AWS re:Invent

NET320-R

The right AWS network architecture for the right reason

Justin Davies
Principal Network Specialist
Amazon Web Services

Bhavin Desai
Senior Network Specialist
Amazon Web Services

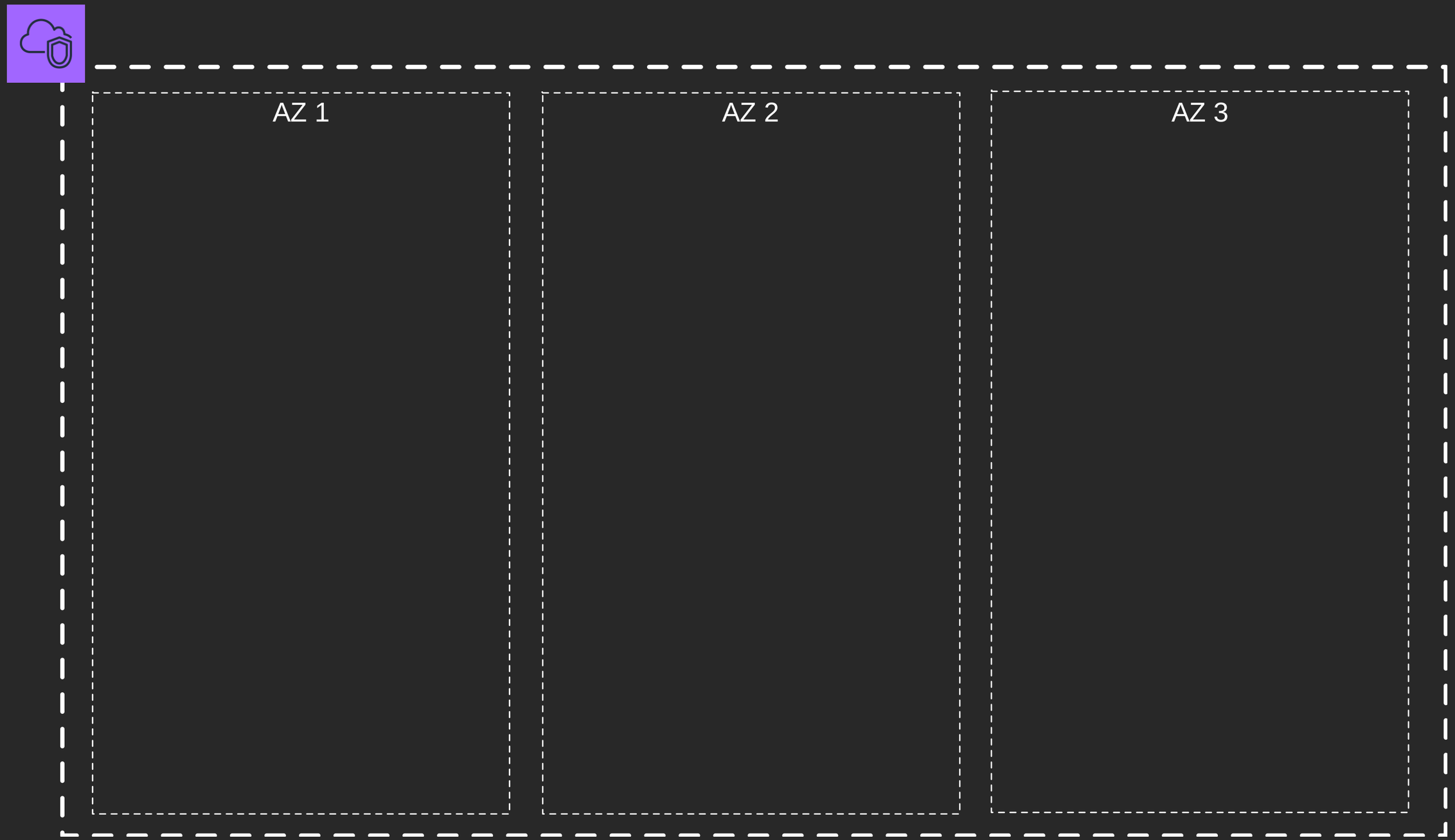
AWS re:Invent

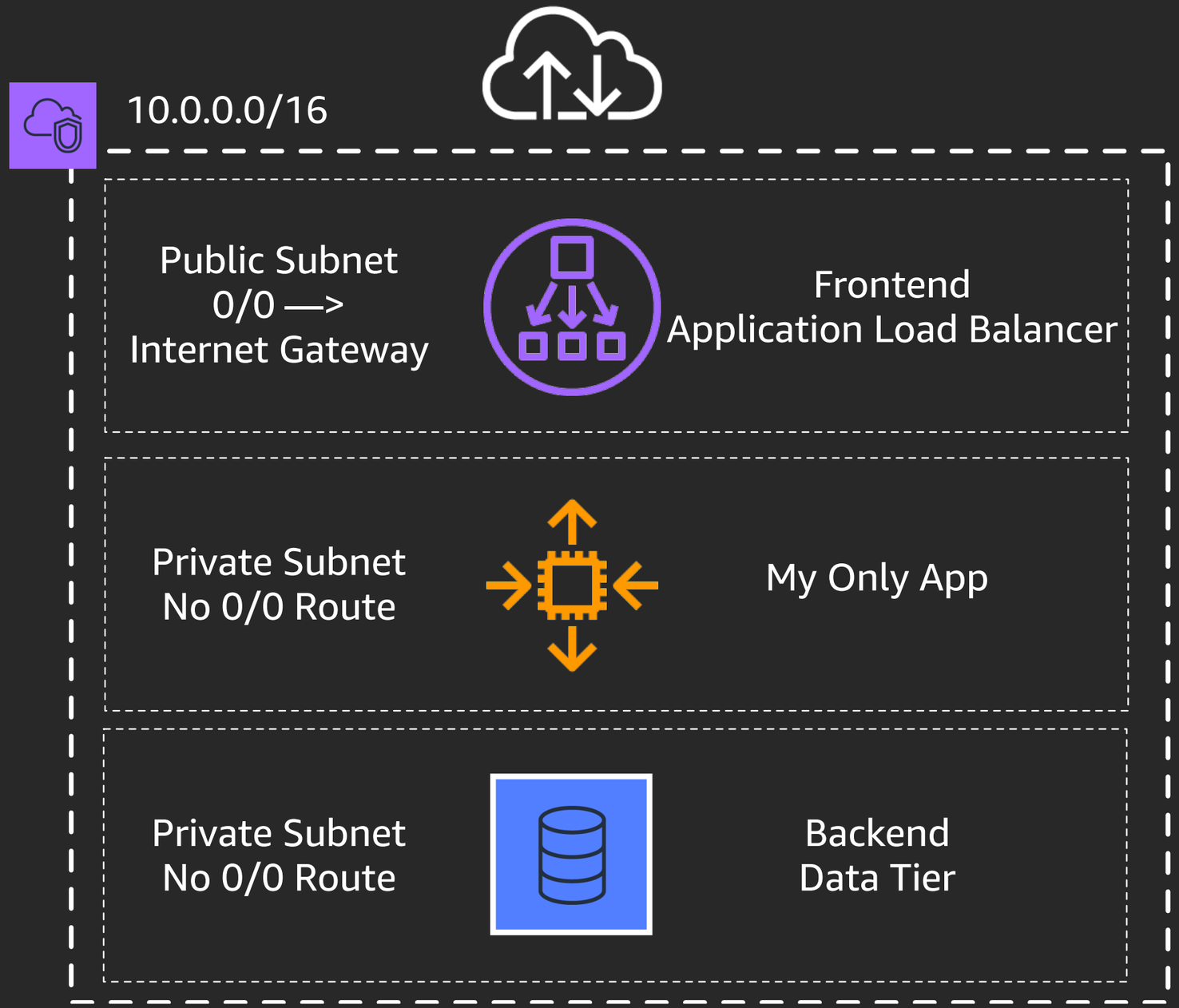
aws

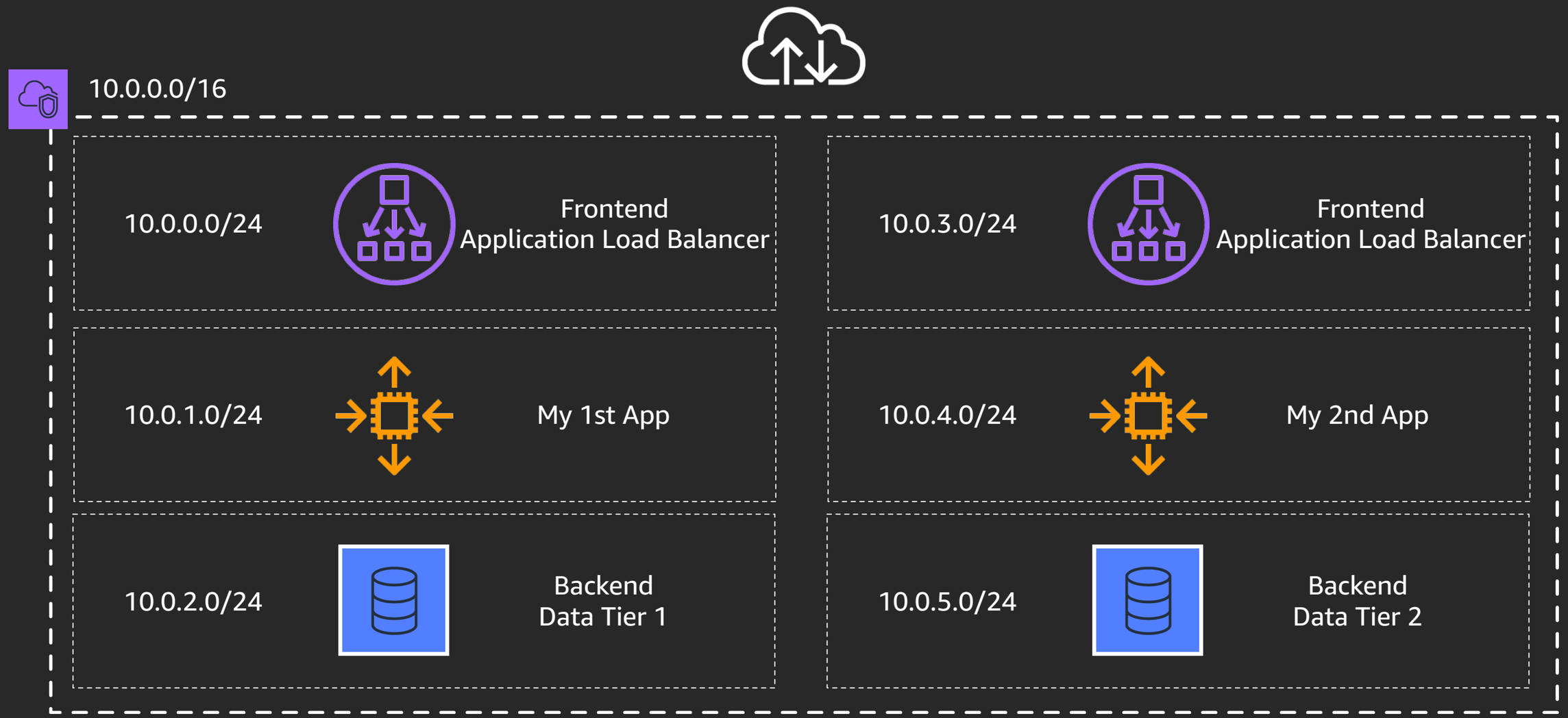
Take a picture, it lasts longer!

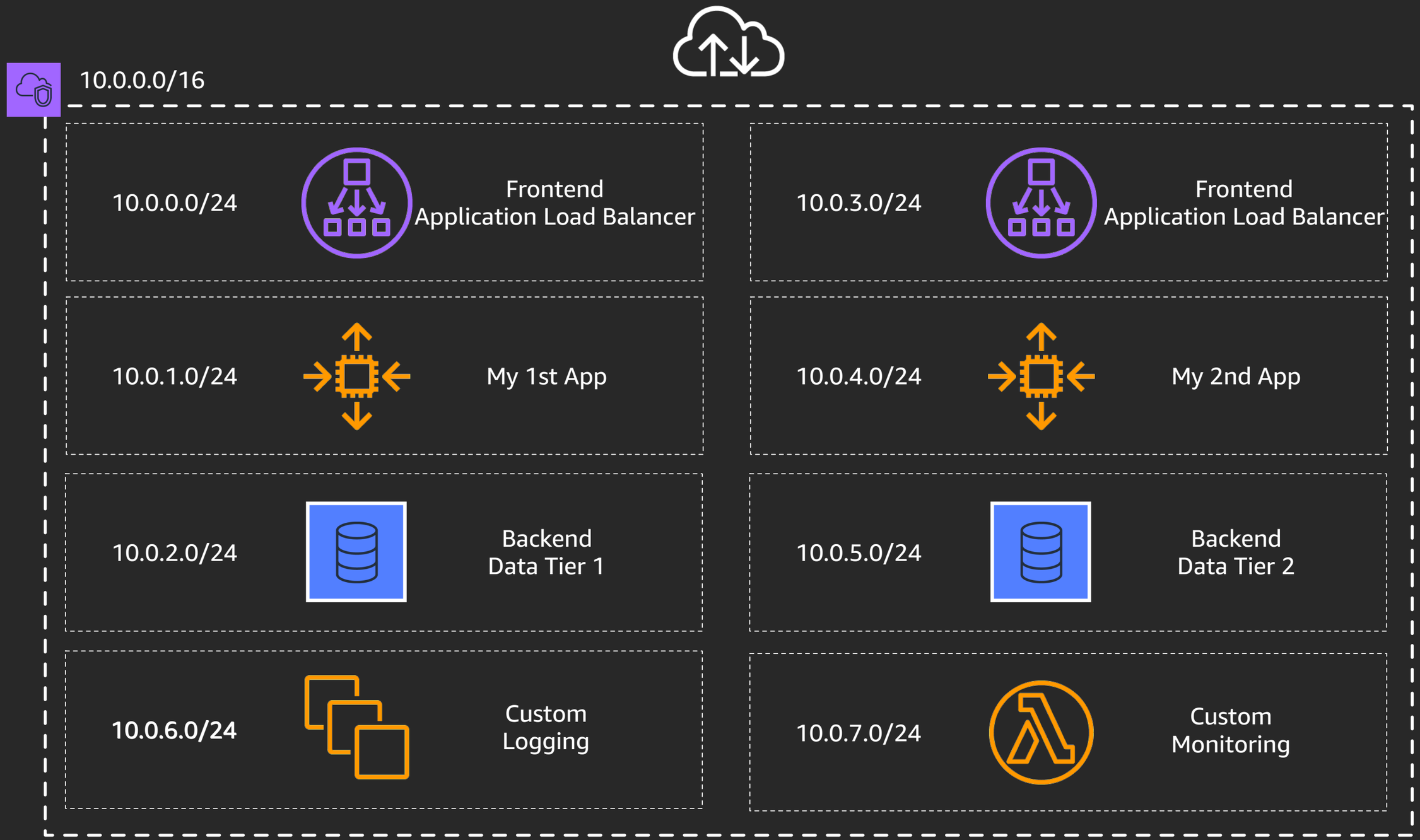
The 4 tenets for success

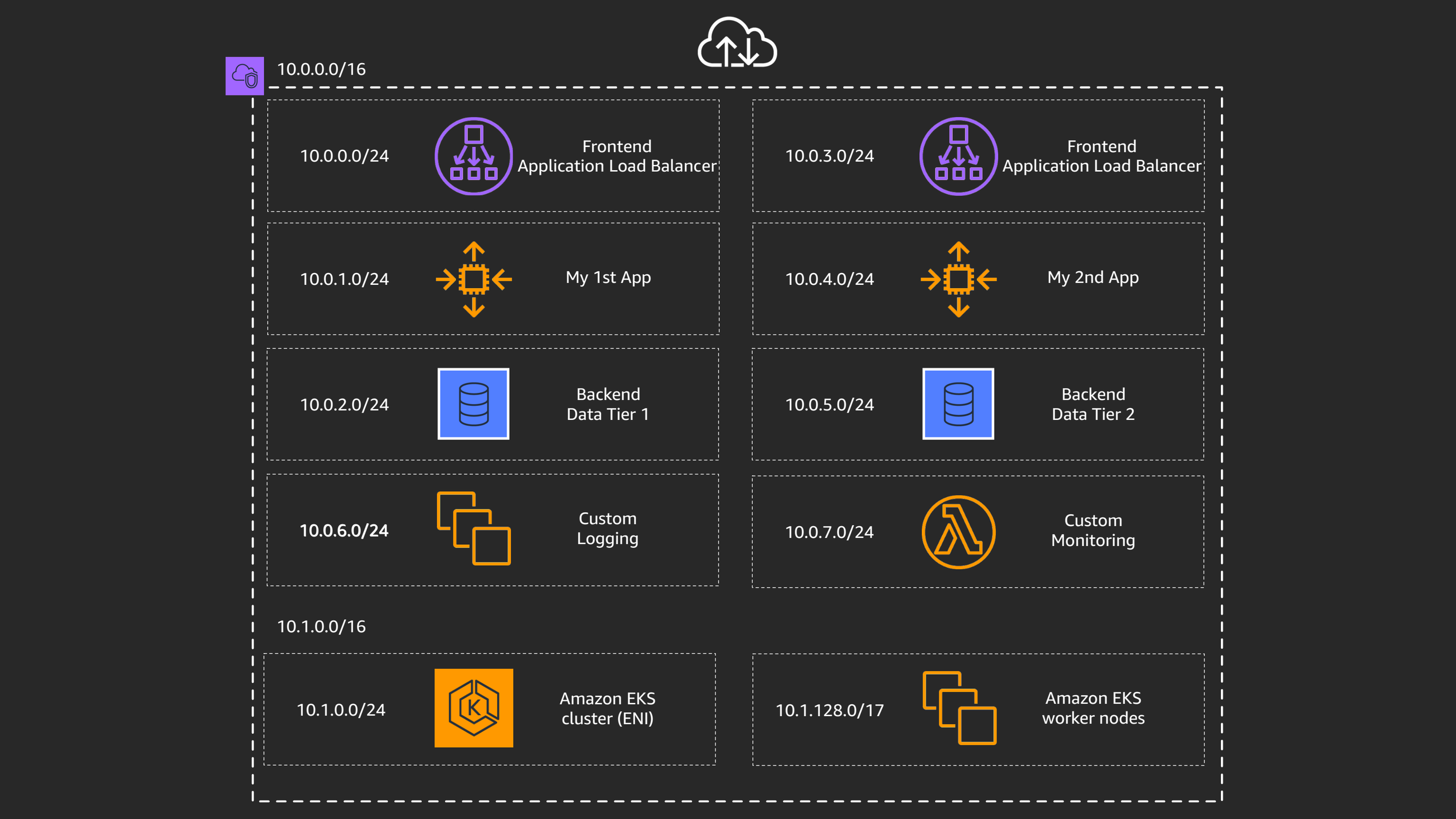
1. Key architectural considerations and design principles
2. Lessons learned & best practices
3. Crucial questions to ask
4. Technical explanations and follow-up documentation

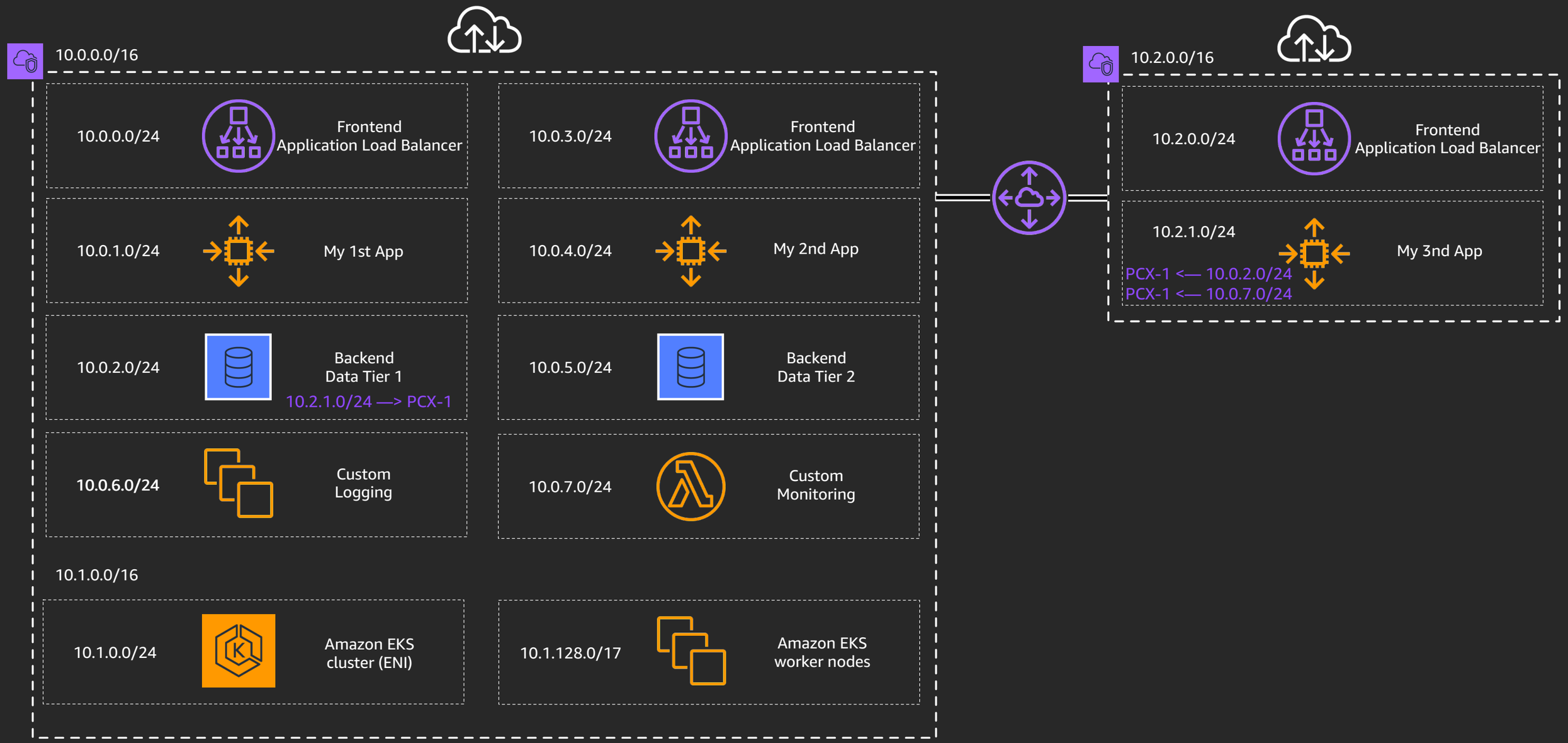

AZ 1
AZ 2
AZ 3

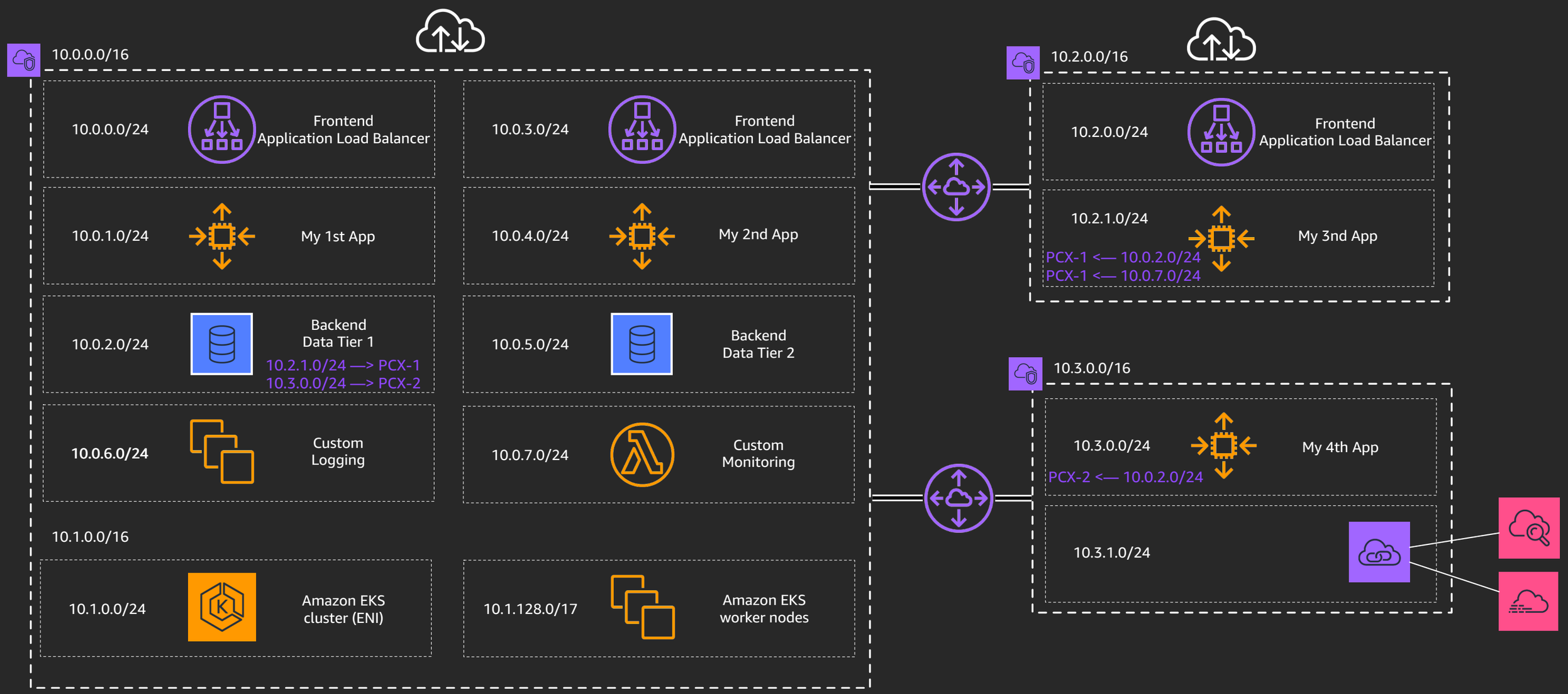

10.0.0.0/16
Public Subnet
0/0 —>
Internet Gateway
Frontend
Application Load Balancer
Private Subnet
No 0/0 Route
My Only App
Private Subnet
No 0/0 Route
Backend
Data Tier

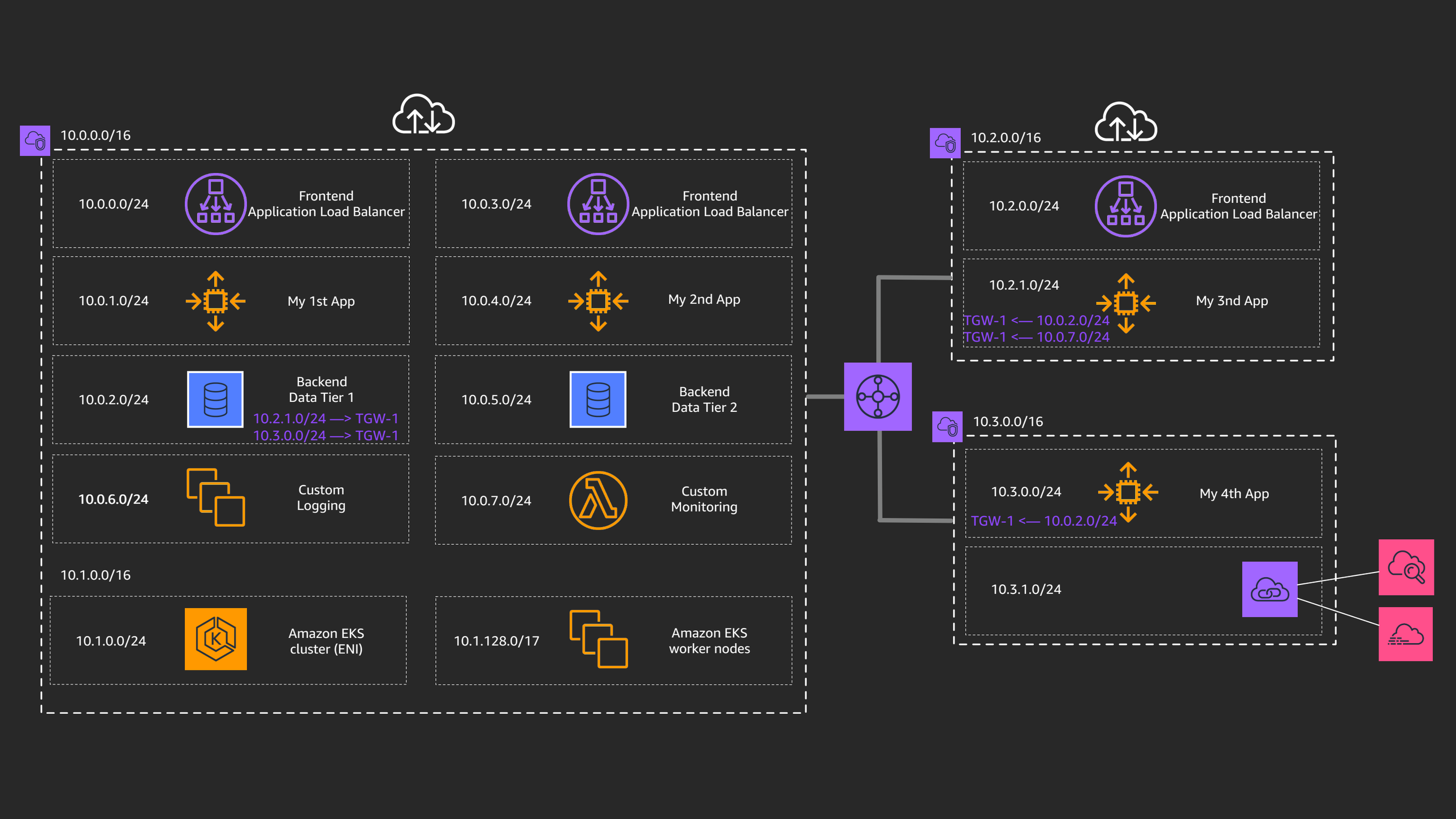

10.0.0.0/16
10.0.0.0/24
Frontend
Application Load Balancer
10.0.3.0/24
Frontend
Application Load Balancer
10.0.1.0/24
My 1st App
10.0.4.0/24
My 2nd App
10.0.2.0/24
Backend
Data Tier 1
10.0.5.0/24
Backend
Data Tier 2

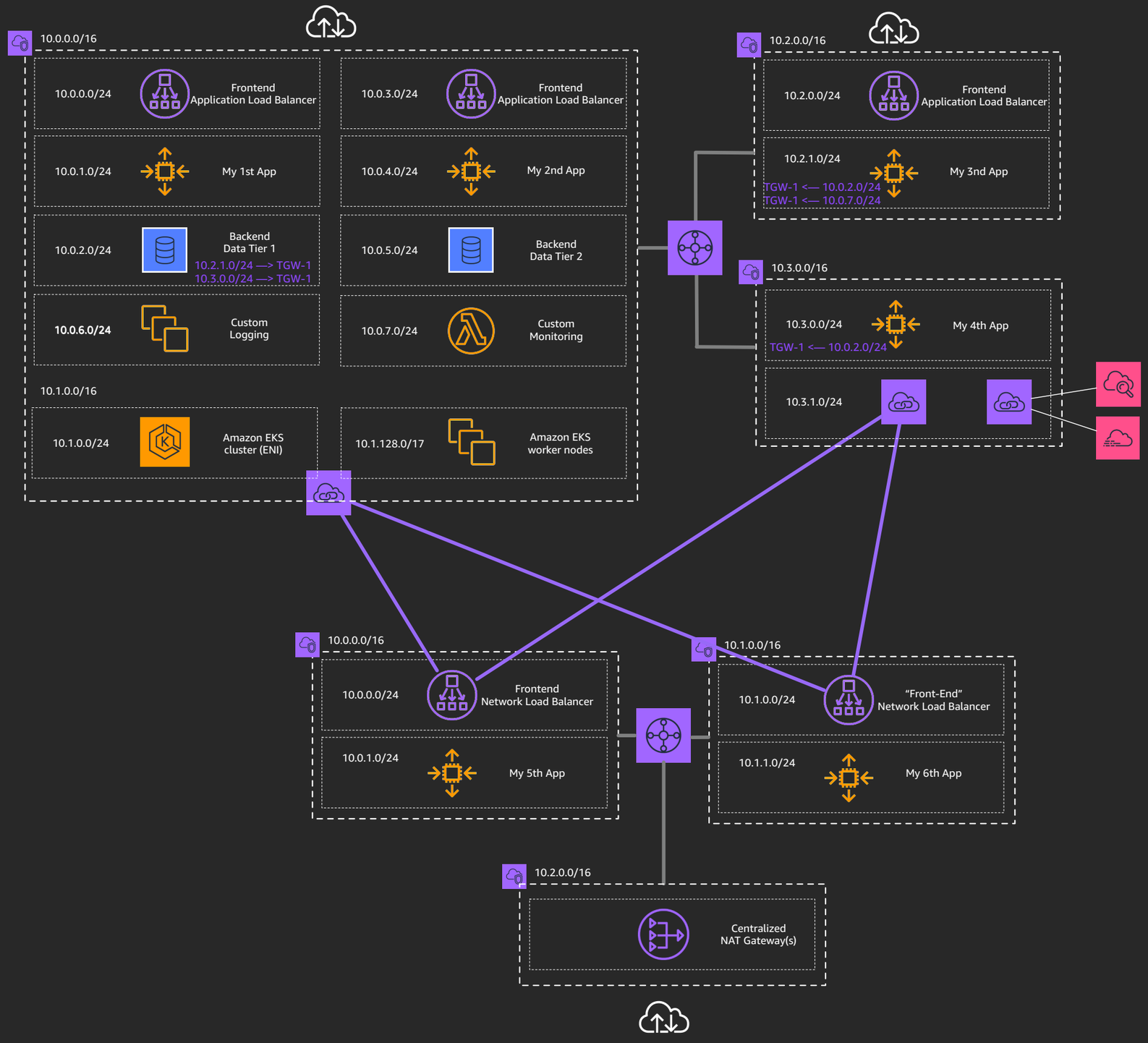

10.0.0.0/16
10.0.0.0/24
Frontend
Application Load Balancer
10.0.3.0/24
Frontend
Application Load Balancer
10.0.1.0/24
My 1st App
10.0.4.0/24
My 2nd App
10.0.2.0/24
Backend
Data Tier 1
10.0.5.0/24
Backend
Data Tier 2
10.0.6.0/24
Custom
Logging
10.0.7.0/24
Custom
Monitoring

