aws SUMMIT ONLINE

3 2

AWS Networking

Building your network from 0 to millions of clients

Sébastien Stormacq

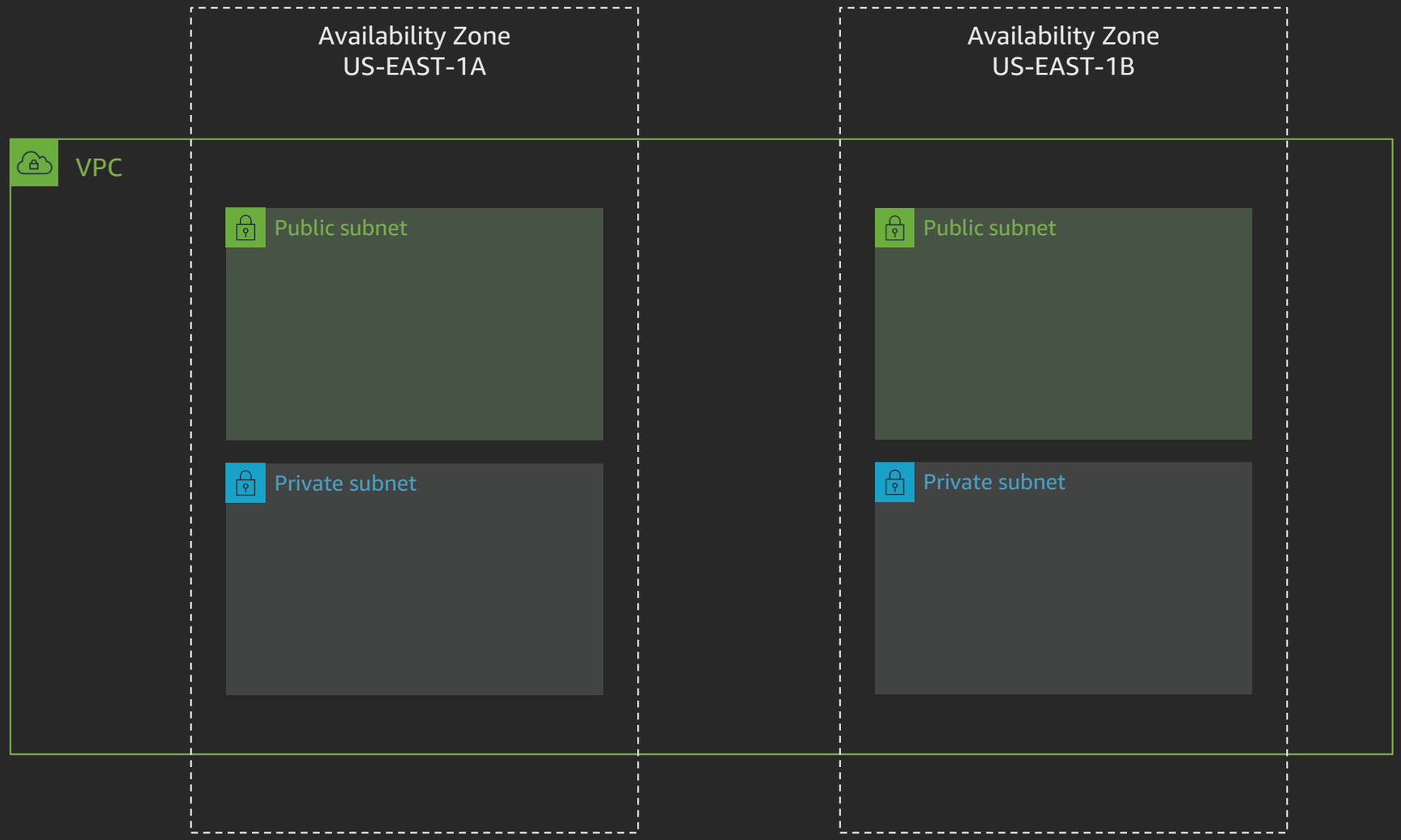
Developer Advocate

Amazon Web Services

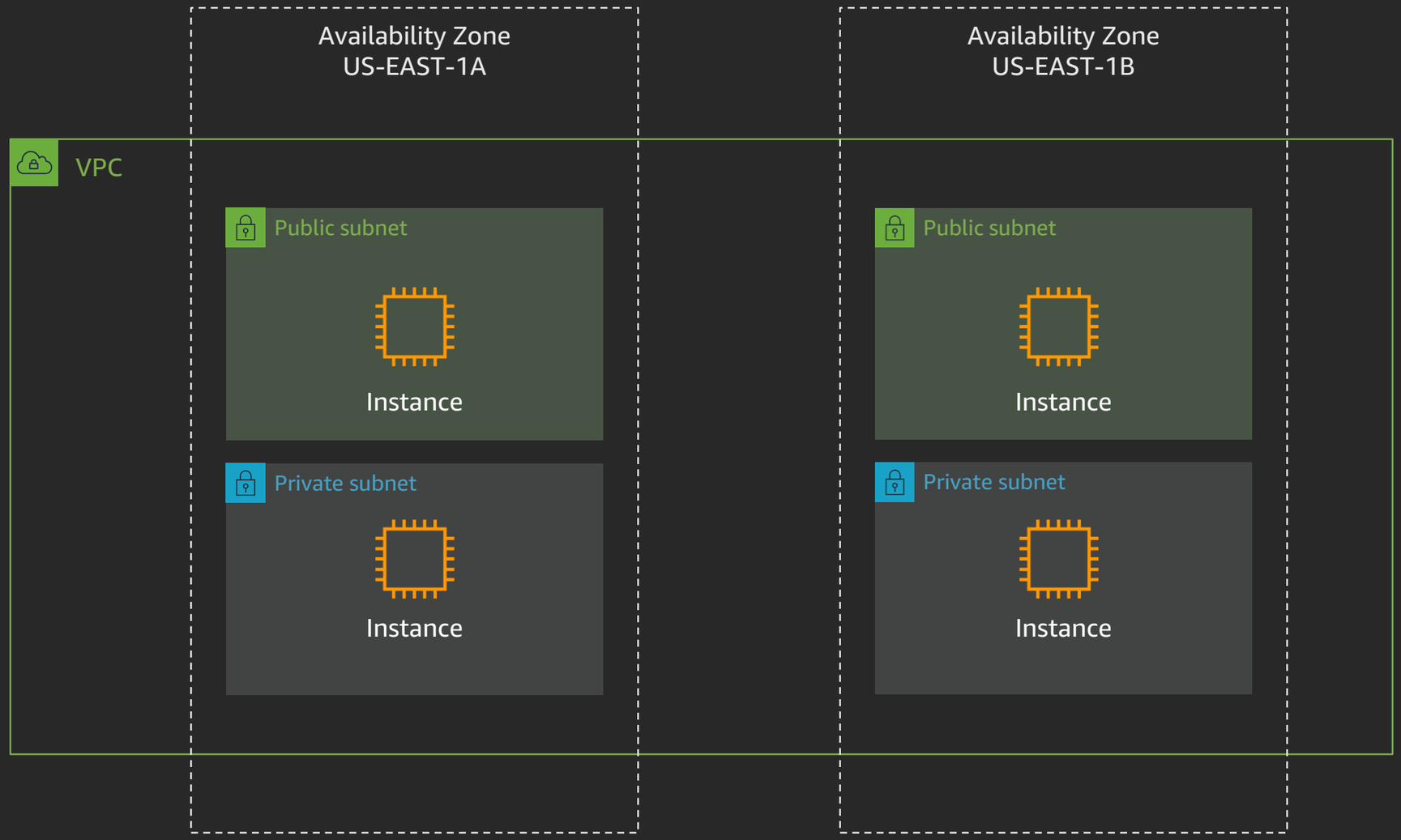
aws SUMMIT ONLINE

Amazon Virtual Private Cloud

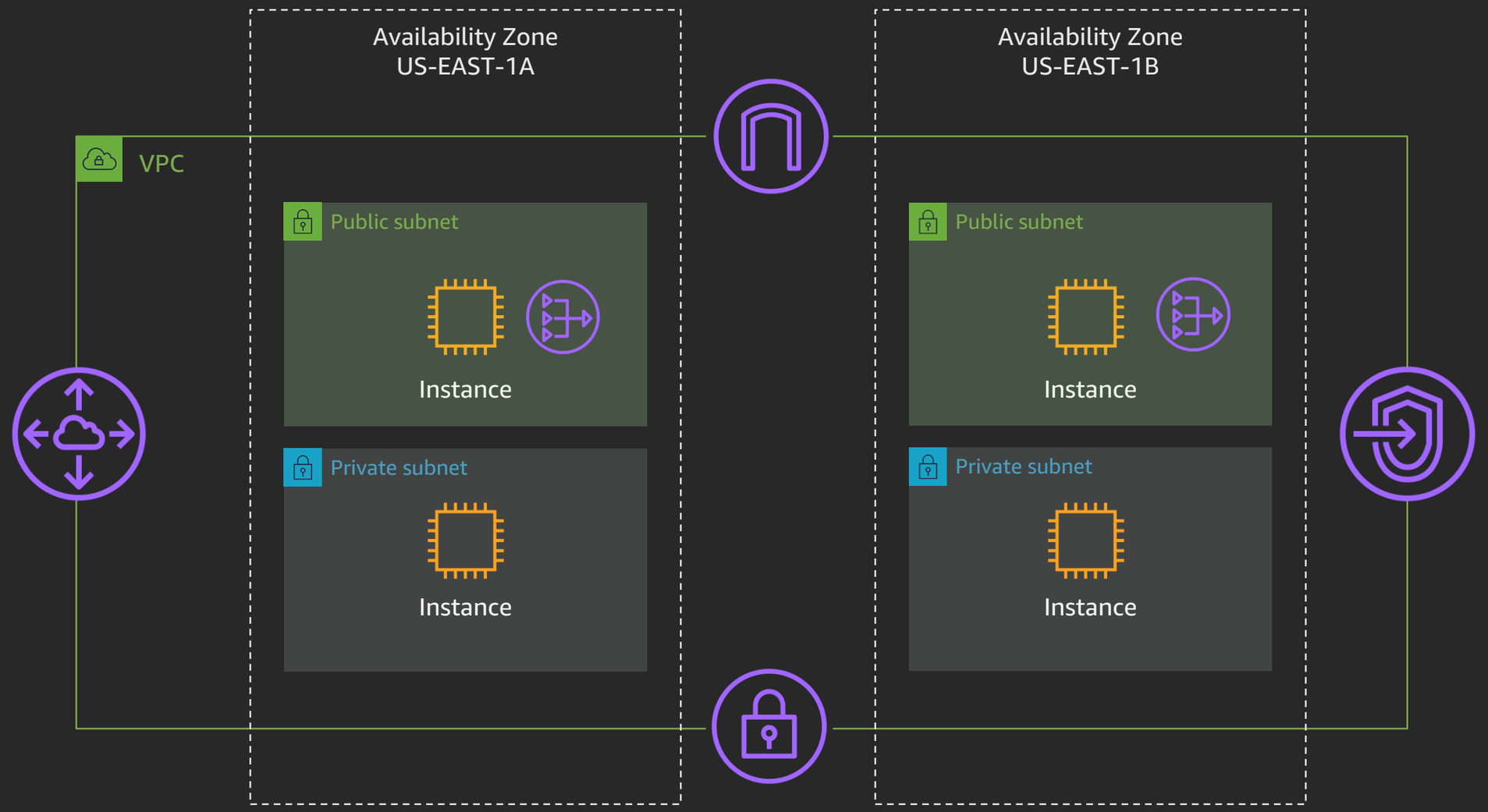
Subnets


Availability Zone
US-EAST-1A
Availability Zone
US-EAST-1B
VPC
Public subnet
Private subnet
Public subnet
Private subnet

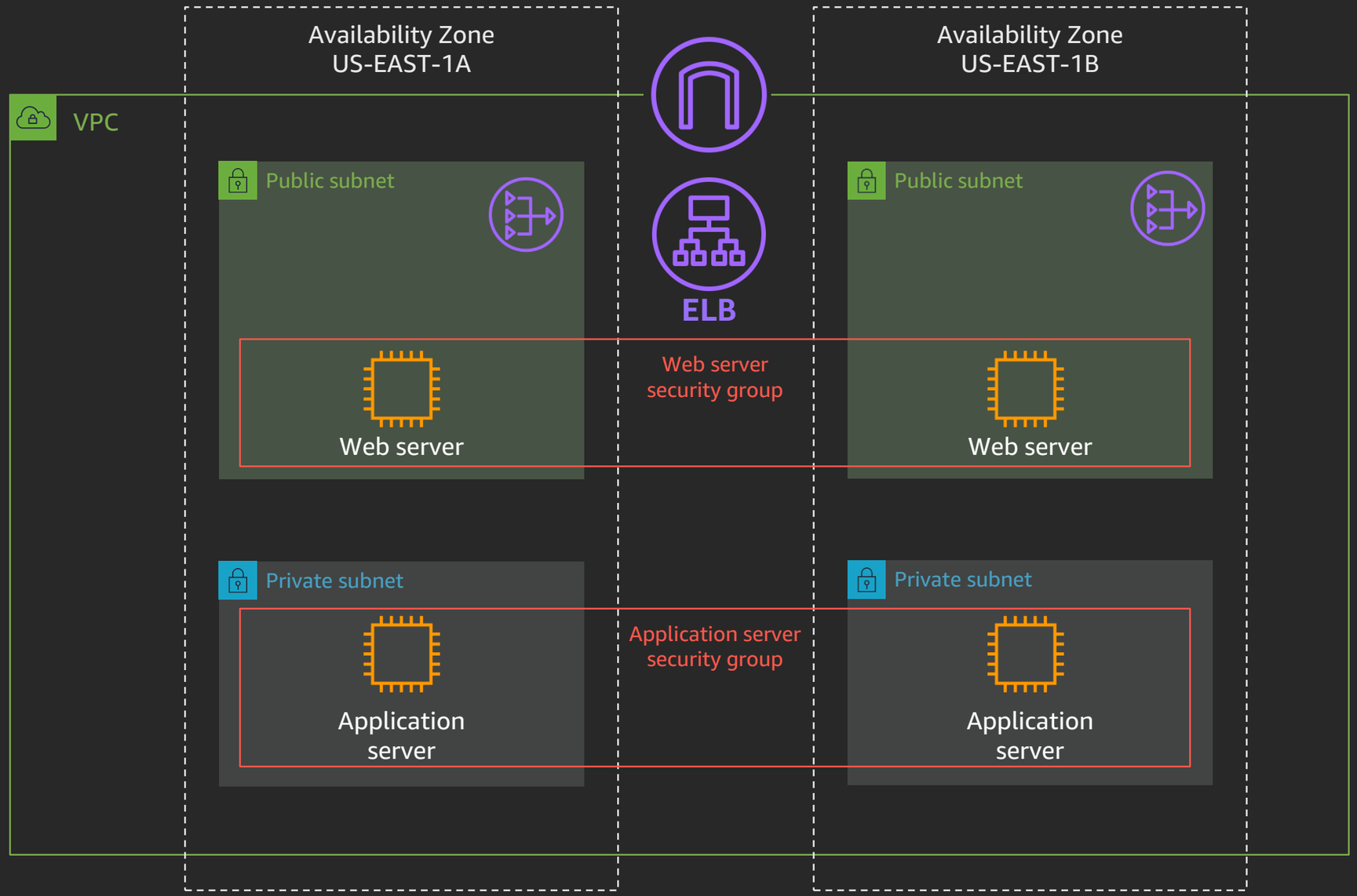
EC2 instances



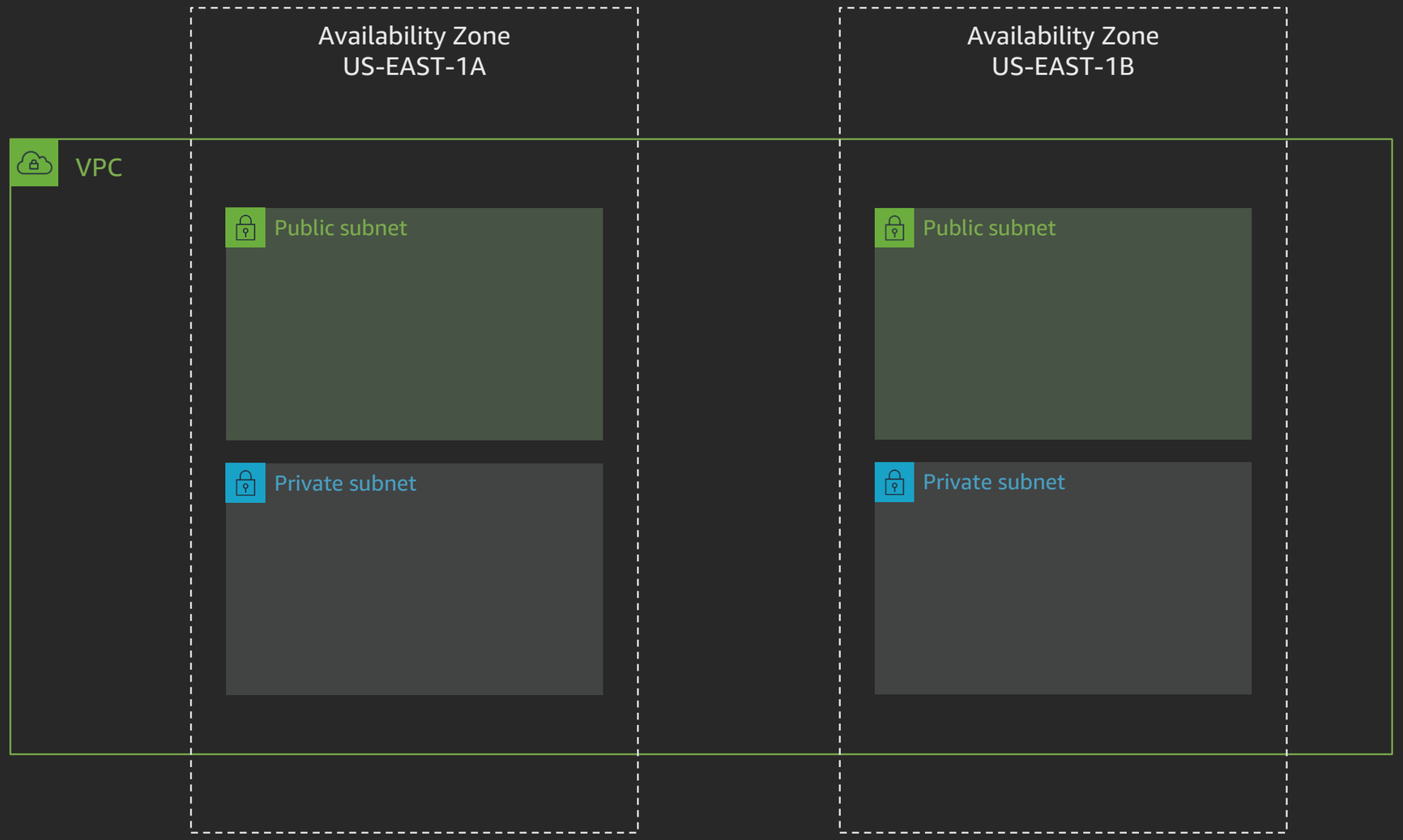
Gateways, endpoints & peering



Example web application



IP addressing



Private IP address range for your VPC – IPv4

- "CIDR" range?
  - Classless inter-domain routing
  - No more class A, B, C
- RFC1918
  - 192.168.0.0 /16
  - 172.16.0.0 /12
  - 10.0.0.0 /8
- How much?
  - /16
  - /28

Updated by: 6761 BEST CURRENT PRACTICE
Errata Exist

Network Working Group — Y. Rekhter, Cisco Systems
Request for Comments: 1918 — B. Moskowitz, Chrysler Corp.
Obsoletes: 1627, 1597 — D. Karrenberg, RIPE NCC
BCP: 5 — G. J. de Groot, RIPE NCC
Category: Best Current Practice — E. Lear, Silicon Graphics, Inc.
February 1996

