port)
- Multiple VPCs with DX gateway
- No bandwidth restraint

Amazon EC2 Customer VPN



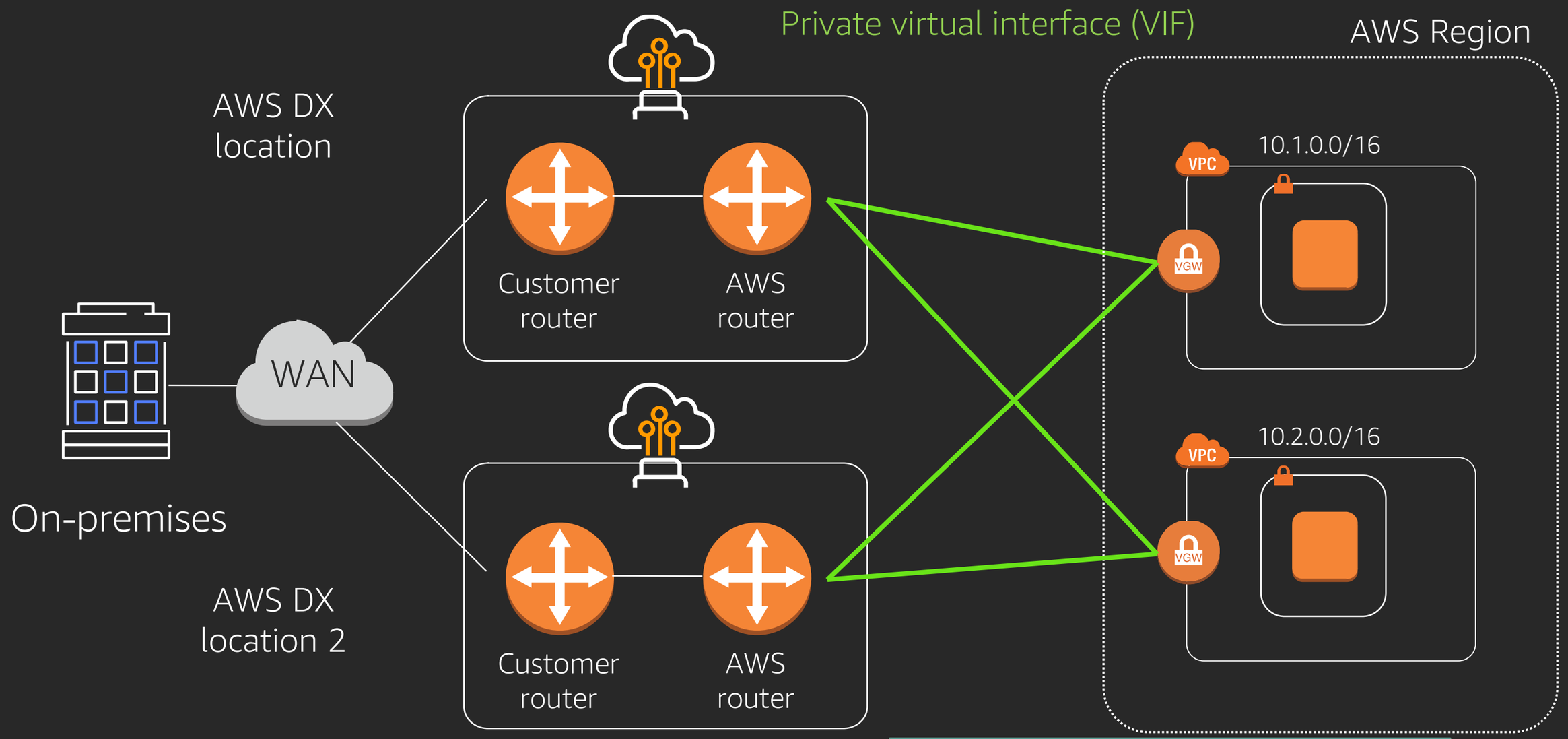
- Per VPC or multiple (Transit VPC)
- Bandwidths vary by instance type
- AWS Marketplace options
- Scalability is generally limited by management complexity

AWS Transit Gateway VPN / DX



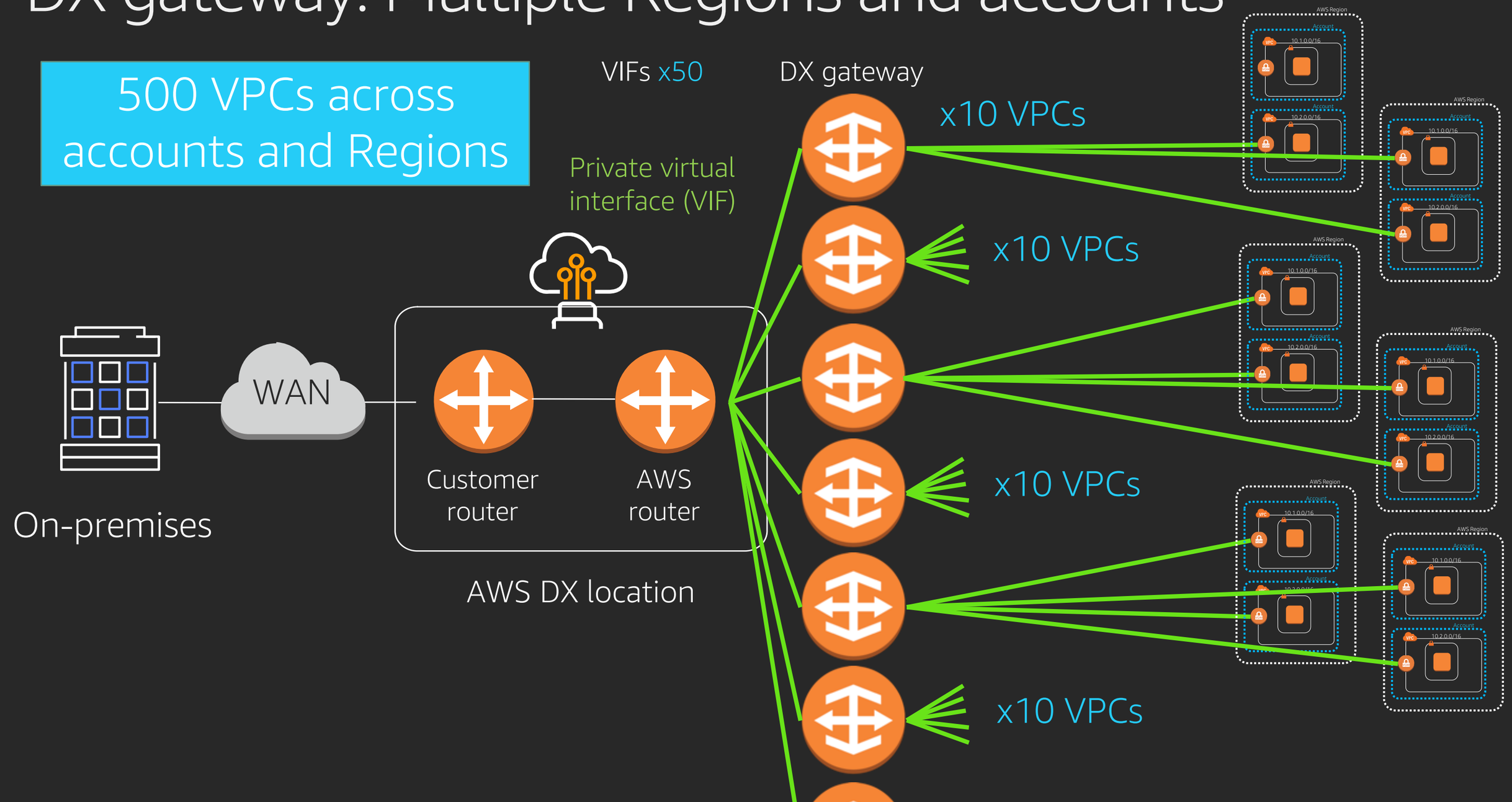
- Multiple VPCs
- Add VPNs as needed
- 1.25 gbps per tunnel
- Native DX support