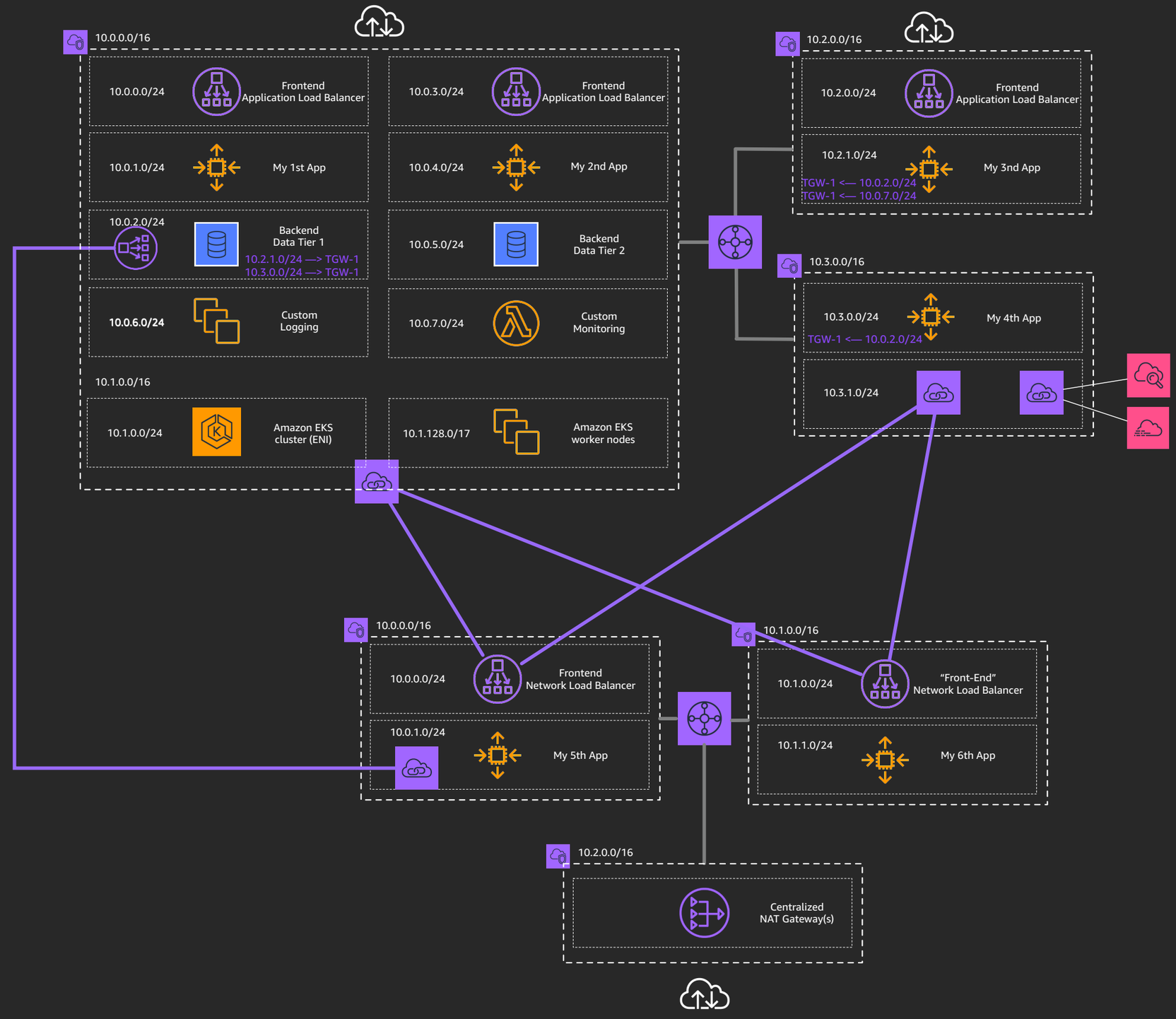

10.0.0.0/16
10.0.0.0/24
Frontend
Application Load Balancer
10.0.3.0/24
Frontend
Application Load Balancer
10.0.1.0/24
My 1st App
10.0.4.0/24
My 2nd App
10.0.2.0/24
Backend
Data Tier 1
10.0.5.0/24
Backend
Data Tier 2
10.0.6.0/24
Custom
Logging
10.0.7.0/24
Custom
Monitoring
10.1.0.0/16
10.1.0.0/24
Amazon EKS
cluster (ENI)
10.1.128.0/17
Amazon EKS
worker nodes

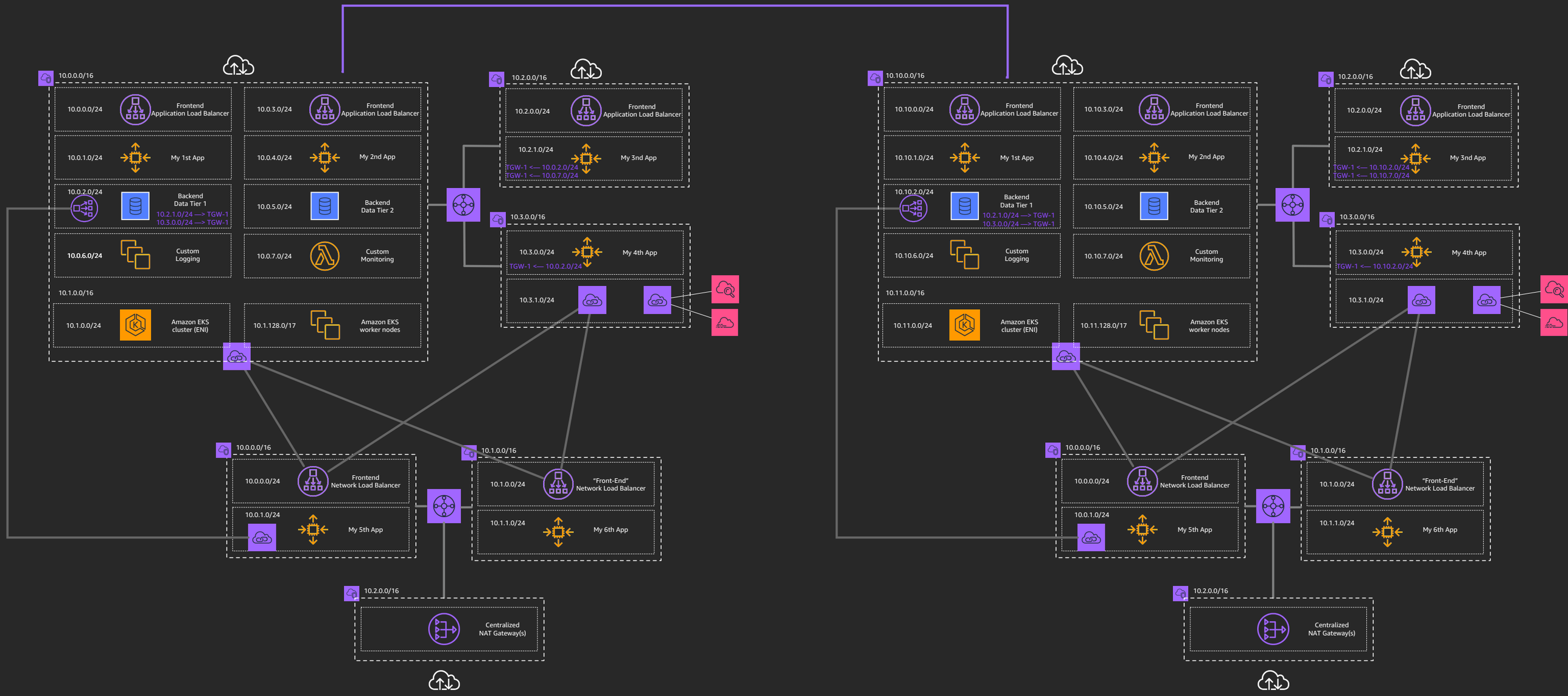

10.0.0.0/16
10.0.0.0/24
Frontend
Application Load Balancer
10.0.3.0/24
Frontend
Application Load Balancer
10.0.1.0/24
My 1st App
10.0.4.0/24
My 2nd App
10.0.2.0/24
Backend
Data Tier 1
10.2.1.0/24 —> PCX-1
10.0.5.0/24
Backend
Data Tier 2
10.0.6.0/24
Custom
Logging
10.0.7.0/24
Custom
Monitoring
10.1.0.0/16
10.1.0.0/24
Amazon EKS
cluster (ENI)
10.1.128.0/17
Amazon EKS
worker nodes
10.2.0.0/16
10.2.0.0/24
Frontend
Application Load Balancer
10.2.1.0/24
My 3nd App
PCX-1 <— 10.0.2.0/24
PCX-1 <— 10.0.7.0/24

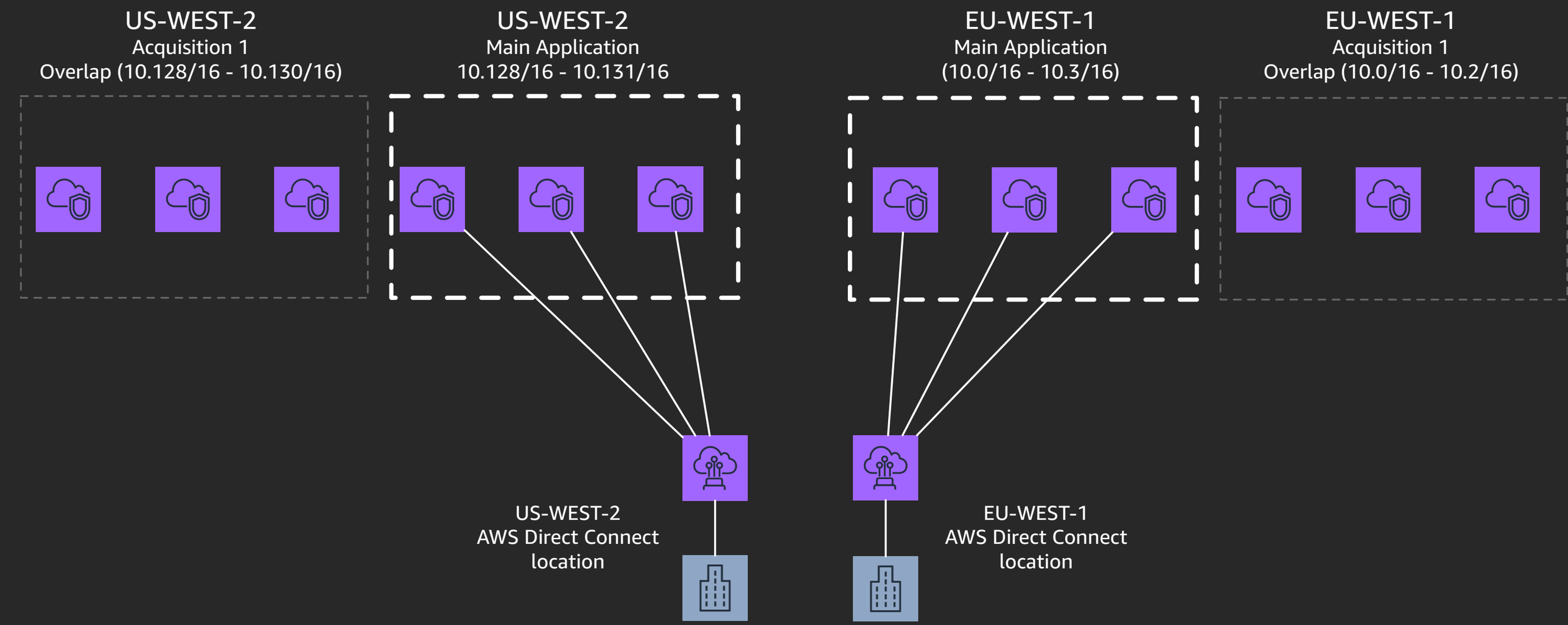

10.0.0.0/16
10.0.0.0/24
Frontend
Application Load Balancer
10.0.3.0/24
Frontend
Application Load Balancer
10.0.1.0/24
My 1st App
10.0.4.0/24
My 2nd App
10.0.2.0/24
Backend
Data Tier 1
10.2.1.0/24 —> PCX-1
10.3.0.0/24 —> PCX-2
10.0.5.0/24
Backend
Data Tier 2
10.0.6.0/24
Custom
Logging
10.0.7.0/24
Custom
Monitoring
10.1.0.0/16
10.1.0.0/24
Amazon EKS
cluster (ENI)
10.1.128.0/17
Amazon EKS
worker nodes
10.2.0.0/16
10.2.0.0/24
Frontend
Application Load Balancer
10.2.1.0/24
My 3nd App
PCX-1 <— 10.0.2.0/24
PCX-1 <— 10.0.7.0/24
10.3.0.0/16
10.3.0.0/24
My 4th App
PCX-2 <— 10.0.2.0/24
10.3.1.0/24

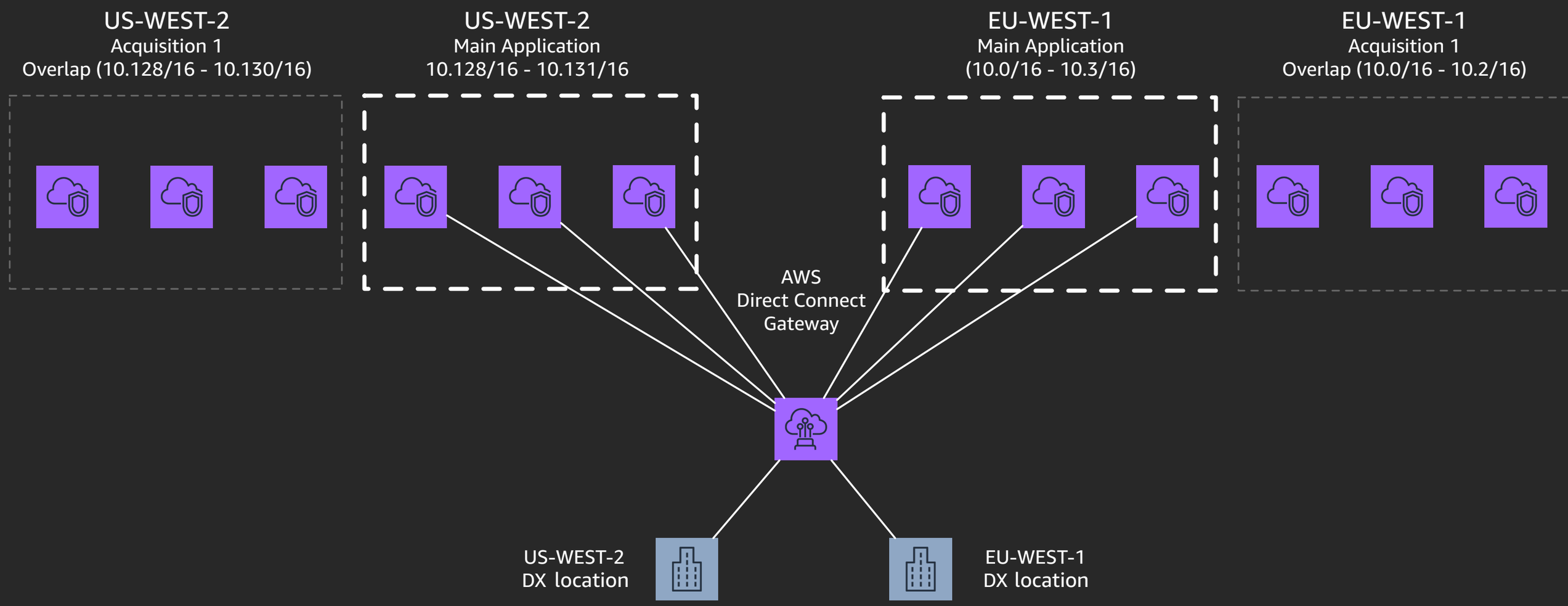

10.0.0.0/16
10.0.0.0/24
Frontend
Application Load Balancer
10.0.3.0/24
Frontend
Application Load Balancer
10.0.1.0/24
My 1st App
10.0.4.0/24
My 2nd App
10.0.2.0/24
Backend
Data Tier 1
10.2.1.0/24 —> TGW-1
10.3.0.0/24 —> TGW-1
10.0.5.0/24
Backend
Data Tier 2
10.0.6.0/24
Custom
Logging
10.0.7.0/24
Custom
Monitoring
10.1.0.0/16
10.1.0.0/24
Amazon EKS
cluster (ENI)
10.1.128.0/17
Amazon EKS
worker nodes
10.2.0.0/16
10.2.0.0/24
Frontend
Application Load Balancer
10.2.1.0/24
My 3nd App
TGW-1 <— 10.0.2.0/24
TGW-1 <— 10.0.7.0/24
10.3.0.0/16
10.3.0.0/24
My 4th App
TGW-1 <— 10.0.2.0/24
10.3.1.0/24

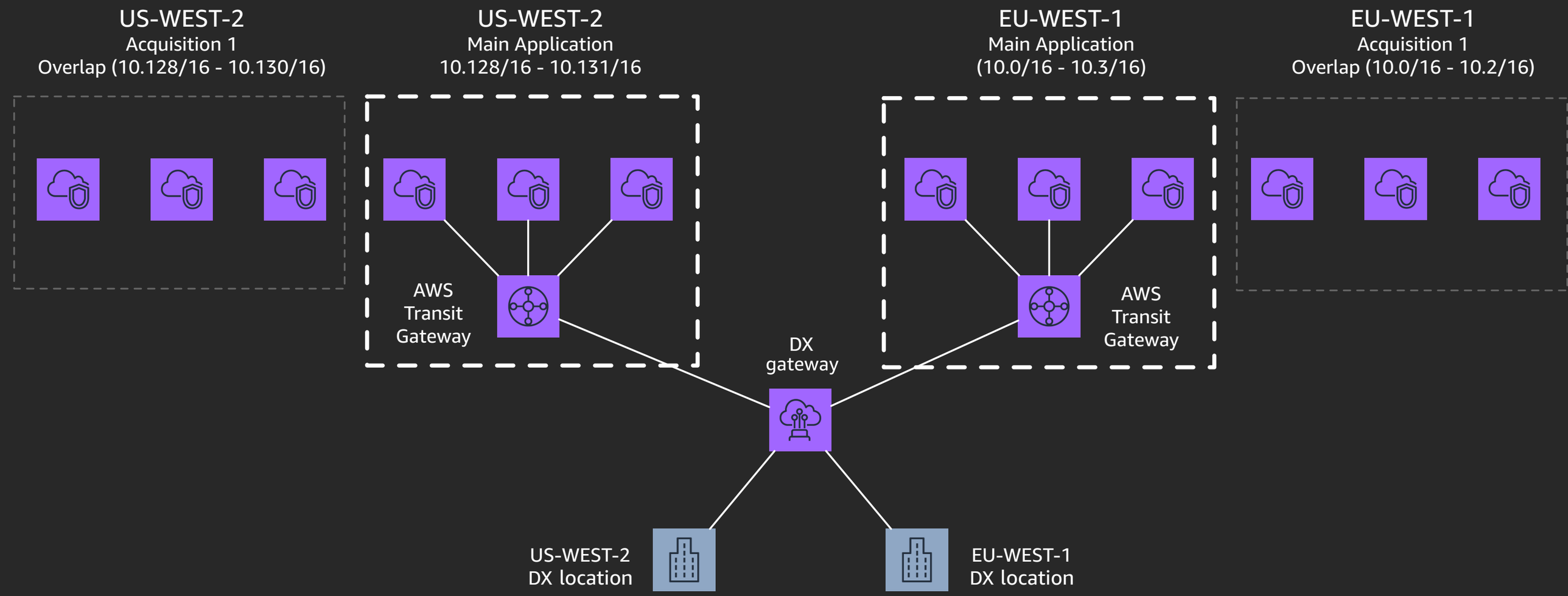

10.0.0.0/16
10.0.0.0/24
Frontend
Application Load Balancer
10.0.3.0/24
Frontend
Application Load Balancer
10.0.1.0/24
My 1st App
10.0.4.0/24
My 2nd App
10.0.2.0/24
Backend
Data Tier 1
10.2.1.0/24 —> TGW-1
10.3.0.0/24 —> TGW-1
10.0.5.0/24
Backend
Data Tier 2
10.0.6.0/24
Custom
Logging
10.0.7.0/24
Custom
Monitoring
10.1.0.0/16
10.1.0.0/24
Amazon EKS
cluster (ENI)
10.1.128.0/17
Amazon EKS
worker nodes
10.2.0.0/16
10.2.0.0/24
Frontend
Application Load Balancer
10.2.1.0/24
My 3nd App
TGW-1 <— 10.0.2.0/24
TGW-1 <— 10.0.7.0/24
10.3.0.0/16
10.3.0.0/24
My 4th App
TGW-1 <— 10.0.2.0/24
10.3.1.0/24
10.0.0.0/16
10.0.0.0/24
Frontend
Network Load Balancer
10.0.1.0/24
My 5th App
10.1.0.0/16
10.1.0.0/24
“Front-End”
Network Load Balancer
10.1.1.0/24
My 6th App
10.2.0.0/16
Centralized
NAT Gateway(s)

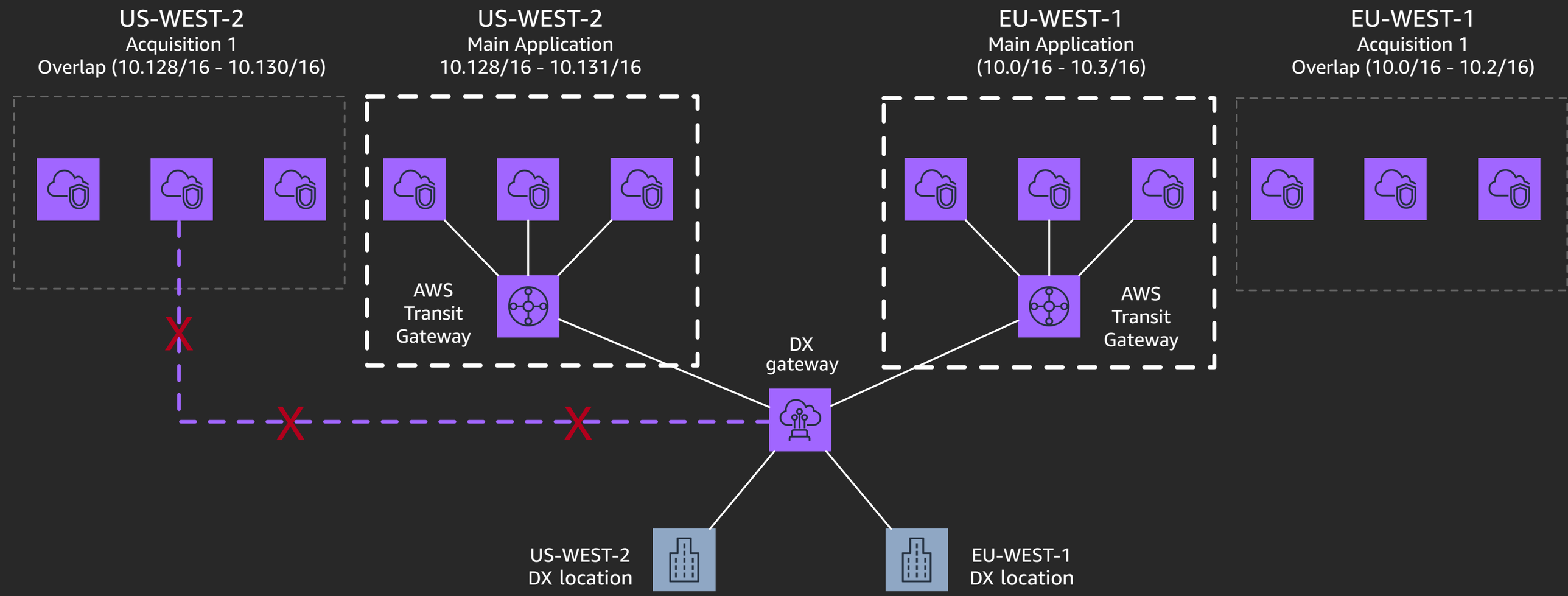

10.0.0.0/16
10.0.0.0/24
Frontend
Application Load Balancer
10.0.3.0/24
Frontend
Application Load Balancer
10.0.1.0/24
My 1st App
10.0.4.0/24
My 2nd App
10.0.2.0/24
Backend
Data Tier 1
10.2.1.0/24 —> TGW-1
10.3.0.0/24 —> TGW-1
10.0.5.0/24
Backend
Data Tier 2
10.0.6.0/24
Custom
Logging
10.0.7.0/24
Custom
Monitoring
10.1.0.0/16
10.1.0.0/24
Amazon EKS
cluster (ENI)
10.1.128.0/17
Amazon EKS
worker nodes
10.2.0.0/16
10.2.0.0/24
Frontend
Application Load Balancer
10.2.1.0/24
My 3nd App
TGW-1 <— 10.0.2.0/24
TGW-1 <— 10.0.7.0/24
10.3.0.0/16
10.3.0.0/24
My 4th App
TGW-1 <— 10.0.2.0/24
10.3.1.0/24
10.0.0.0/16
10.0.0.0/24
Frontend
Network Load Balancer
10.0.1.0/24
My 5th App
10.1.0.0/16
10.1.0.0/24
“Front-End”
Network Load Balancer
10.1.1.0/24
My 6th App
10.2.0.0/16
Centralized
NAT Gateway(s)

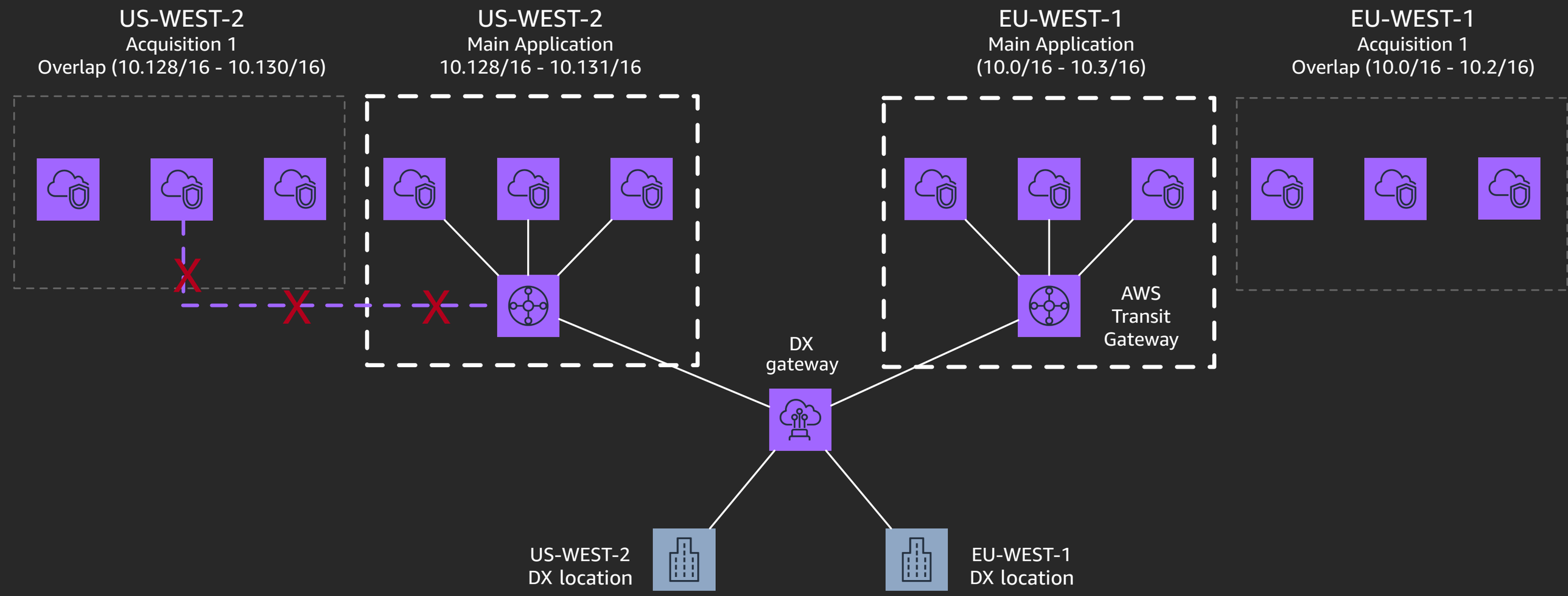

10.0.0.0/16
10.0.0.0/24
Frontend Application Load Balancer
10.0.3.0/24
Frontend Application Load Balancer
10.0.1.0/24
My 1st App
10.0.4.0/24
My 2nd App
10.0.2.0/24
Backend Data Tier 1
10.2.1.0/24 —> TGW-1
10.3.0.0/24 —> TGW-1
10.0.5.0/24
Backend Data Tier 2
10.0.6.0/24
Custom Logging
10.0.7.0/24
Custom Monitoring
10.1.0.0/16
10.1.0.0/24
Amazon EKS cluster (ENI)
10.1.128.0/17
Amazon EKS worker nodes
10.2.0.0/16
10.2.0.0/24
Frontend Application Load Balancer
10.2.1.0/24
My 3nd App
TGW-1 <— 10.0.2.0/24
TGW-1 <— 10.0.7.0/24
10.3.0.0/16
10.3.0.0/24
My 4th App
TGW-1 <— 10.0.2.0/24
10.3.1.0/24
10.0.0.0/16
10.0.0.0/24
Frontend Network Load Balancer
10.0.1.0/24
My 5th App
10.1.0.0/16
10.1.0.0/24
"Front-End" Network Load Balancer
10.1.1.0/24
My 6th App
10.2.0.0/16
Centralized NAT Gateway(s)
10.10.0.0/16
10.10.0.0/24
Frontend Application Load Balancer
10.10.3.0/24
Frontend Application Load Balancer
10.10.1.0/24
My 1st App
10.10.4.0/24
My 2nd App
10.10.2.0/24
Backend Data Tier 1
10.2.1.0/24 —> TGW-1
10.3.0.0/24 —> TGW-1
10.10.5.0/24
Backend Data Tier 2
10.10.6.0/24
Custom Logging
10.10.7.0/24
Custom Monitoring
10.11.0.0/16
10.11.0.0/24
Amazon EKS cluster (ENI)
10.11.128.0/17
Amazon EKS worker nodes
10.2.0.0/16
10.2.0.0/24
Frontend Application Load Balancer
10.2.1.0/24
My 3nd App
TGW-1 <— 10.10.2.0/24
TGW-1 <— 10.10.7.0/24
10.3.0.0/16
10.3.0.0/24
My 4th App
TGW-1 <— 10.10.2.0/24
10.3.1.0/24
10.0.0.0/16
10.0.0.0/24
Frontend Network Load Balancer
10.0.1.0/24
My 5th App
10.1.0.0/16
10.1.0.0/24
"Front-End" Network Load Balancer
10.1.1.0/24
My 6th App
10.2.0.0/16
Centralized NAT Gateway(s)

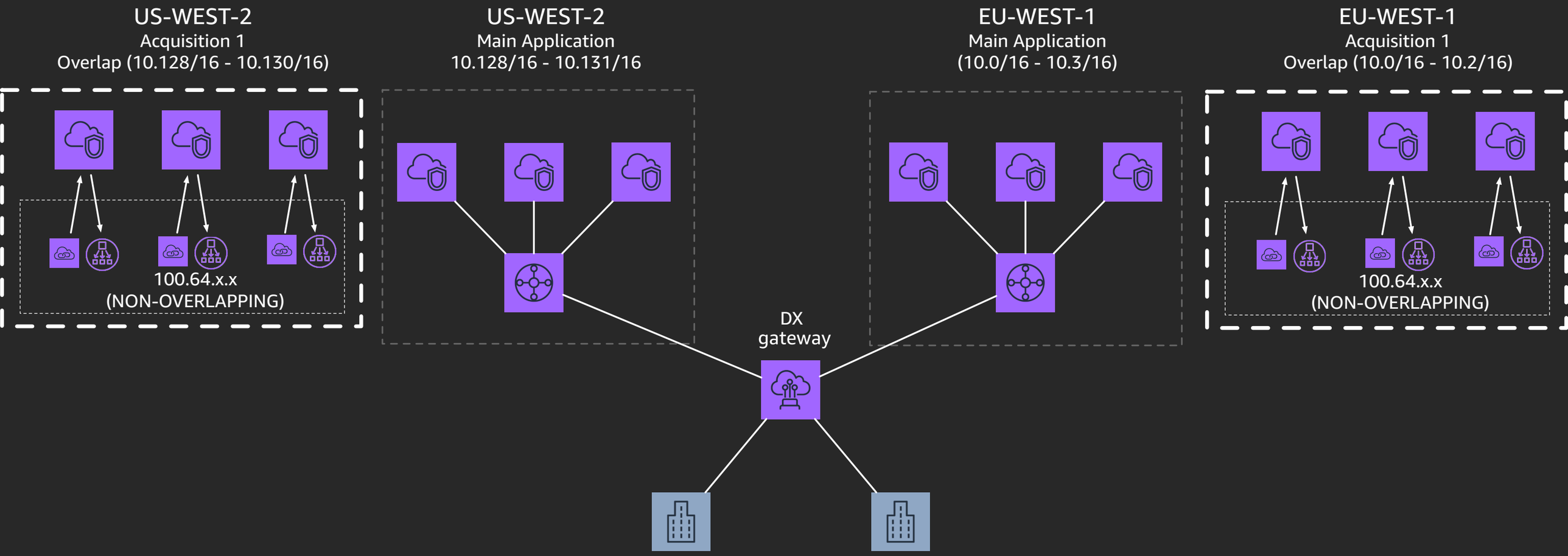

US-WEST-2
Acquisition 1
Overlap (10.128/16 - 10.130/16)
US-WEST-2
Main Application
10.128/16 - 10.131/16
EU-WEST-1
Main Application
(10.0/16 - 10.3/16)
EU-WEST-1
Acquisition 1
Overlap (10.0/16 - 10.2/16)
US-WEST-2
AWS Direct Connect
location
EU-WEST-1
AWS Direct Connect
location

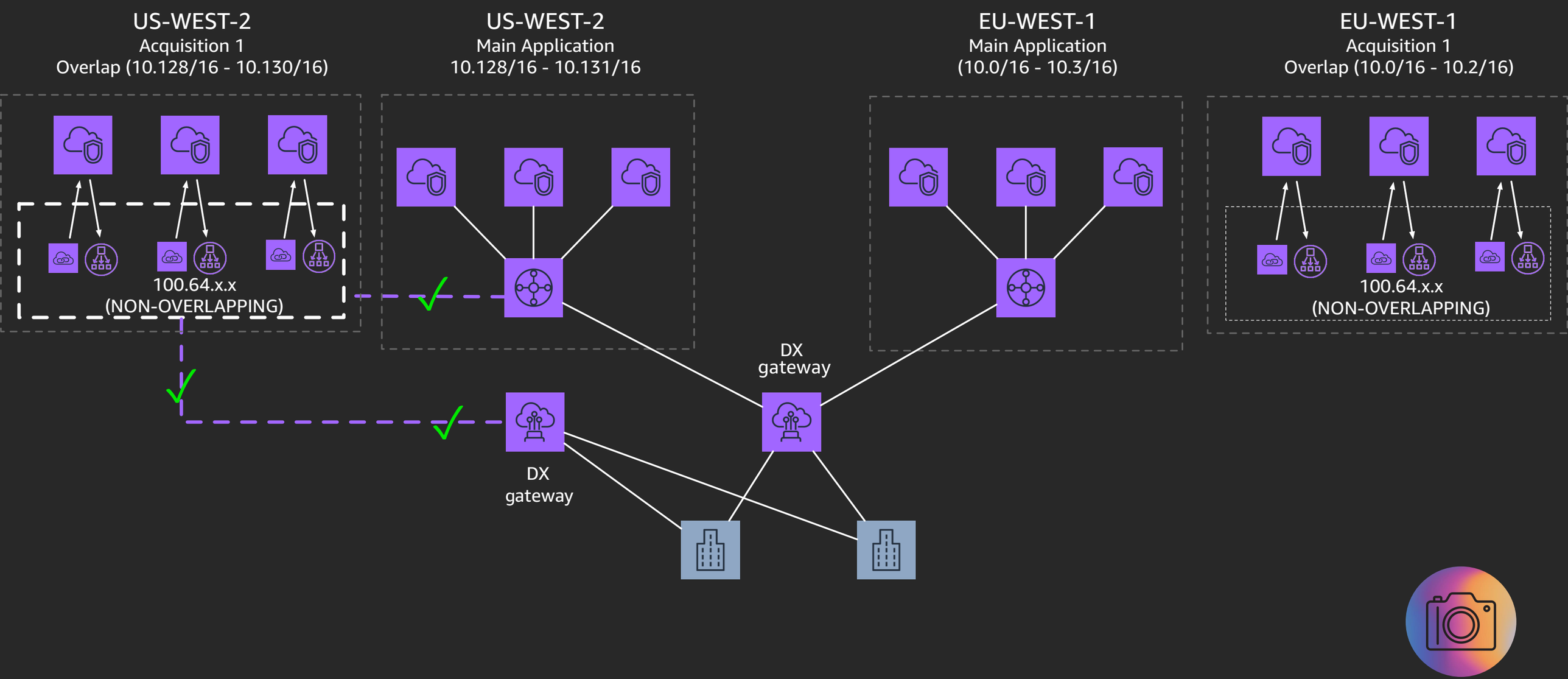

US-WEST-2
Acquisition 1
Overlap (10.128/16 - 10.130/16)
US-WEST-2
Main Application
10.128/16 - 10.131/16
EU-WEST-1
Main Application
(10.0/16 - 10.3/16)
EU-WEST-1
Acquisition 1
Overlap (10.0/16 - 10.2/16)
AWS
Direct Connect
Gateway
US-WEST-2
DX location
EU-WEST-1
DX location


US-WEST-2
Acquisition 1
Overlap (10.128/16 - 10.130/16)
US-WEST-2
Main Application
10.128/16 - 10.131/16
EU-WEST-1
Main Application
(10.0/16 - 10.3/16)
EU-WEST-1
Acquisition 1
Overlap (10.0/16 - 10.2/16)
AWS
Transit
Gateway
AWS
Transit
Gateway
DX
gateway
US-WEST-2
DX location
EU-WEST-1
DX location


US-WEST-2
Acquisition 1
Overlap (10.128/16 - 10.130/16)
US-WEST-2
Main Application
10.128/16 - 10.131/16
EU-WEST-1
Main Application
(10.0/16 - 10.3/16)
EU-WEST-1
Acquisition 1
Overlap (10.0/16 - 10.2/16)
AWS
Transit
Gateway
AWS
Transit
Gateway
DX
gateway
US-WEST-2
DX location
EU-WEST-1
DX location


US-WEST-2
Acquisition 1
Overlap (10.128/16 - 10.130/16)
US-WEST-2
Main Application
10.128/16 - 10.131/16
EU-WEST-1
Main Application
(10.0/16 - 10.3/16)
EU-WEST-1
Acquisition 1
Overlap (10.0/16 - 10.2/16)
AWS
Transit
Gateway
DX
gateway
US-WEST-2
DX location
EU-WEST-1
DX location


US-WEST-2
Acquisition 1
Overlap (10.128/16 - 10.130/16)
100.64.x.x
(NON-OVERLAPPING)
US-WEST-2
Main Application
10.128/16 - 10.131/16
EU-WEST-1
Main Application
(10.0/16 - 10.3/16)
EU-WEST-1
Acquisition 1
Overlap (10.0/16 - 10.2/16)
100.64.x.x
(NON-OVERLAPPING)
DX
gateway


US-WEST-2
Acquisition 1
Overlap (10.128/16 - 10.130/16)
US-WEST-2
Main Application
10.128/16 - 10.131/16
EU-WEST-1
Main Application
(10.0/16 - 10.3/16)
EU-WEST-1
Acquisition 1
Overlap (10.0/16 - 10.2/16)
100.64.x.x
(NON-OVERLAPPING)
100.64.x.x
(NON-OVERLAPPING)
DX
gateway
DX
gateway

"Those who never change their minds, never change anything."