Address Allocation for Private Internets

Status of this Memo

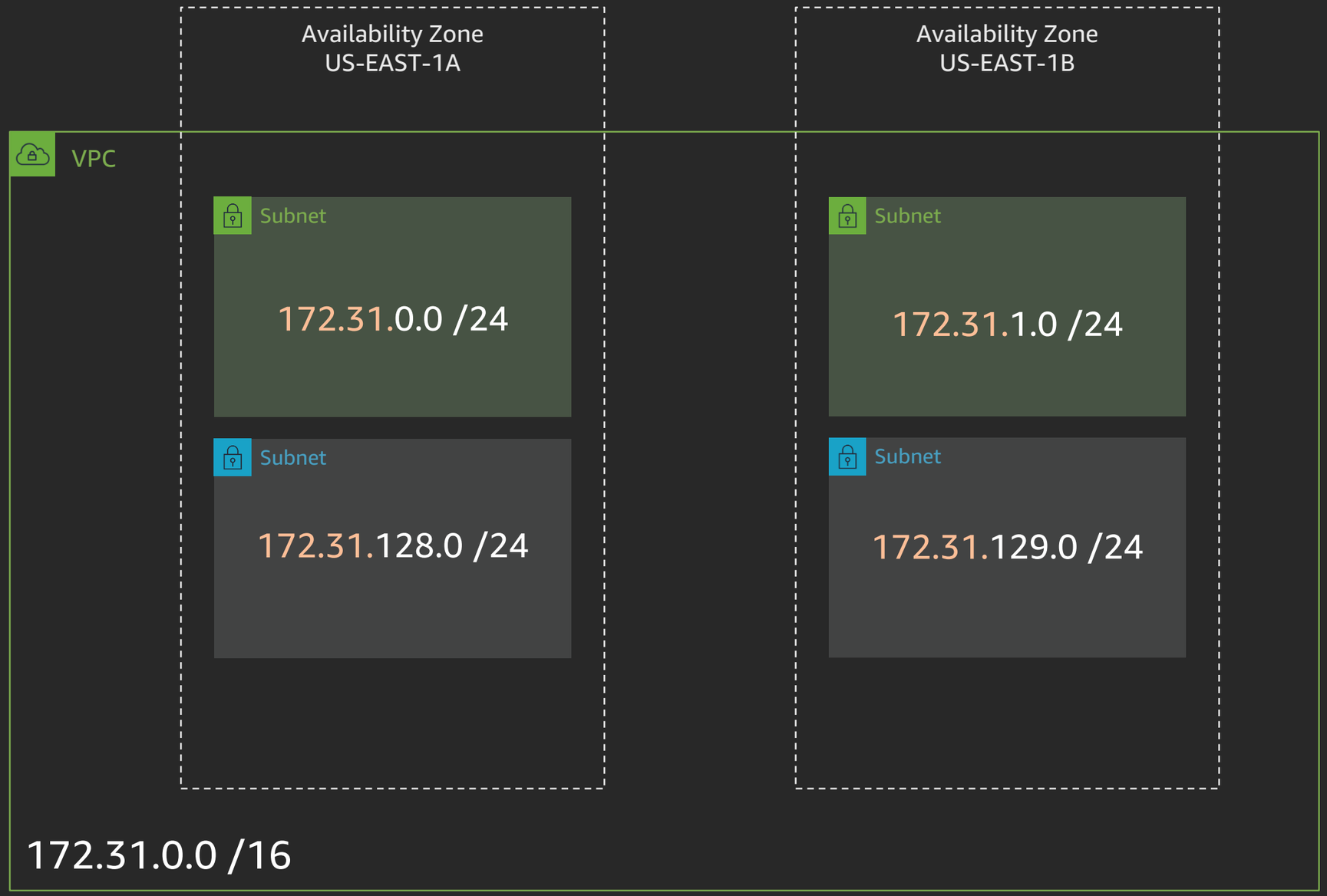
This document specifies an Internet Best Current Practices for the Internet Community, and requests discussion and suggestions for improvements. Distribution of this memo is unlimited.

1. Introduction

For the purposes of this document, an enterprise is an entity autonomously operating a network using TCP/IP and in particular determining the addressing plan and address assignments within that network.

This document describes address allocation for private internets. The allocation permits full network layer connectivity among all hosts inside an enterprise as well as among all public hosts of different enterprises. The cost of using private internet address space is the potentially costly effort to renumber hosts and networks between public and private.

Where to use IPv4 addresses?



IPv6 basics

IPv6: Colon-separated hextet notation + CIDR

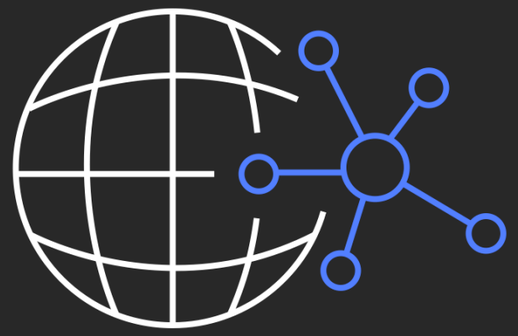
2001:0db8:0ec2:0000:0000:0000:0000:0001/64	0000:0000:0000:0000:0000:0000:0000:0001/128
2001:db8:ec2:0:0:0:0:1/64	0:0:0:0:0:0:0:1/128
2001:db8:ec2::1/64	::1/128

Unicast addresses

Loopback address	::1
Link local address (LLA)	fe80::/10 (fe80::/64 in practice)
Global unicast address (GUA)	2600:1f16:14d:6300::/64

Multicast addresses (ff00::/8)

All nodes	ff02::1
All routers	ff02::2
Solicited node	ff02::1:ff00:0/104



IPv6 on AWS

- /56 VPC
- /64 subnets
- Dual stack
- Link local address and global unicast address required

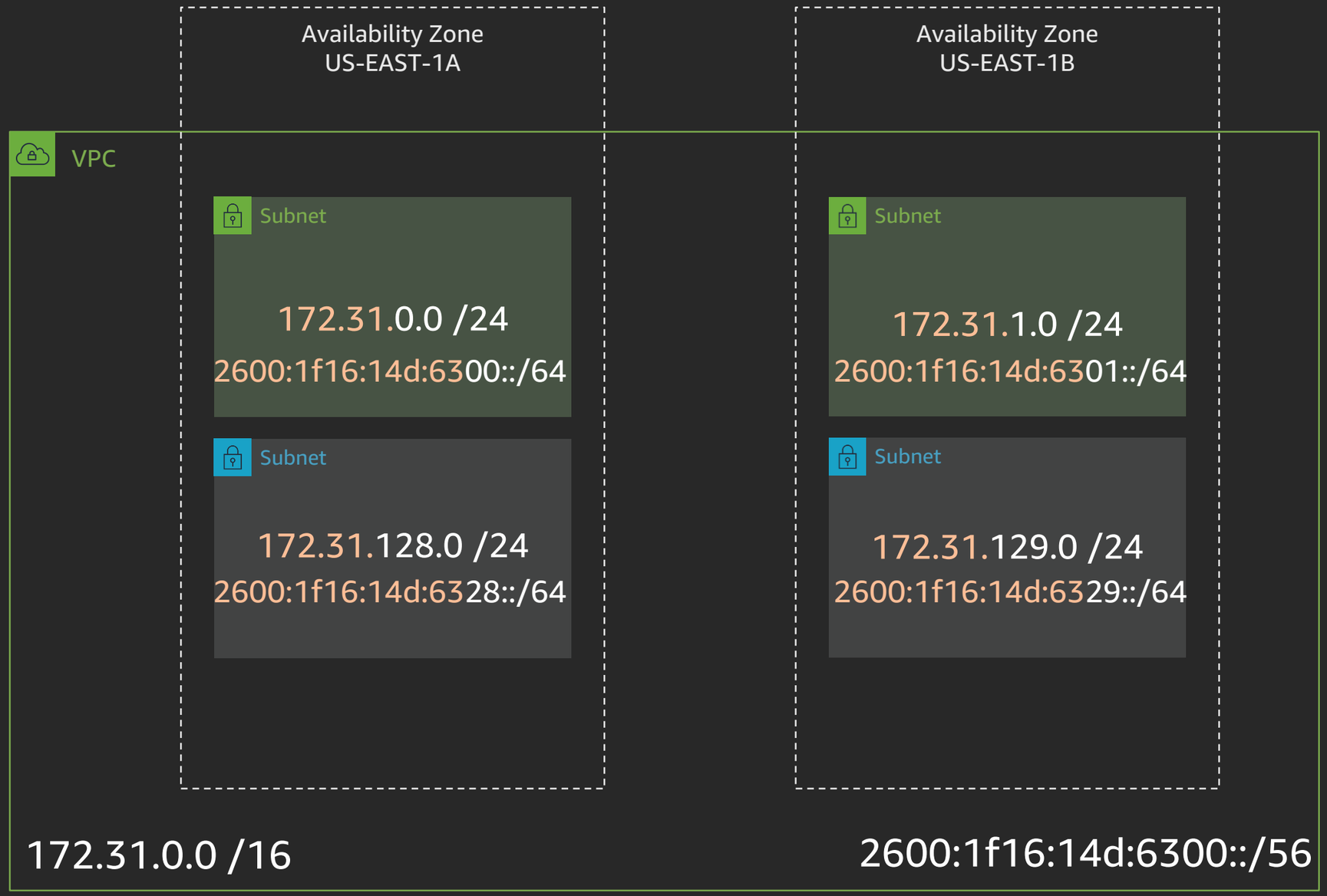
```
$ ifconfig
eth0      Link encap:Ethernet  HWaddr 0E:A2:04:52:2A:44
          inet addr:172.31.0.250  Bcast:172.31.0.255  Mask:255.255.255.0
          inet6 addr: fe80::ca2:4ff:fe52:2a44/64 Scope:Link
          inet6 addr: 2600:1f16:14d:6300:7965:9a71:653a:822b/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:9001  Metric:1
          RX packets:35090 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12411 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:49899286 (47.5 MiB)  TX bytes:840649 (820.9 KiB)
```

IPv4 private address

IPv6 link local address (private)

IPv6 global unicast address (public)

Where to use IPv6 addresses?