DX direct to VPCs

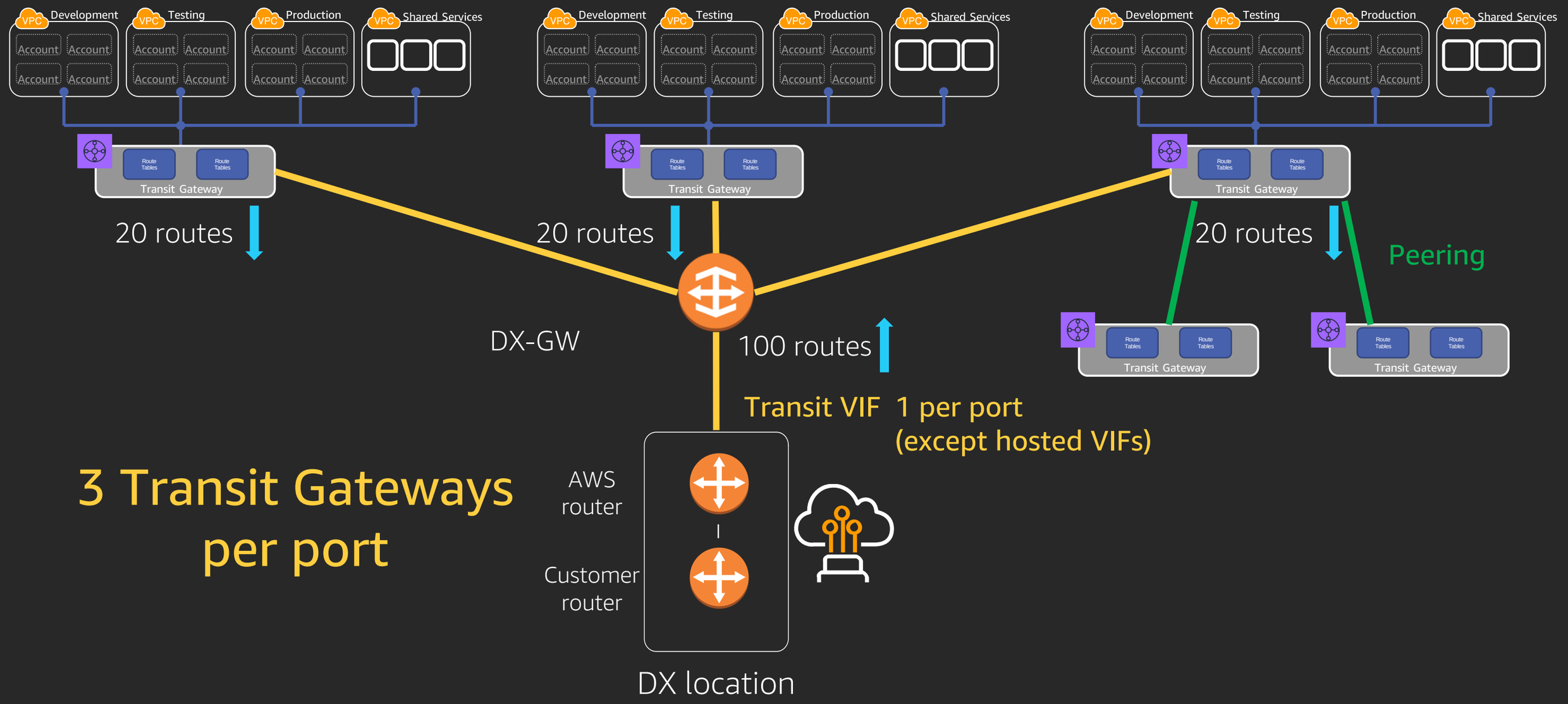


Up to 50 VIFs per port

DX gateway: Multiple Regions and accounts



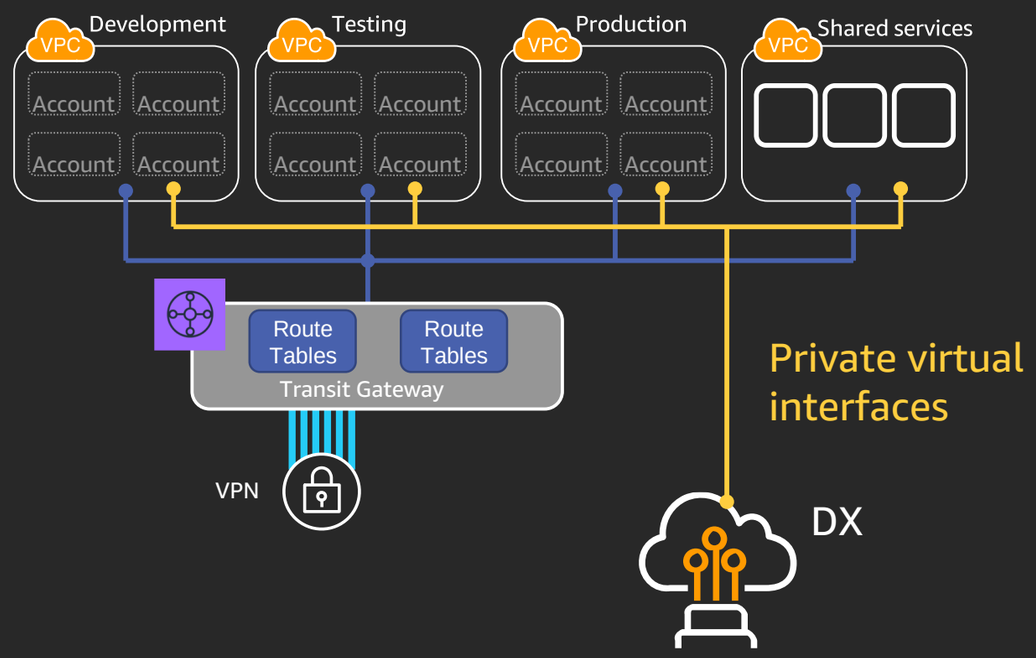
Transit virtual interface



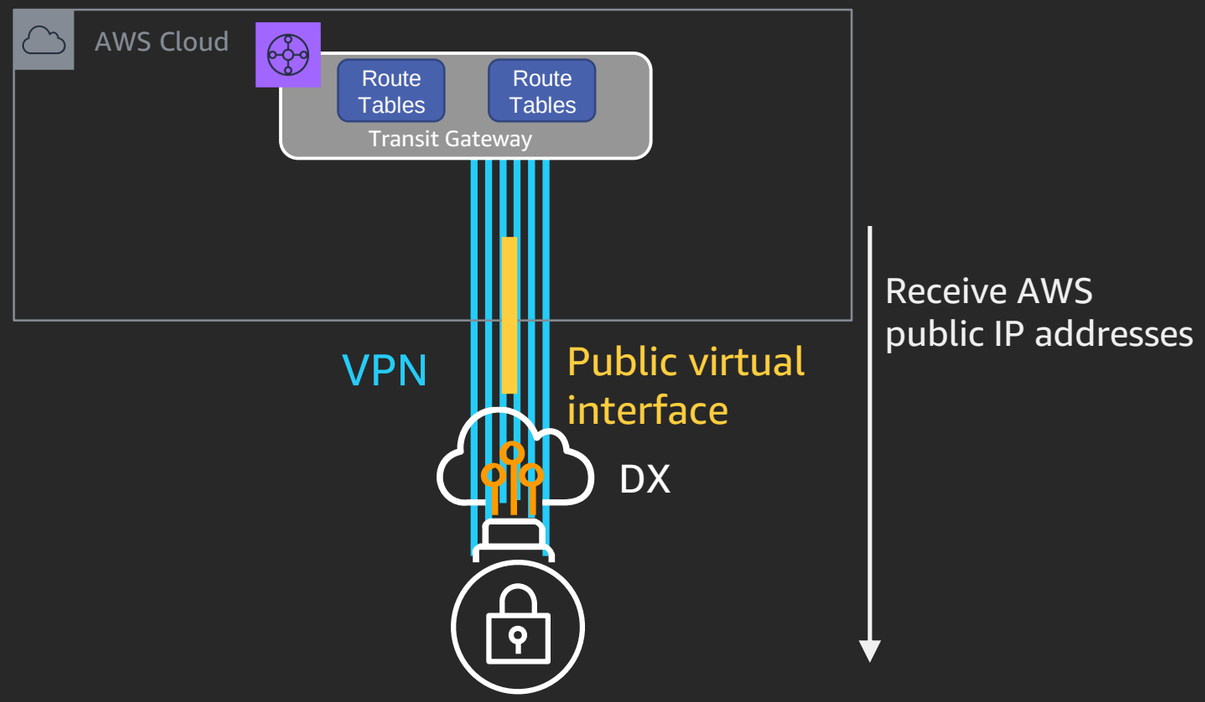
3 Transit Gateways per port

Other DX and Transit Gateway options

Use DX in parallel



VPN over a public virtual interface



Use cases: Add VPN backup to existing VPCs
Avoid ingress Transit Gateway charges
Scaling beyond 3 Transit Gateways

Use cases: Encryption over DX
Scaling beyond 3 Transit Gateways

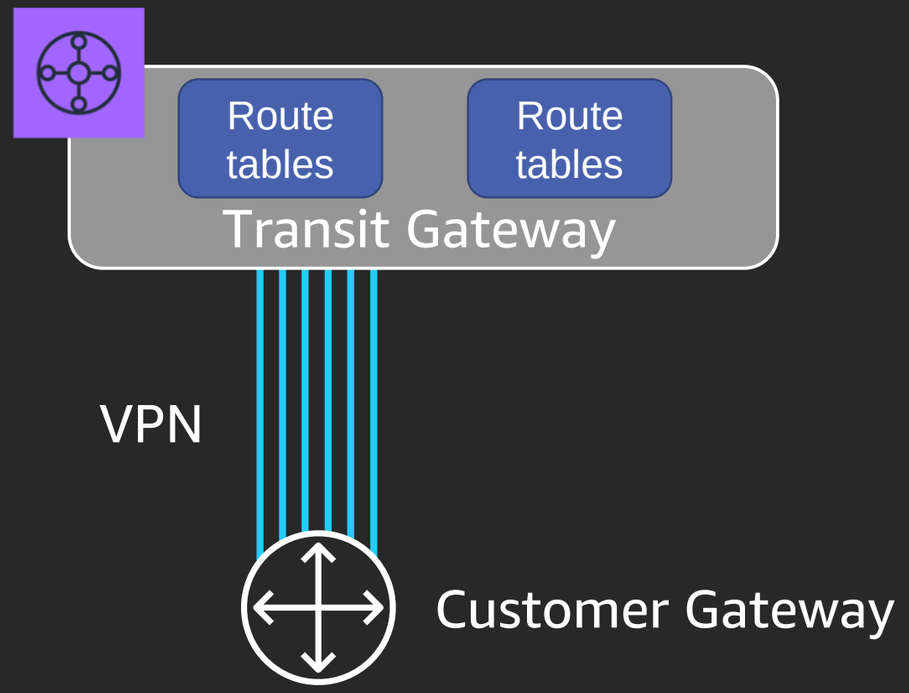
VPN with Transit Gateway: Add more bandwidth

Support for spreading traffic across up to 50 gbps

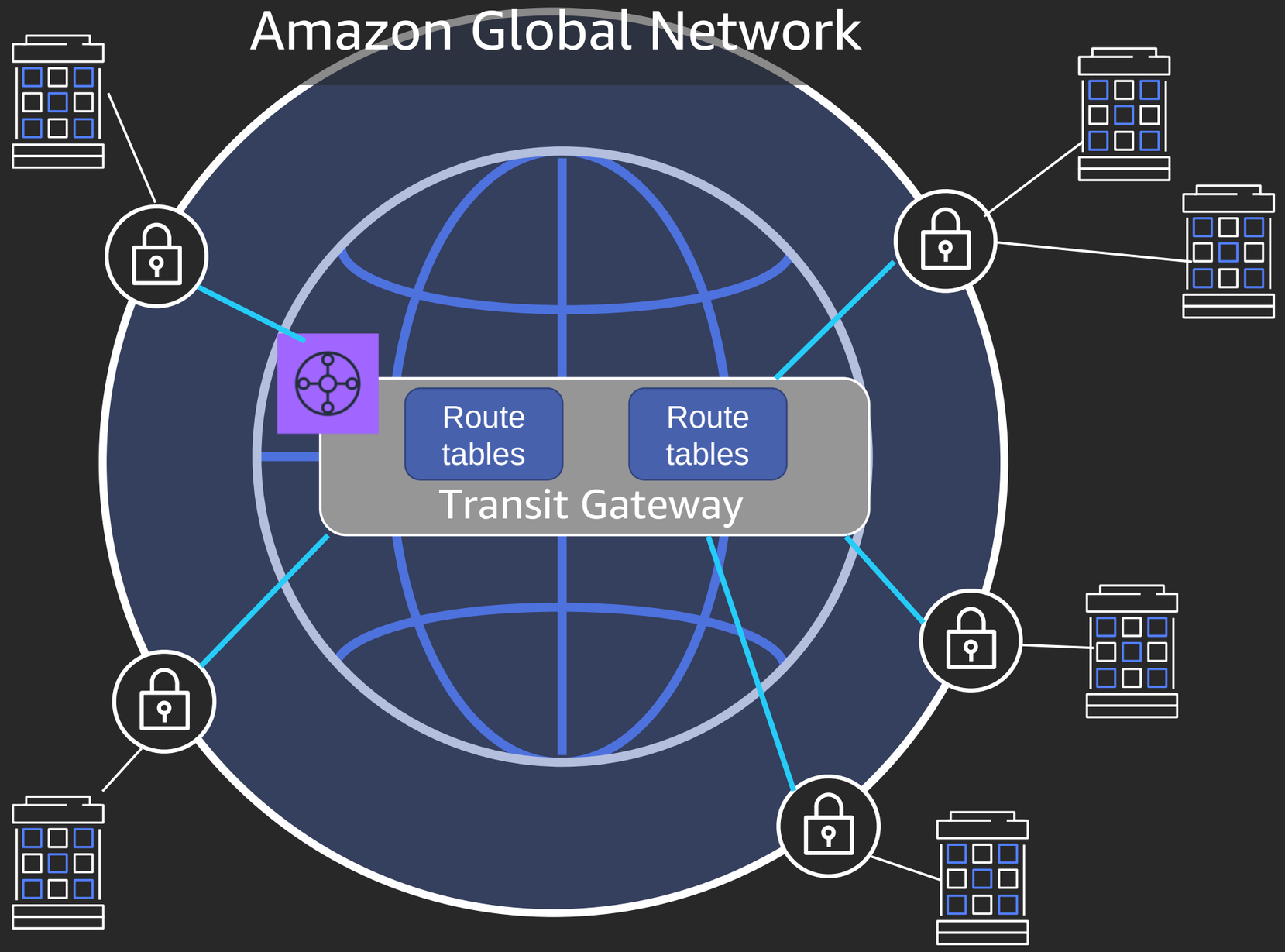
- Equal Cost Multi-Path (ECMP) support with BGP multi-path
- 1.25 gbps flow limits, so split traffic into smaller flows and use multi-part uploads

Check your on-premises support

- Multi-path BGP required
- ECMP support, amount of equal paths, reverse-path forwarding/spoofing checks