Winston S. Churchill

Agenda

What does it mean to be “well-architected”?

Flat network architectures

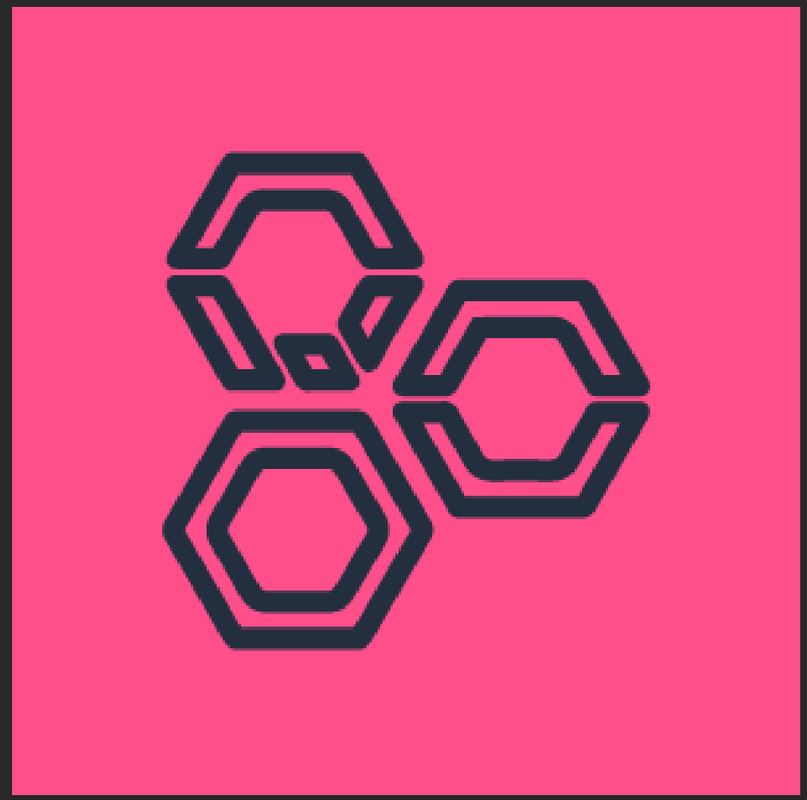
Segmented network architectures

Hybrid connectivity

How to be a solutions architect for your organization

The AWS Well-Architected Framework

Developed to help cloud architects build **secure**, **high-performing**, **resilient**, and **efficient** infrastructure for their applications . . .



The AWS Well-Architected Framework

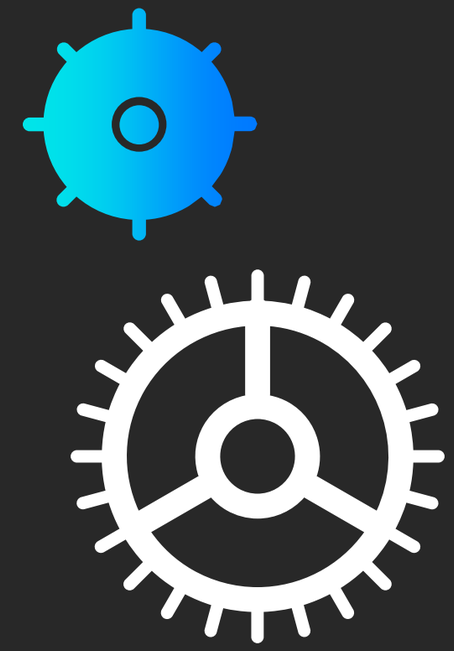
Operational excellence

Security

Reliability

Performance efficiency

Cost optimization



The AWS Well-Architected Framework

Operational excellence

Security

Reliability

Performance efficiency

Cost optimization

The AWS Well-Architected Framework

Operational excellence

Security

Reliability

Performance efficiency

Cost optimization

The AWS Well-Architected Framework

Operational excellence

Security

Reliability

Performance efficiency

Cost optimization

The AWS Well-Architected Framework

Operational excellence

Security

Reliability

Performance efficiency

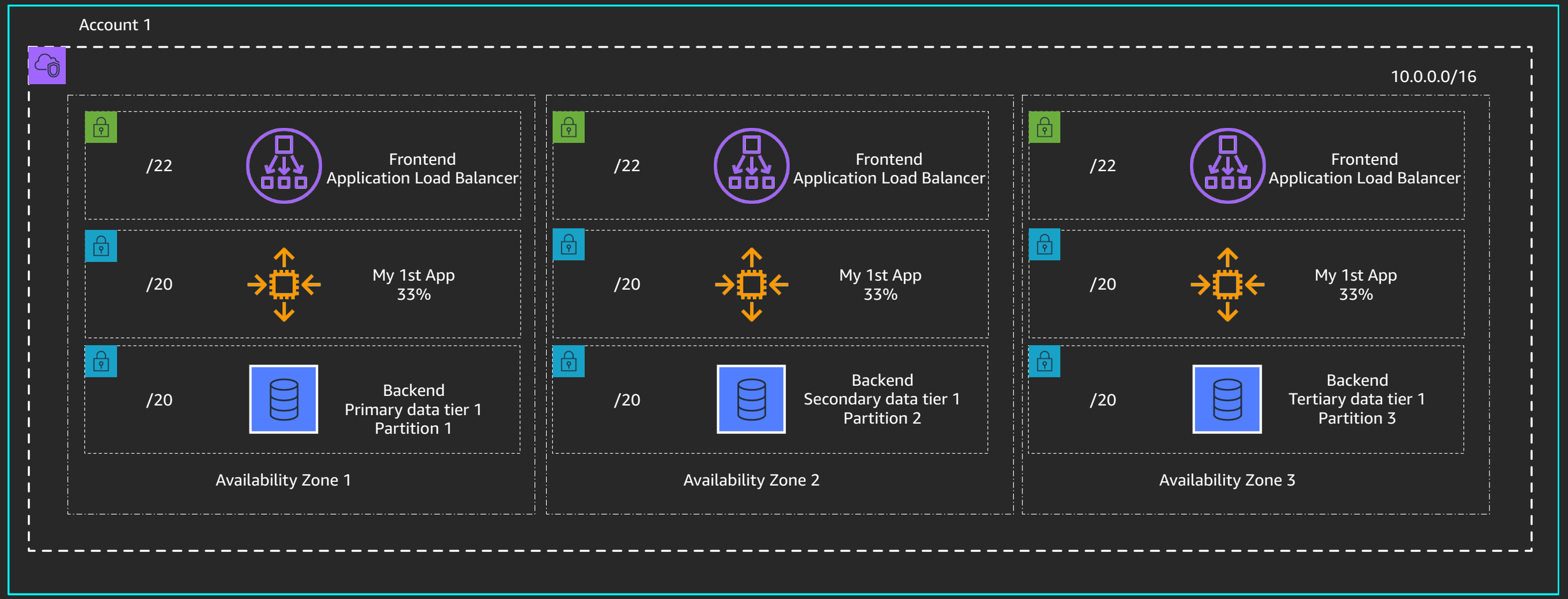
Cost optimization

Flat network architectures:
Single Amazon Virtual Private Cloud (Amazon VPC)

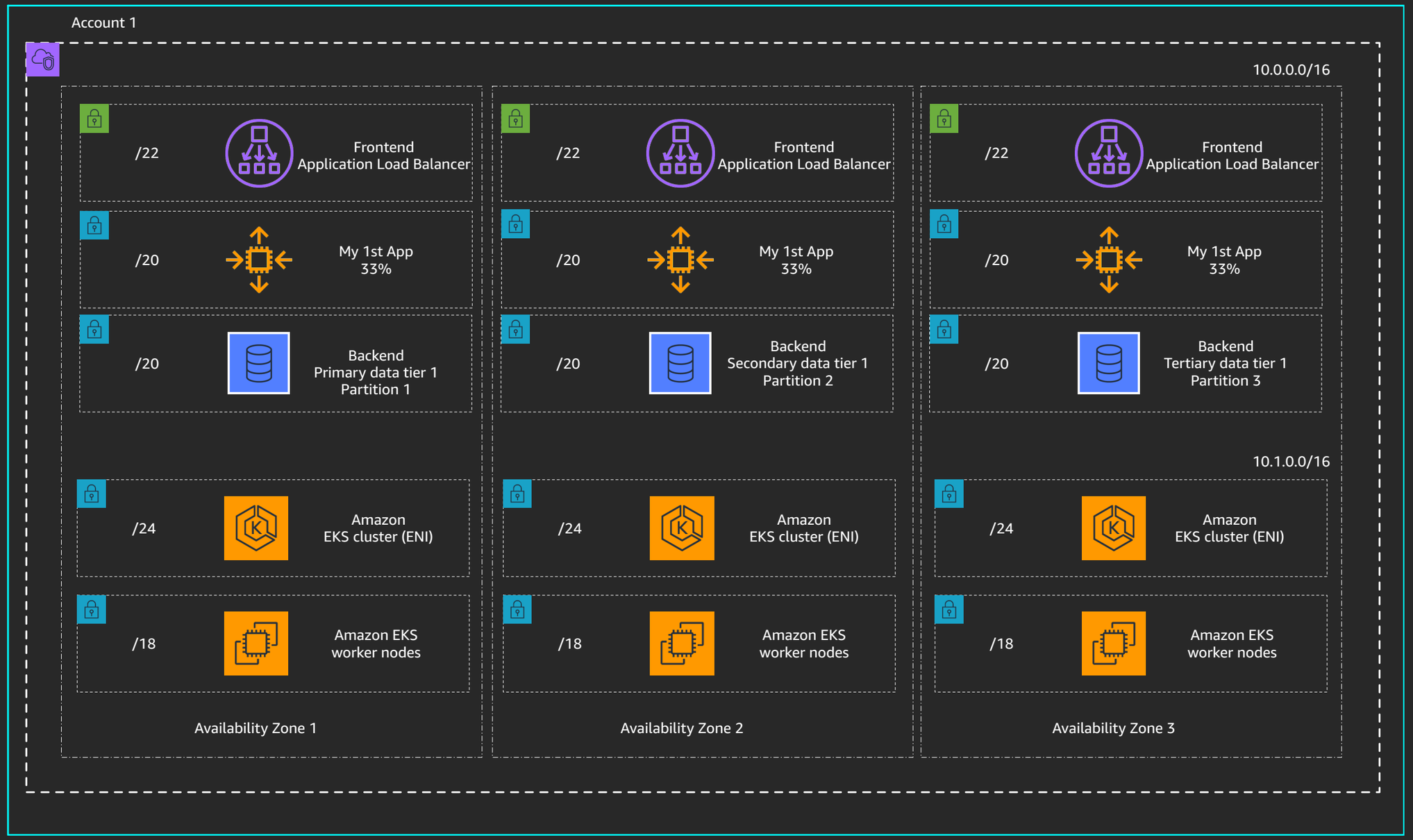
AWS re:Invent

aws

Single VPC—flat(ish) VPC



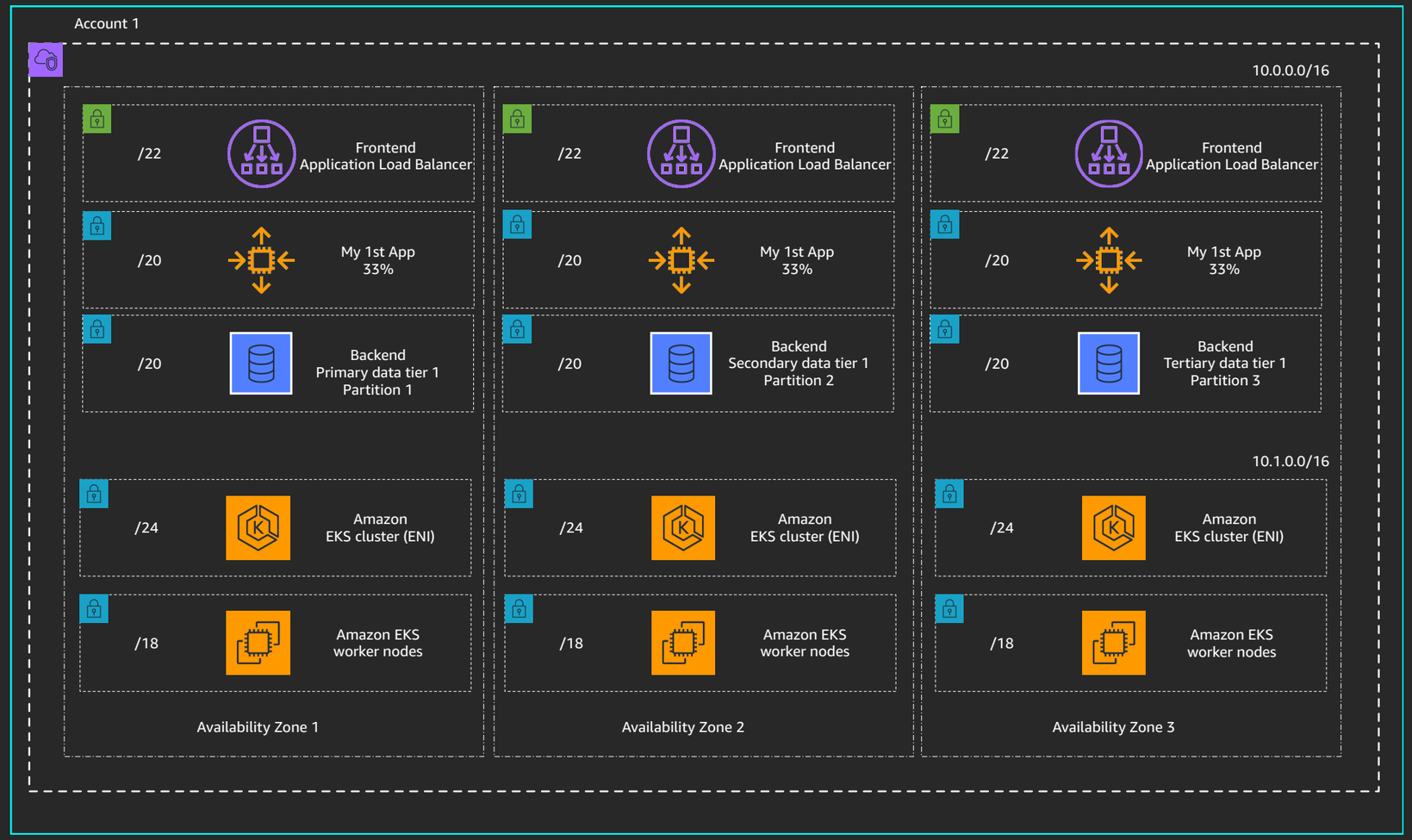
Single VPC—with multiple CIDRs



Is a single-VPC architecture right for me?

The good . . .

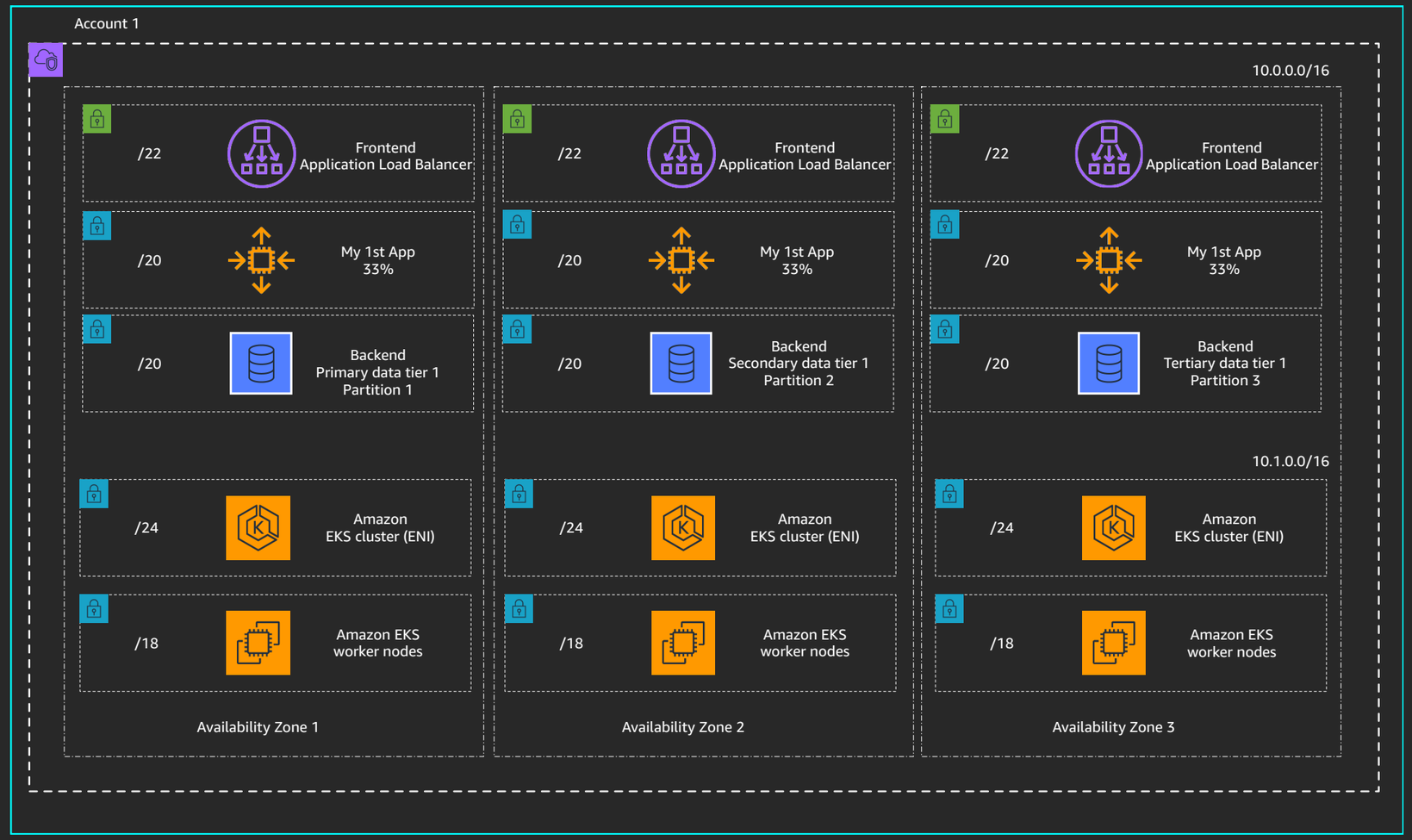
- Simple setup
- Flat network
- No inter-VPC cost
- 5 * /16 CIDRs ~327,680 IPs
- Segmentation with route tables and NACLs



Is a single-VPC architecture right for me?

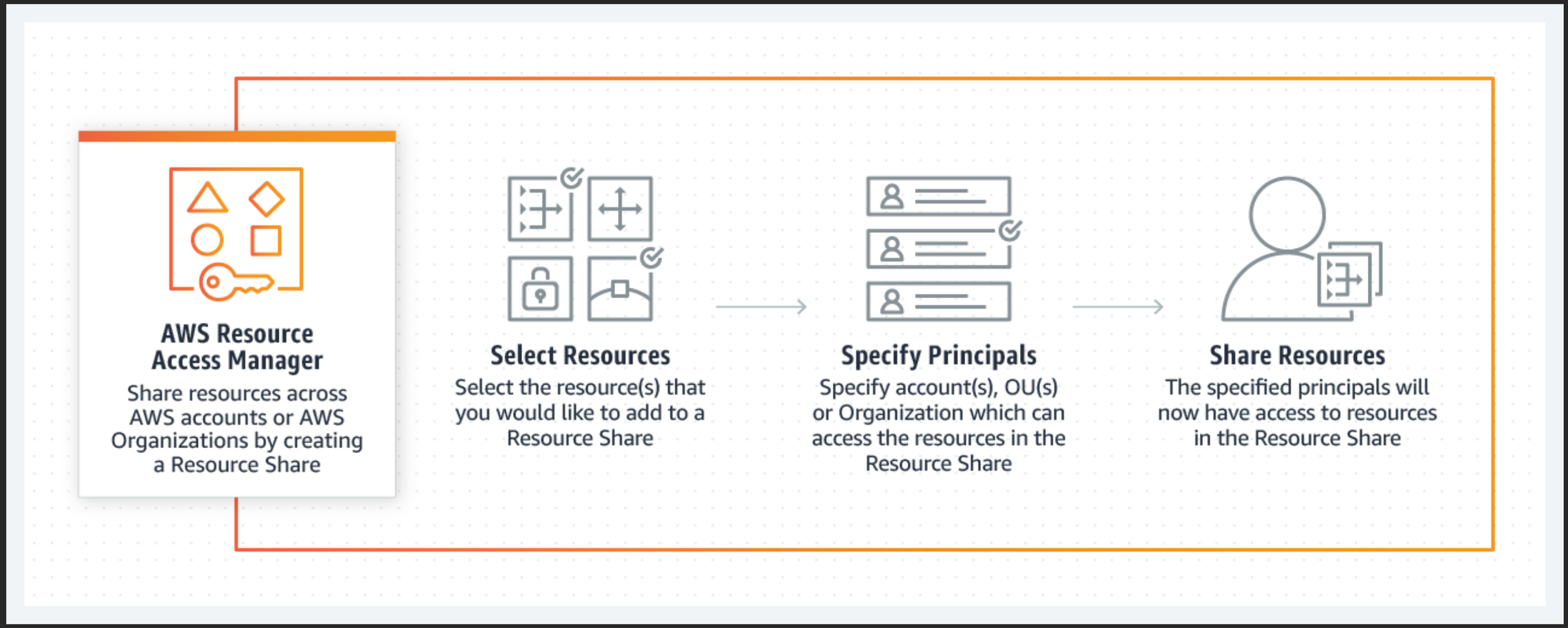
The bad . . .

- Complexity at scale
- Shared service limits
- Blast radius
- Policy complexity
- Cost allocation complexity



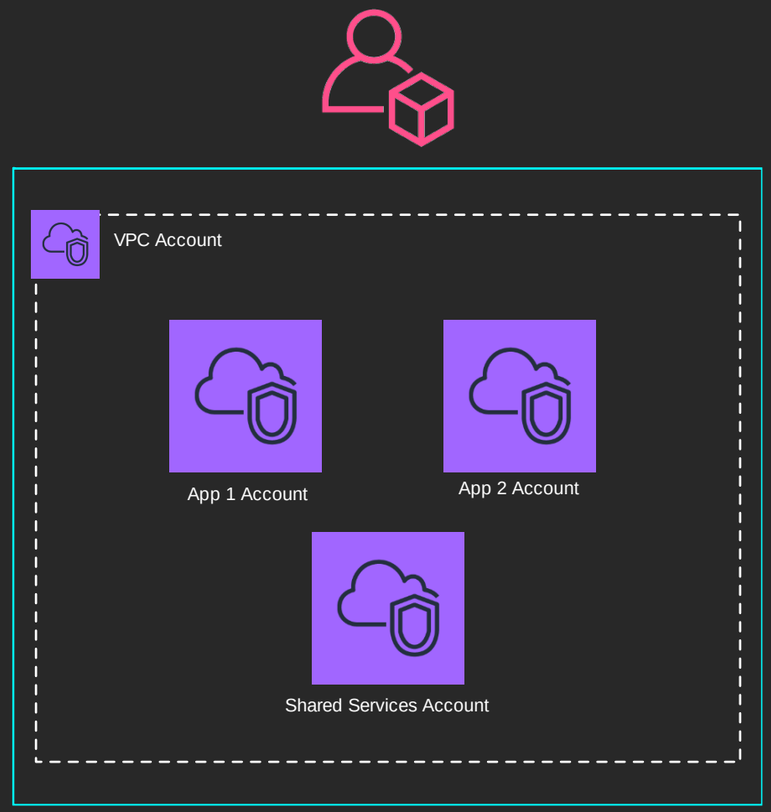
AWS Resource Access Manager

Share AWS resources between accounts
within your AWS Organization



Single VPC—with Resource Sharing “shared VPC”

- VPC
- Subnets & route tables
- Network ACLs
- AWS PrivateLink and VPC endpoints
- VPC peering
- AWS Transit Gateway attachments
- Virtual private gateway
- NAT gateway
- Internet gateway



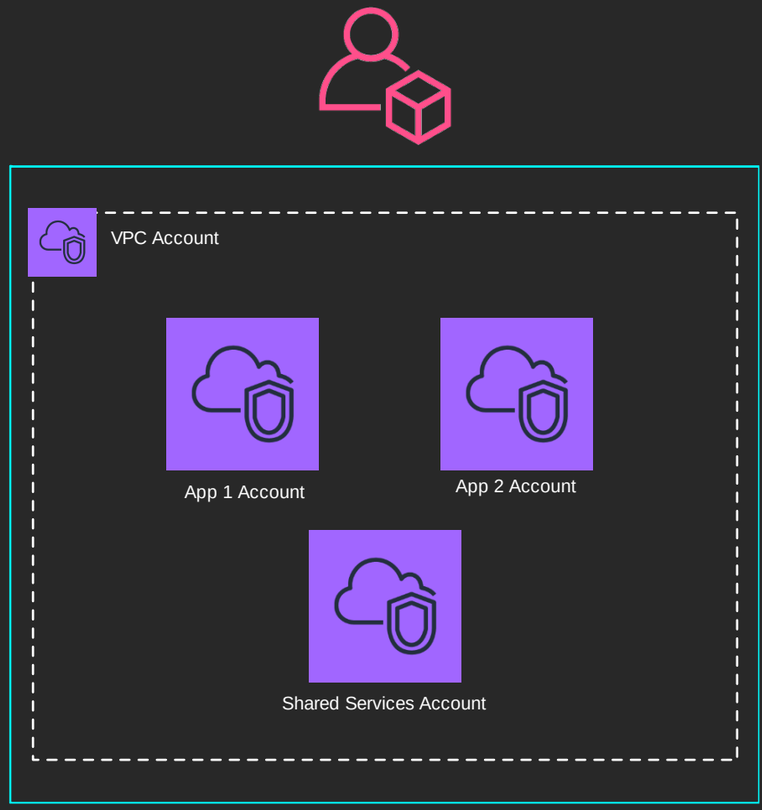
Owner Permissions

Single VPC—with Resource Sharing “shared VPC”

EC2 instances

Databases & managed services

Elastic Load Balancing



Participant Permissions

Single VPC—with Resource Sharing “shared VPC”