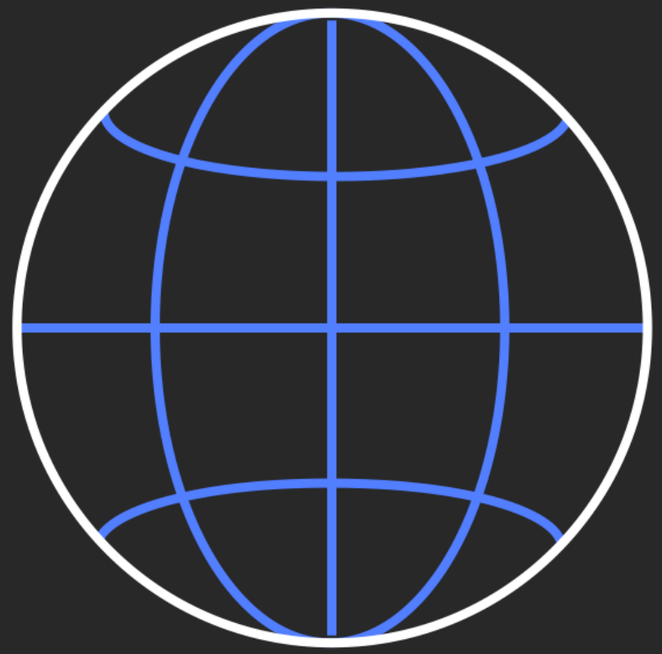


The “5 things” required for internet traffic

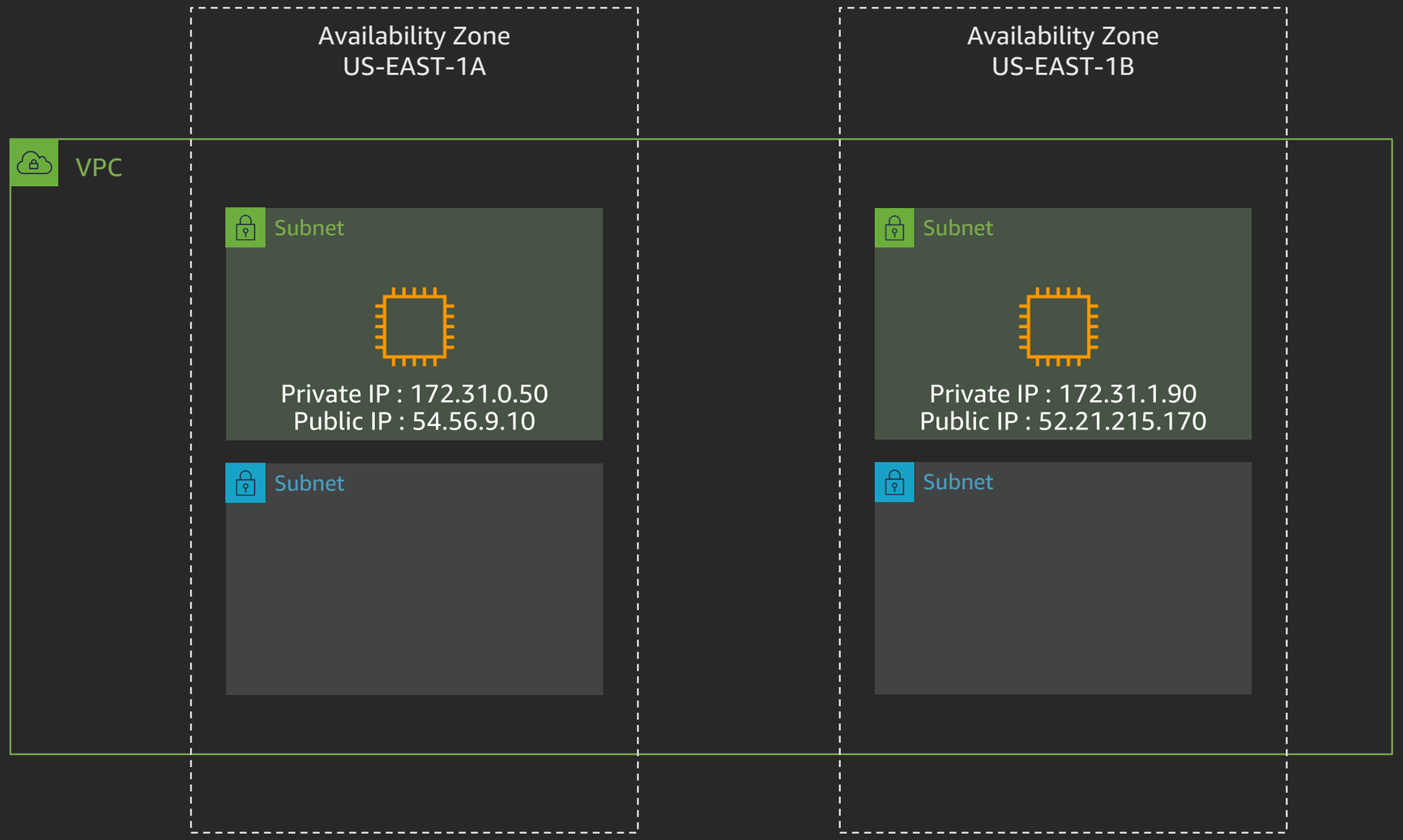
1. Public IP address
2. Internet gateway attached to a VPC
3. Route to an internet gateway
4. Network ACL Allow rule
5. Security group Allow rule

Public IP addresses for your instances

- Auto-assign public IP addresses
- Elastic IP addresses
  - Amazon Elastic IP address pool
  - Bring Your Own IP (BYOIP) pool



Public IP addresses



Internet access

172.16.0.0
172.16.1.0
172.16.2.0

Create internet gateway | Actions

Filter by tags and attributes or search by keyword

Name	ID	State	VPC
	igw-09ef761d872b…	attached	vpc-0bcb5110cf0c…

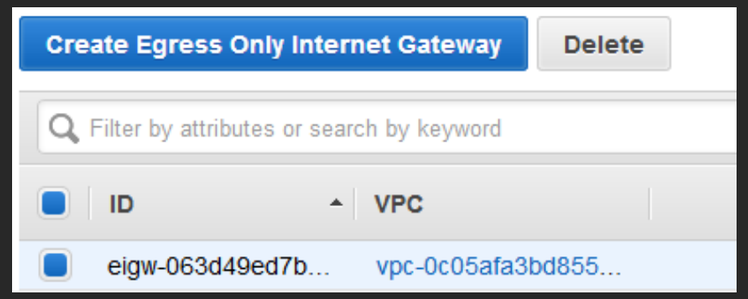
Destination	Target	Status	Propagated
172.31.0.0/16	local	Active	No
2600:1f16:14d:6300::/56	local	Active	No
0.0.0.0/0	igw-09ef761d872bd7540	Active	No
::/0	igw-09ef761d872bd7540	Active	No

"To get to the IPv4 internet (0.0.0.0/0) go via the internet gateway"

"To get to the IPv6 internet (::/0) go via the internet gateway"

Internet access

172.16.0.0
172.16.1.0
172.16.2.0



Destination	Target	Status	Propagated
172.31.0.0/16	local	Active	No
2600:1f16:14d:6300::/56	local	Active	No
0.0.0.0/0	igw-09ef761d872bd7540	Active	No
::/0	eigw-063d49ed7bb0f8c36	Active	No

"To get to the IPv6 internet (::/0) go via the egress-only internet gateway (EIGW)"

Different routes for different subnets

172.16.0.0
172.16.1.0
172.16.2.0

Public subnet

Destination	Target	Status	Propagated
172.31.0.0/16	local	Active	No
2600:1f16:14d:6300::/56	local	Active	No
0.0.0.0/0	igw-09ef761d872bd7540	Active	No
::/0	igw-09ef761d872bd7540	Active	No

"To get to the internet, go via the internet gateway"

Private subnet

Destination	Target	Status	Propagated
172.31.0.0/16	local	Active	No
2600:1f16:14d:6300::/56	local	Active	No

"To get to anything in the VPC, stay local. No route anywhere else."

Public & private subnets

Private subnet

A

Private IP : 172.31.128.75

Public subnet

B

Private IP : 172.31.0.50
Public IP : 54.56.9.10

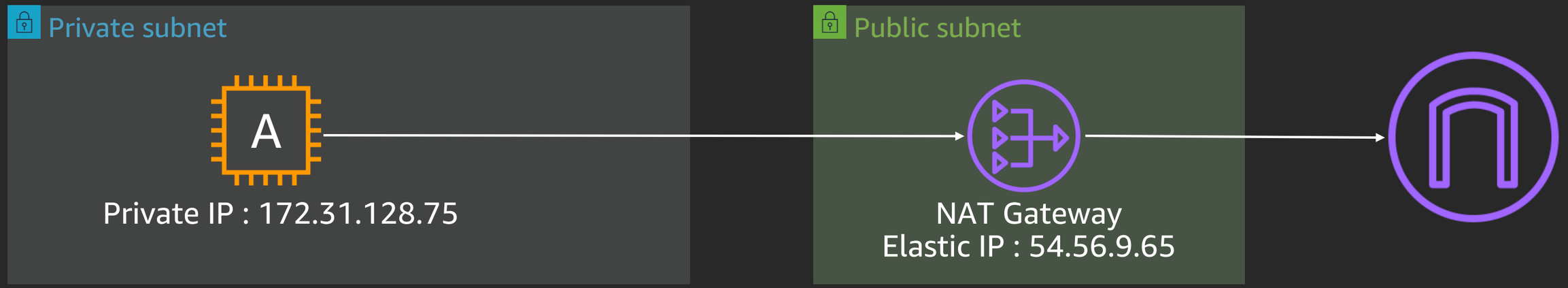
Destination	Target	Status	Propagated
172.31.0.0/16	local	Active	No
2600:1f16:14d:6300::/56	local	Active	No

Destination	Target	Status	Propagated
172.31.0.0/16	local	Active	No
2600:1f16:14d:6300::/56	local	Active	No
0.0.0.0/0	igw-09ef761d872bd7540	Active	No
::/0	igw-09ef761d872bd7540	Active	No

"Instance A has a path to and from instance B."

"Instance B has a path to and from the internet."

Network address translation (NAT) gateway



Destination	Target	Status	Propagated
172.31.0.0/16	local	Active	No
0.0.0.0/0	nat-0964c62a07d6491f5	Active	No

Destination	Target	Status	Propagated
172.31.0.0/16	local	Active	No
2600:1f16:14d:6300::/56	local	Active	No
0.0.0.0/0	igw-09ef761d872bd7540	Active	No
::/0	igw-09ef761d872bd7540	Active	No

The route table for the private subnet says to send all IPv4 internet traffic to the NAT gateway.