Accelerated VPN



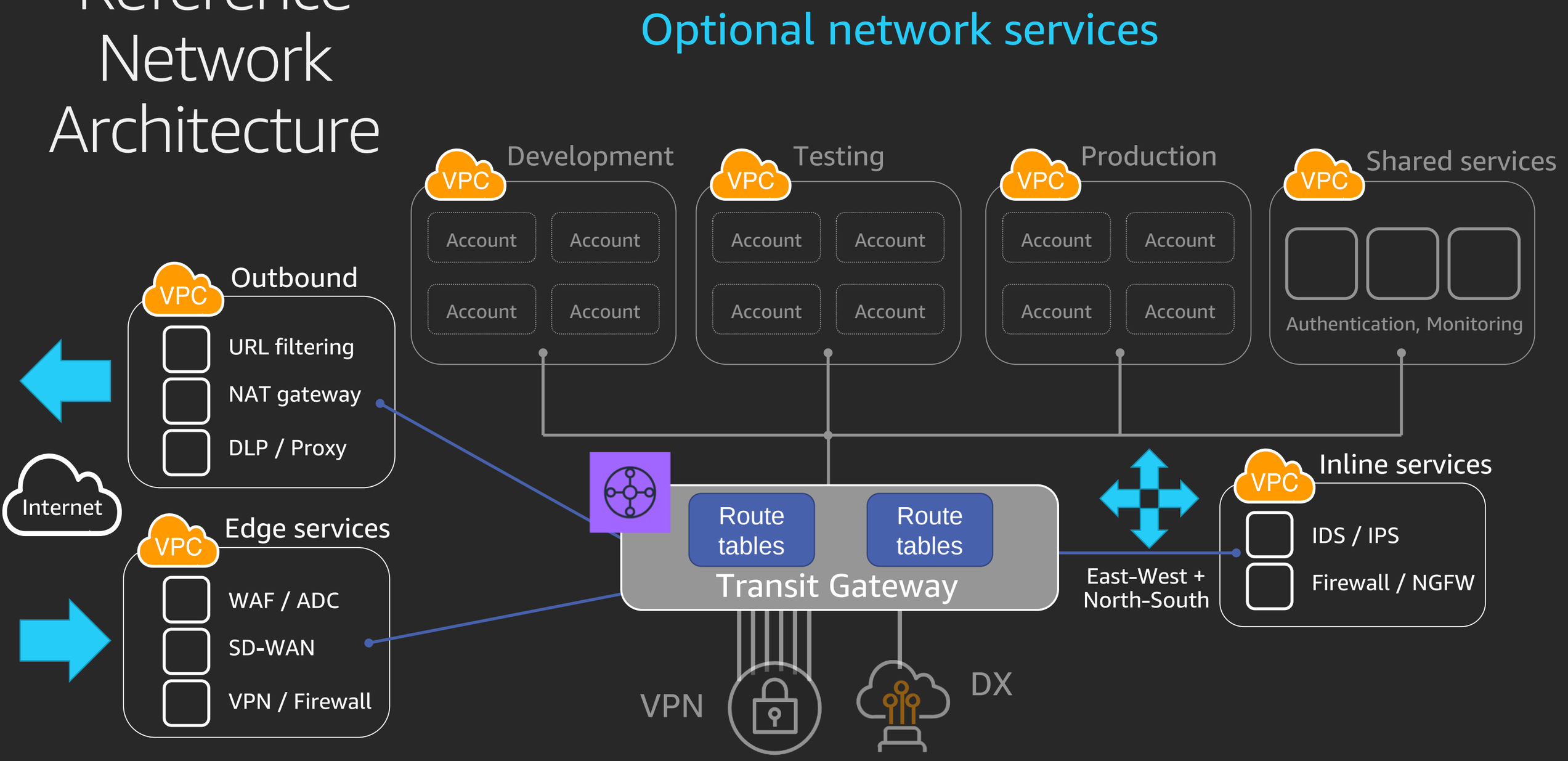
New

Leverage Amazon's Global Network

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

Network services

Reference Network Architecture



Interface method

Method one: Interface attachment

Spoke route table

Route	Destination
0.0.0.0/0	tgw-xxxxxxxxx

Outbound VPC route table

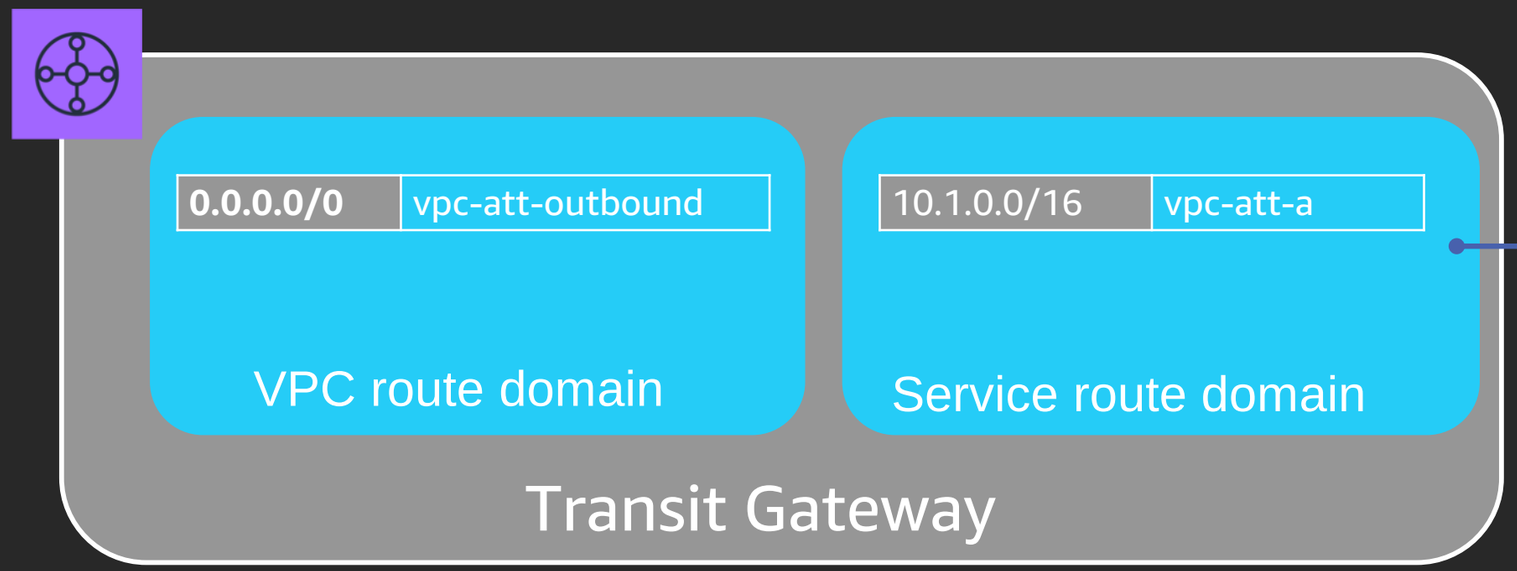
Route	Destination
10.0.0.0/8	tgw-xxxxxxxxx
0.0.0.0/0	igw-xxxxxxxxx

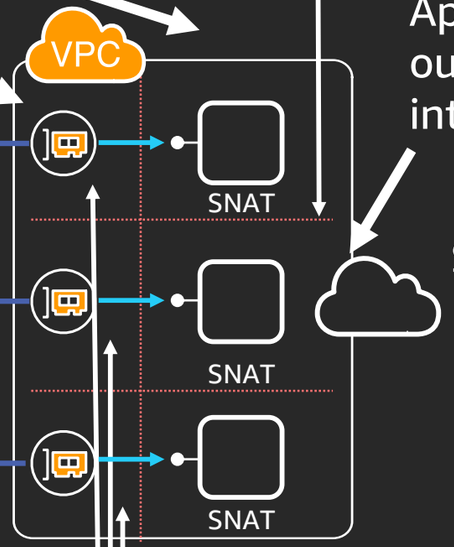
Create return route to VPCs

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
0.0.0.0/0	tgw-xxxxxxxxx

Outbound VPC route table

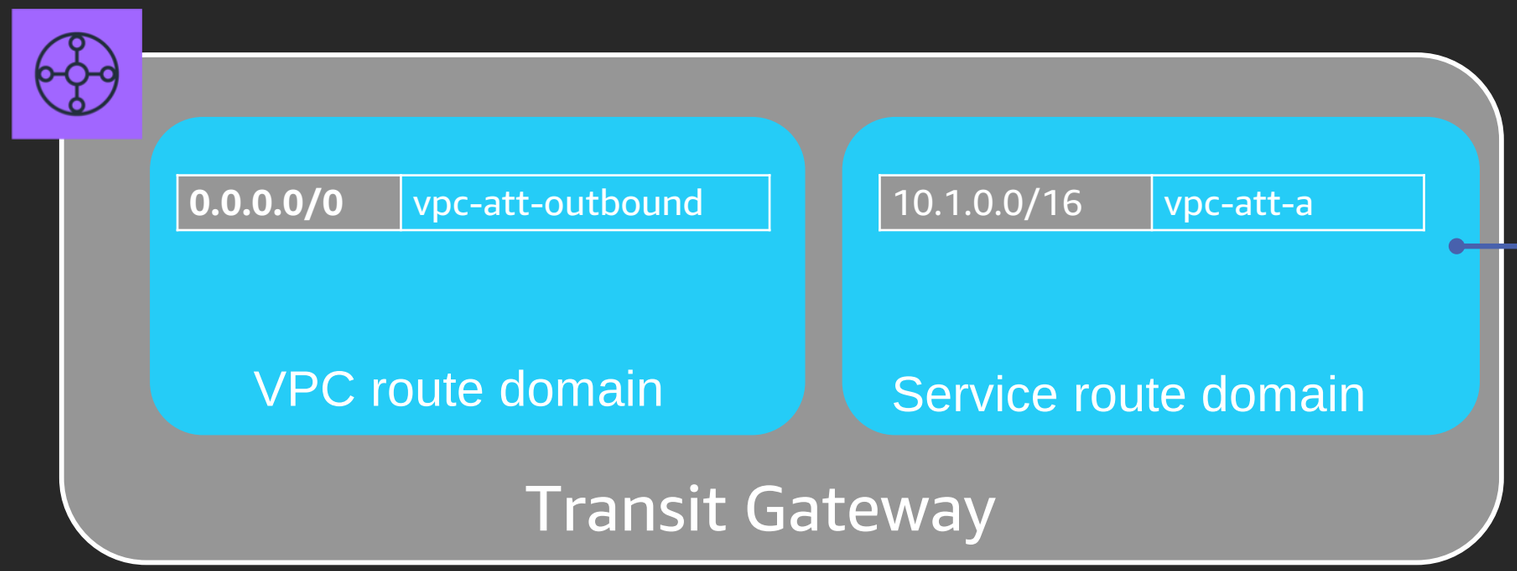
Route	Destination
10.0.0.0/8	tgw-xxxxxxxxx
0.0.0.0/0	igw-xxxxxxxxx

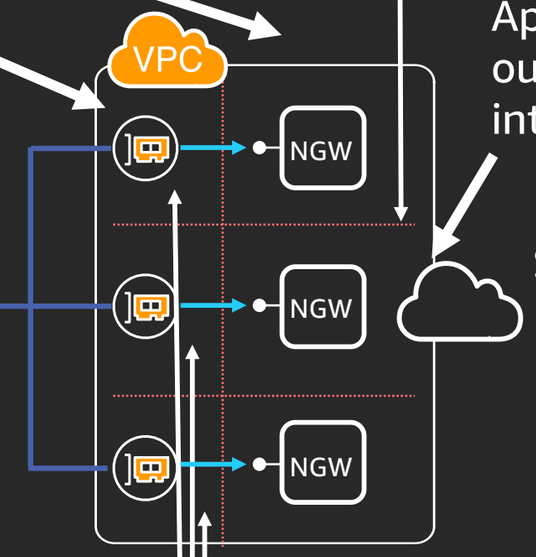
Create return route to VPCs

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	ngw-xxxxxxx

Interface method

Interface insertion design notes

Simpler pattern, DIY health checks

Potentially limited to performance of a single instance (worst-case scenario). Configure your own high availability checks.