VPC Account

10.0.0.0/16

/22 /22 /22

/20 /20 /20

/20 /20 /20

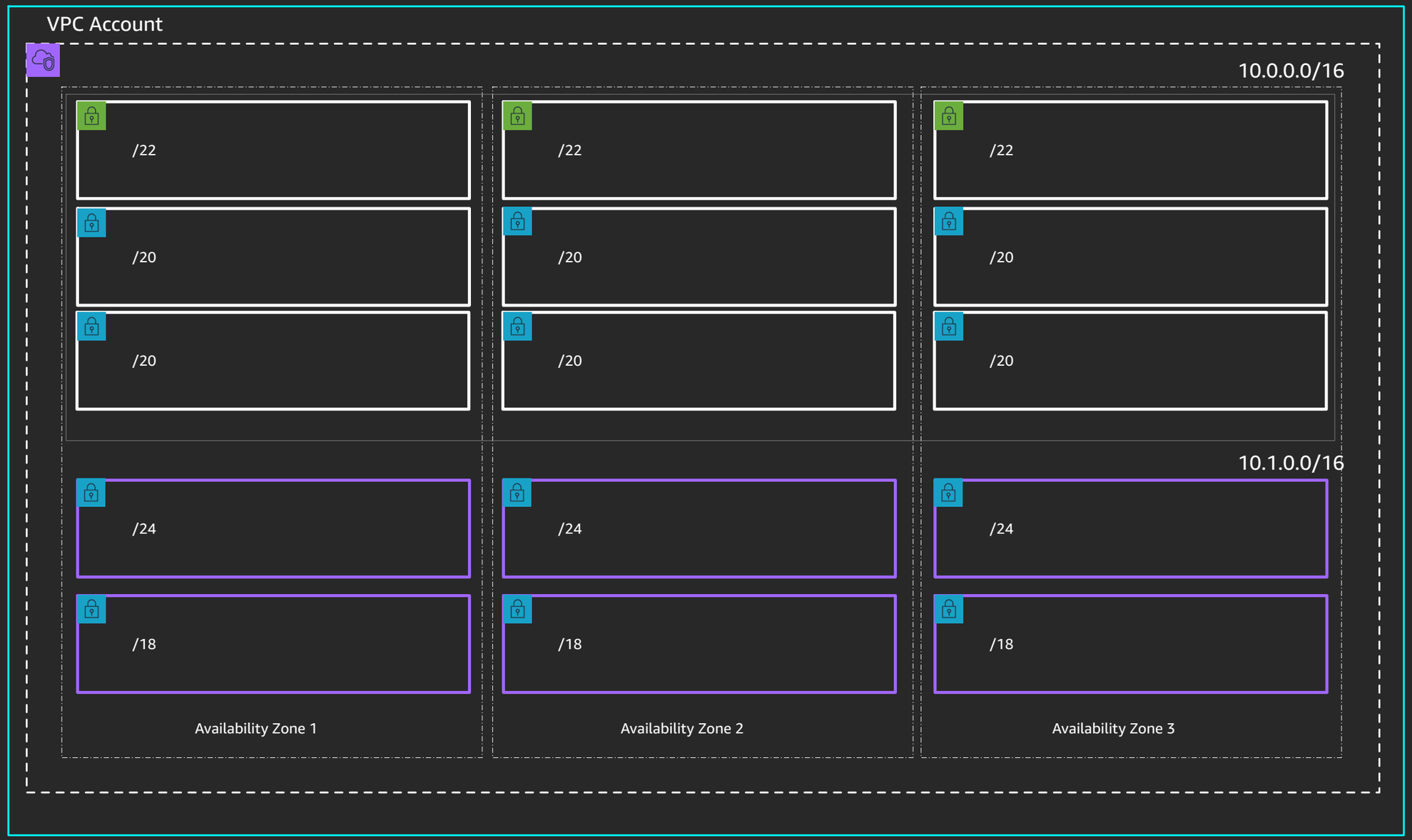
10.1.0.0/16

/24 /24 /24

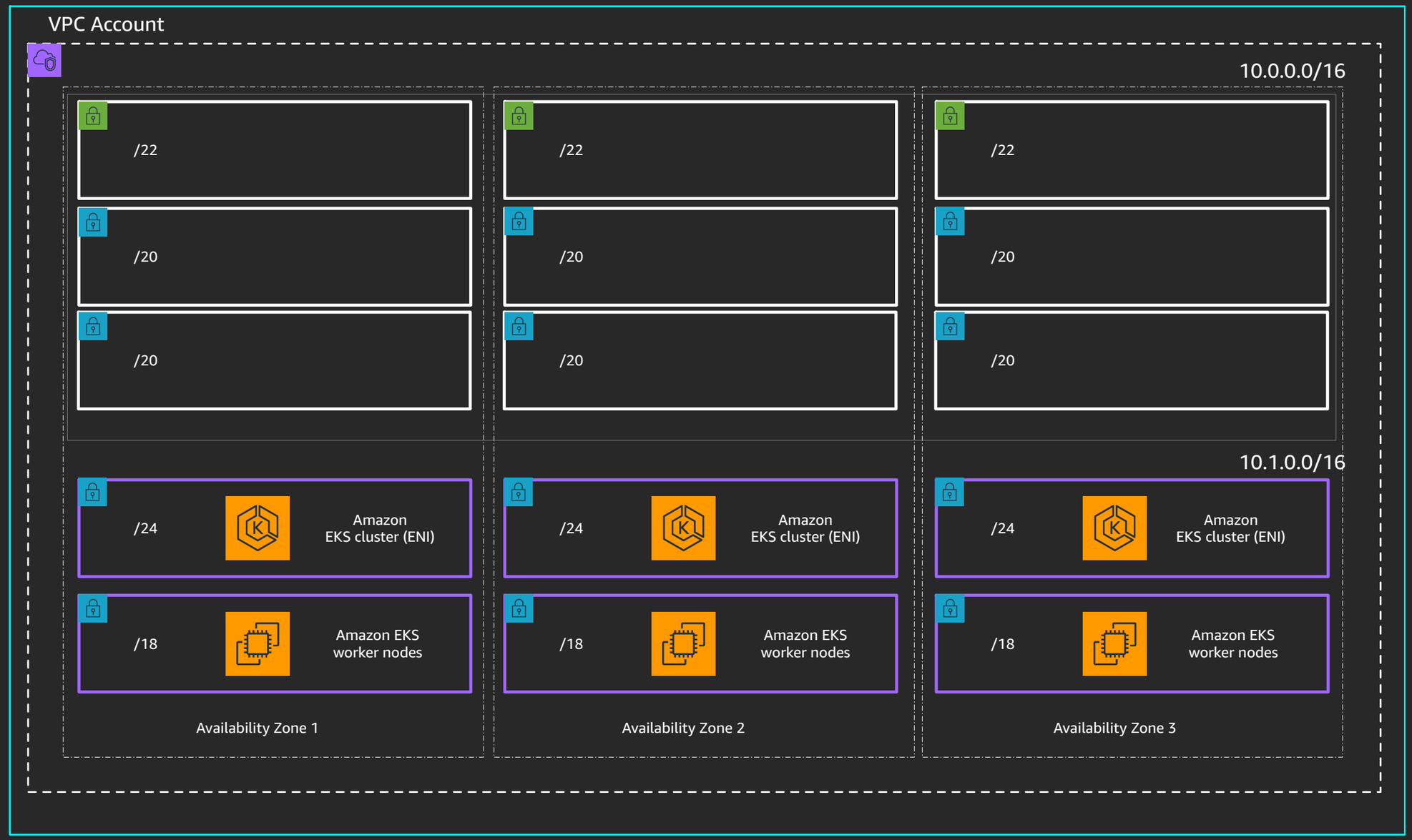
/18 /18 /18

Availability Zone 1 Availability Zone 2 Availability Zone 3

Single VPC—with Resource Sharing “shared VPC”



Single VPC—with Resource Sharing “shared VPC”

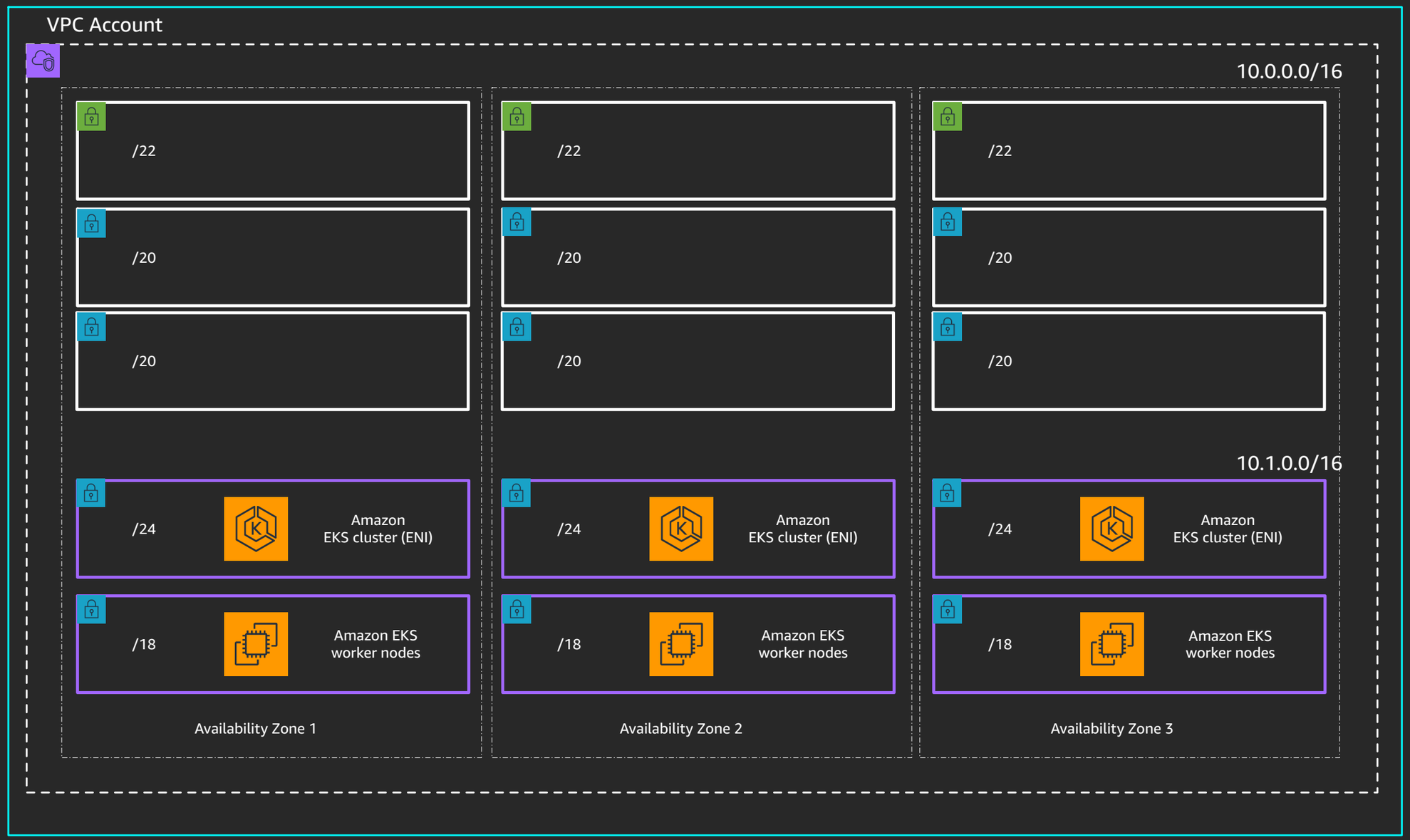


A1

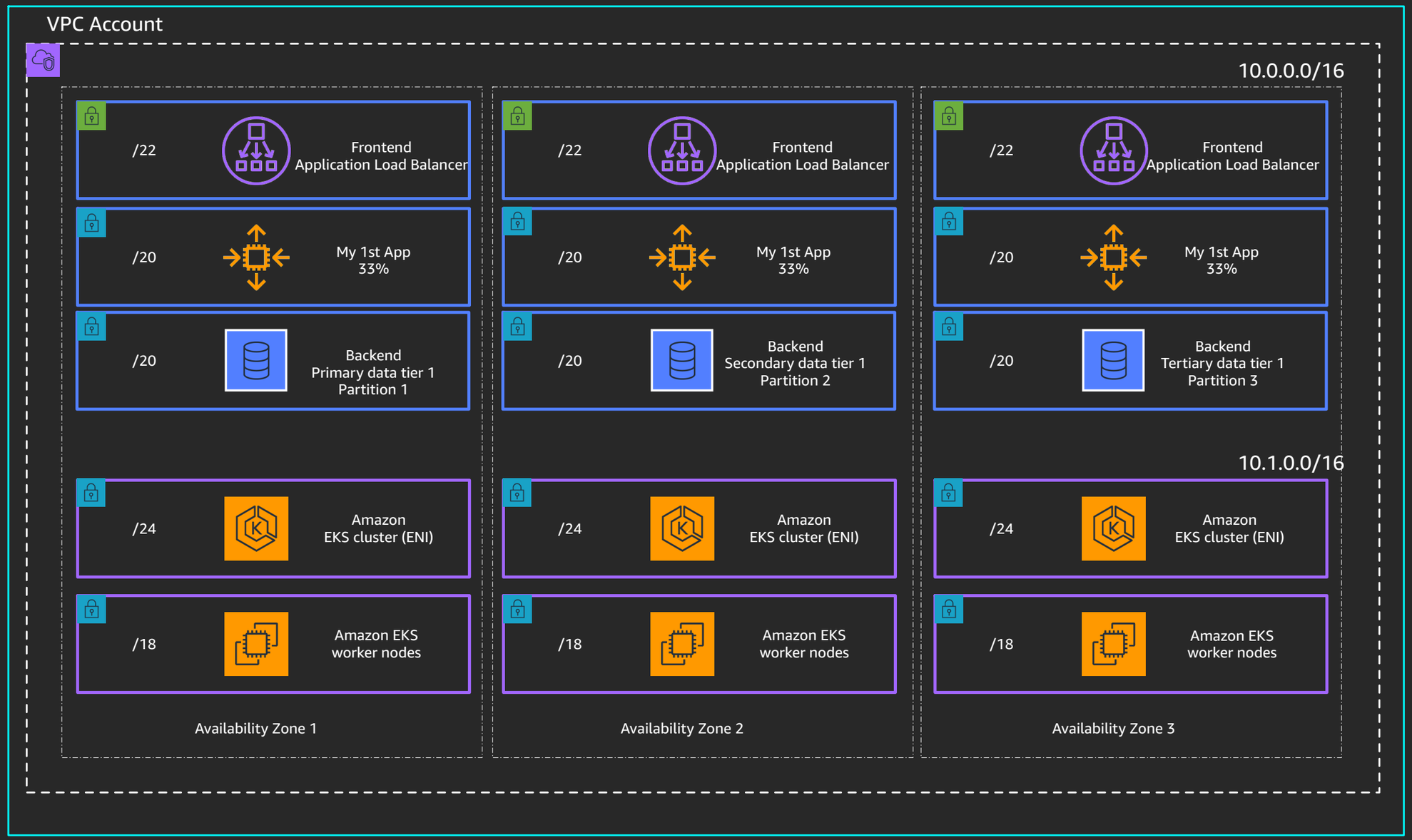
Amazon RDS

Account Purple

Single VPC—with Resource Sharing “shared VPC”



Single VPC—with Resource Sharing “shared VPC”



A1

Amazon RDS

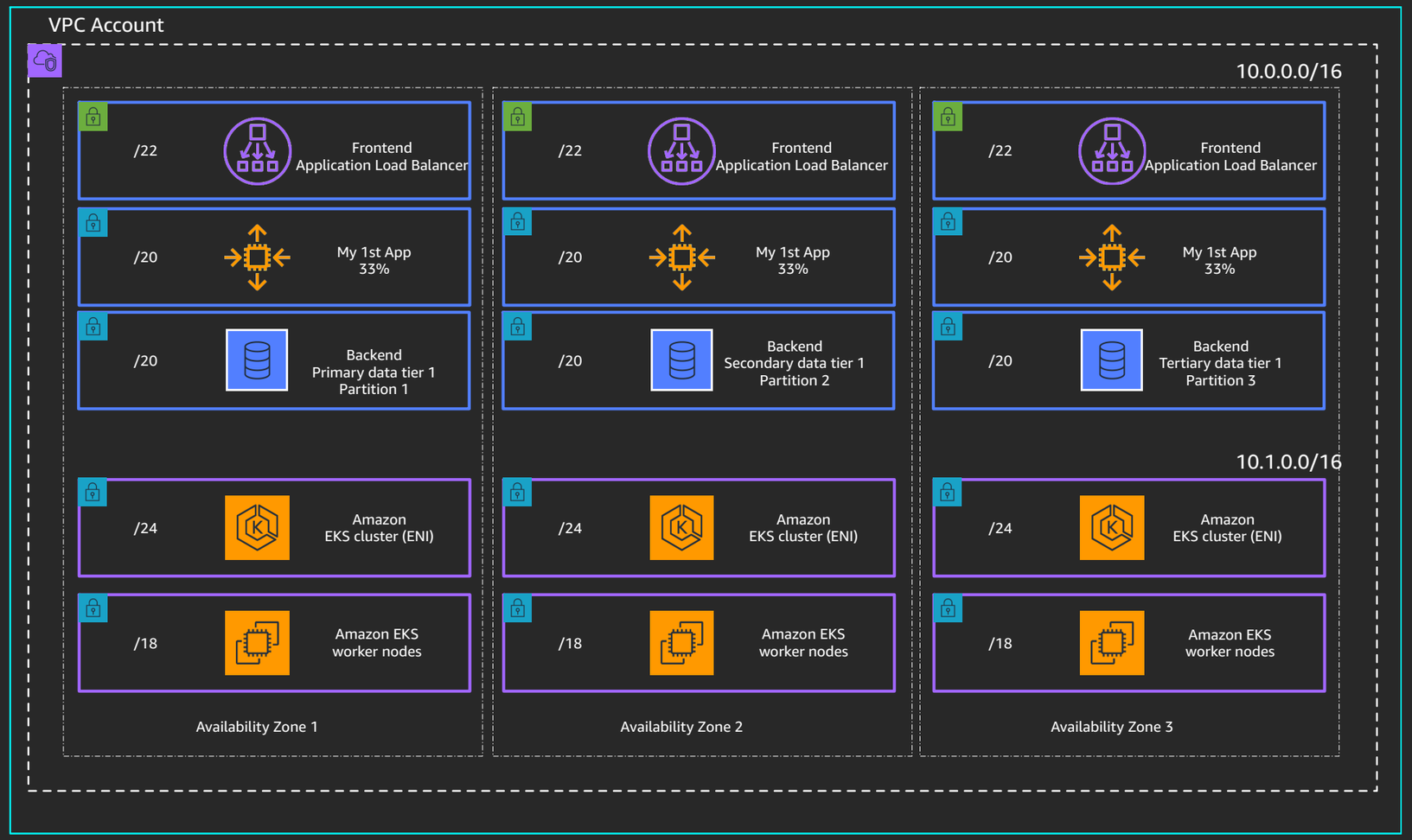
Account Blue

Account Purple

Is a shared VPC architecture right for me?

The good . . .

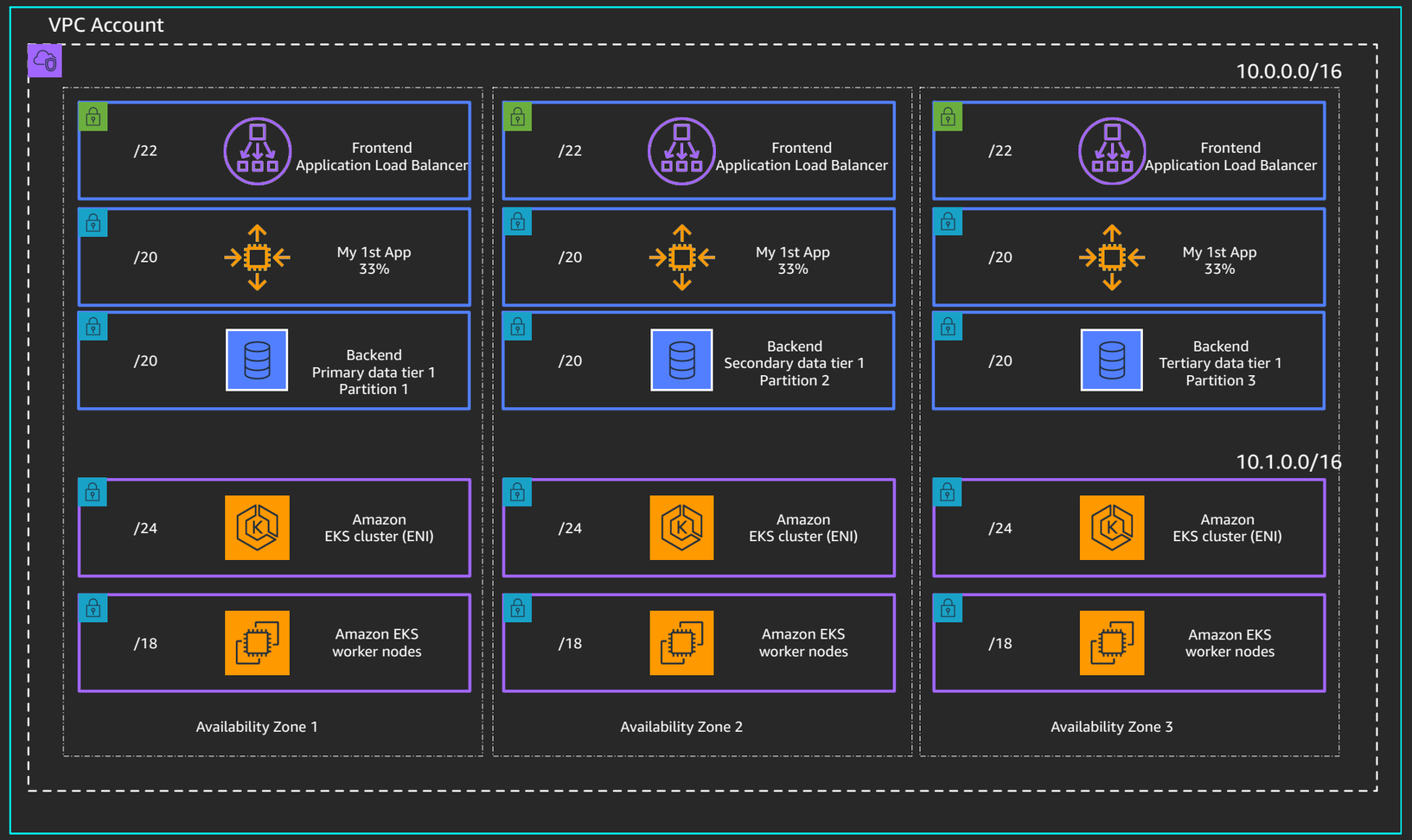
- Simple setup, even simpler for “participants”
- Account-specific cost allocation
- Shared DNS
- Application owners have limited access control



Is a shared VPC architecture right for me?

The bad . . .

- Application owners have limited access control?
- VPC-specific limits may still be a limiting factor
- Single VPC blast radius

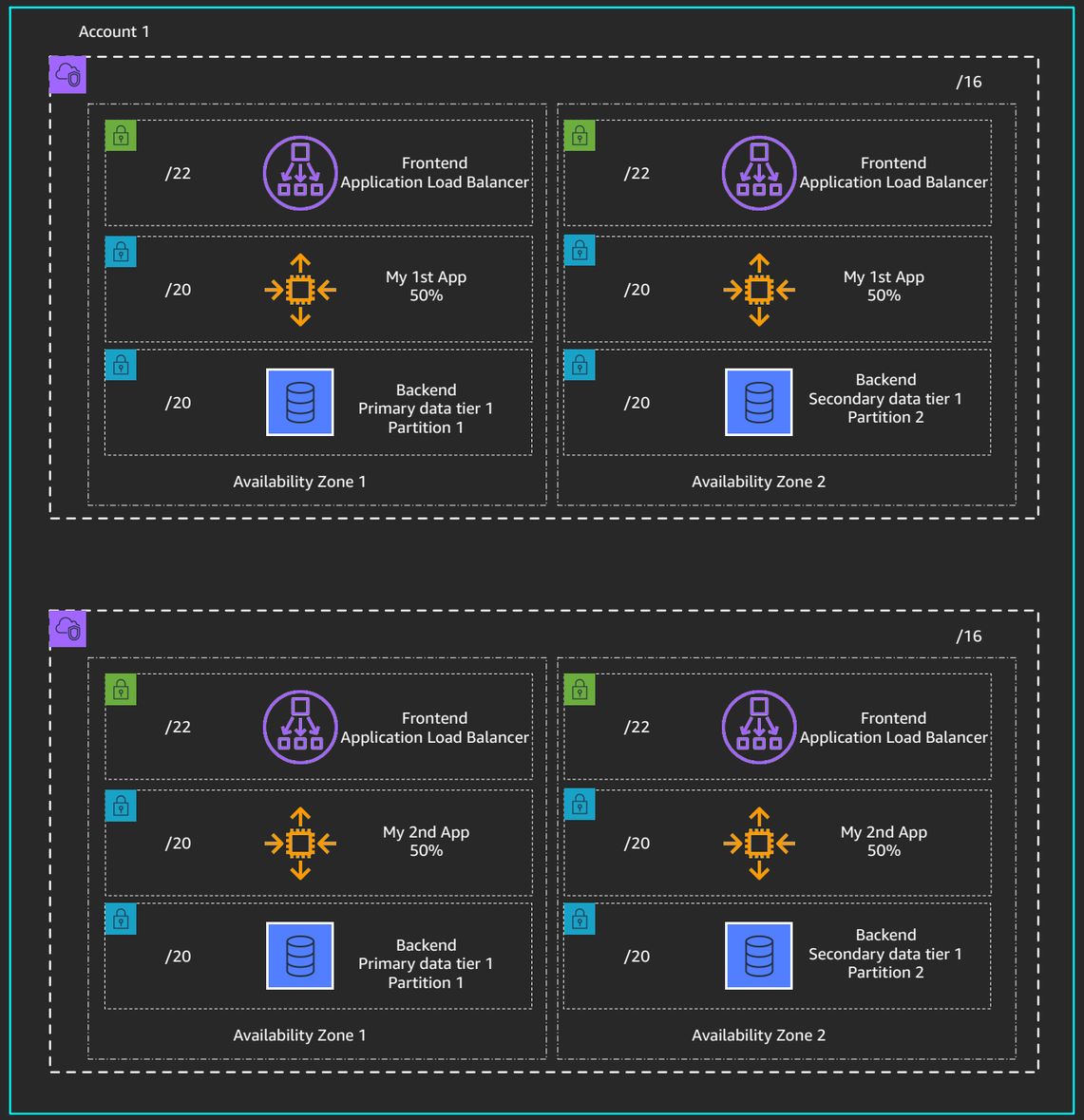


Segmented network architectures:
Multi Amazon VPC

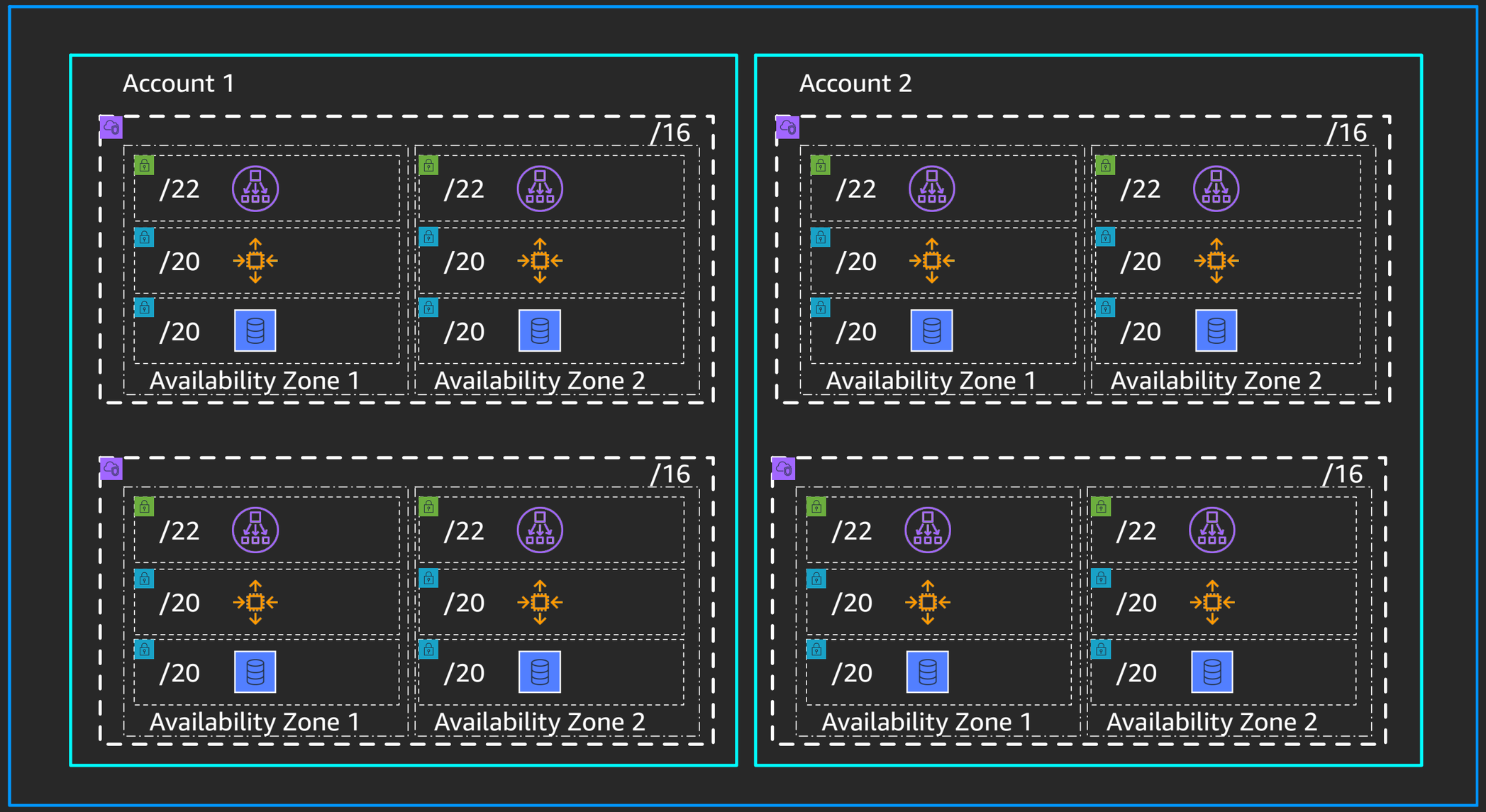
AWS re:Invent

aws

Multiple VPC—single account



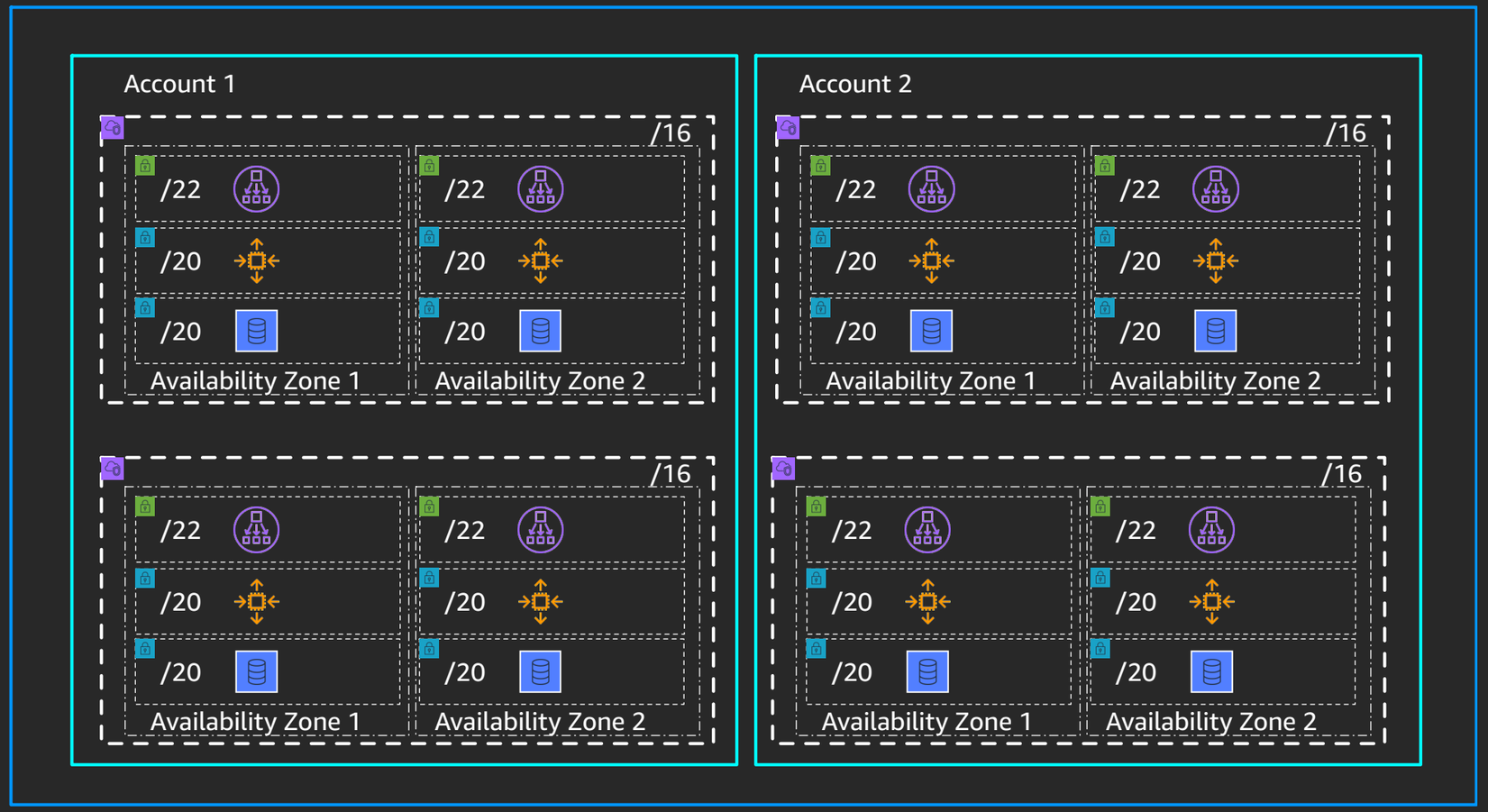
Multiple VPC—multiple accounts



Is a multi-VPC & multi-account architecture right for me?

The good . . .

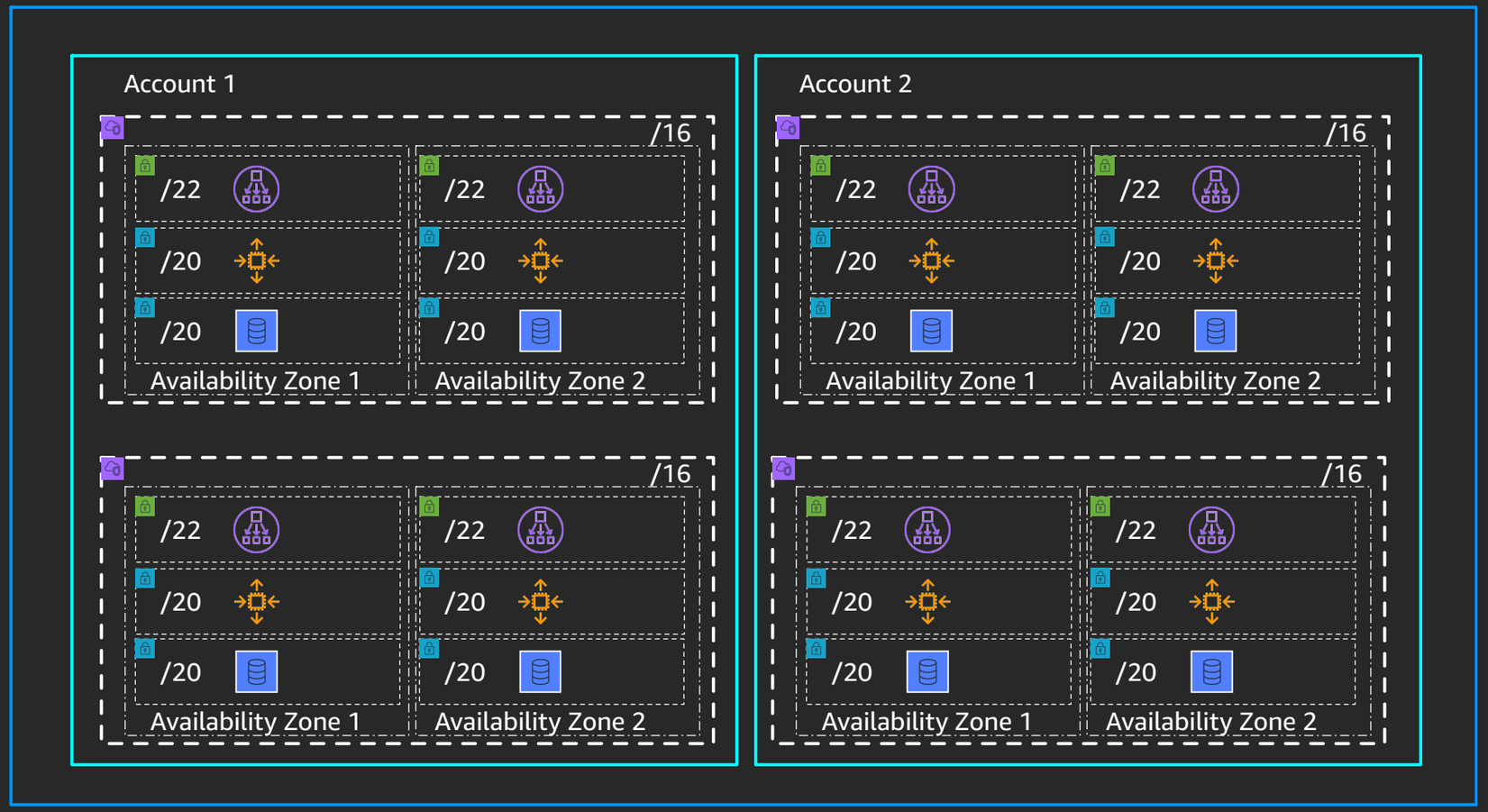
- Isolated blast radius
- Fine-grained access control
- Granular cost allocation
- Distributed/isolated service limits



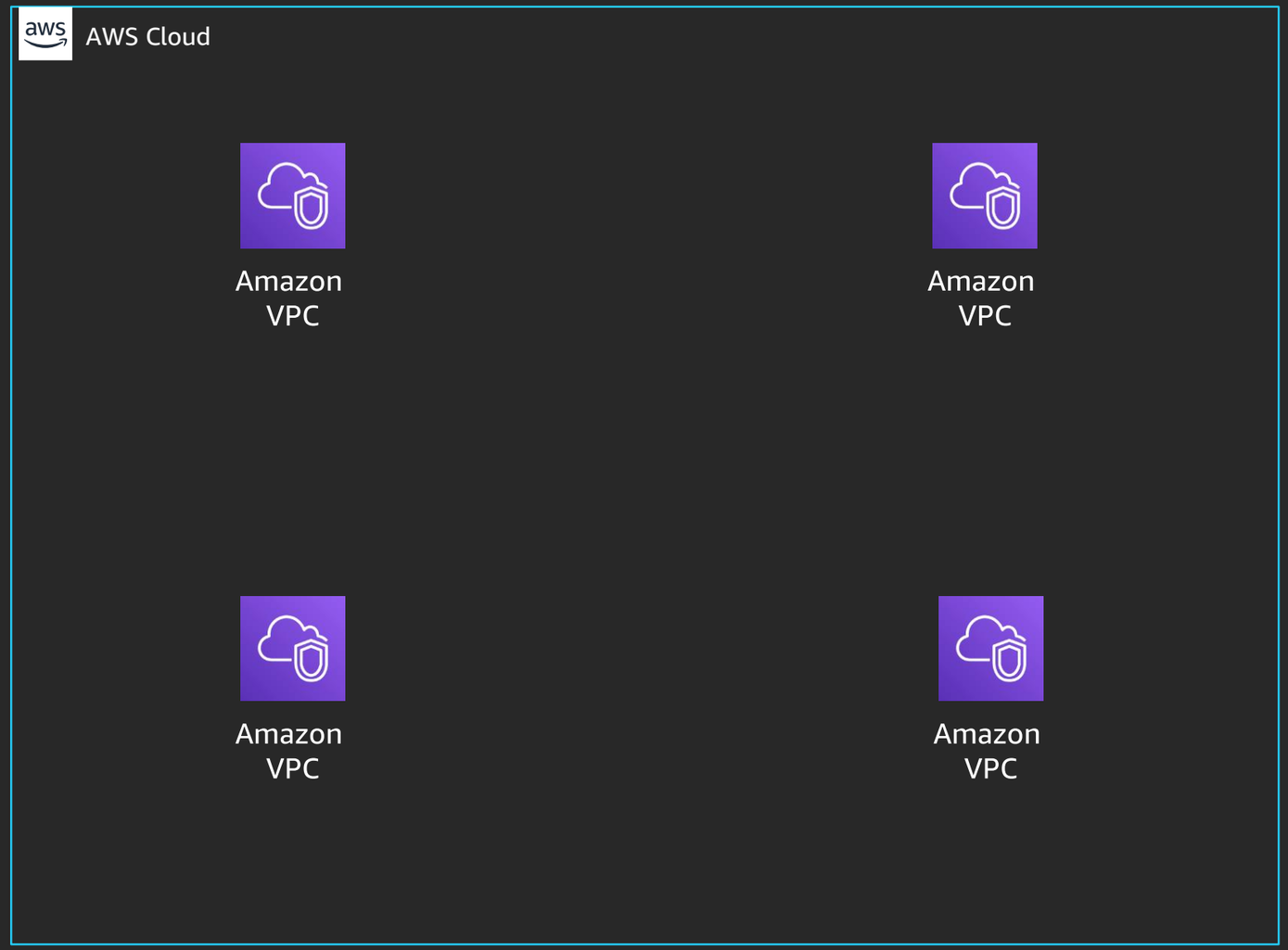
Is a multi-VPC & multi-account architecture right for me?