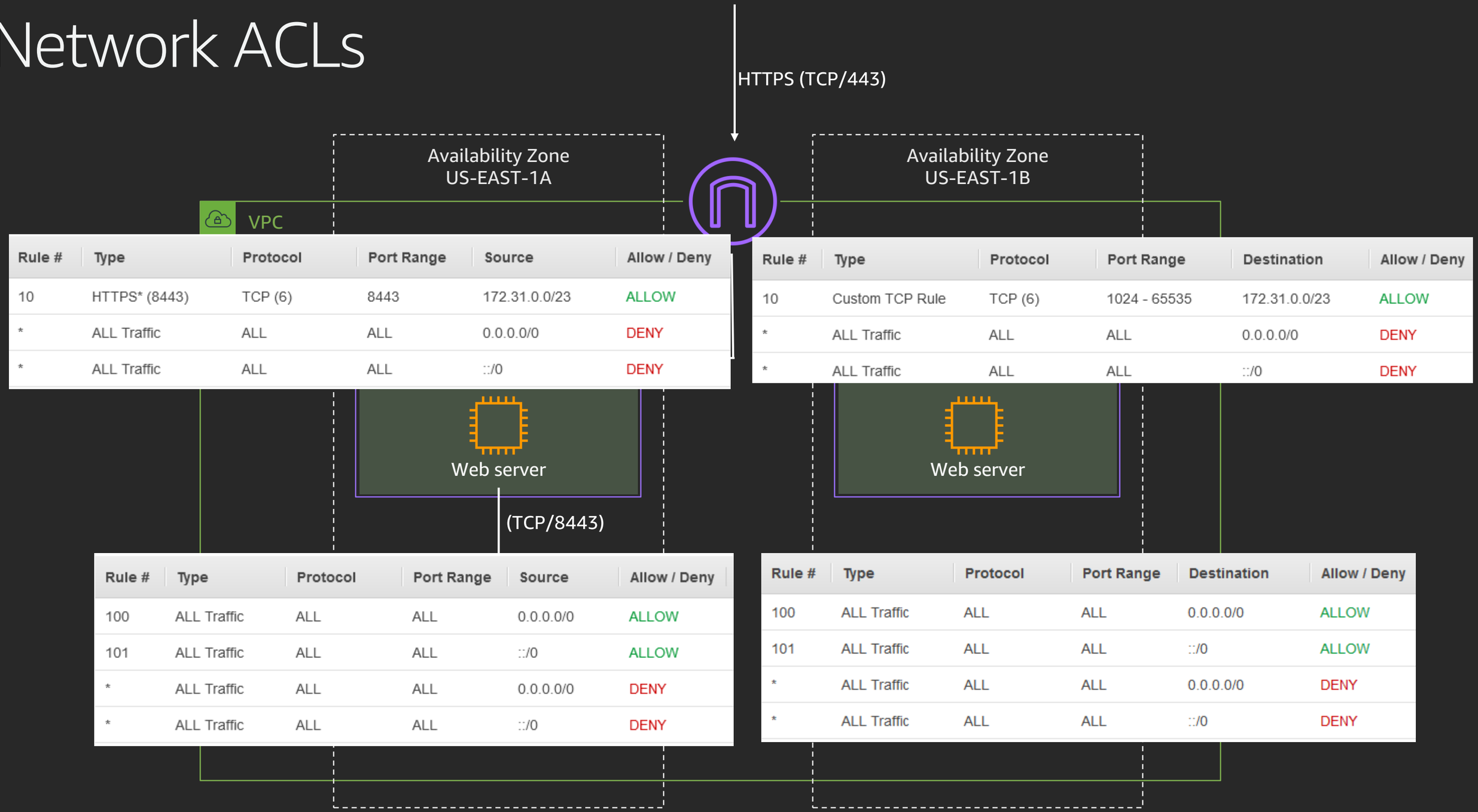
The NAT gateway translates all traffic that it receives such that it appears to come from itself.

The route table for the public subnet says to send all internet traffic to the internet gateway.

Network security

- Network ACLs
- Security groups
- VPC Flow Logs
- Amazon VPC Traffic Mirroring

Network ACLs



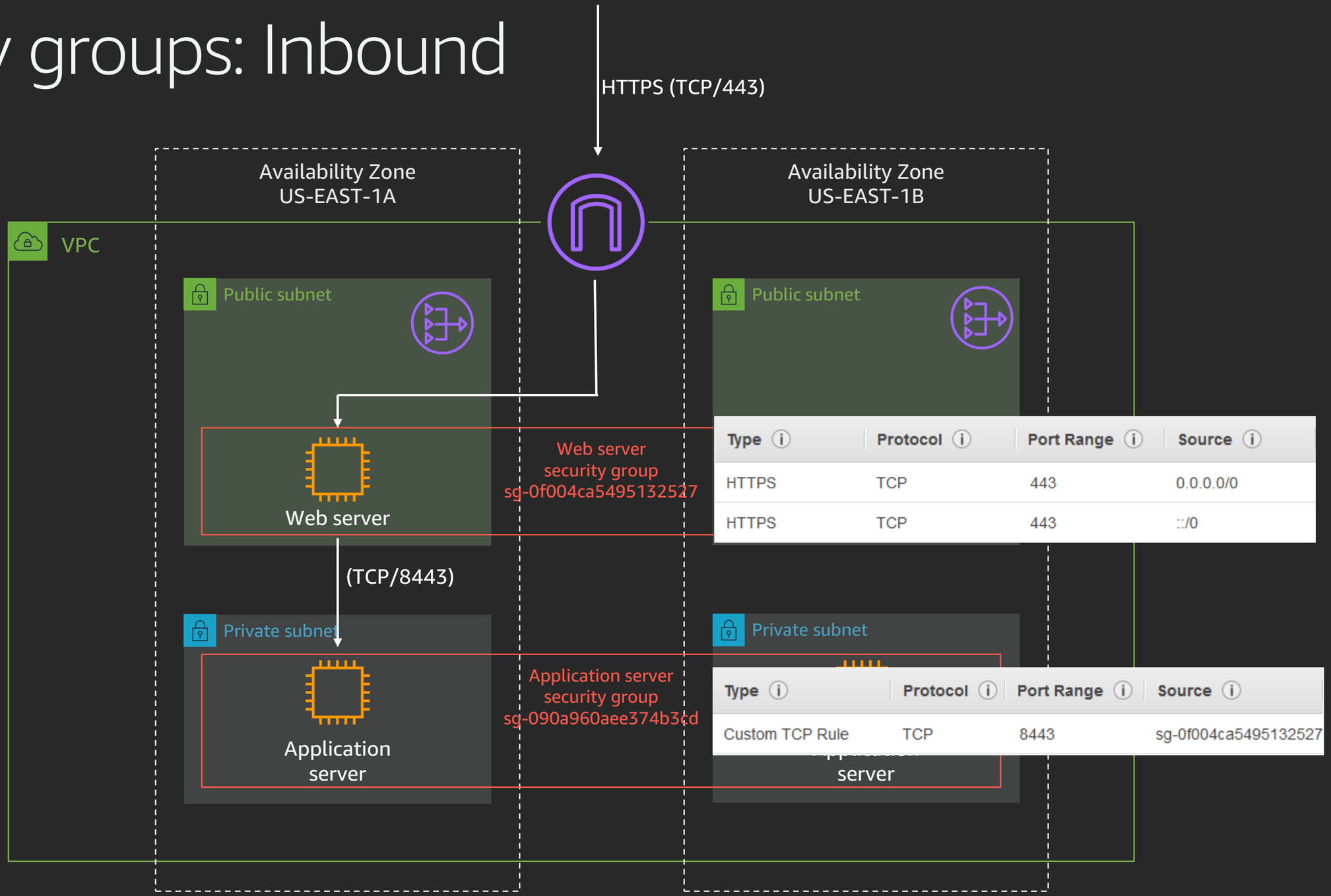
Rule #	Type	Protocol	Port Range	Source	Allow / Deny
10	HTTPS* (8443)	TCP (6)	8443	172.31.0.0/23	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY
*	ALL Traffic	ALL	ALL	::/0	DENY

Rule #	Type	Protocol	Port Range	Destination	Allow / Deny
10	Custom TCP Rule	TCP (6)	1024 - 65535	172.31.0.0/23	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY
*	ALL Traffic	ALL	ALL	::/0	DENY

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
101	ALL Traffic	ALL	ALL	::/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY
*	ALL Traffic	ALL	ALL	::/0	DENY

Rule #	Type	Protocol	Port Range	Destination	Allow / Deny
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
101	ALL Traffic	ALL	ALL	::/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY
*	ALL Traffic	ALL	ALL	::/0	DENY