Instance must SNAT or use 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 or active-standby to guarantee the return flow to the same instance

VPN method

Method two: VPN attachment

Spoke route table

Route	Destination
0.0.0.0/0	tgw-xxxxxxxxx

Outbound VPC route table

Route	Destination
0.0.0.0/0	igw-xxxxxxxxx

Load balance traffic across many VPN tunnels

VPC routes will be advertised over BGP

ECMP VPN

Apply SNAT outbound to the internet

Service VPC

0.0.0.0/0	Outbound VPC VPN

VPC route domain

10.1.0.0/16	vpc-att-a

Service route domain

Transit Gateway

VPC

SNAT

SNAT

SNAT

BGP prefix	Next hop
0.0.0.0/0	Local IP

BGP advertisement

VPN method

VPN insertion design notes

Instance must be able to support:

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

Horizontally scalable service pattern, more overhead

Preferred method if the service supports BGP, VPN, and 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

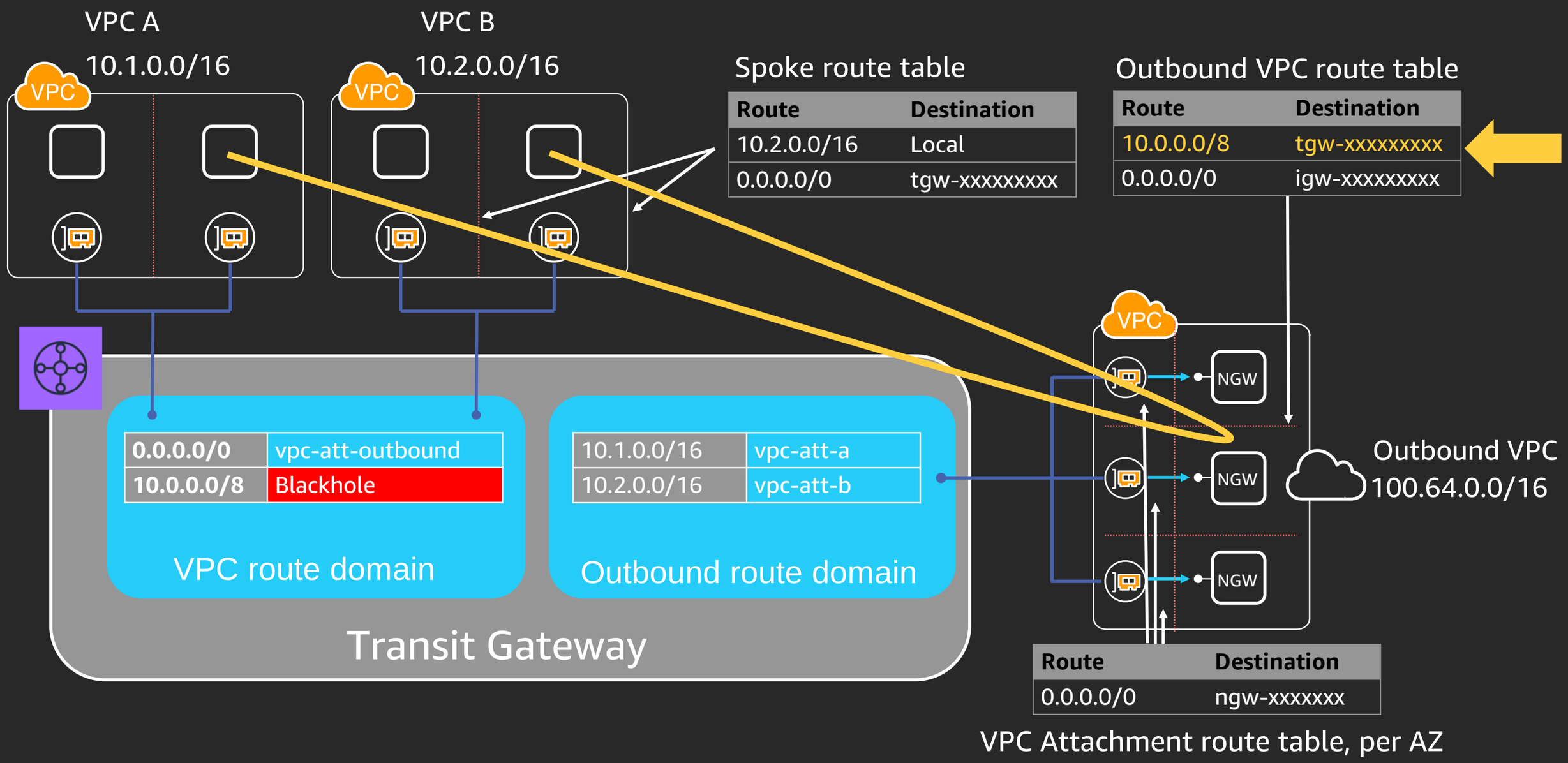
- Use Source NAT or active-standby to guarantee the return flow to the same instance

Outbound services: Interface

Interface method

Use cases:

NAT gateways, services without VPN support

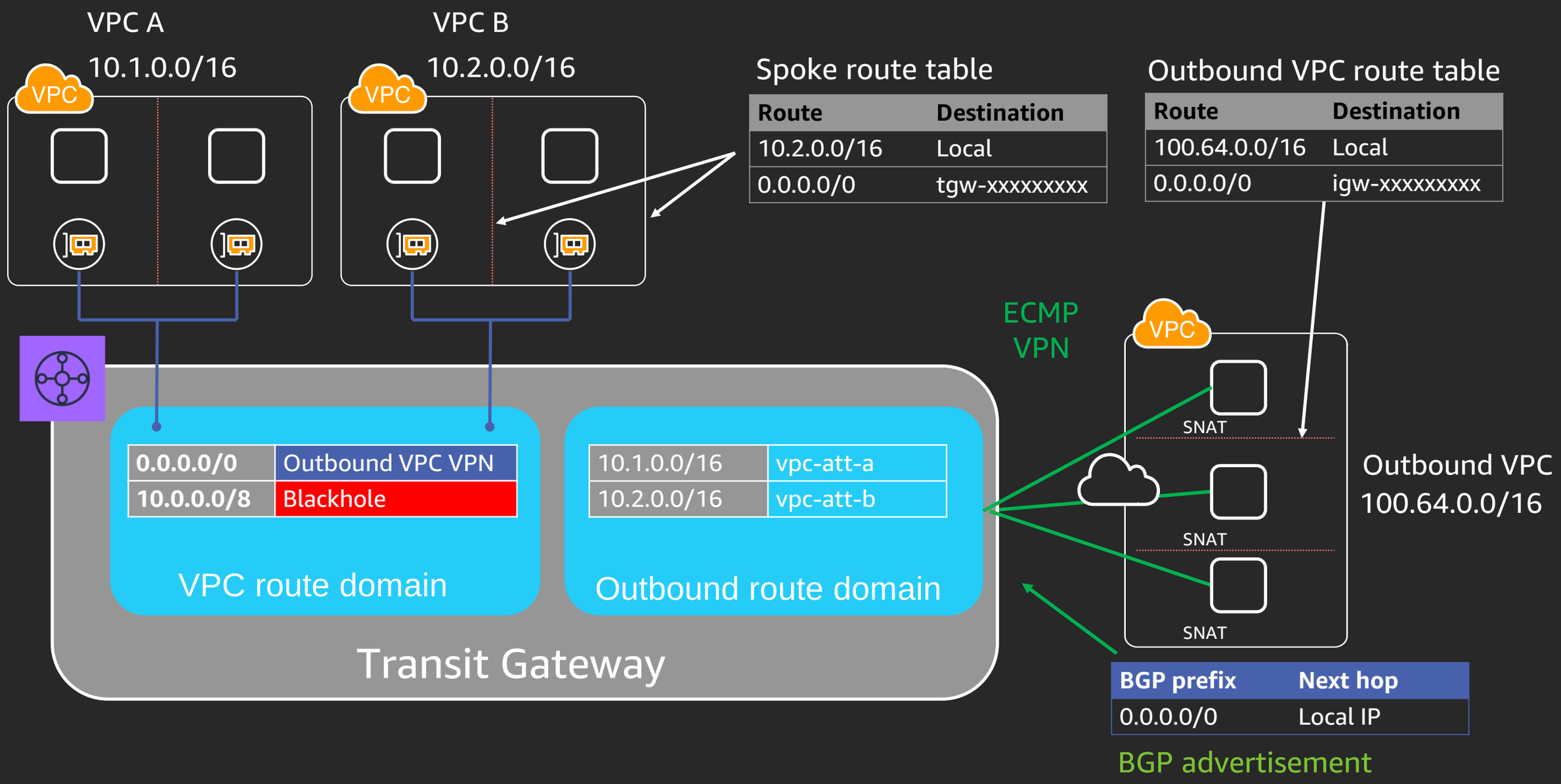


Outbound services: VPN

VPN method

Use cases:

URL filtering, firewalls, IPS, web proxy services



Ingress services

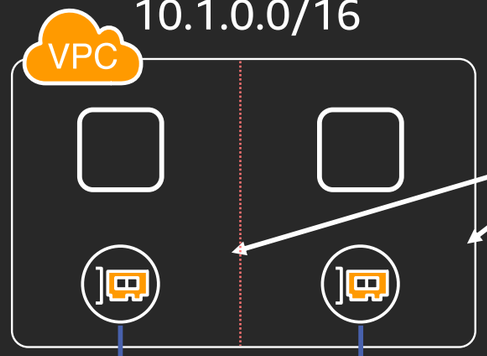
VPN method

Use cases:

WAF, inspection, Load balancing

VPC A

10.1.0.0/16

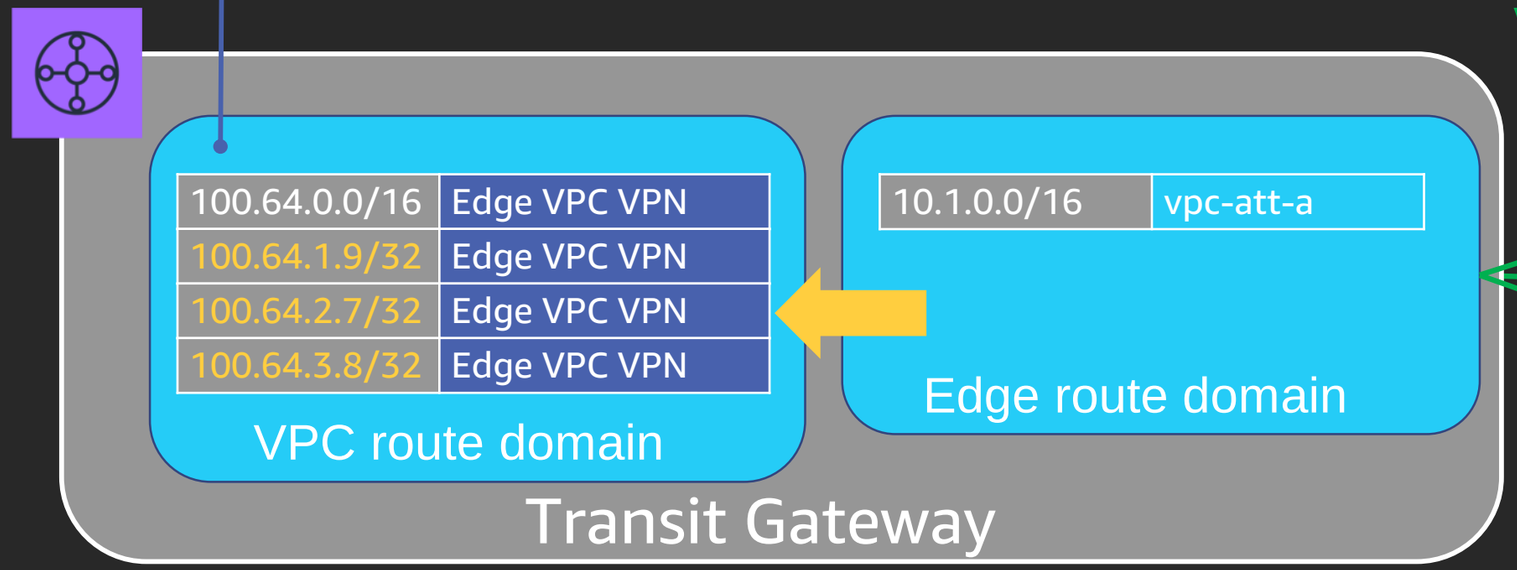


Spoke route table

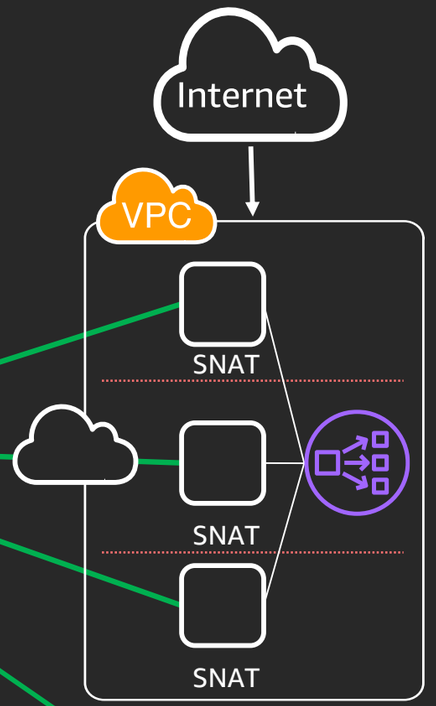
Route	Destination
10.1.0.0/16	Local
100.64.0.0/16	tgw-xxxxxxxxx