The bad . . .

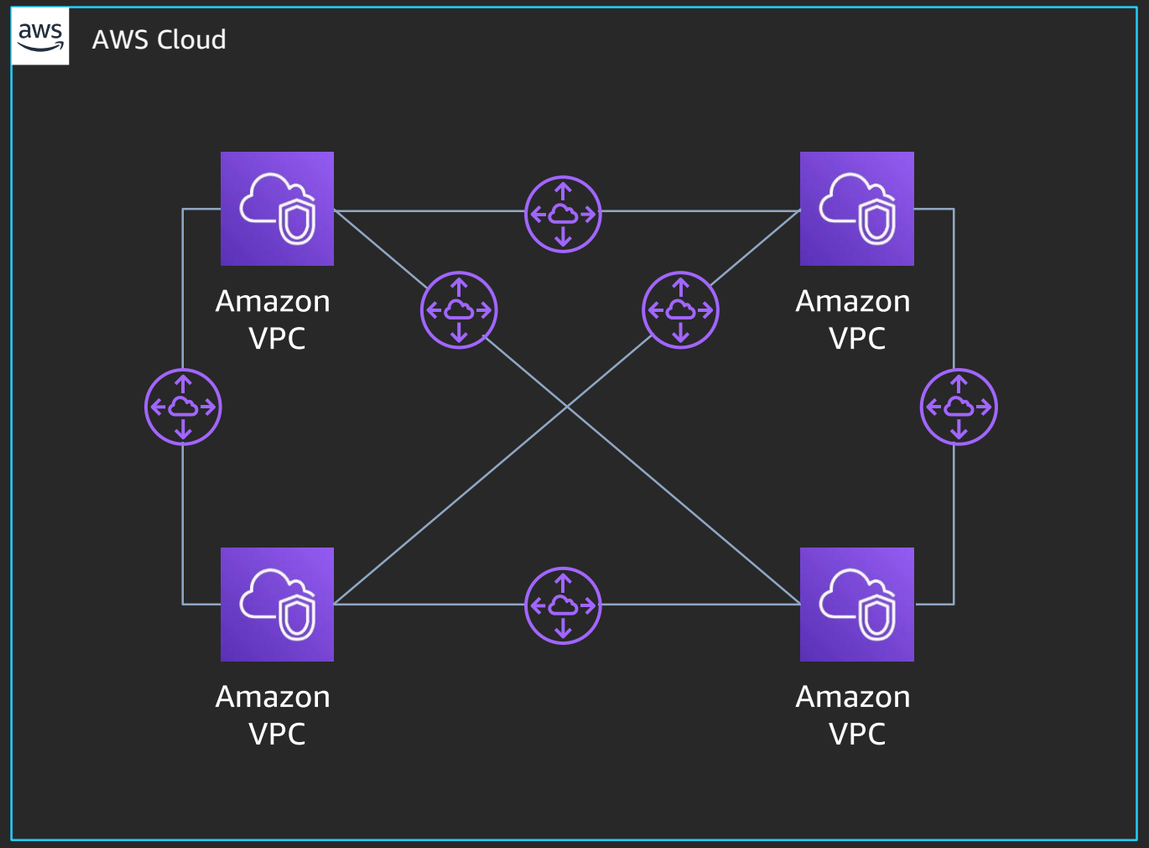
- Complexity
- How do you handle IP management?
- How do you handle DNS?
- How do you manage access control between accounts and VPCs?



VPC-to-VPC connectivity



VPC peering



The good . . .

- Simple at small scale
- Referenceable security groups
- Intra & inter-region support
- Scalable and fully managed redundancy

The bad . . .

- Does not support “transitive routing”
- Complex management at scale

Full mesh: How many Amazon VPC peering connections do I need (full mesh)?

$$\frac{n(n-1)}{2}$$

VPC x 100

Full mesh: How many Amazon VPC peering connections do I need (full mesh)?

4,950

VPC x 100

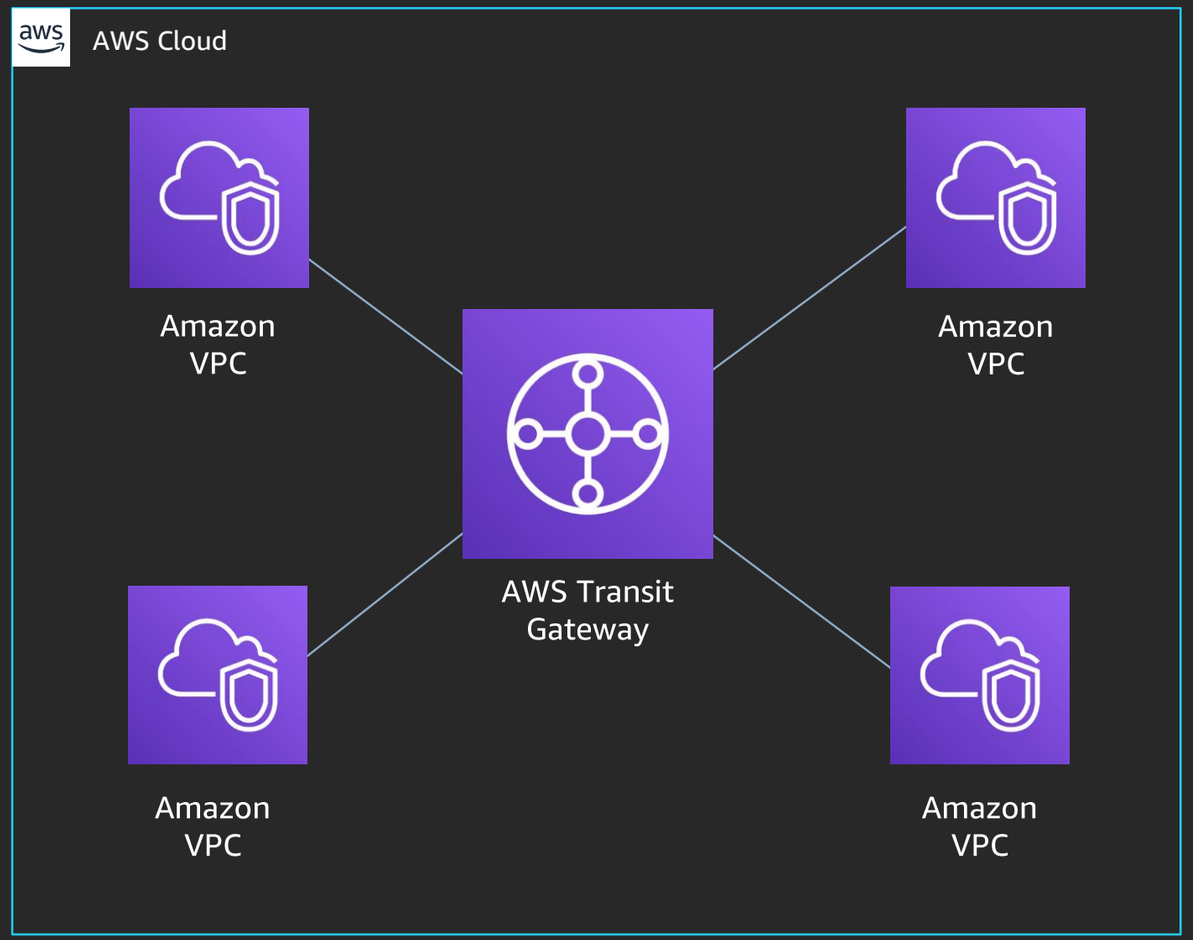
Static routes per Amazon VPC route table

1,000

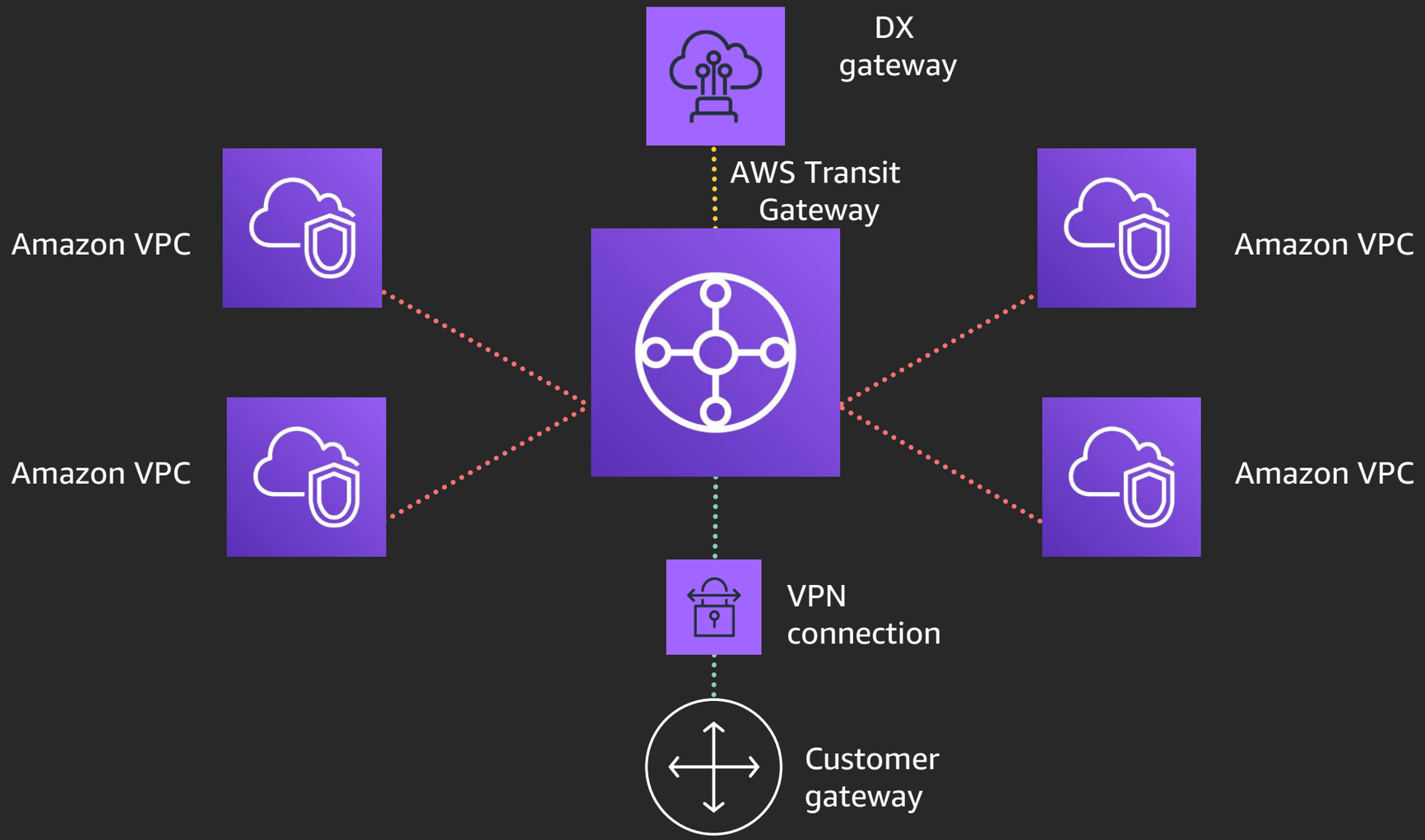
Amazon VPC peering connections per Amazon VPC

125

Connecting VPCs at scale—AWS Transit Gateway



AWS Transit Gateway—edge connectivity



Static routes per AWS Transit Gateway

10,000

AWS Transit Gateway attachments per transit gateway

5,000

AWS Transit Gateway

Regional service

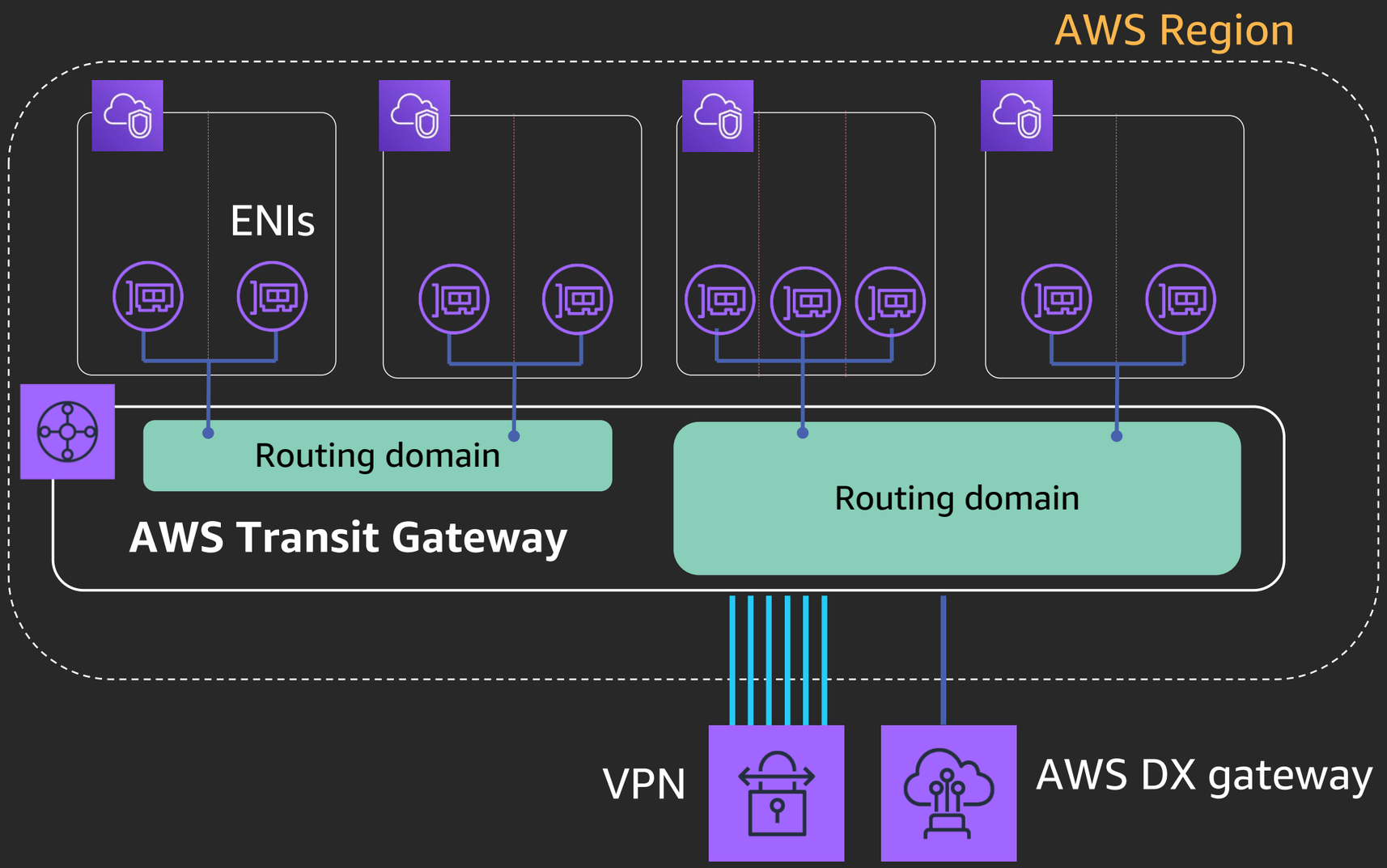
- Centralize VPN and DX gateway

Scalable

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

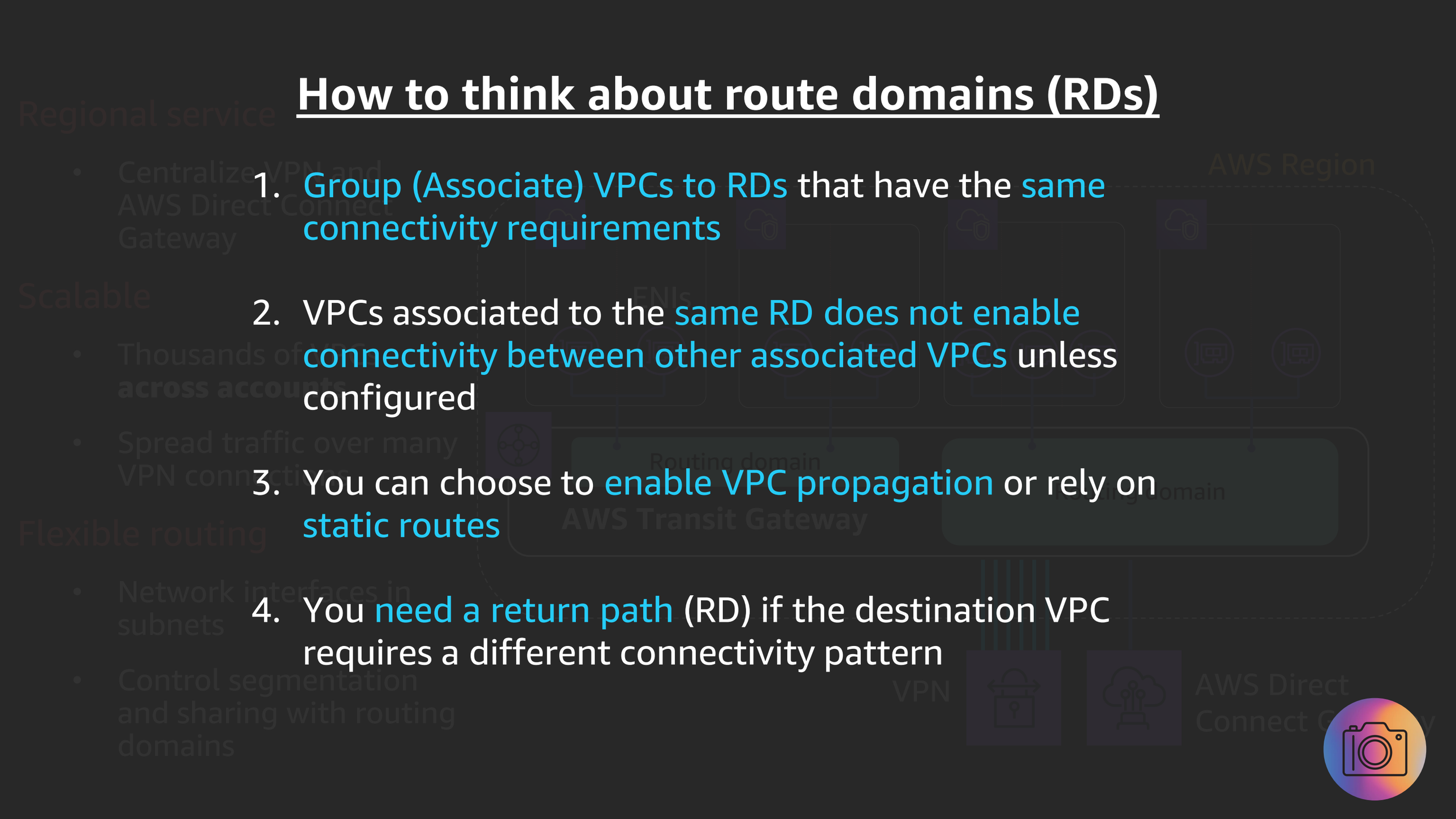
Flexible routing

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

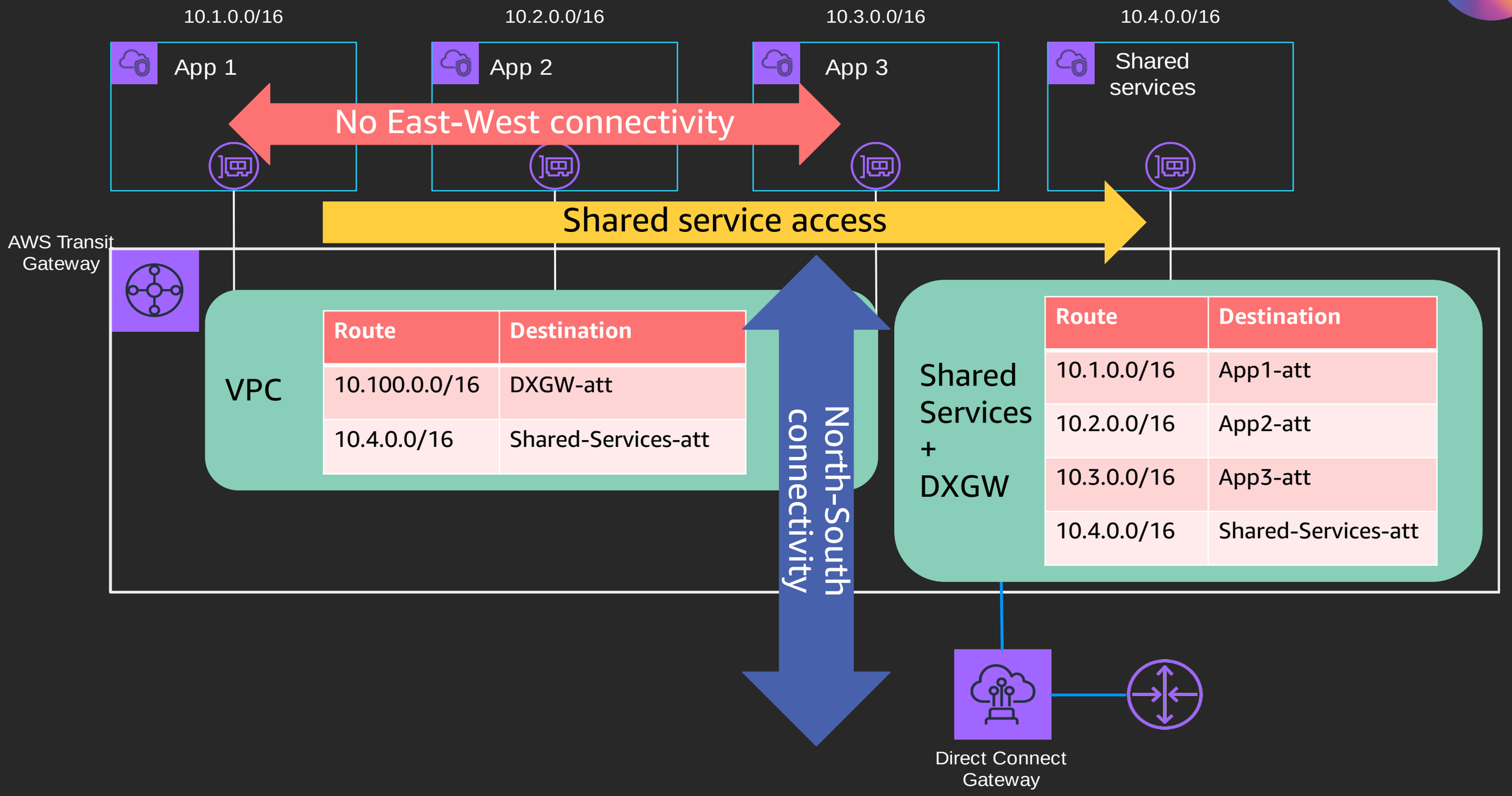


How to think about route domains (RDs)

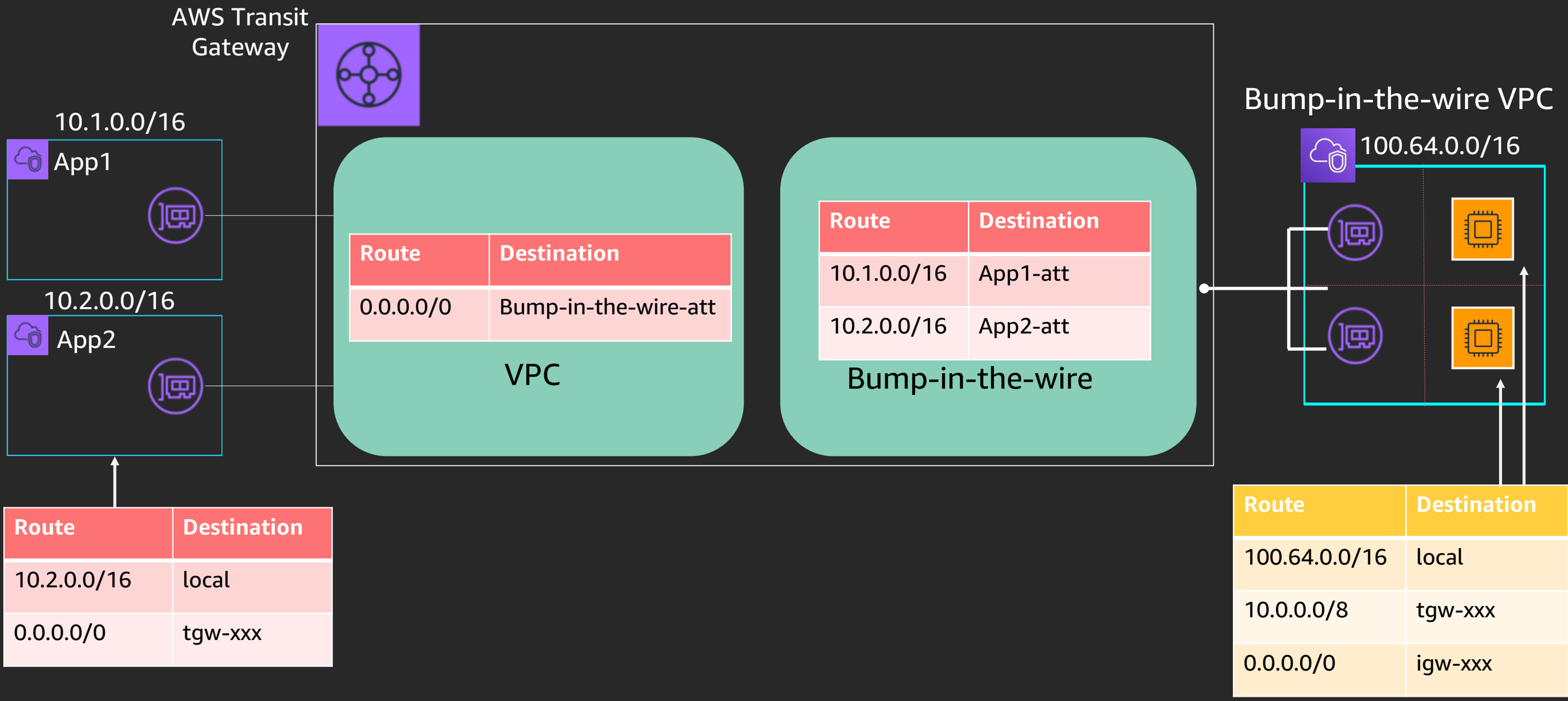
1. Group (Associate) VPCs to RDs that have the same connectivity requirements
2. VPCs associated to the same RD does not enable connectivity between other associated VPCs unless configured
3. You can choose to enable VPC propagation or rely on static routes
4. You need a return path (RD) if the destination VPC requires a different connectivity pattern



Shared services VPC



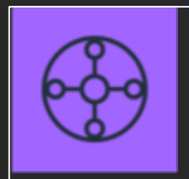
Bump-in-the-wire—VPC attachment



Bump-in-the-wire—VPC attachment

Apply **SRC_NAT** to firewall/IDS

AWS Transit Gateway



10.1.0.0/16

App1

10.2.0.0/16

App2

Route	Destination
0.0.0.0/0	Bump-in-the-wire-att

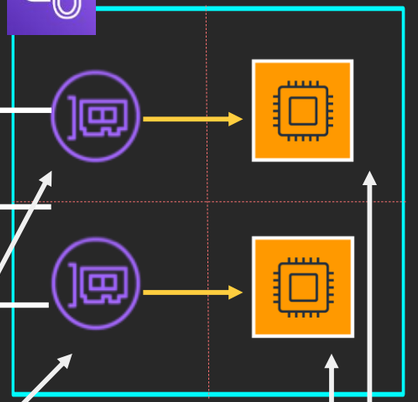
VPC

Route	Destination
10.1.0.0/16	App1-att
10.2.0.0/16	App2-att

Bump-in-the-wire

Bump-in-the-wire VPC

100.64.0.0/16



Route	Destination
10.2.0.0/16	local
0.0.0.0/0	tgw-xxx

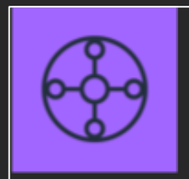
Route	Destination
100.64.0.0/16	local
0.0.0.0/0	ENI-xxx

Route	Destination
100.64.0.0/16	local
10.0.0.0/8	tgw-xxx
0.0.0.0/0	igw-xxx

Bump-in-the-wire—VPC attachment

Active/Standby Model
NO **SRC_NAT**
to firewall/IDS

AWS Transit Gateway



10.1.0.0/16

App1

10.2.0.0/16

App2

Route	Destination
0.0.0.0/0	Bump-in-the-wire-att

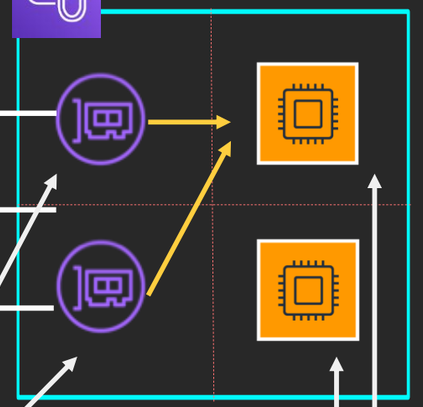
VPC

Route	Destination
10.1.0.0/16	App1-att
10.2.0.0/16	App2-att

Bump-in-the-wire

Bump-in-the-wire VPC

100.64.0.0/16



Route	Destination
10.2.0.0/16	local
0.0.0.0/0	tgw-xxx

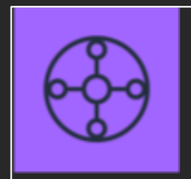
Route	Destination
100.64.0.0/16	local
0.0.0.0/0	**ENI-xxx (AZ1)**

Route	Destination
100.64.0.0/16	local
10.0.0.0/8	tgw-xxx
0.0.0.0/0	igw-xxx

Bump-in-the-wire—VPC attachment

Active/Standby Model
NO **SRC_NAT**
to firewall/IDS

AWS Transit Gateway



10.1.0.0/16

App1

10.2.0.0/16

App2

Route	Destination
0.0.0.0/0	Bump-in-the-wire-att

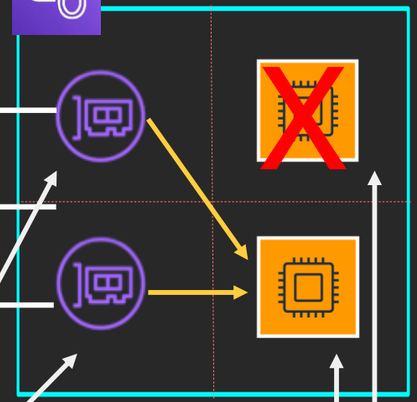
VPC

Route	Destination
10.1.0.0/16	App1-att
10.2.0.0/16	App2-att

Bump-in-the-wire

Bump-in-the-wire VPC

100.64.0.0/16



Route	Destination
10.2.0.0/16	local
0.0.0.0/0	tgw-xxx

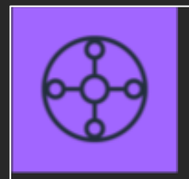
Route	Destination
100.64.0.0/16	local
0.0.0.0/0	**ENI-xxx (AZ2)**