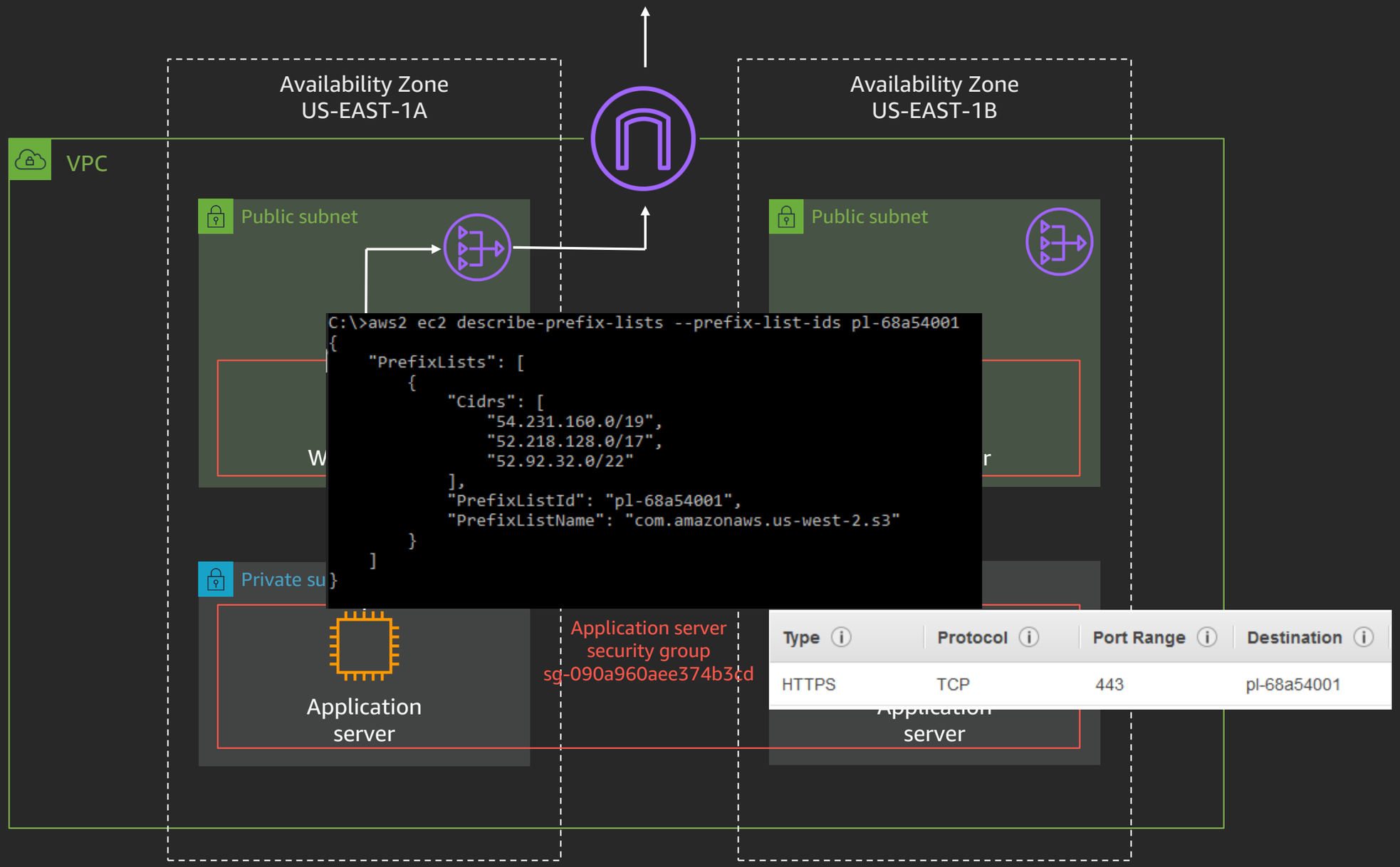
Security groups: Inbound



Type	Protocol	Port Range	Source
HTTPS	TCP	443	0.0.0.0/0
HTTPS	TCP	443	::/0

Type	Protocol	Port Range	Source
Custom TCP Rule	TCP	8443	sg-0f004ca5495132527

Security groups: Outbound




```
C:\>aws2 ec2 describe-prefix-lists --prefix-list-ids pl-68a54001
{
    "PrefixLists": [
        {
            "Cidrs": [
                "54.231.160.0/19",
                "52.218.128.0/17",
                "52.92.32.0/22"
            ],
            "PrefixListId": "pl-68a54001",
            "PrefixListName": "com.amazonaws.us-west-2.s3"
        }
    ]
}
```

Type	Protocol	Port Range	Destination
HTTPS	TCP	443	pl-68a54001

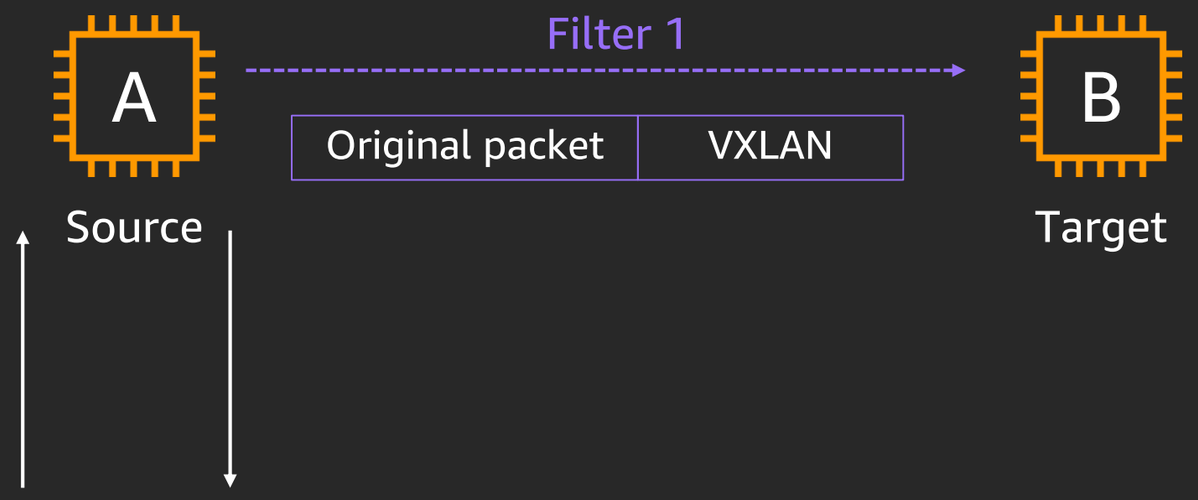
VPC Flow Logs

- Amazon CloudWatch Logs or Amazon S3
- Does not impact throughput or latency
- Apply to VPC, subnet, or elastic network interface
- Accepted, rejected, or all traffic

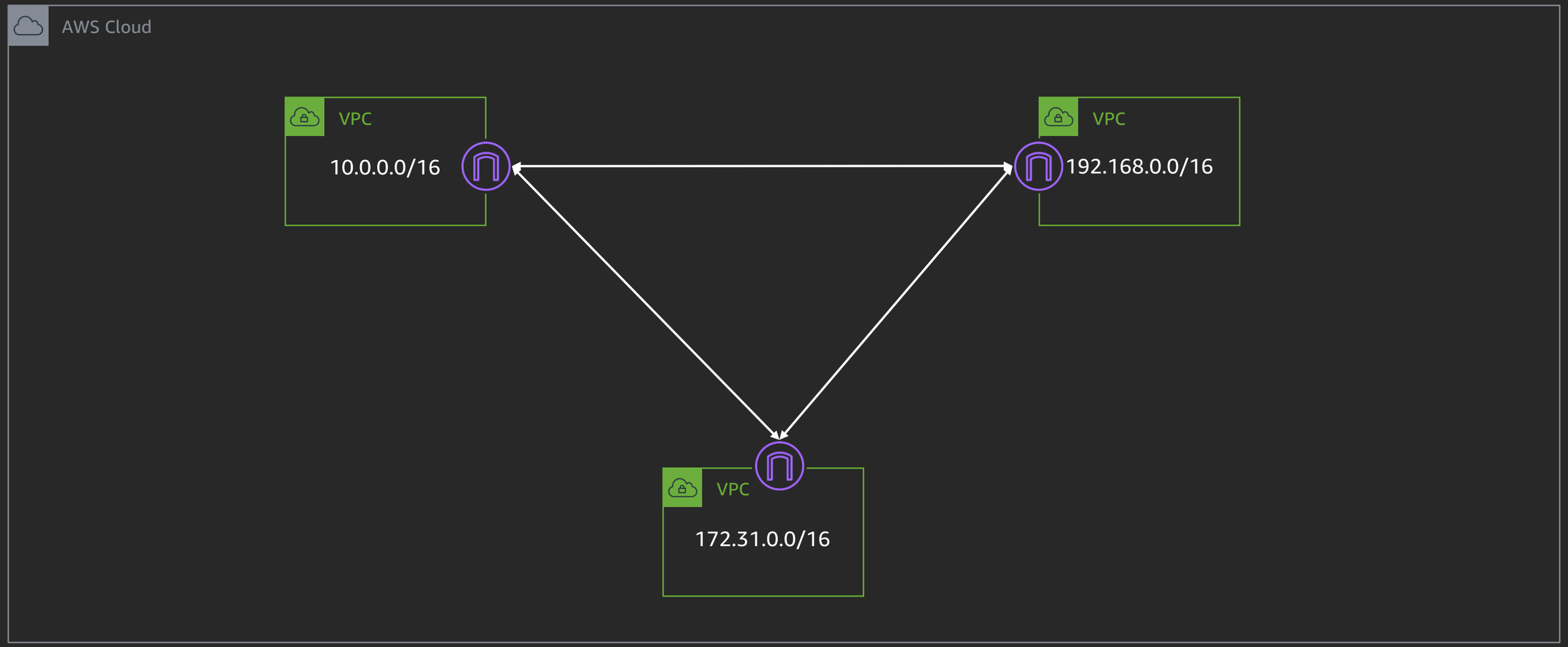
version	3
account-id	384767312345
interface-id	eni-0b62d5e000e412345
srcaddr	108.56.192.231
dstaddr	172.31.0.202
srcport	50565
dstport	80
protocol	6
packets	7
bytes	751
start	1573704396
end	1573704455
action	ACCEPT
log-status	OK
vpc-id	vpc-0af48868ceeb12345
subnet-id	subnet-02ab634d2e4c12345
instance-id	i-0a998a68301112345
tcp-flags	3
type	IPv4
pkt-srcaddr	108.56.192.231
pkt-dstaddr	172.31.0.202