Edge VPC route table

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



ECMP VPN



Edge VPC

100.64.0.0/16

Optional ELB

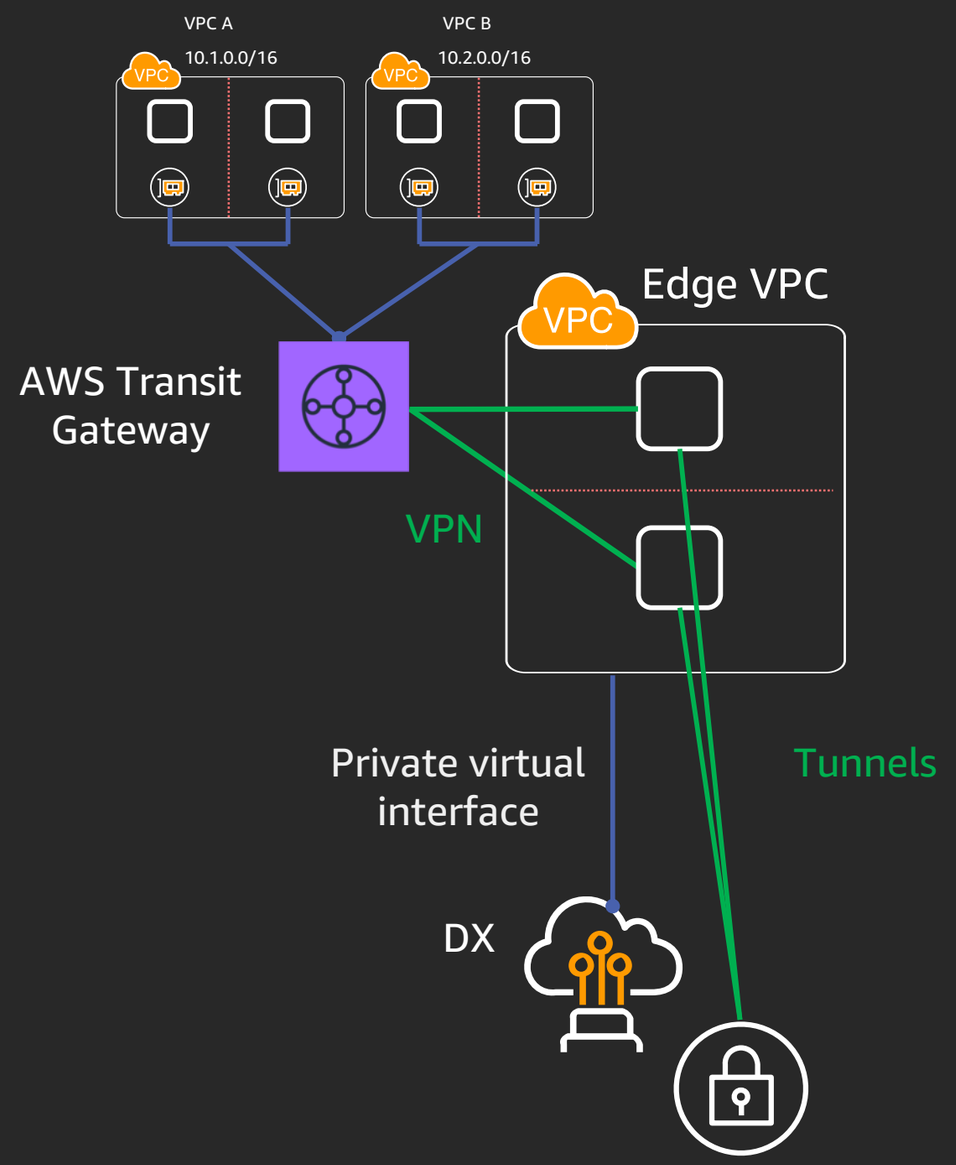
BGP advertisement

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

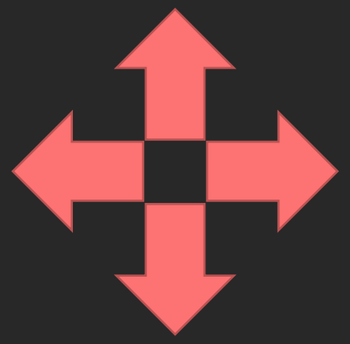
Edge services: SDWAN, VPN, Firewalls

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



Reminder:



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

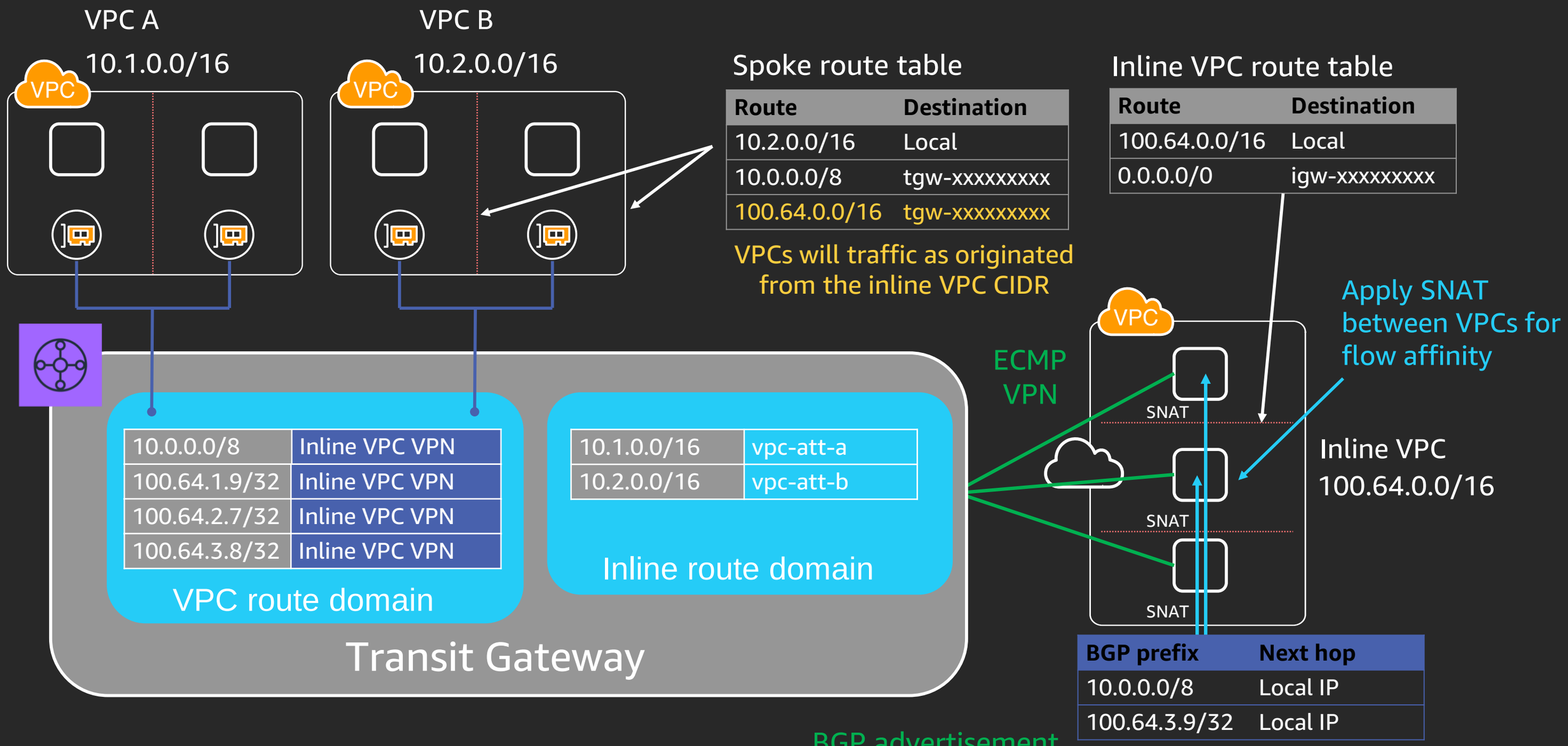
Remember to evaluate operational processes, alternatives, and automation

Inline service: VPN

VPN method

Use cases:

Intrusion detection/prevention (IDS/IPS), firewalls



Transit Gateway partners

aviatrix

Check Point
SOFTWARE TECHNOLOGIES LTD.