Route	Destination
100.64.0.0/16	local
10.0.0.0/8	tgw-xxx
0.0.0.0/0	igw-xxx

Bump-in-the-wire—VPN attachment

Apply **SRC_NAT**
to firewall/IDS/proxy

AWS Transit Gateway



10.1.0.0/16

App1

10.2.0.0/16

App2

Route	Destination
0.0.0.0/0	Bump-in-the-wire-att

VPC

Route	Destination
10.1.0.0/16	App1-att
10.2.0.0/16	App2-att

Bump-in-the-wire

Bump-in-the-wire VPC

100.64.0.0/16

ECMP VPN

Inline Firewall

Route	Destination
10.2.0.0/16	local
0.0.0.0/0	tgw-xxx
100.64.0.0/16	tgw-xxx

BGP Prefix	Next-hop
0.0.0.0/0	Local IP

Route	Destination
100.64.0.0/16	local
0.0.0.0/0	igw-xxx

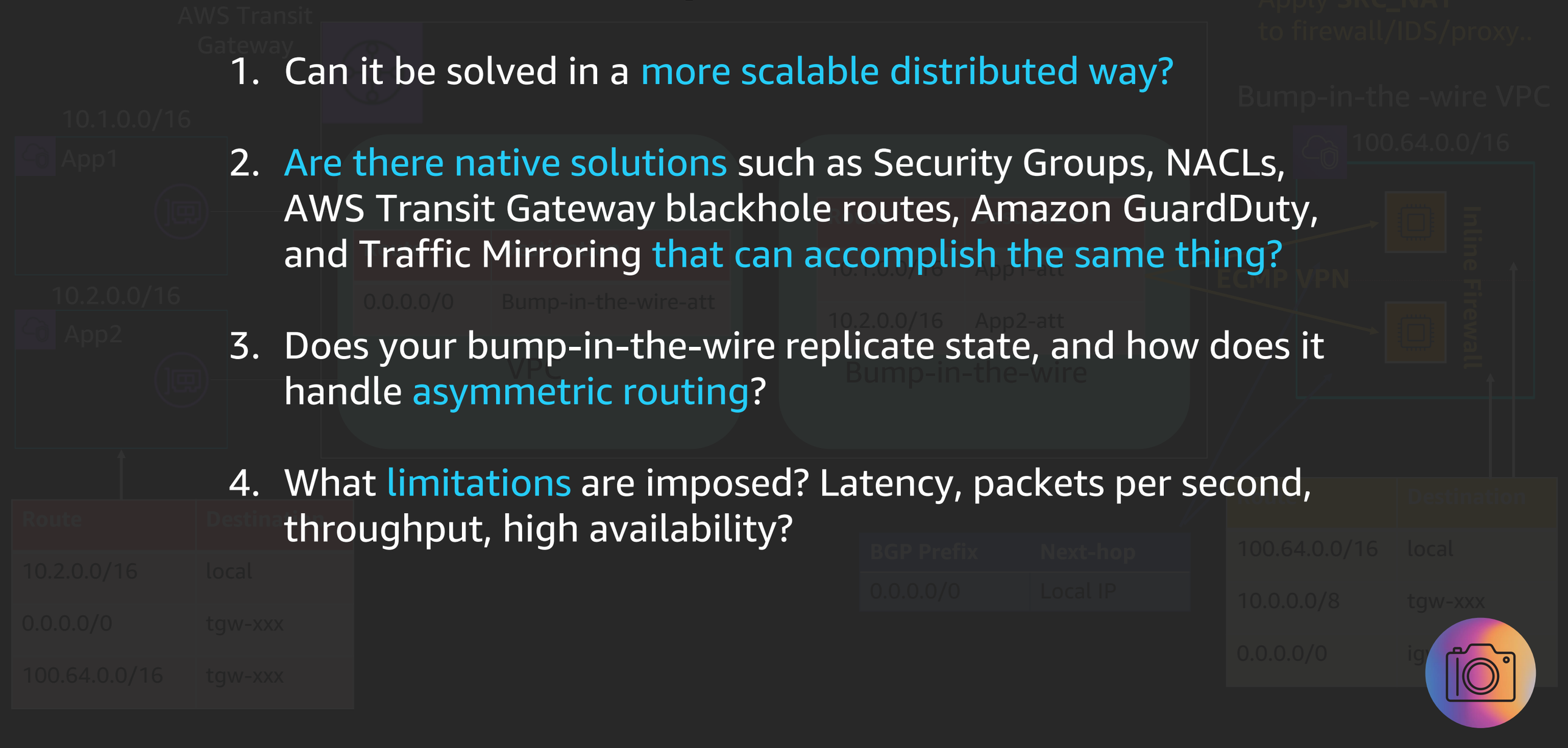
Bump-in-the-wire

1. Can it be solved in a more scalable distributed way?

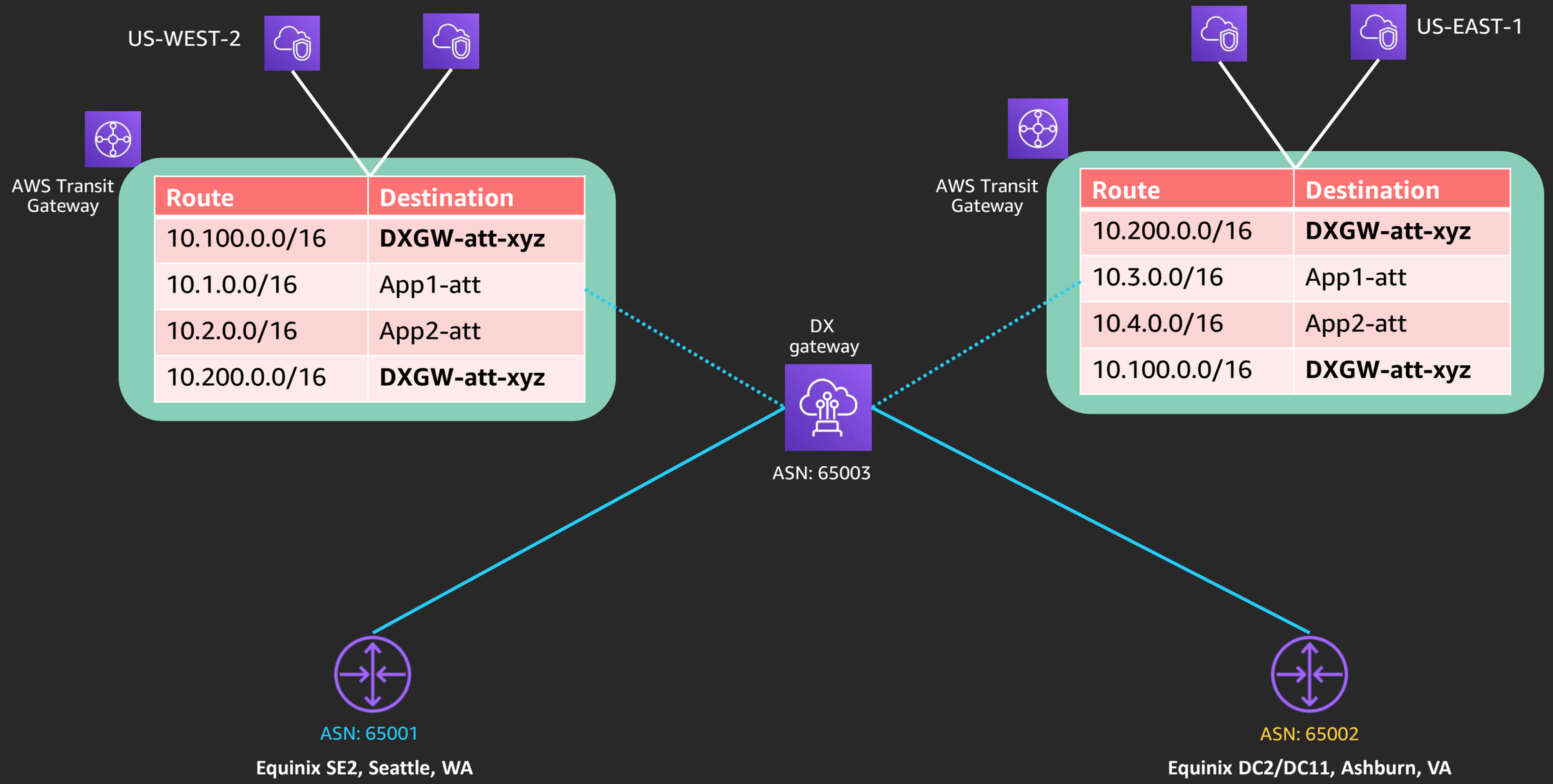
2. Are there native solutions such as Security Groups, NACLs, AWS Transit Gateway blackhole routes, Amazon GuardDuty, and Traffic Mirroring that can accomplish the same thing?

3. Does your bump-in-the-wire replicate state, and how does it handle asymmetric routing?

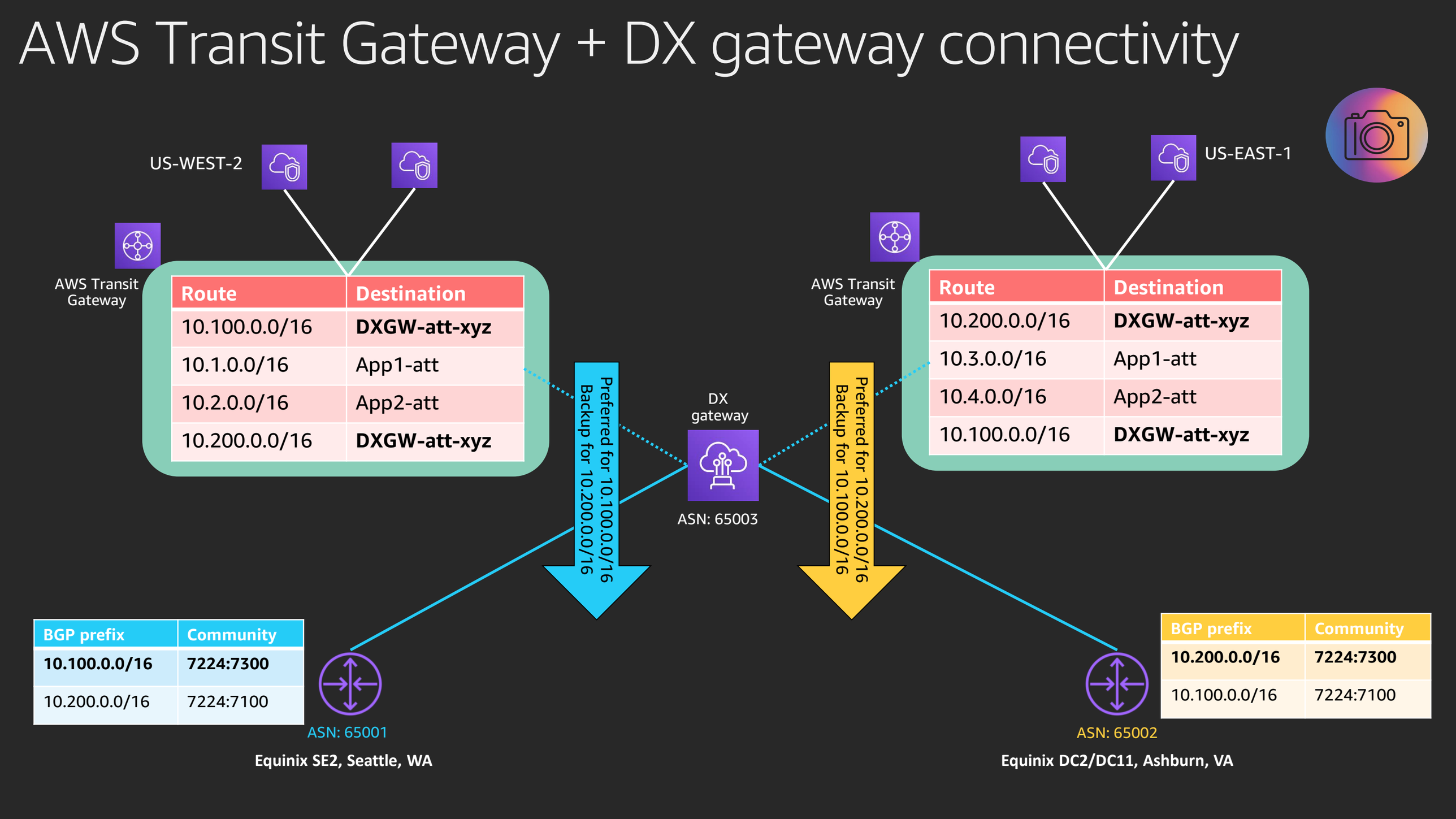
4. What limitations are imposed? Latency, packets per second, throughput, high availability?



AWS Transit Gateway + DX gateway connectivity

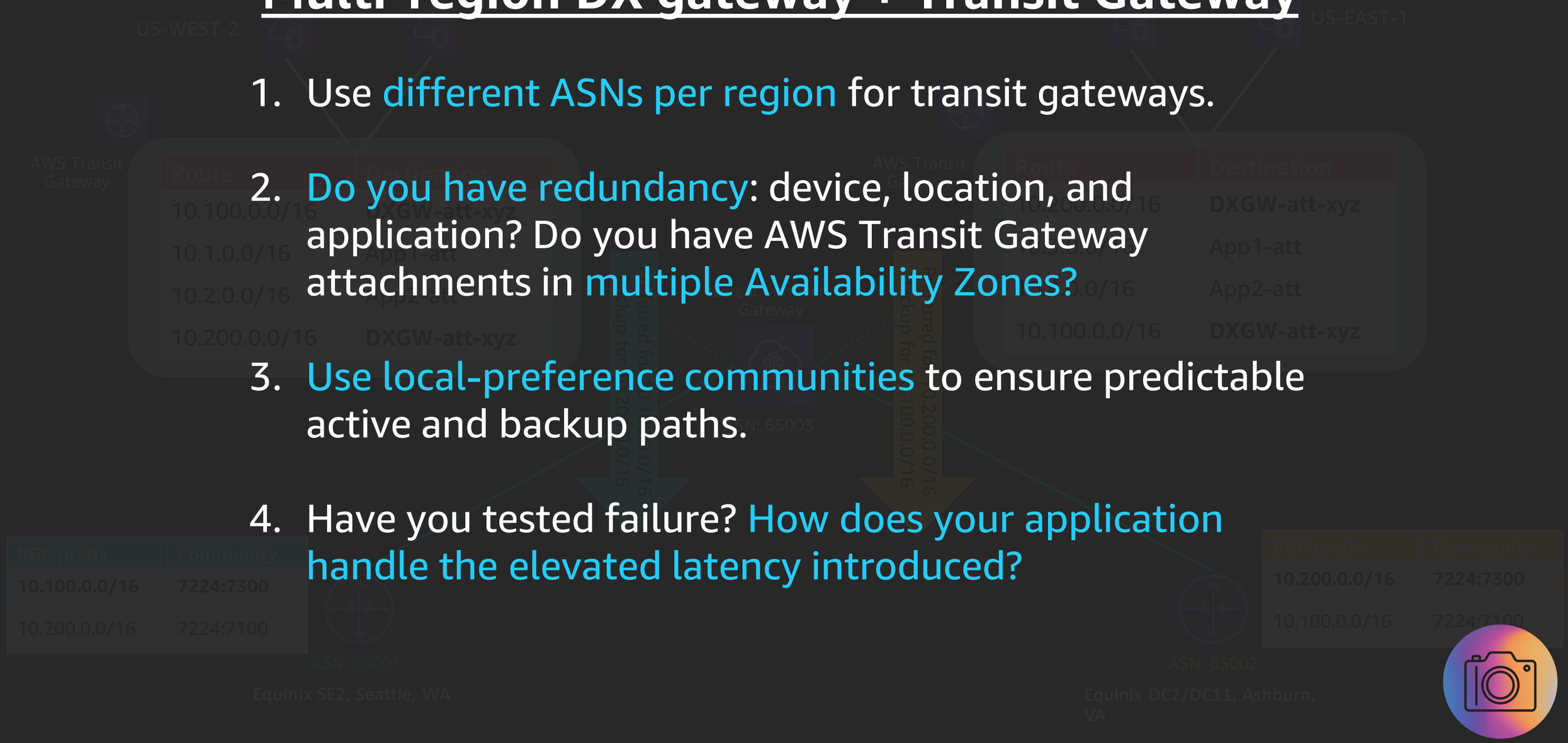


AWS Transit Gateway + DX gateway connectivity

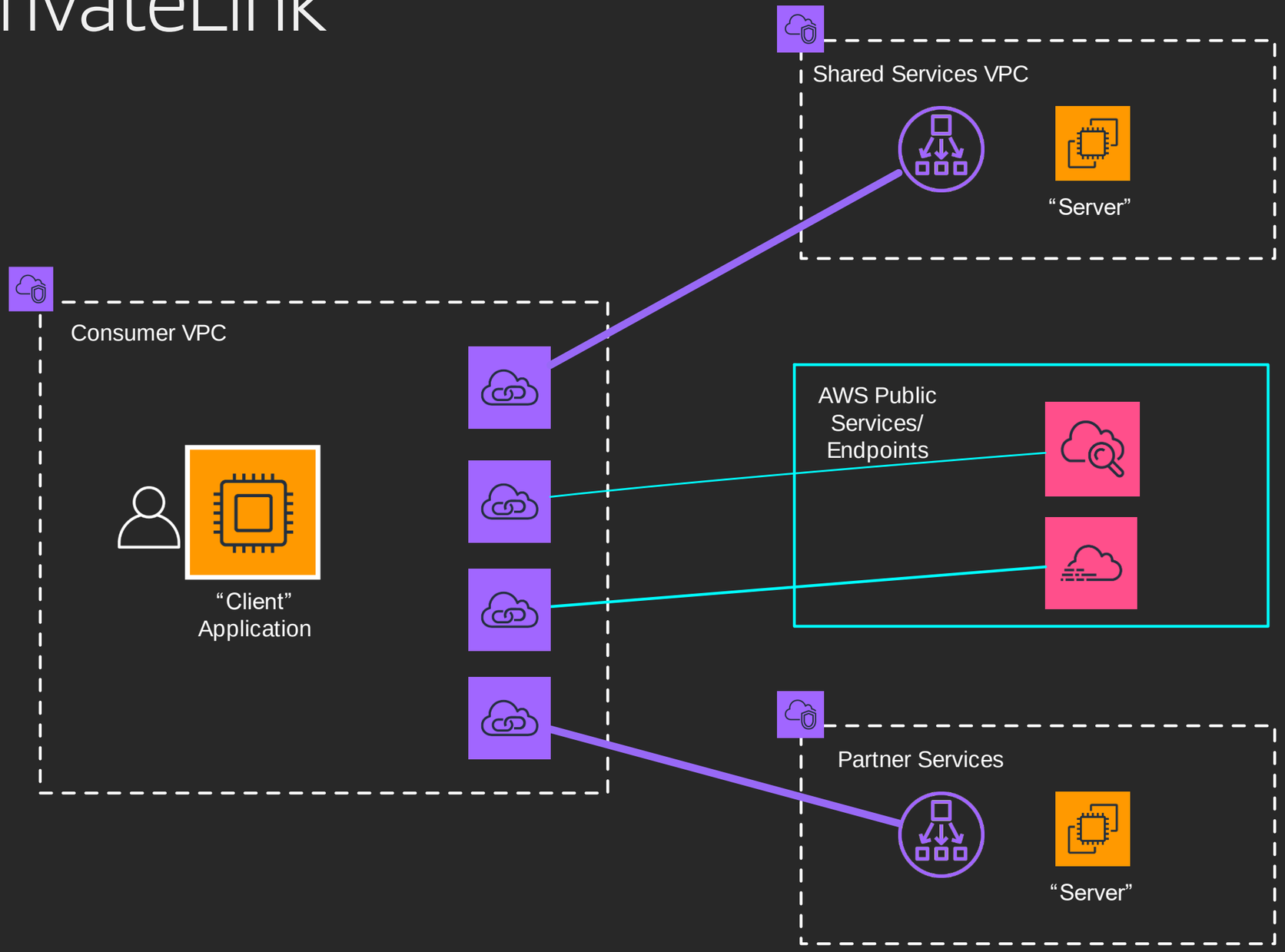


Multi-region DX gateway + Transit Gateway

1. Use different ASNs per region for transit gateways.
2. Do you have redundancy: device, location, and application? Do you have AWS Transit Gateway attachments in multiple Availability Zones?
3. Use local-preference communities to ensure predictable active and backup paths.
4. Have you tested failure? How does your application handle the elevated latency introduced?



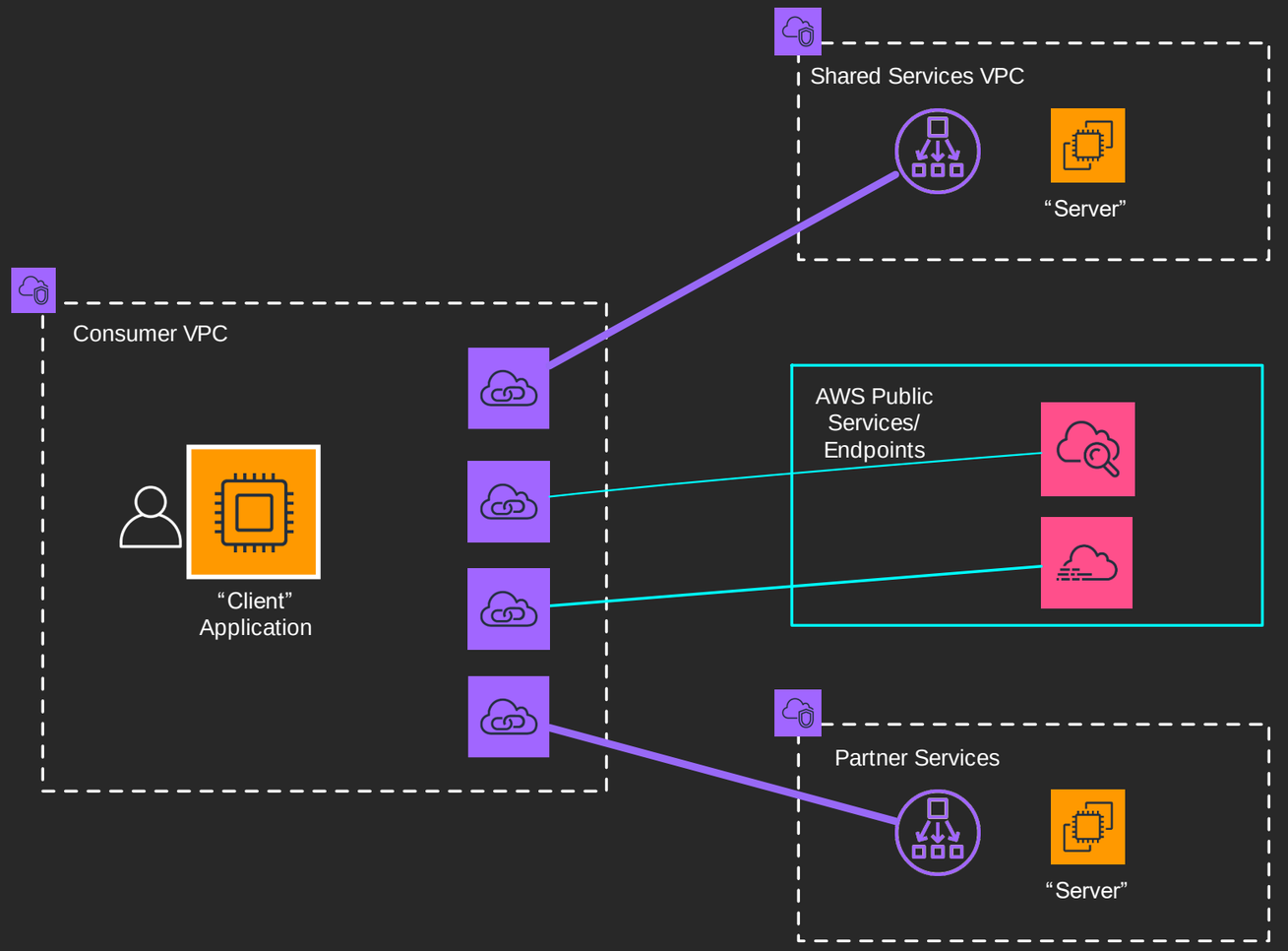
AWS PrivateLink



AWS PrivateLink

The good . . .

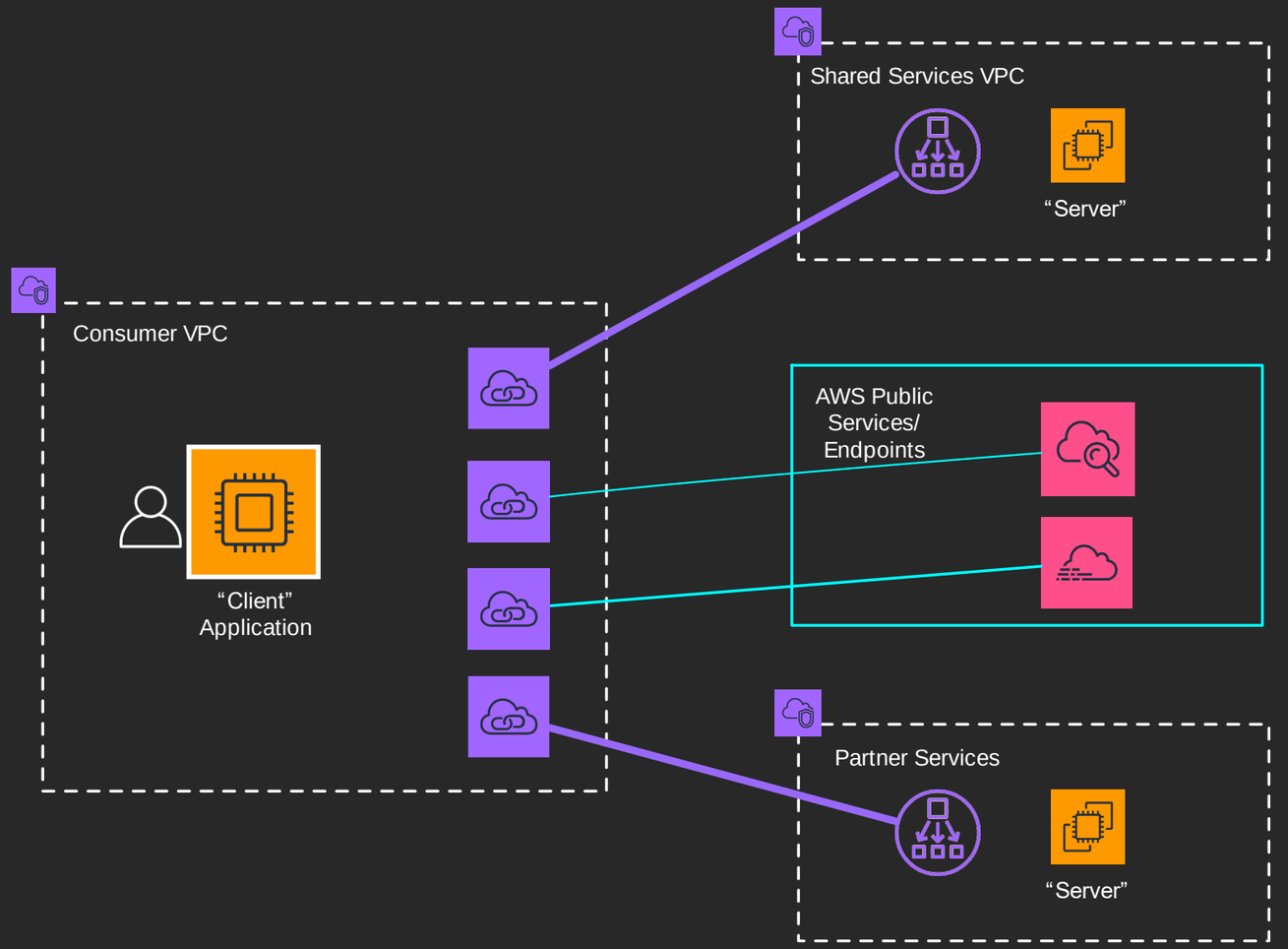
- Simplifies and limits exposure to and from shared services
- Removes overlapping IP constraints
- Dynamically scales to millions of TPSs
- Accessible over all AWS mediums and across accounts



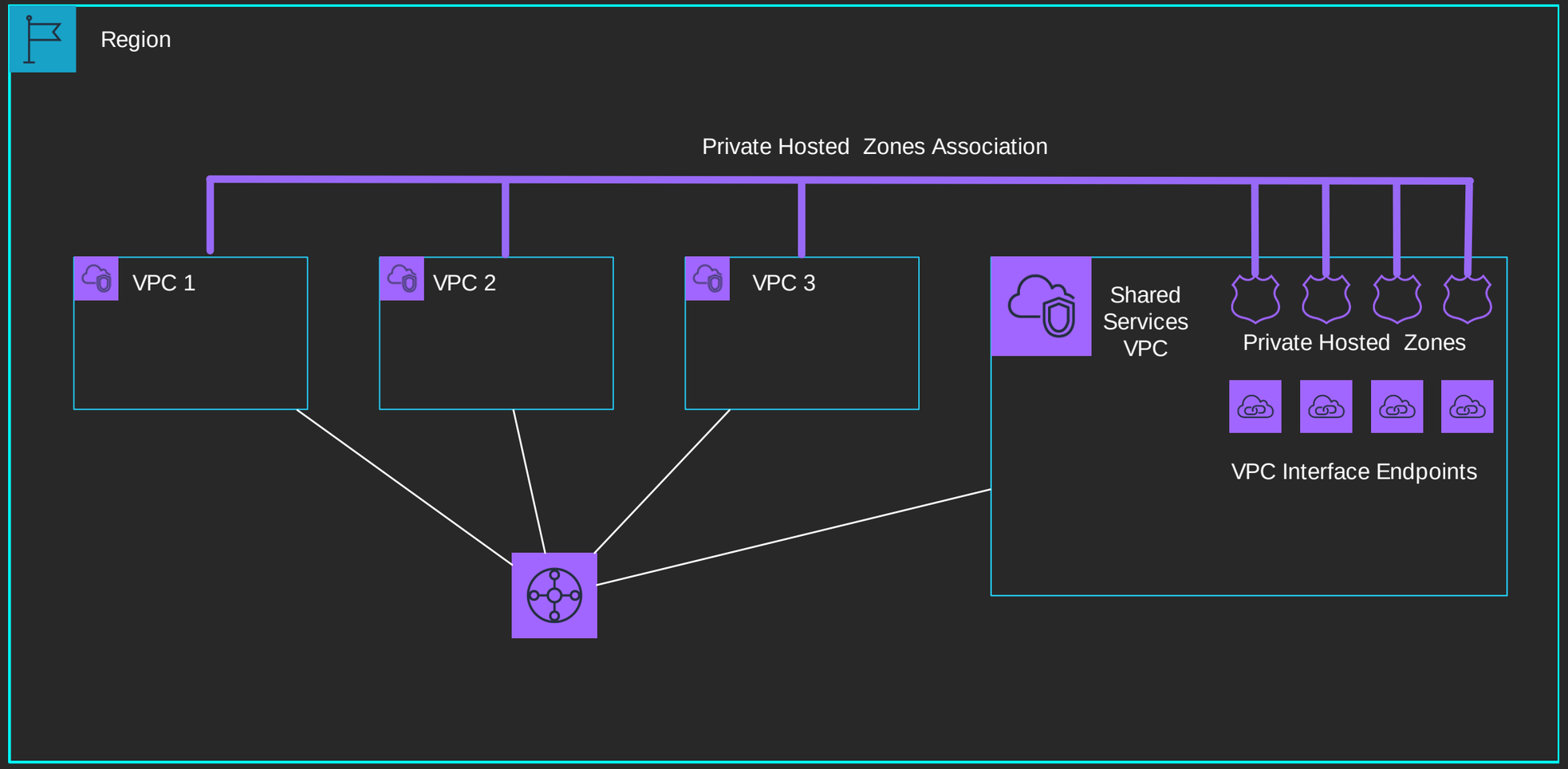
AWS PrivateLink

The bad…

- Supports only TCP client/server application today
- Requires a Network Load Balancer in the “Server VPC”
- DNS can be complex when designed in a centralized fashion