Amazon VPC Traffic Mirroring

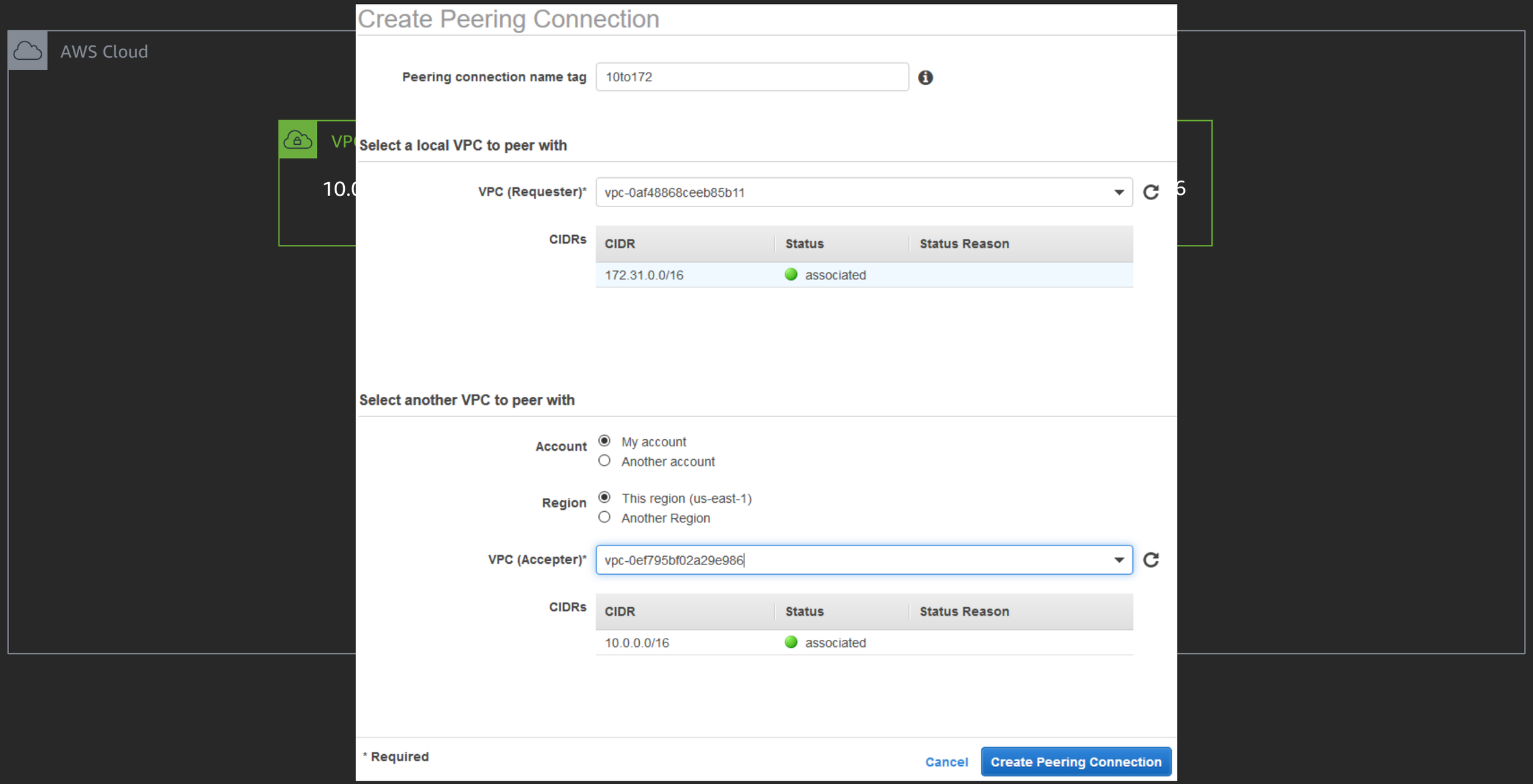
- Mirror to another elastic network interface or Network Load Balancer with UDP listener
- Packet copy; shares interface bandwidth
- Traffic mirror filters to define “interesting traffic”
- Traffic mirror session is the combination of source, target, and filter



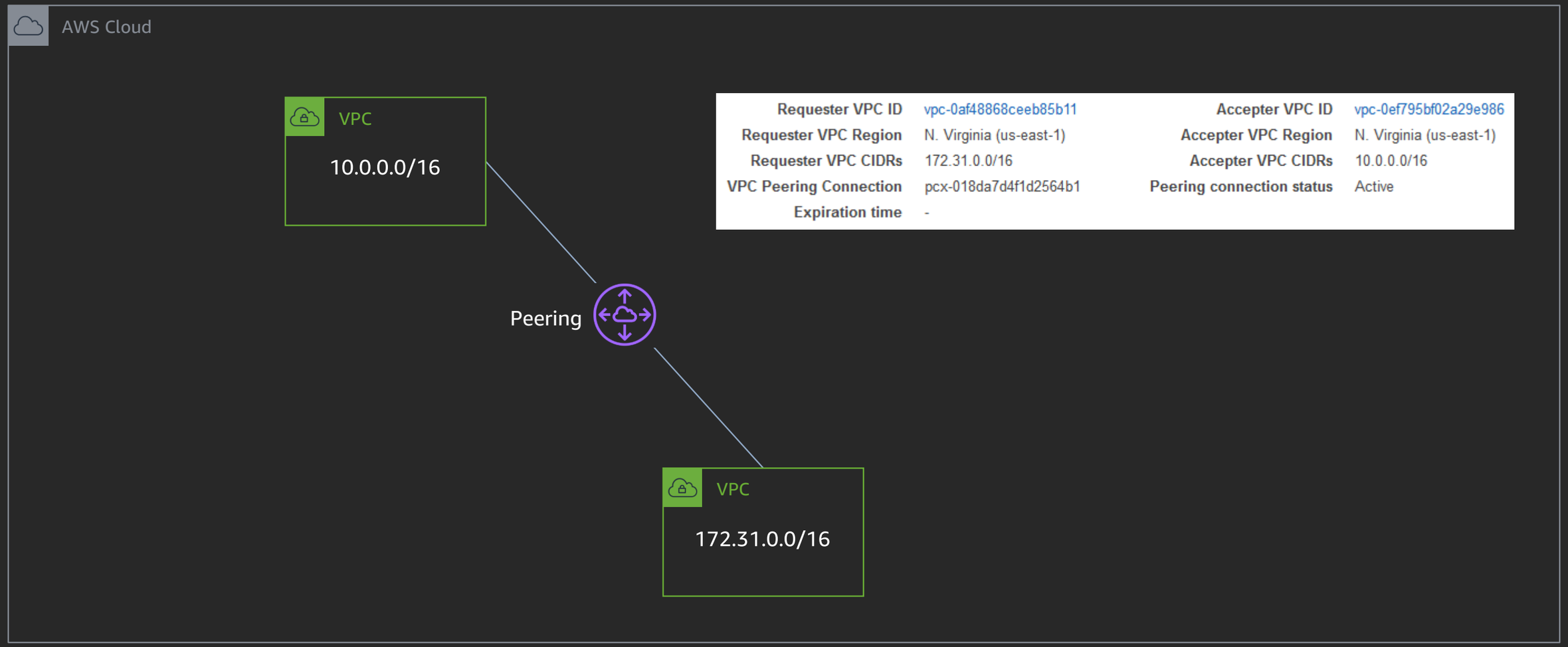
Connecting between VPCs



VPC peering: Same Region

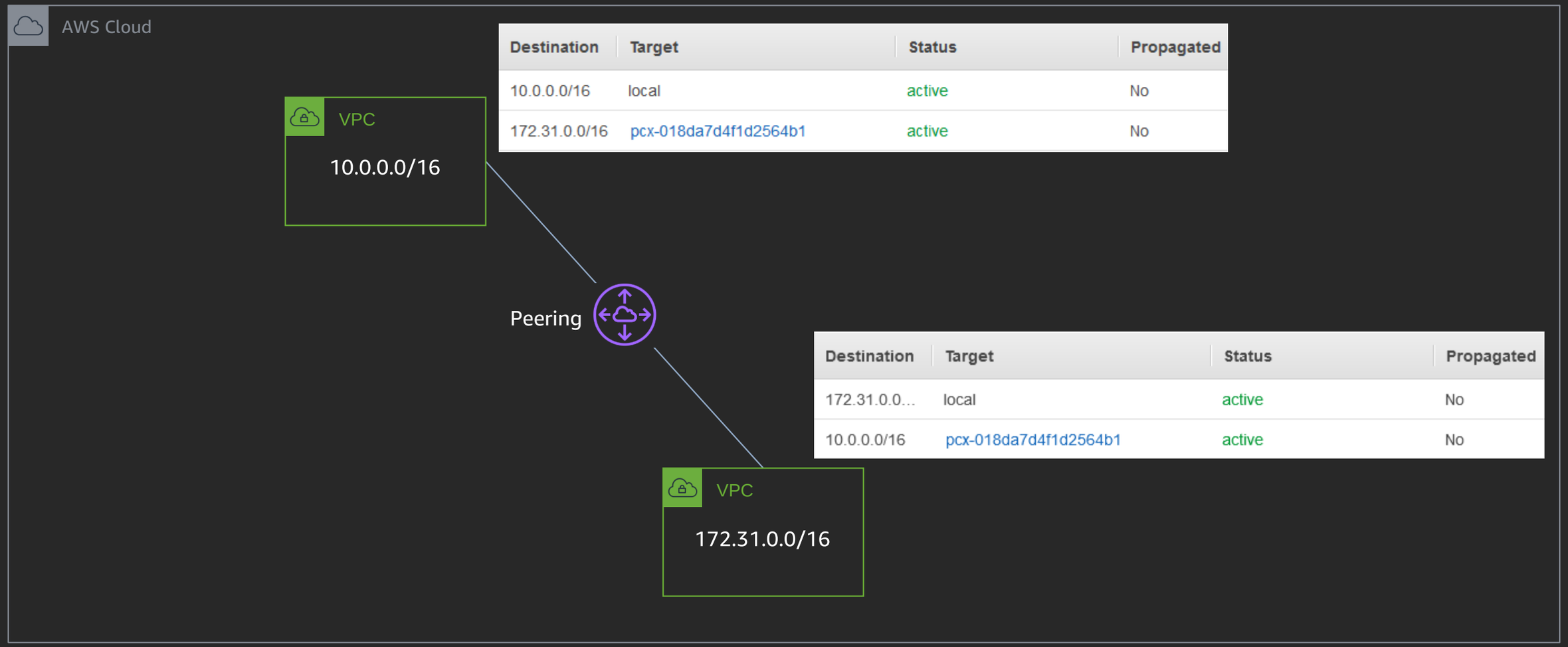


VPC peering: Same Region



Requester VPC ID	vpc-0af48868ceeb85b11	Accepter VPC ID	vpc-0ef795bf02a29e986
Requester VPC Region	N. Virginia (us-east-1)	Accepter VPC Region	N. Virginia (us-east-1)
Requester VPC CIDRs	172.31.0.0/16	Accepter VPC CIDRs	10.0.0.0/16
VPC Peering Connection	pcx-018da7d4f1d2564b1	Peering connection status	Active
Expiration time	-		