CISCO
vEdge SD-WAN
CSR 1000v

f5

FORTINET

paloalto NETWORKS
VM-Series

Symantec
ProxySG

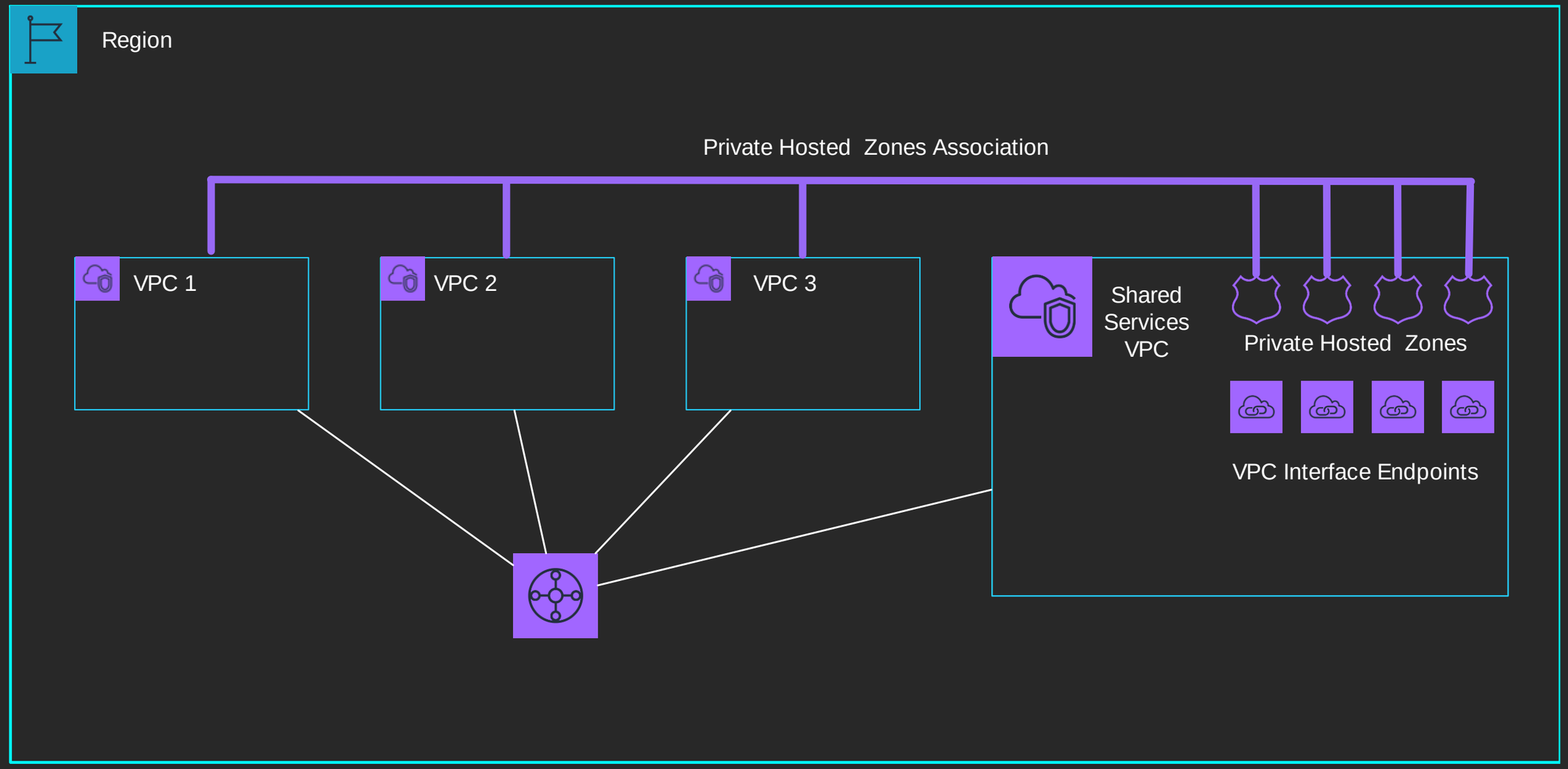
HashiCorp
Terraform

Hewlett Packard Enterprise

silver peak

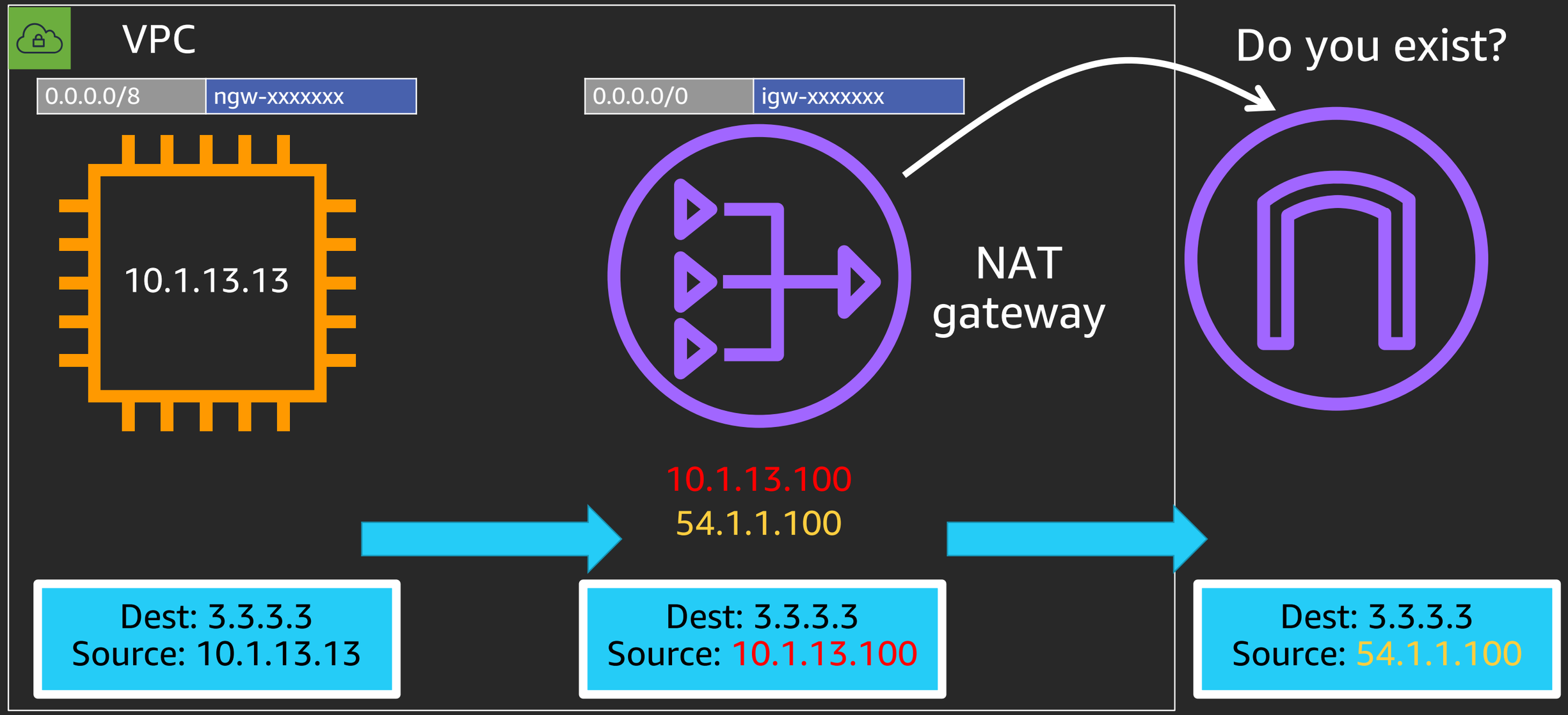
CITRIX

Centralizing PrivateLink with Transit Gateway

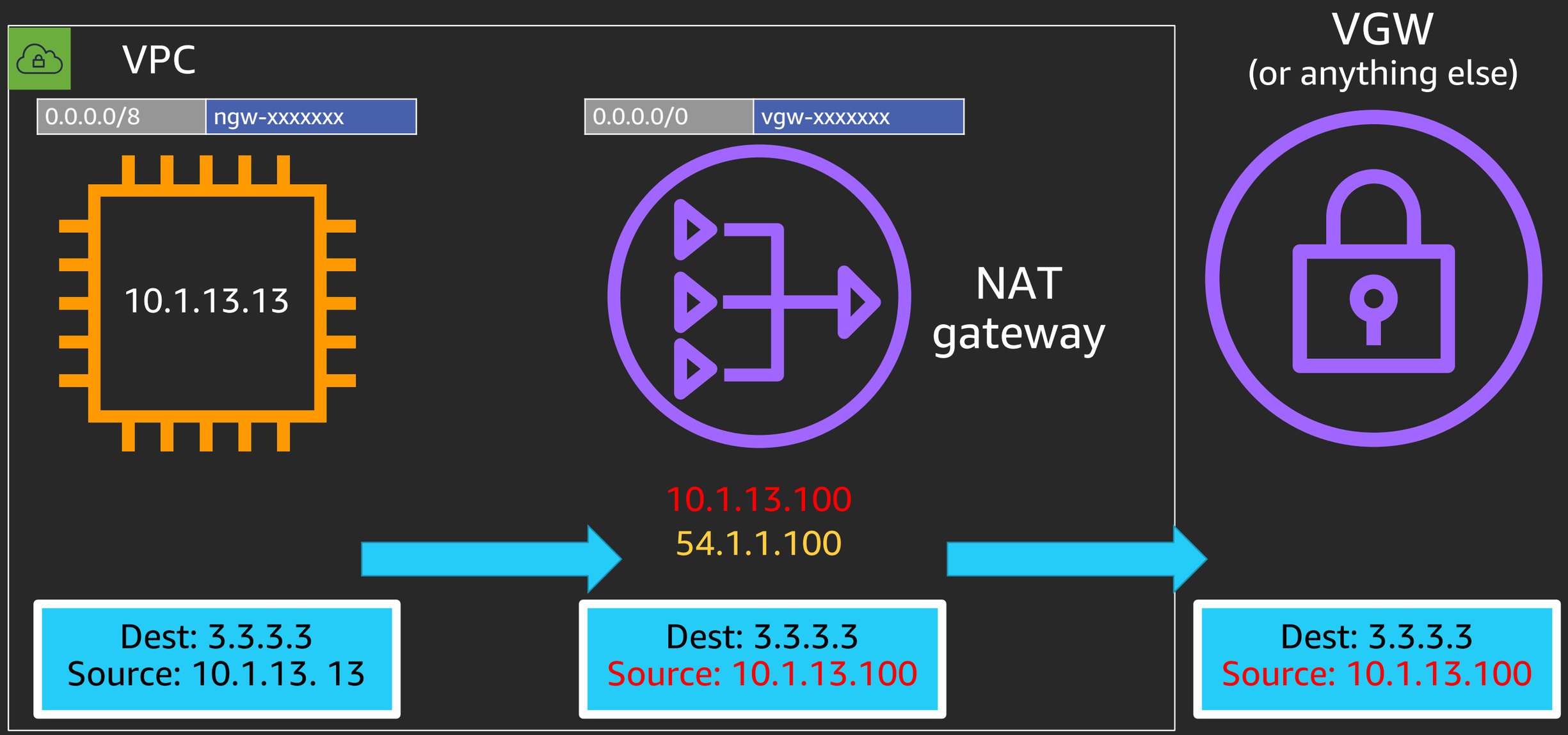