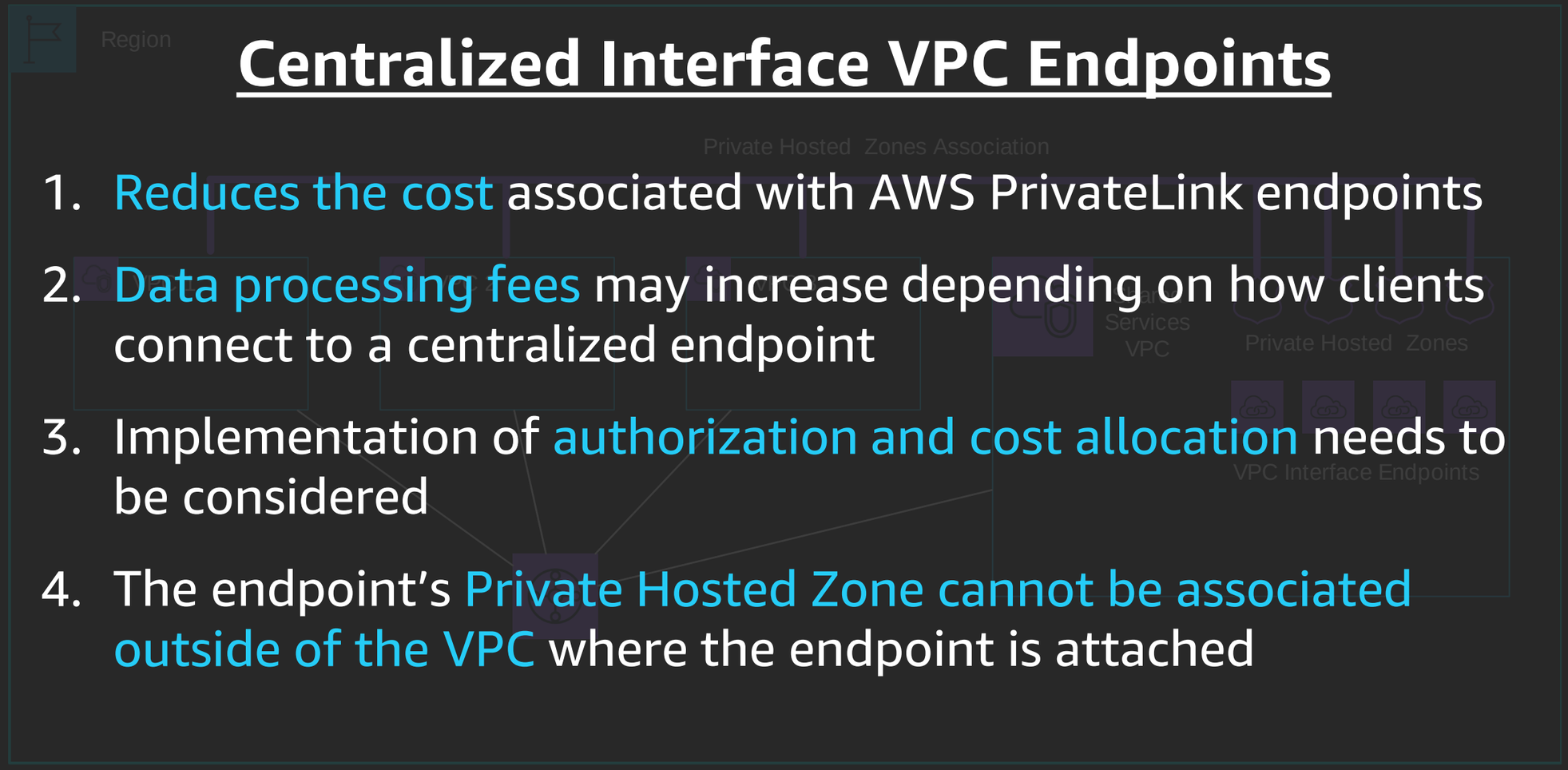


Centralizing Interface VPC Endpoints



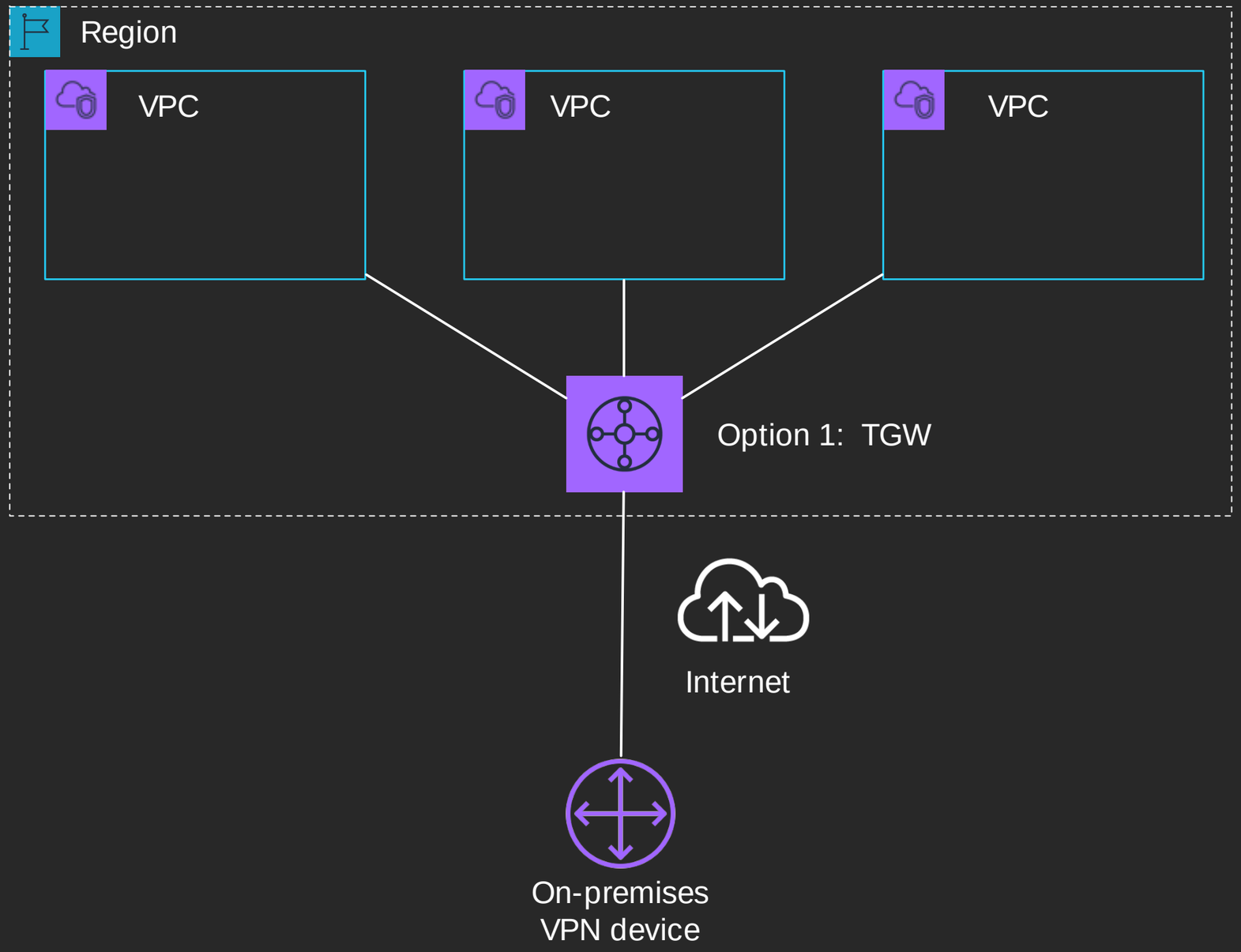
Centralized Interface VPC Endpoints

1. Reduces the cost associated with AWS PrivateLink endpoints
2. Data processing fees may increase depending on how clients connect to a centralized endpoint
3. Implementation of authorization and cost allocation needs to be considered
4. The endpoint's Private Hosted Zone cannot be associated outside of the VPC where the endpoint is attached

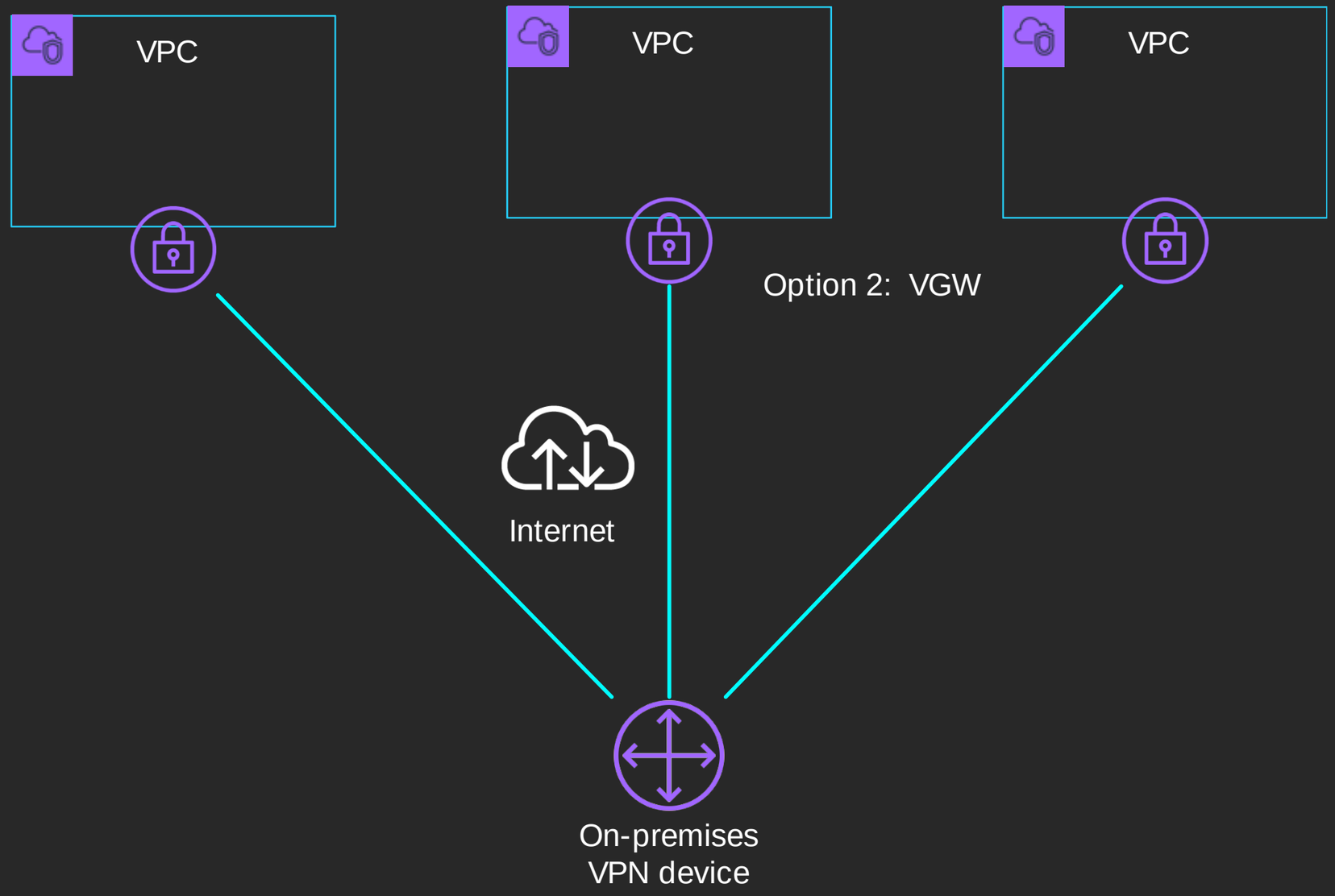


Hybrid connectivity

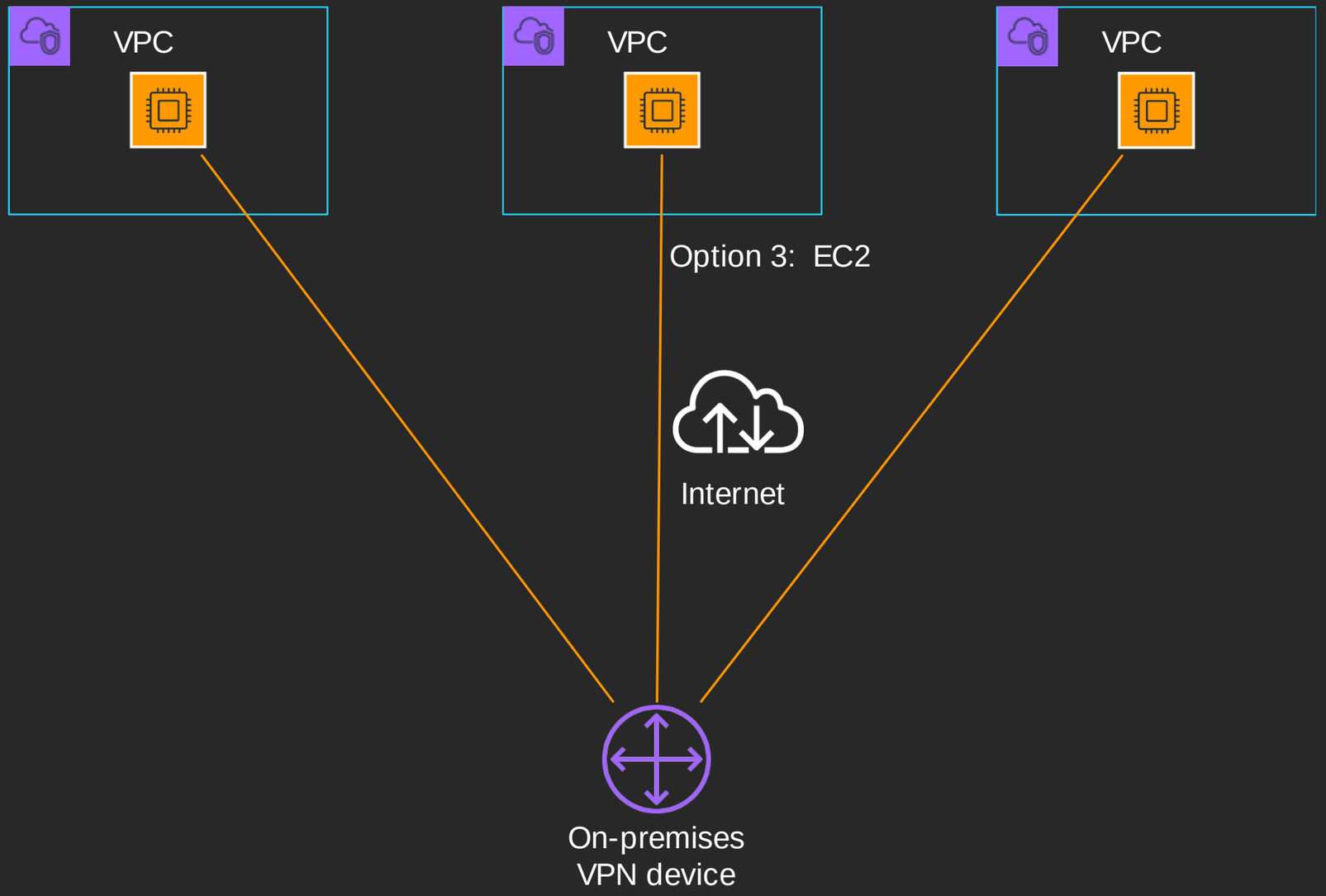
AWS site-to-site VPN with AWS Transit Gateway



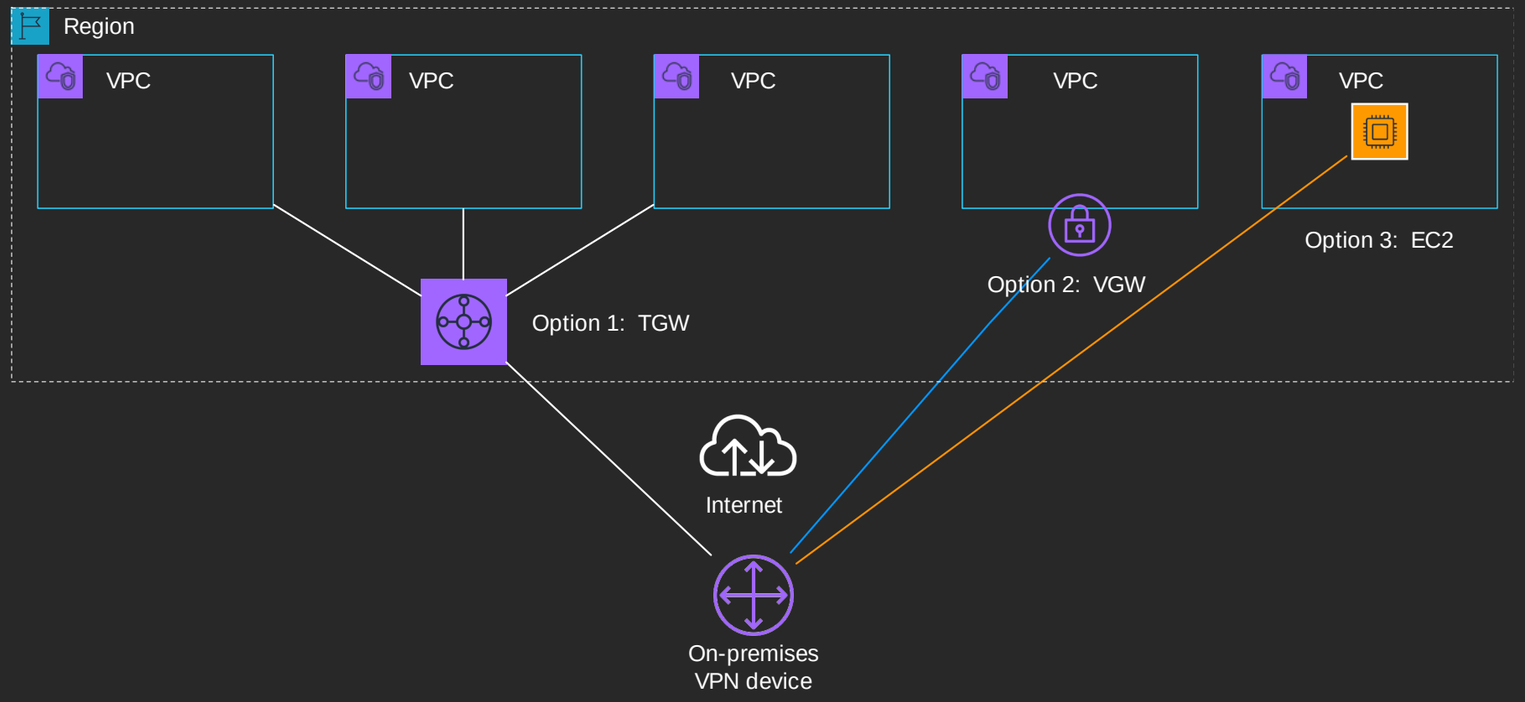
AWS site-to-site VPN with virtual private gateway



AWS site-to-site VPN with EC2 (software-based VPN)



AWS site-to-site VPN



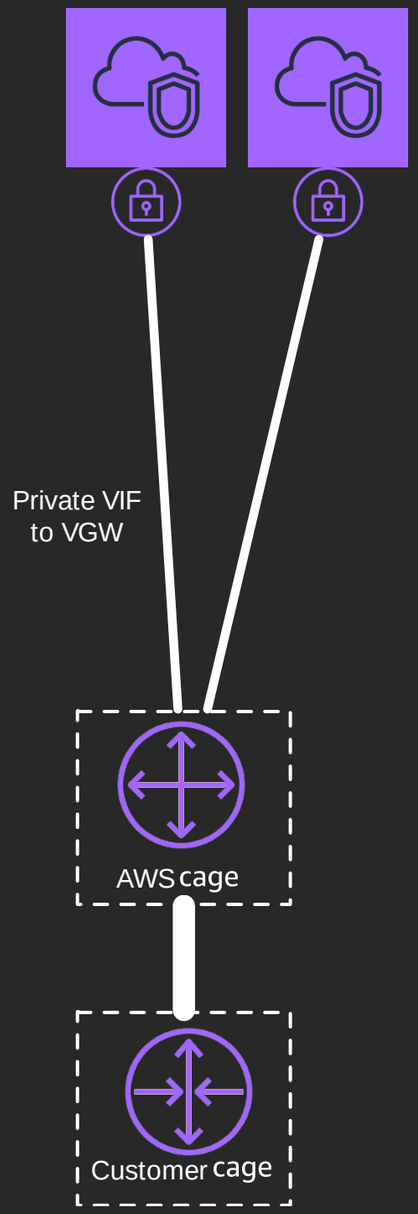
Limits

- 1 VPN connection = Many VPCs
- Throughput per tunnel = 1.25Gbps
- ECMP VPN support
  1.25 *2 = 2.5…1.25*8 = 10
- AWS Transit Gateway data processing fee

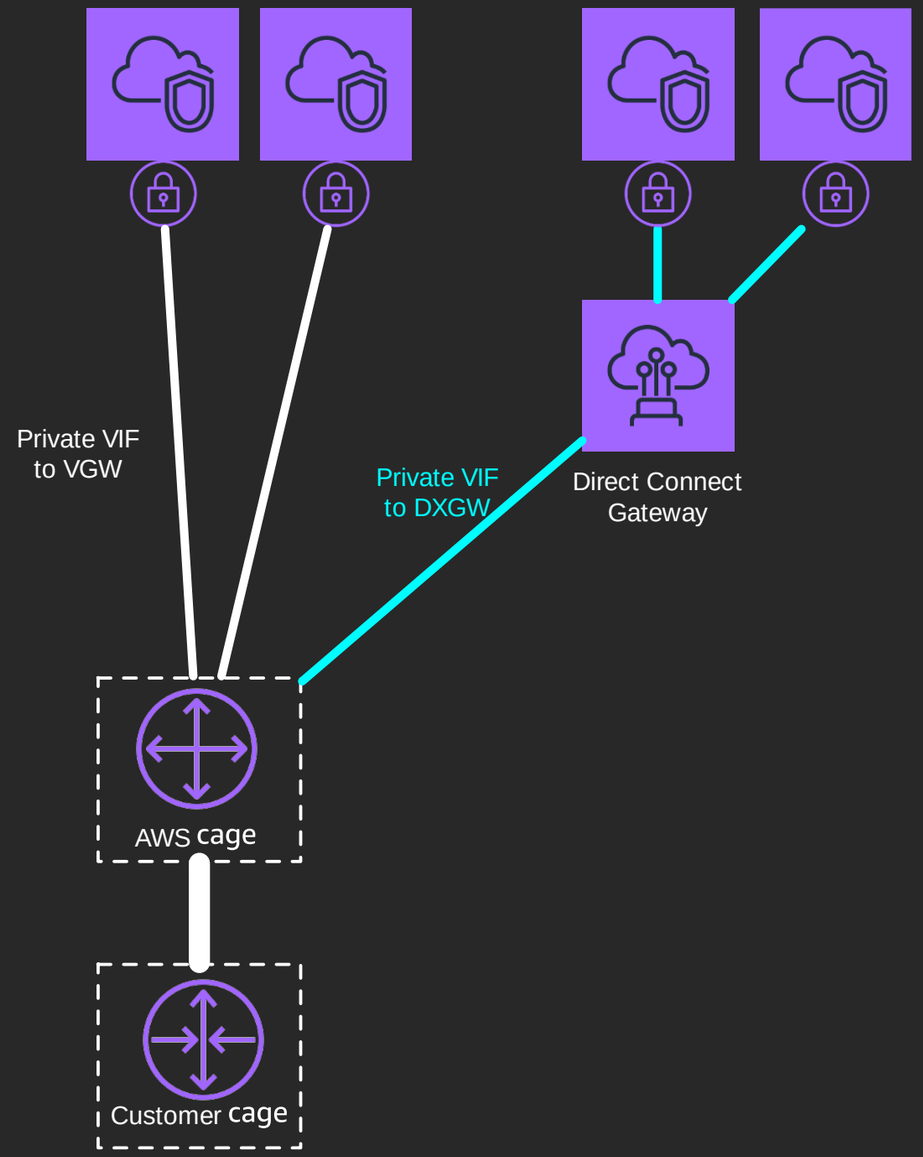
- 1 VPN connection = 1 VPC
- Throughput per tunnel = 1.25Gbps

- 1 VPN connection = 1 VPC
- Throughput depends on EC2 instance type & is vendor-specific
- Self-managed high availability and failover

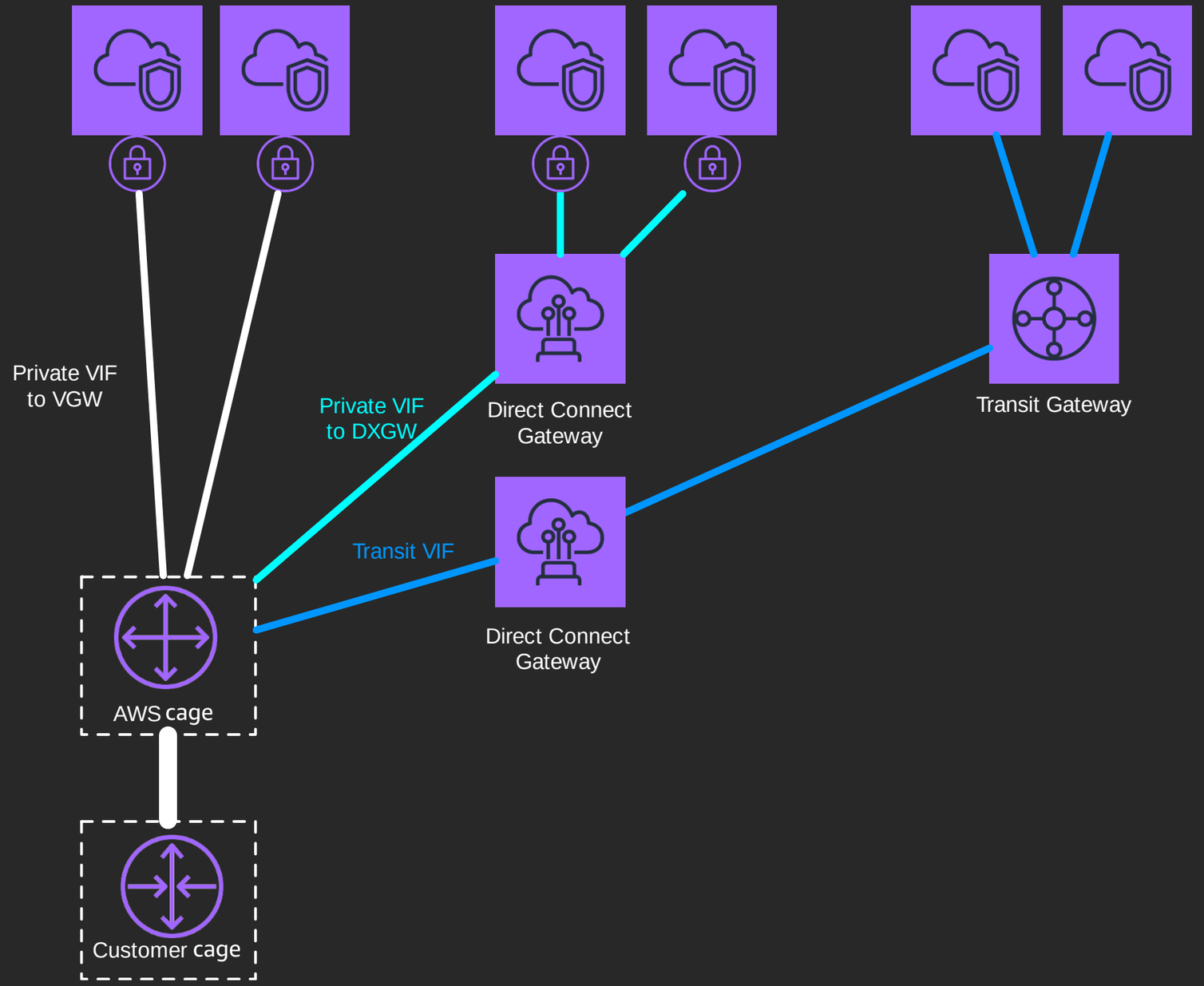
DX



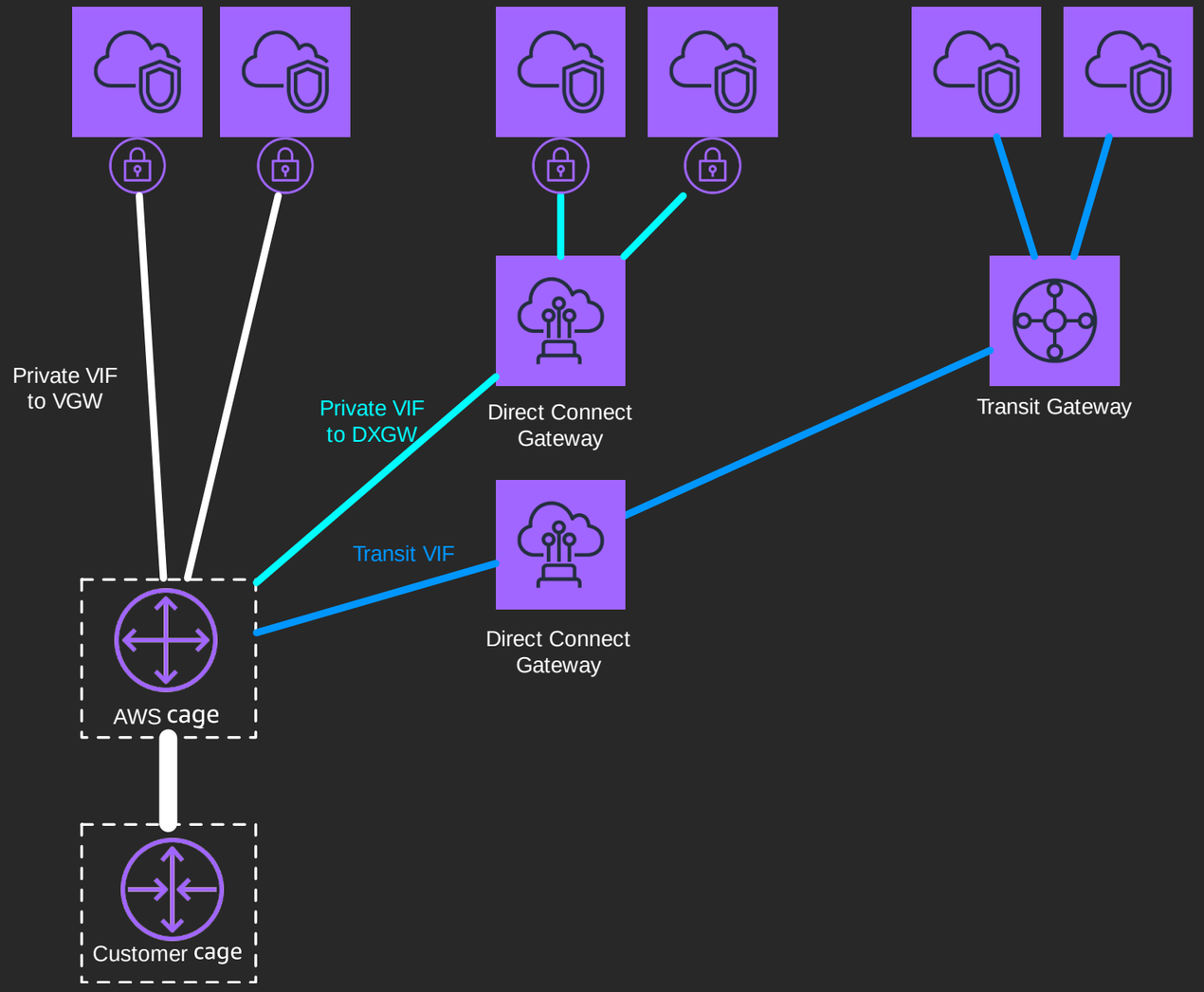
DX gateway


Private VIF
to VGW
Private VIF
to DXGW
Direct Connect
Gateway
AWS cage
Customer cage

DX gateway with AWS Transit Gateway



DX gateway with AWS Transit Gateway



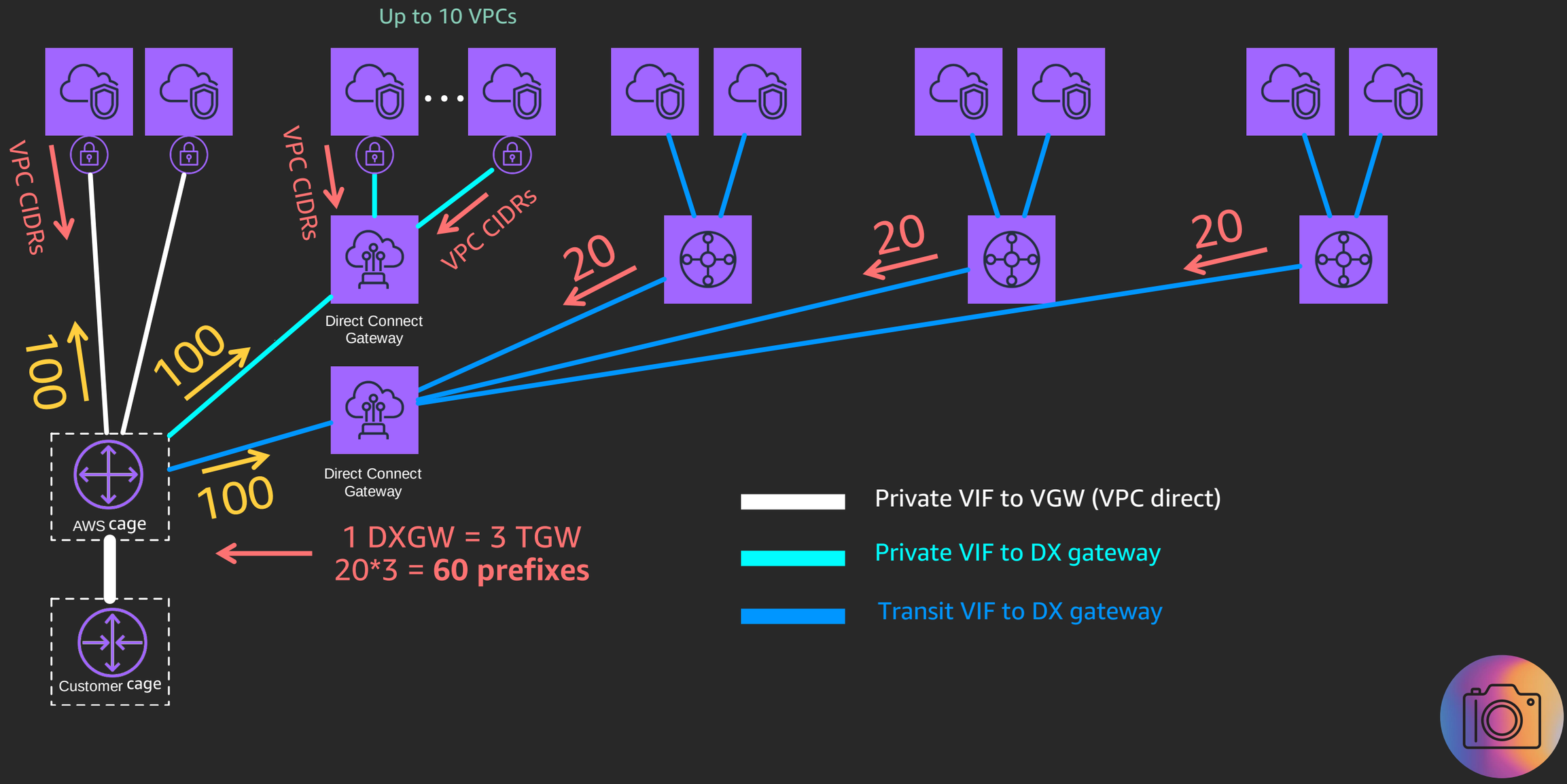
Limits

- 50 private virtual interface (VIF) per DX connection
  1 private = 1 VGW = 1 VPC
  1 BGP = 1 VPC

- 1 private = 1 DXGW
  1 DXGW = 10 VPC
  1 BGP = 10 VPC

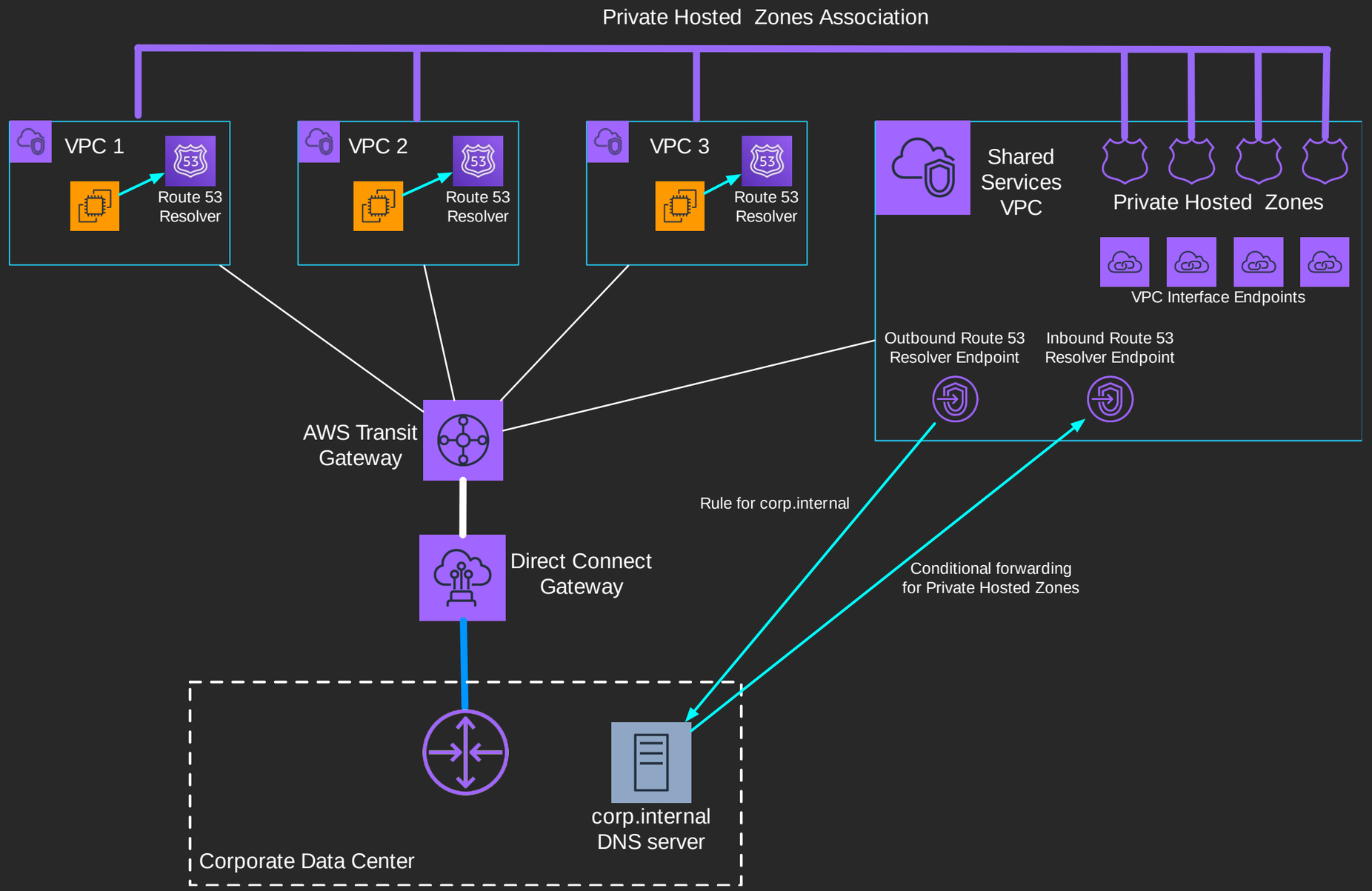
- 1 transit VIF per DX connection
  1 transit VIF = DXGW
  1 DXGW = 3 TGWs
  1 BGP = 1,000s of VPCs

DX gateway with AWS Transit Gateway



What about my DNS?

Hybrid DNS resolution using Amazon Route 53 Resolver



Hybrid DNS with Route 53 Resolver

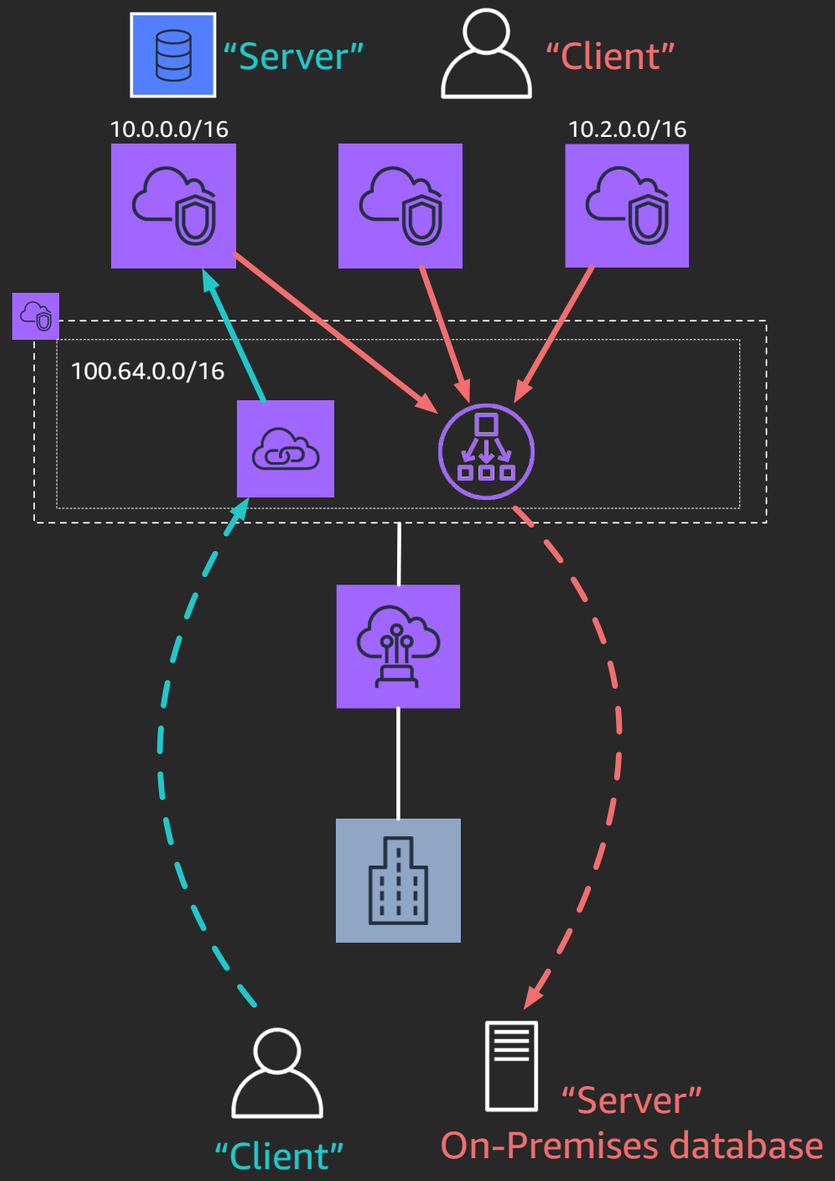
1. Scalable way to forward queries to alternate DNS resolvers
2. Not a substitute for Hosted Zone associations for inter-VPC resolution
3. Deploy Resolver endpoints (ENI's) in multiple AZs
4. Each endpoint ENI can handle ~10,000 QPS

Hybrid DNS whitepaper—Updated September 2019

https://d1.awsstatic.com/whitepapers/hybrid-cloud-dns-options-for-vpc.pdf

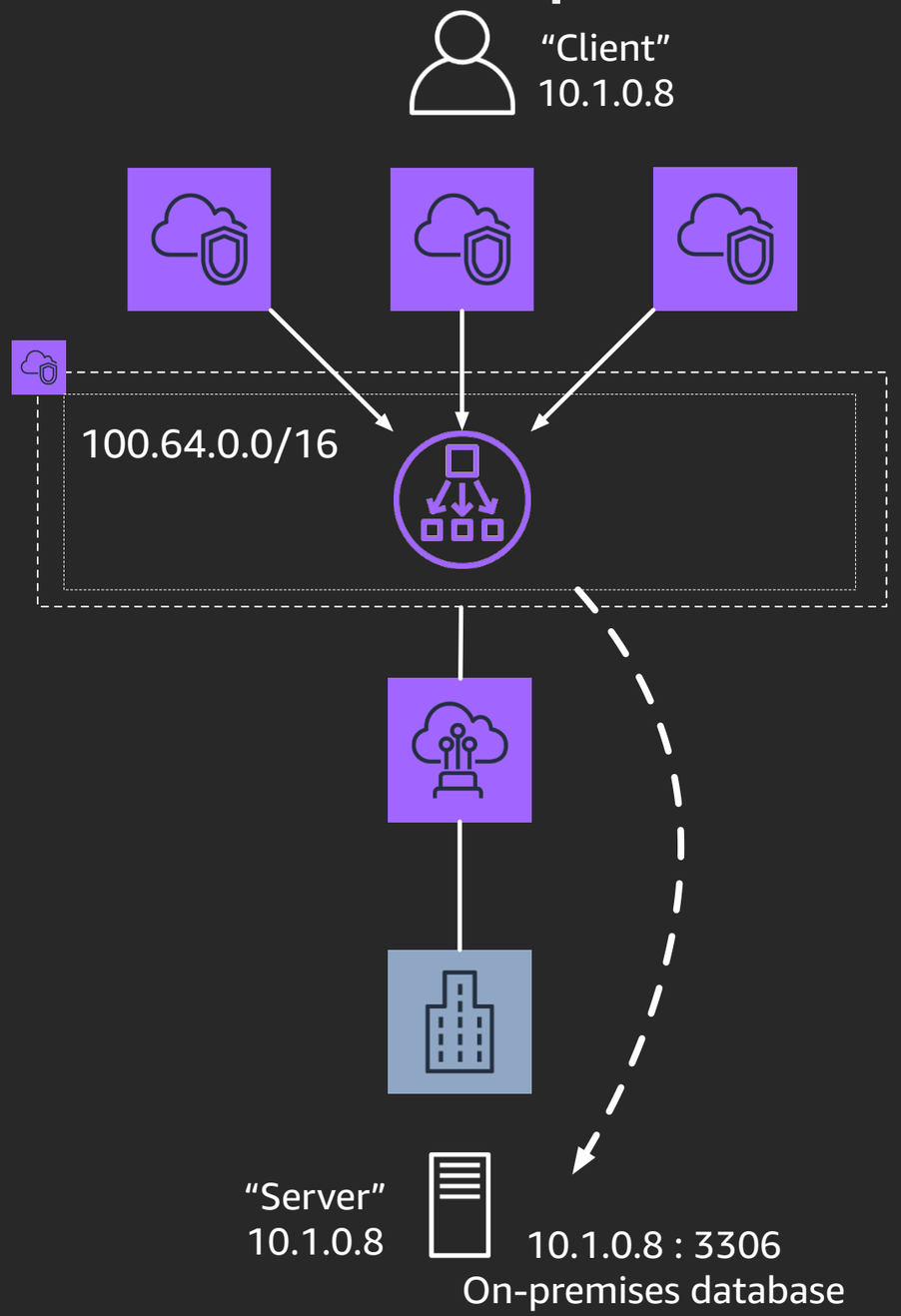
How do I access dependencies on-premises that have overlapping IPs?

Exposing hybrid services with overlapping IP networks

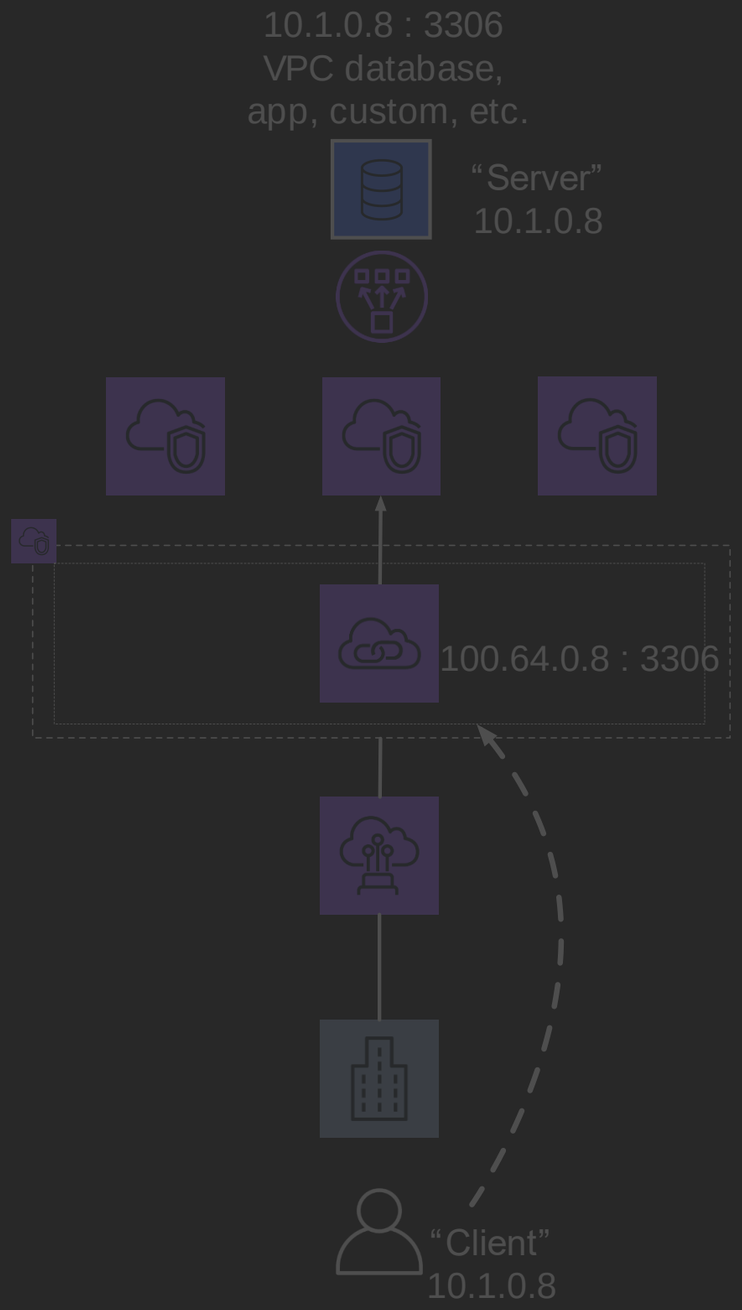


1. How many services do you want to expose on-premises, in the cloud?
2. What is the on-premises target? Is it redundant and scalable?
3. How does the target handle connections from the same source-IP address?
4. How is client/server identity maintained?

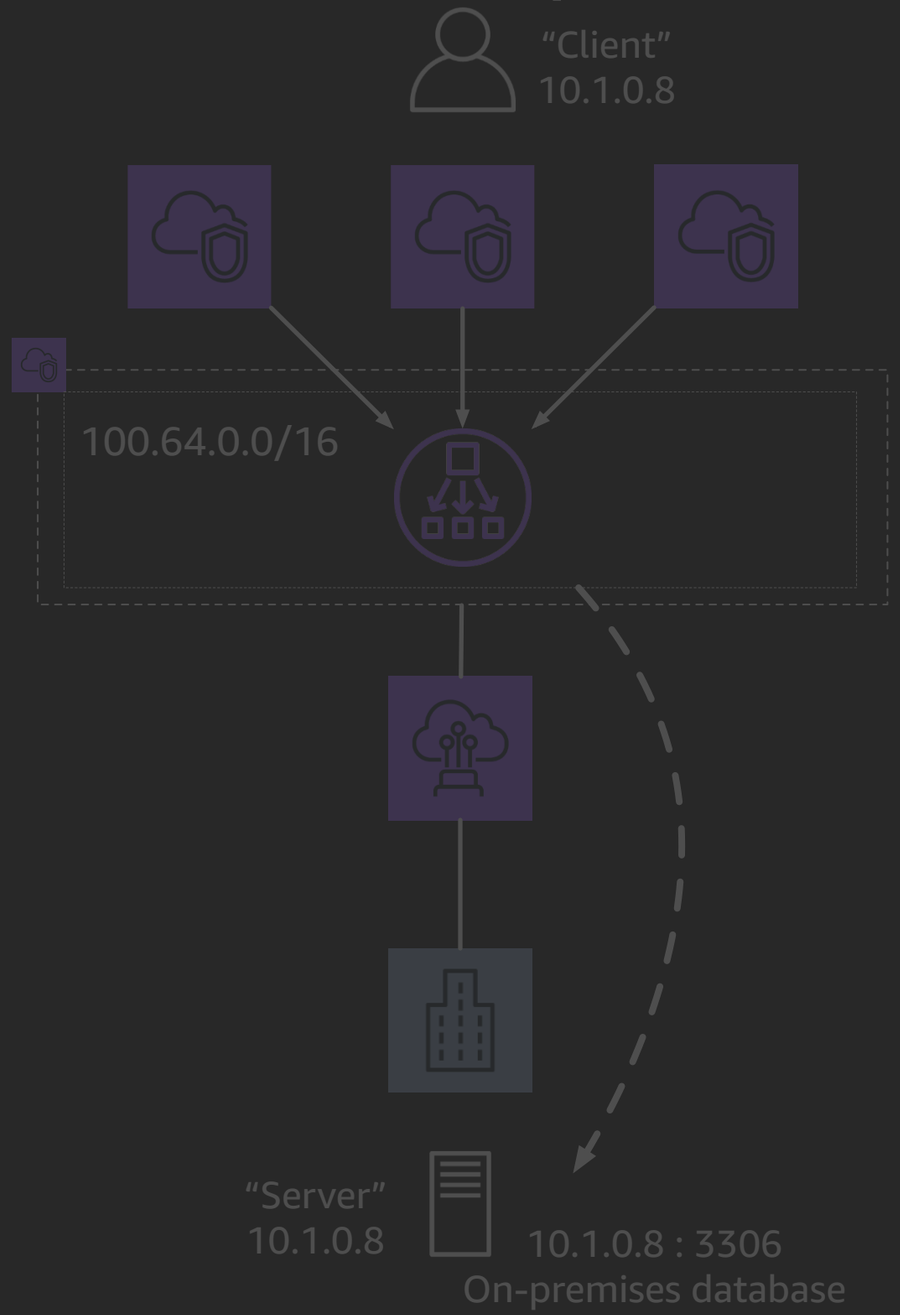
AWS to On-premises



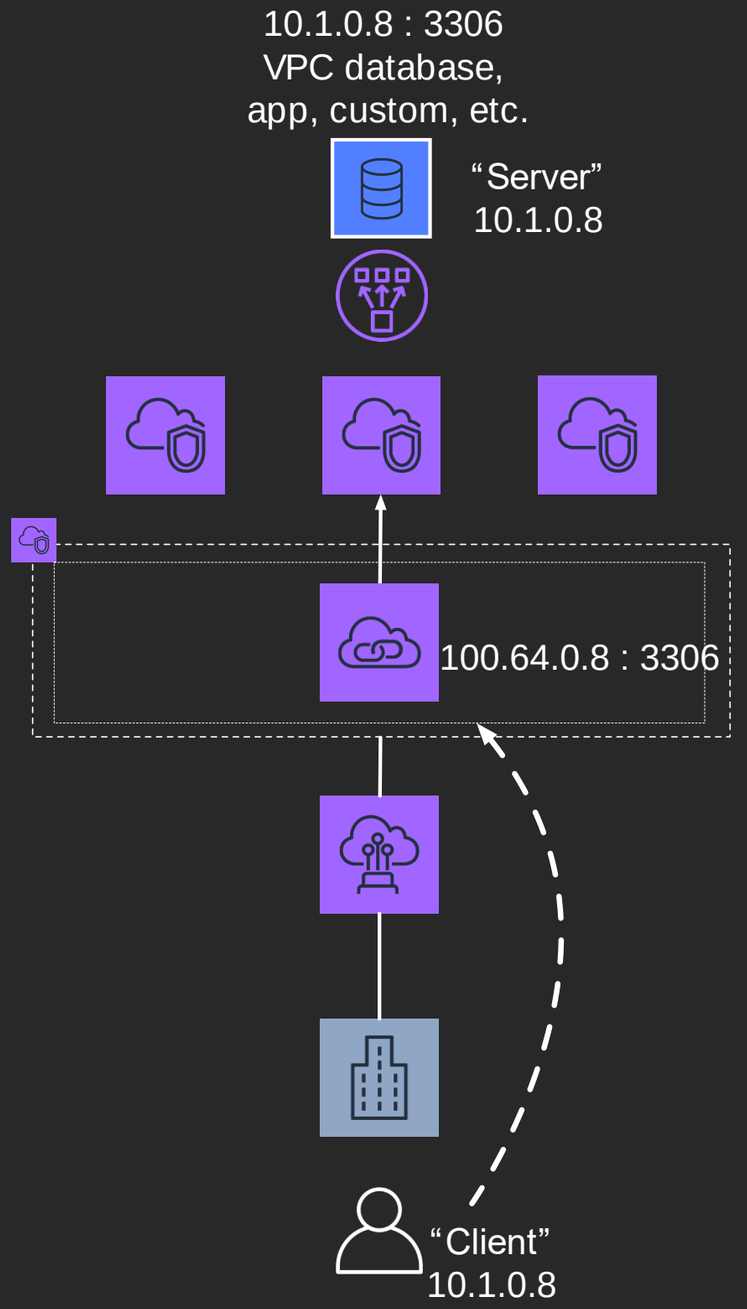
On-premises to AWS



AWS to On-premises



On-premises to AWS



How to be a solutions architect for your company

AWS re:Invent

aws

What is an SA?

"Someone who solves business problems with technical solutions by working with customers, developers, operations teams, and other stakeholders for a product or service.

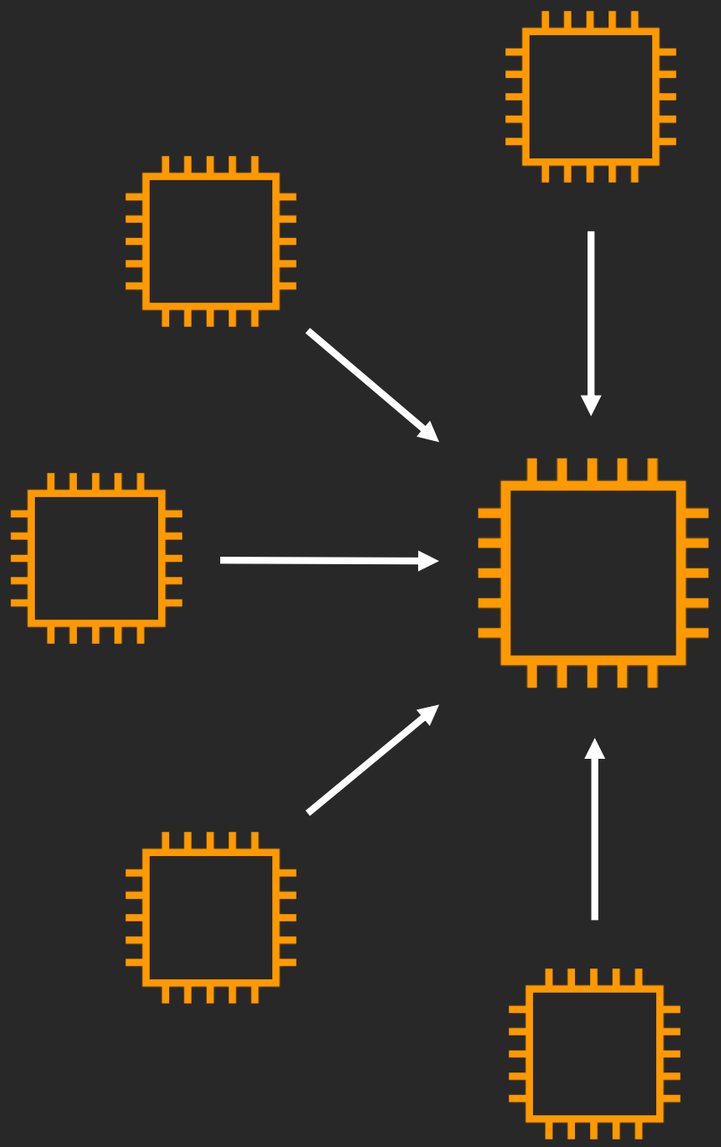
Question-asker, thinker, tinkerer, builder!"

-Bhavin Desai

How do I identify the *right architecture*? What questions should I be asking?

What are the scaling factors for the workload?

Throughput, latency, I/O, TPS?

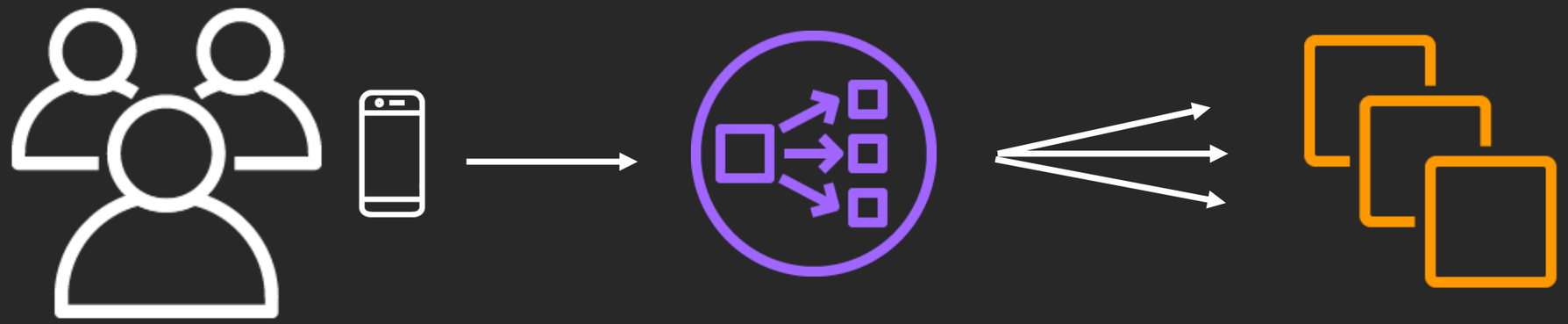


What is the dependency matrix?

How does (x) communicate with other services?

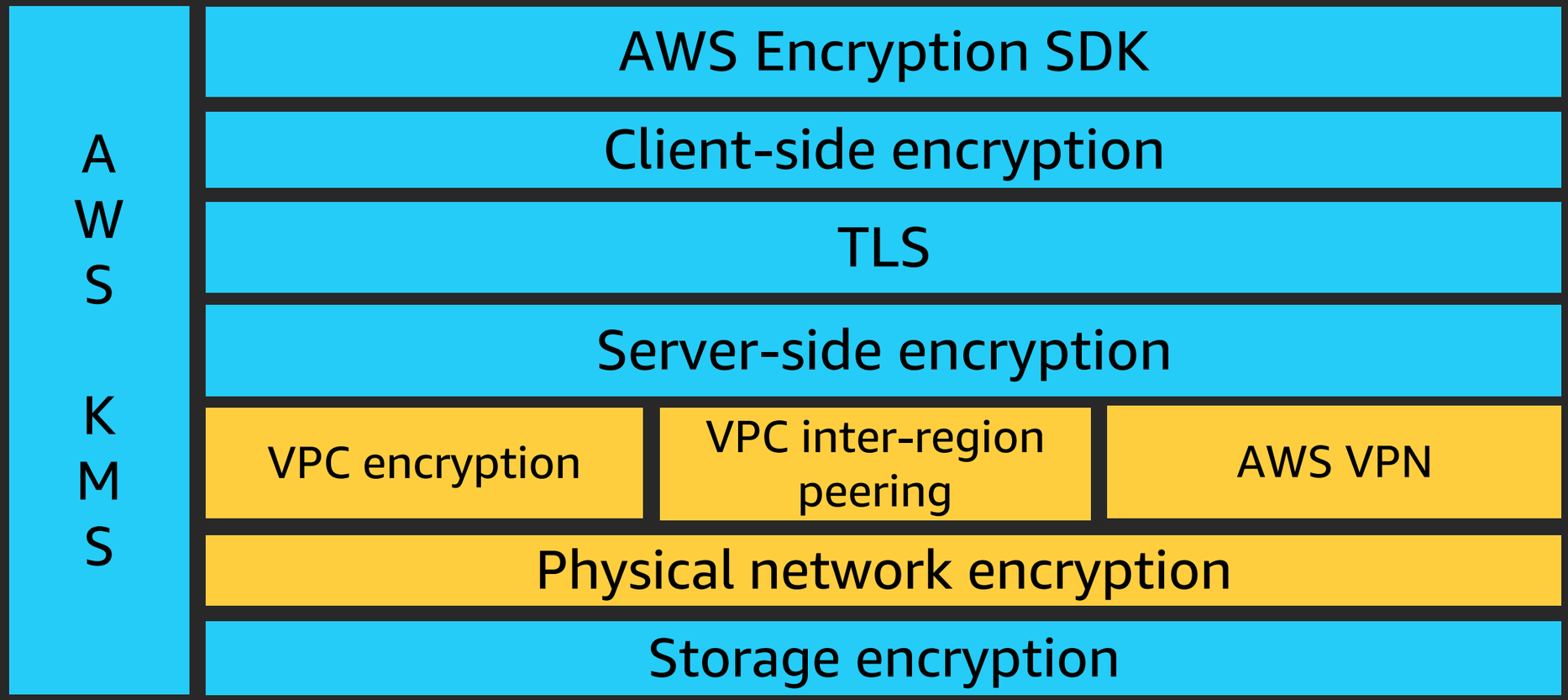
What is the source and what is the destination?

What type of transactions are involved?



Are there compliance or regulatory constraints?

Encryption at AWS



What is the applications SLA?

How do you handle degradation?

Is it better to fail some or degrade all?

General rules of thumb (designed for)

Single AZ: 99.9% availability

Single Region: 99.99% availability

Multi-Region

99.999% is possible

But many customers use multiple Regions for 99.95%

with a bounded recovery time of (x)

General rules of thumb (designed for)

Single AZ: 99.9% availability

Single Region: 99.99% availability

Multi-Region

99.999% is possible

But many customers use multiple Regions for 99.95%

with a bounded recovery time of (x)

Define your priorities

Define your priorities

Embrace “tiny bubbles”

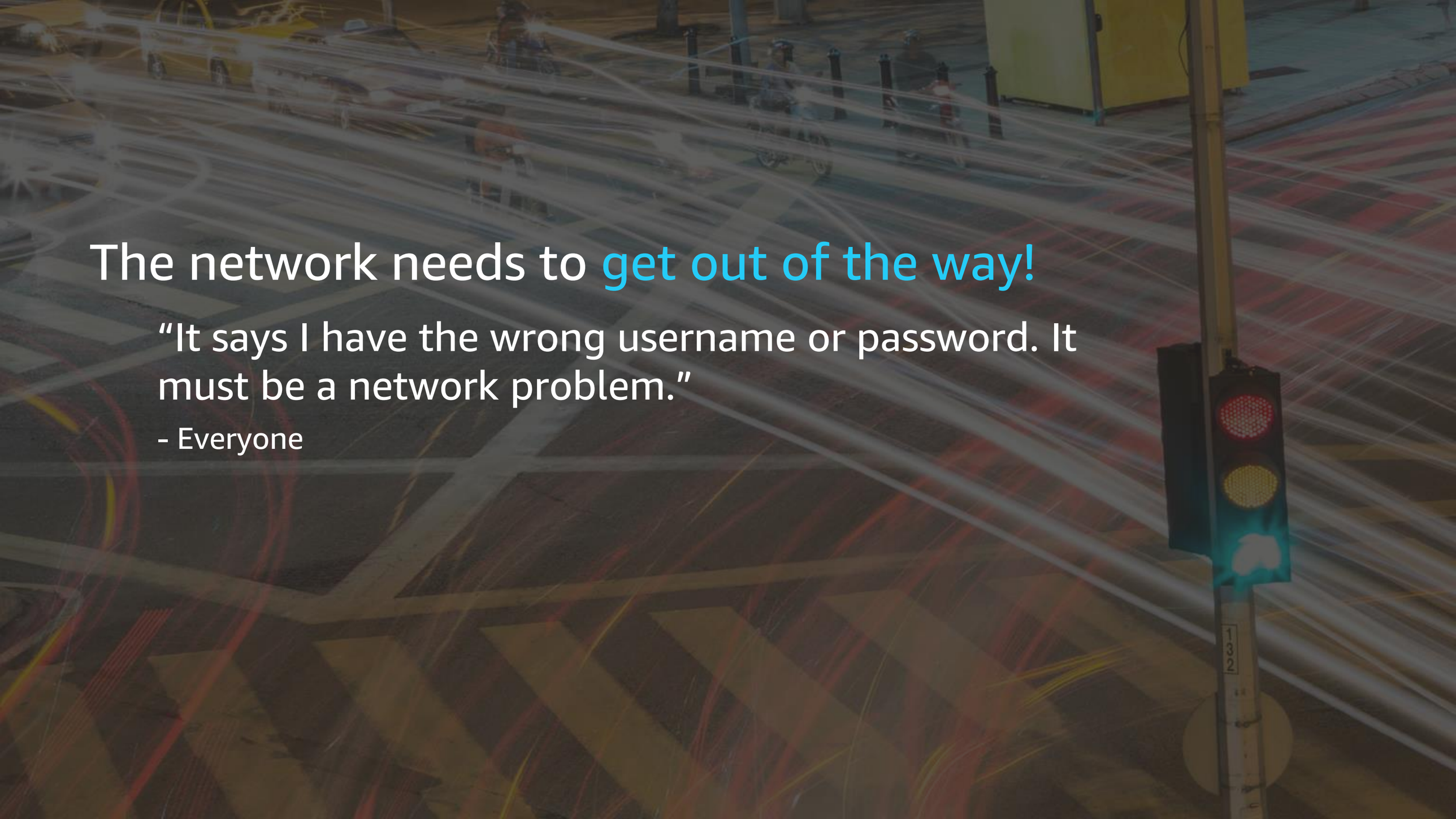
“You can’t stop the future. You can’t rewind the past. The only way to learn the secret … is to press play.”

- Jay Asher, Thirteen Reasons Why

Embrace “tiny bubbles”

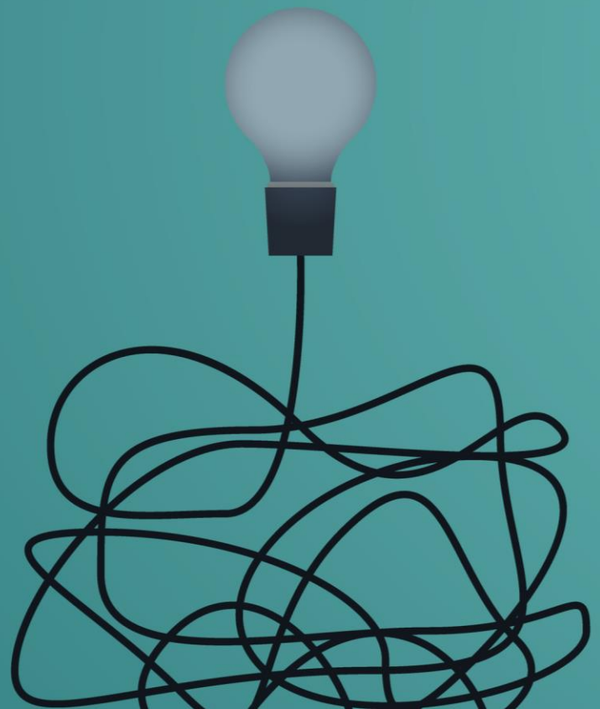
“You can’t stop the future. You can’t rewind the past. The only way to learn the secret … is to press play.”

- Jay Asher, Thirteen Reasons Why



The network needs to get out of the way!

> “It says I have the wrong username or password. It must be a network problem.”
>
> - Everyone



Keep it simple, silly!

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!

AWS re:Invent

aws

Please complete the session survey in the mobile app.