For more: NET321 Wednesday, Dec 4, 1:00 PM - 2:00 PM

Let's go to NAT school

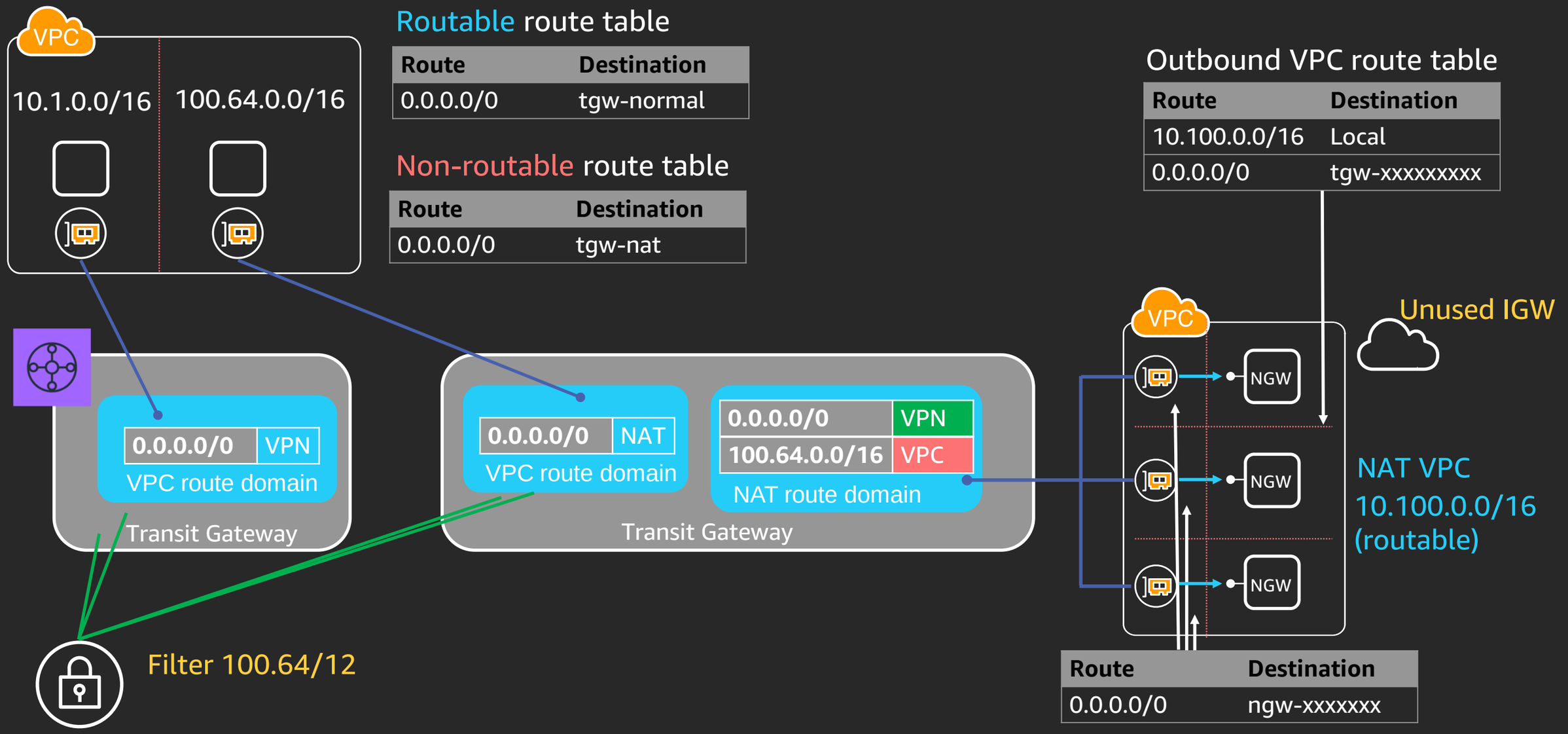


Let's go to NAT school



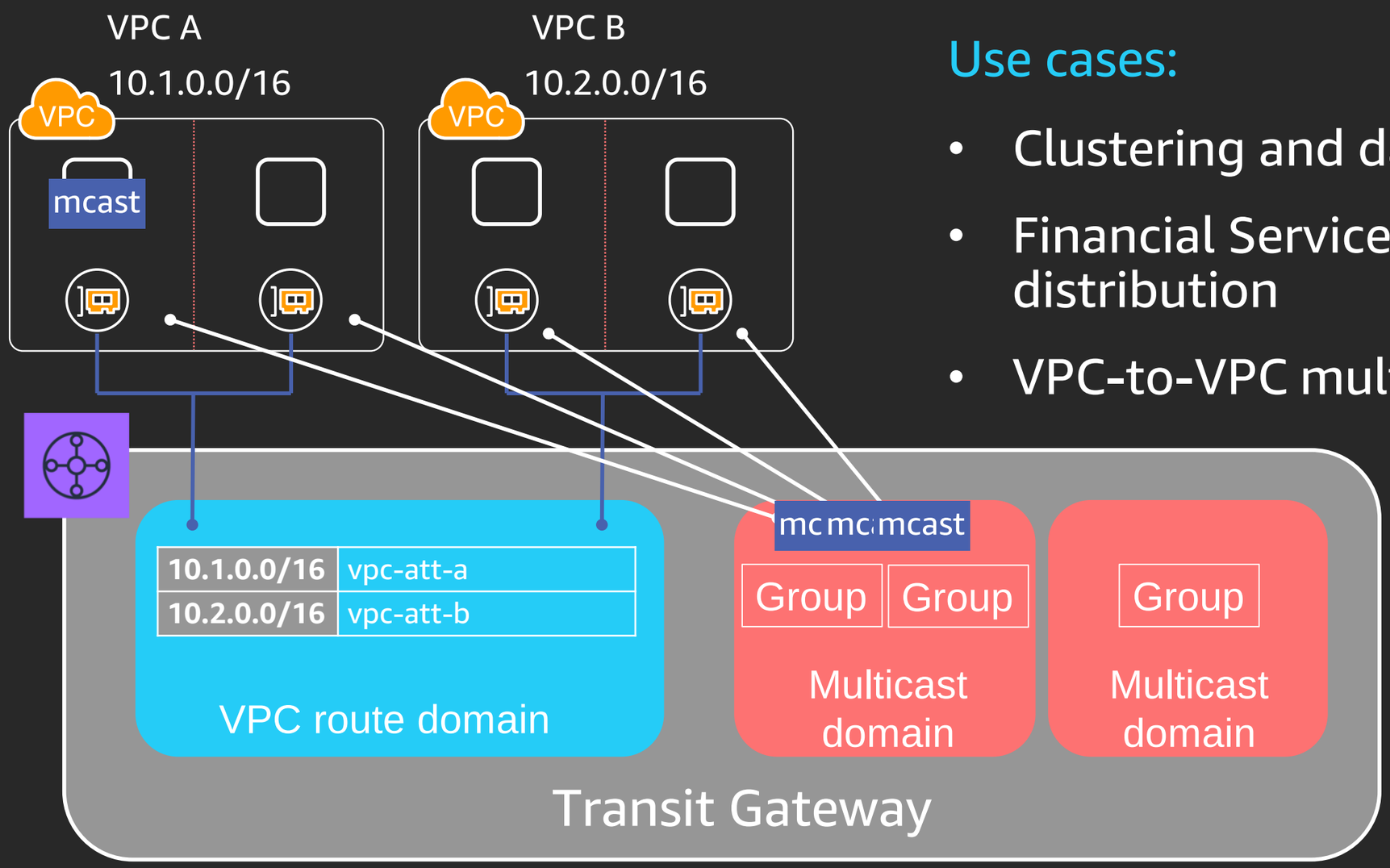
Centralized IP address preservation

Interface method



Multicast on AWS Transit Gateway

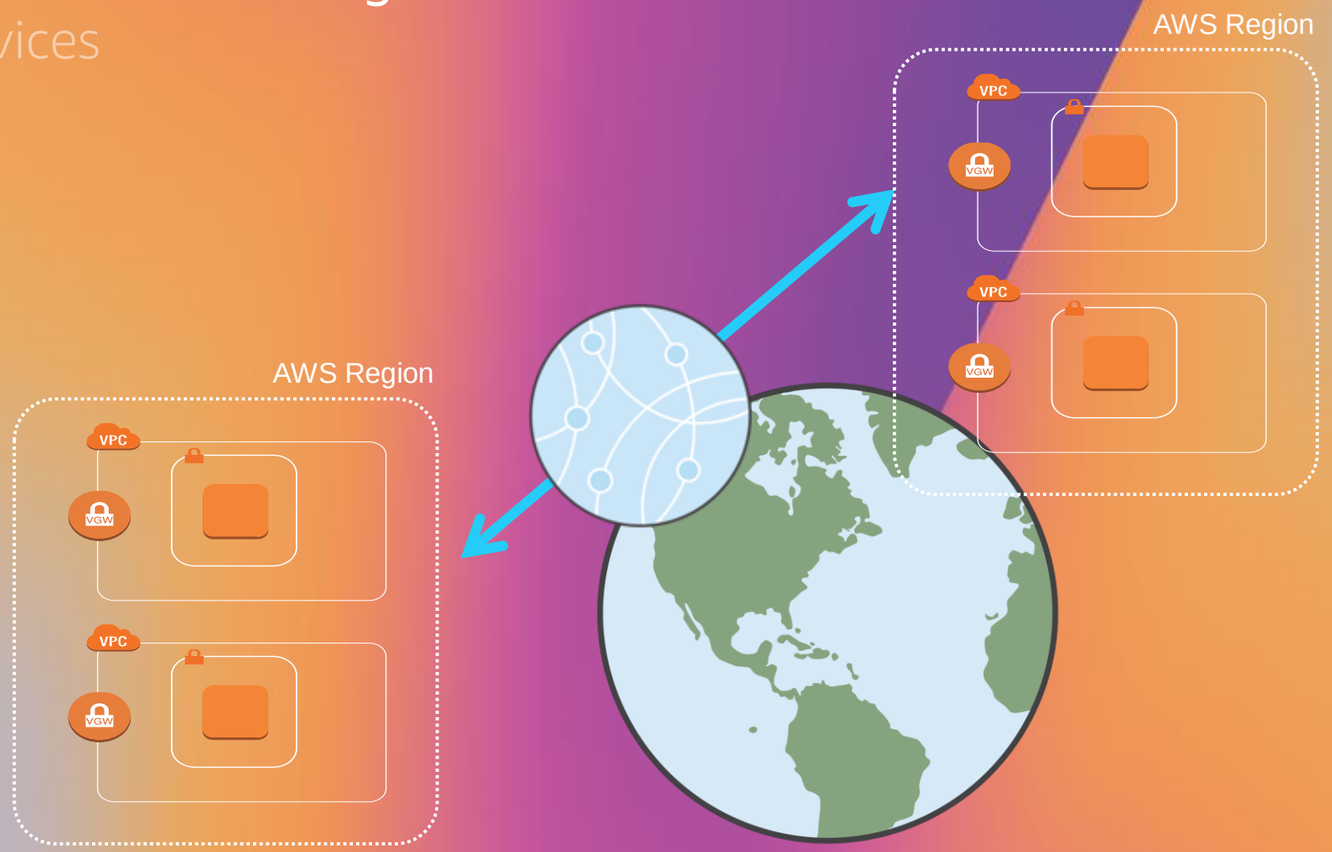
New



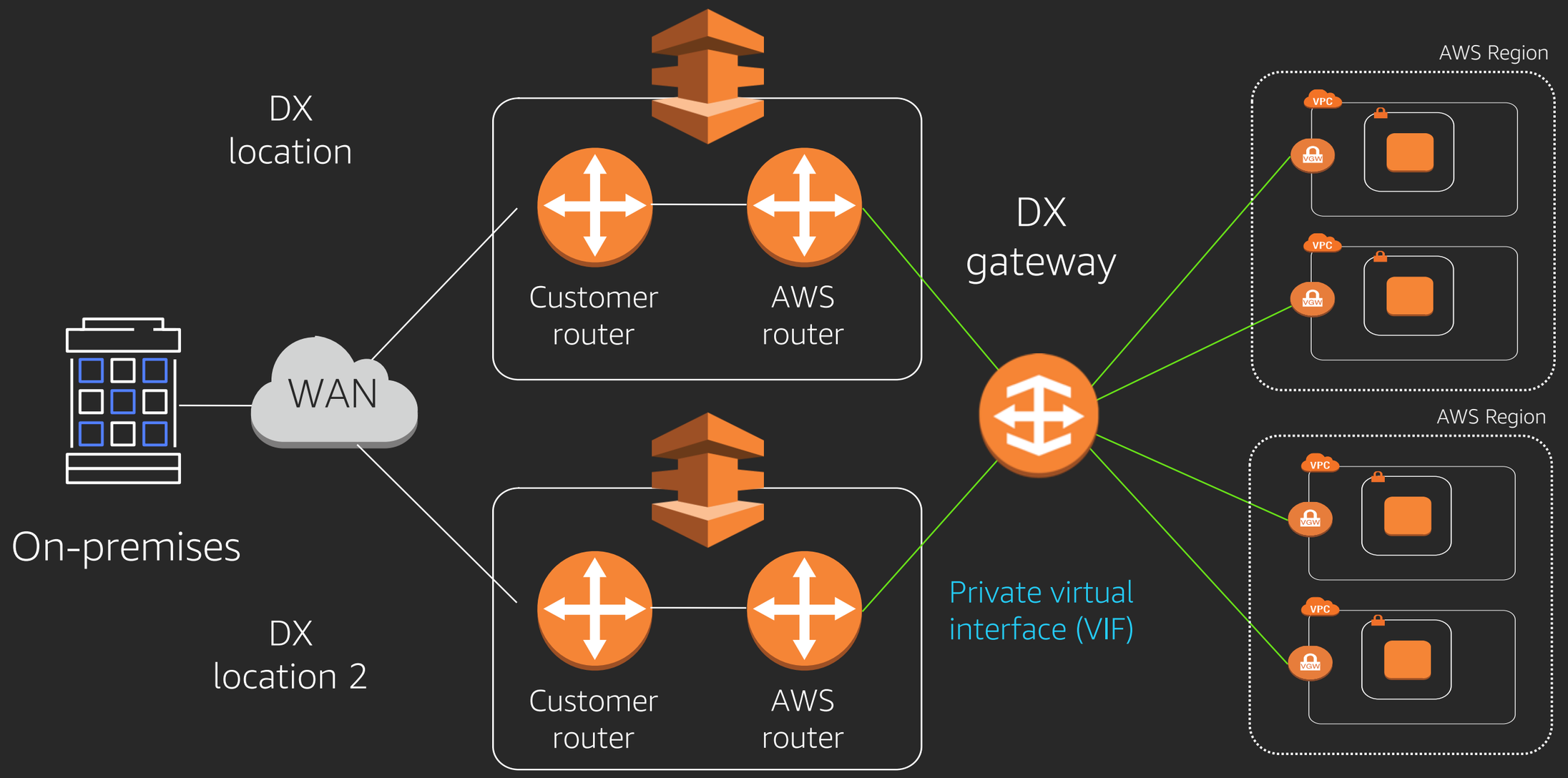
Use cases:

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

Multiple Regions

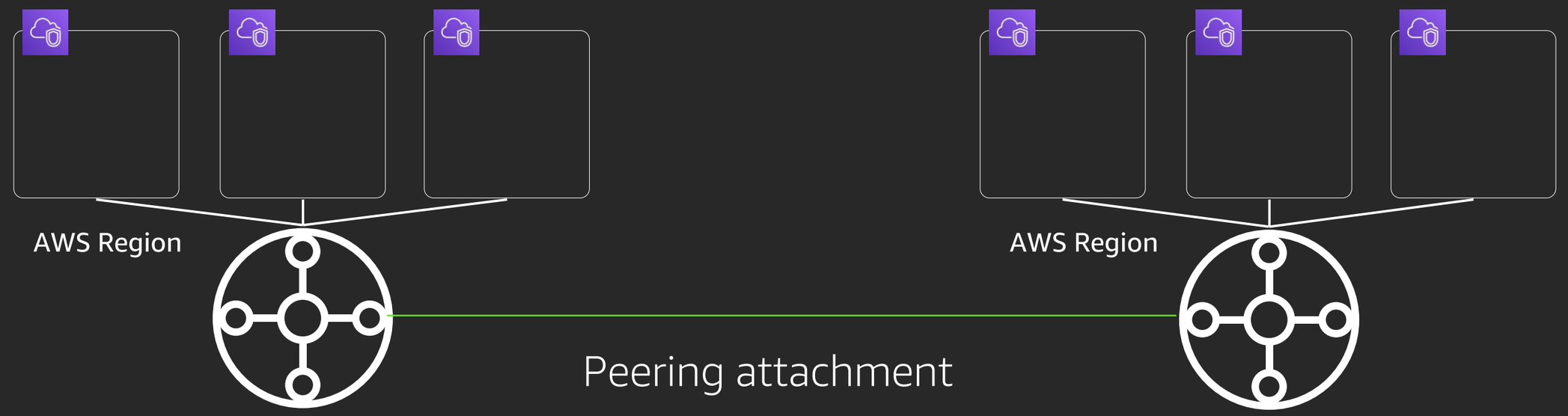


On-premises to multiple Regions



Cross-region Transit Gateway peering

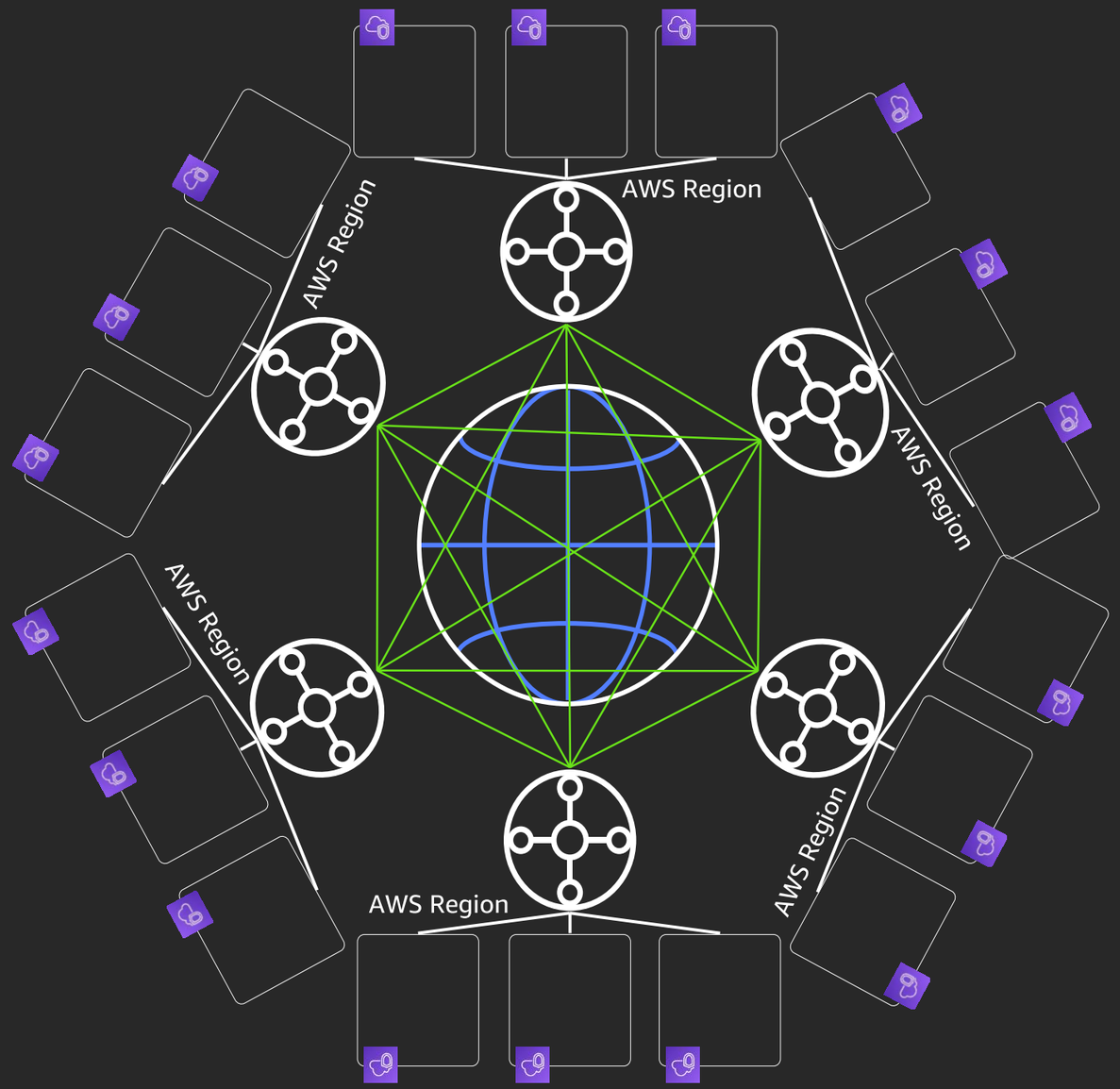
New



- 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



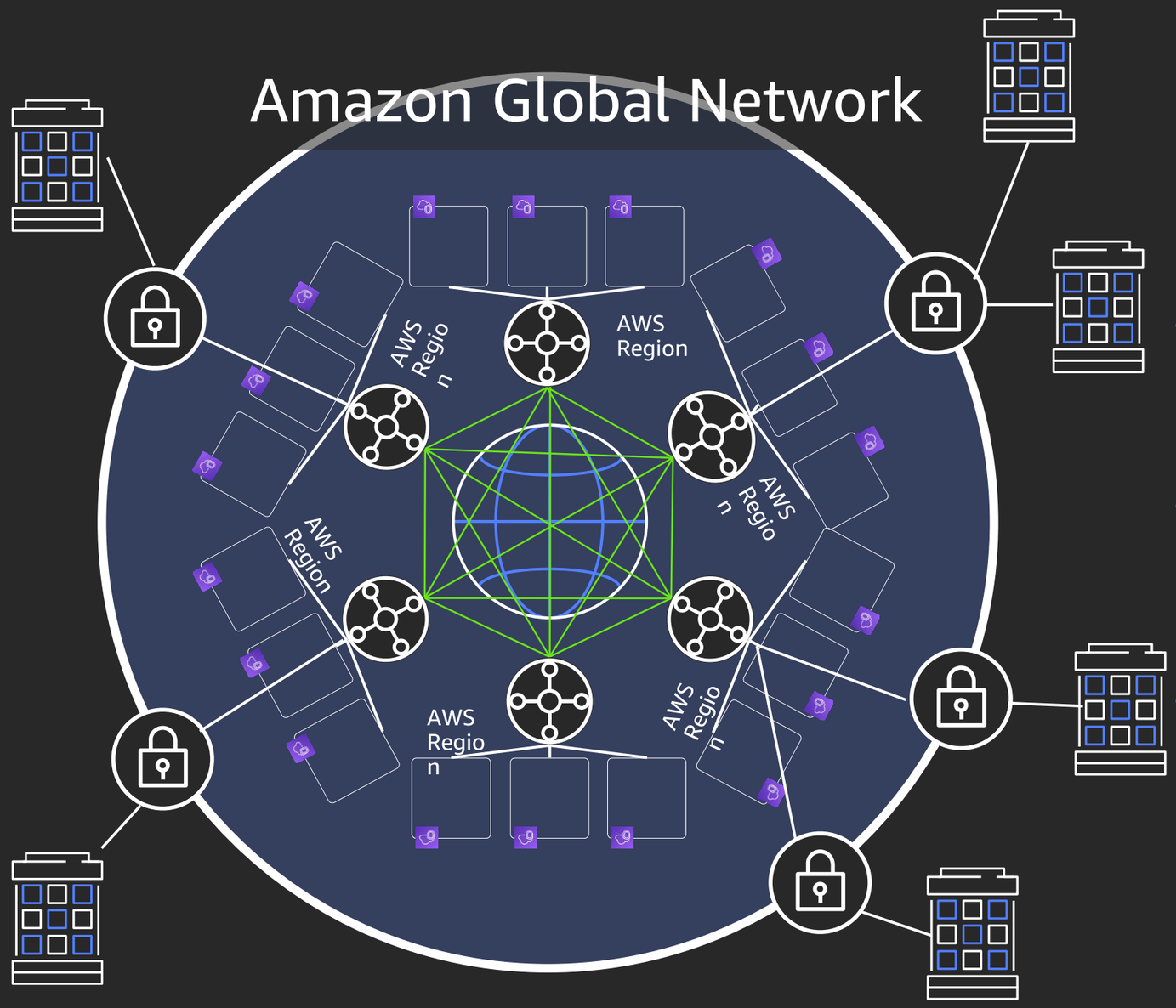
Full mesh network across multiple regions with static peering

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

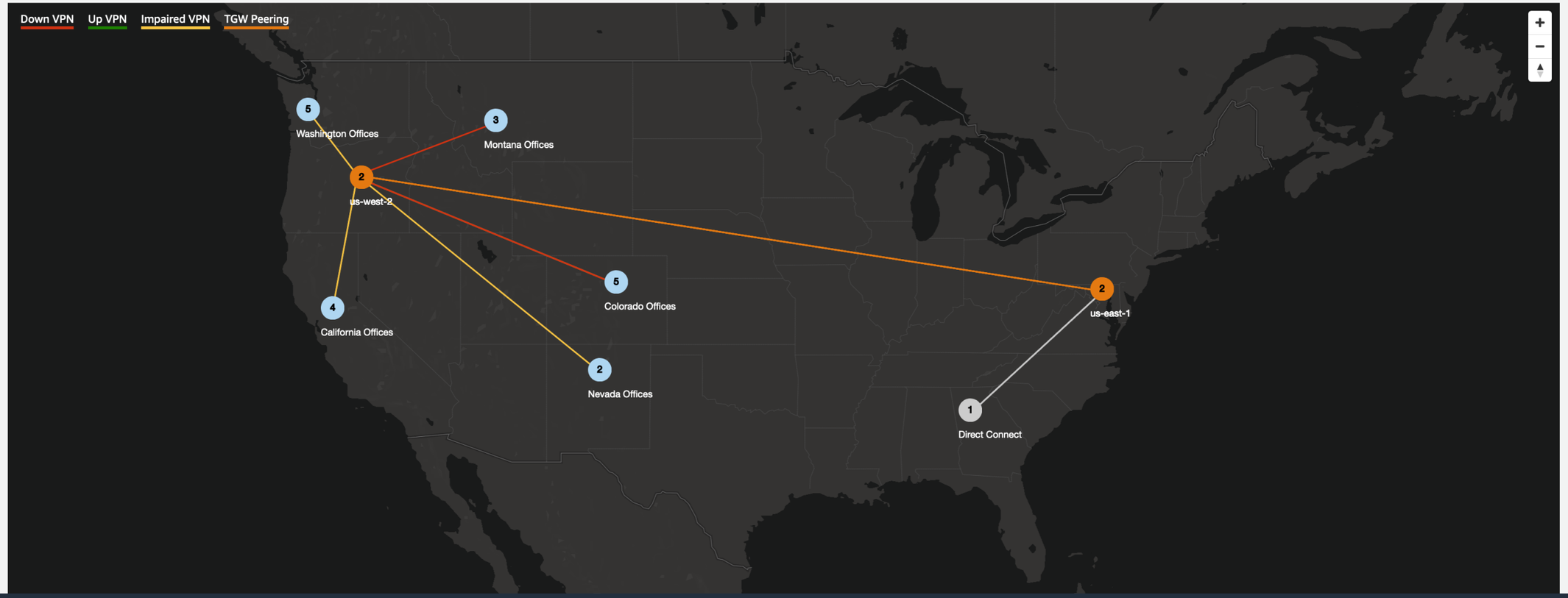
aws Services Resource Groups test_tho... 2... Select a Region Support

Network manager > Global networks > My Global Network > Geographic

Overview | Details | Geographic | Topology | Events | Monitoring

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



Cost

Costs in AWS Transit Gateway architectures

AWS Transit Gateway costs (N. Virginia):

- **$0.05/hour per attachment ~ $36.50/month**
- **$0.02 per GB data processed (sender)**

Example pricing for N. Virginia. For pricing, refer to:

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

https://aws.amazon.com/ec2/pricing/

https://aws.amazon.com/directconnect/pricing/

Notes:

- VPC peering is $0.01/GB in and out, $0.02 total. Similar to TGW.
- Ingress data has no additional cost. VPN and DX-GW attachments to TGW incur $0.02/GB data processing.
- VPN to TGW integration method is considered intra-Region public transfer, $0.01/GB each direction, same as cross-AZ transfer.
- To reduce VPC peering costs, look at using VPC sharing

Conclusions

Takeaways

We have tools and architectures that horizontally scale to many VPCs

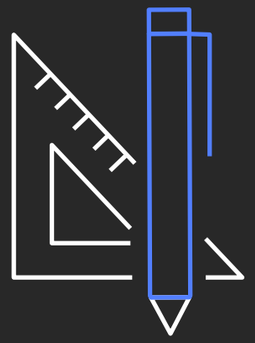
There's wiggle room for your specific use cases

Use services in combination to meet scale and security requirements

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



Learn networking with AWS Training and Certification

Resources created by the experts at AWS to help you build and validate networking skills

Free digital courses cover topics related to networking and content delivery, including Introduction to Amazon CloudFront and Introduction to Amazon VPC

Validate expertise with the
AWS Certified Advanced Networking - Specialty exam

Visit aws.amazon.com/training/paths-specialty

aws training and certification

Thank you!

Nick Matthews

@nickpowpow

AWS re:Invent

aws

Please complete the session survey in the mobile app.

AWS re:Invent

aws