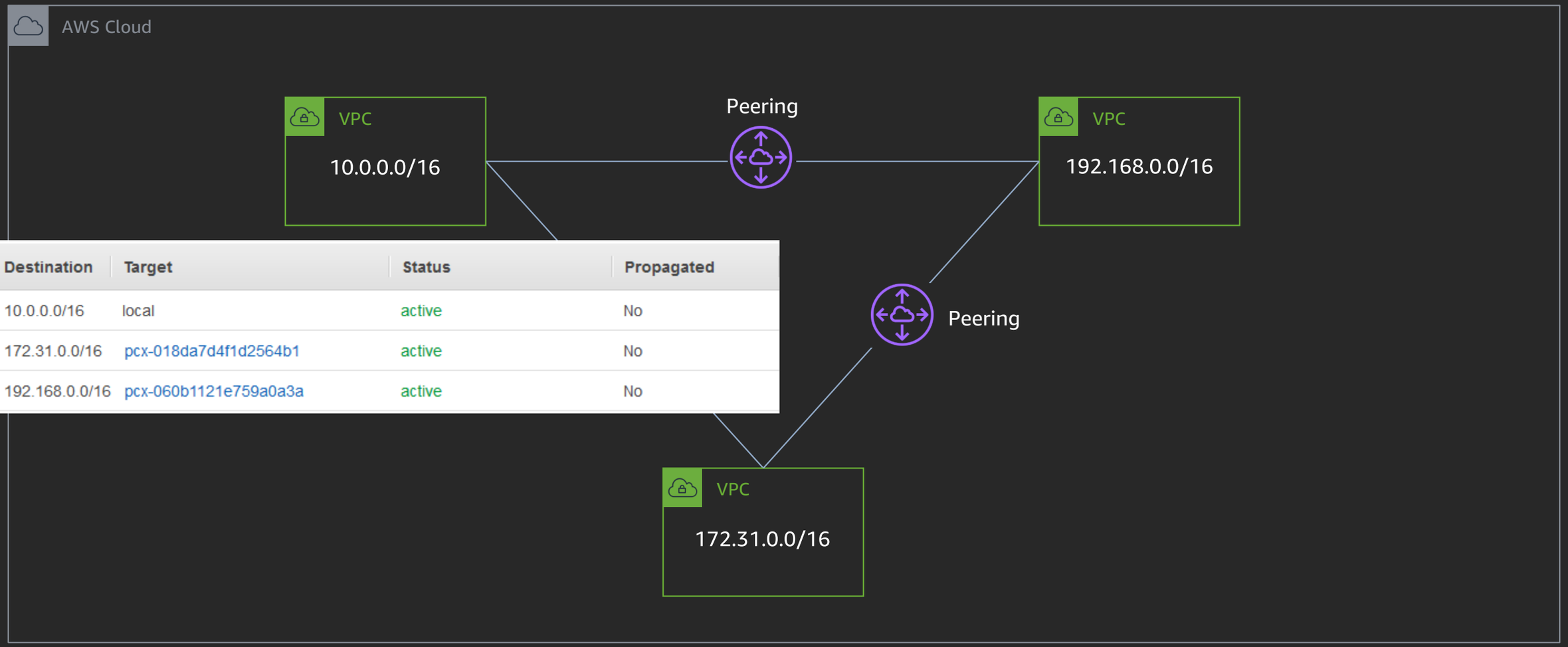
VPC peering: Same Region



Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
172.31.0.0/16	pcx-018da7d4f1d2564b1	active	No

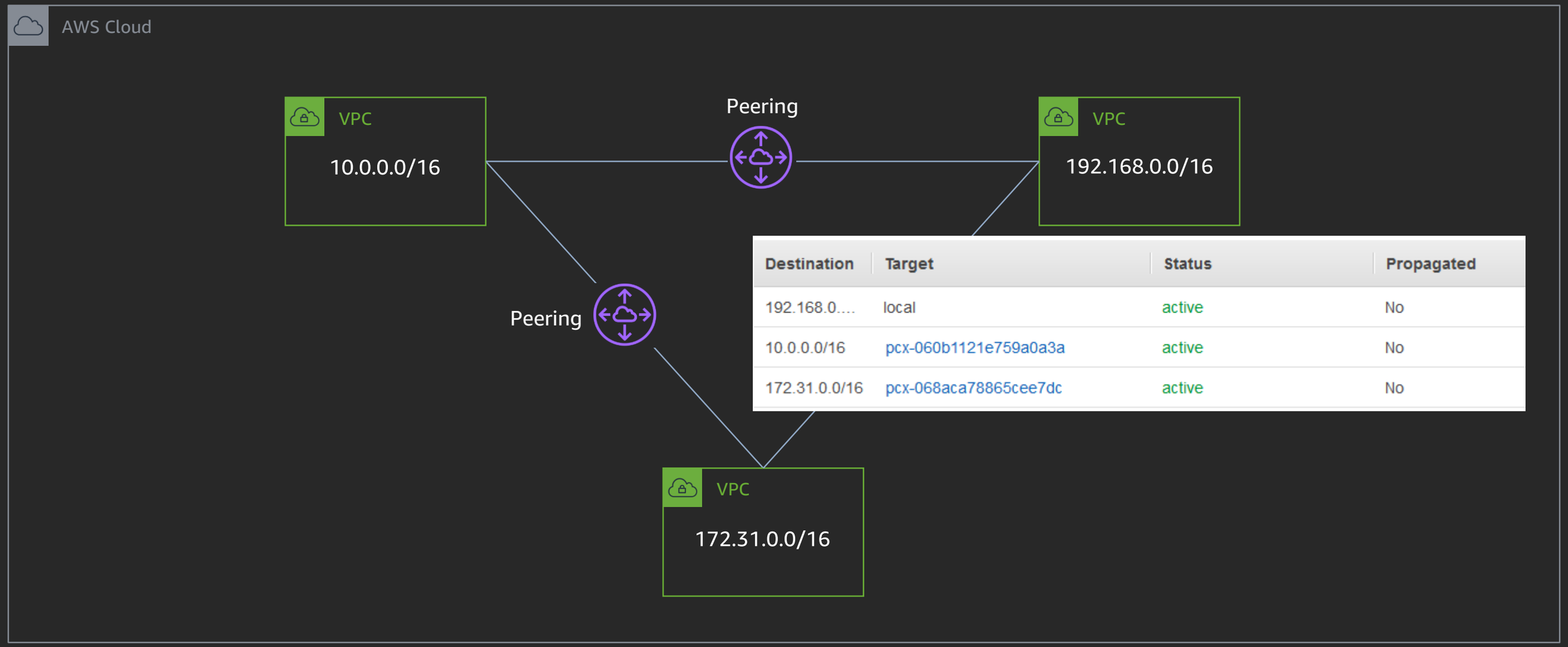
Destination	Target	Status	Propagated
172.31.0.0...	local	active	No
10.0.0.0/16	pcx-018da7d4f1d2564b1	active	No

VPC peering: Same Region



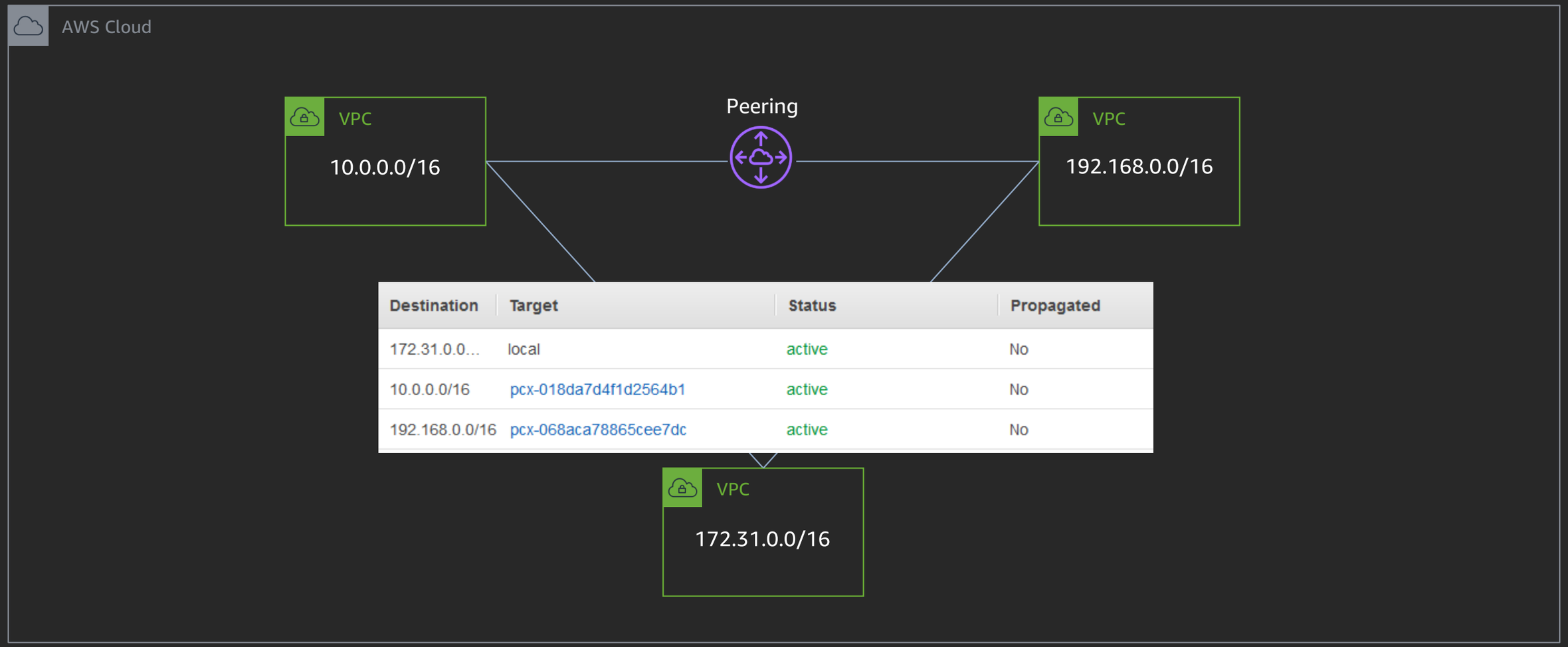
Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
172.31.0.0/16	pcx-018da7d4f1d2564b1	active	No
192.168.0.0/16	pcx-060b1121e759a0a3a	active	No

VPC peering: Same Region



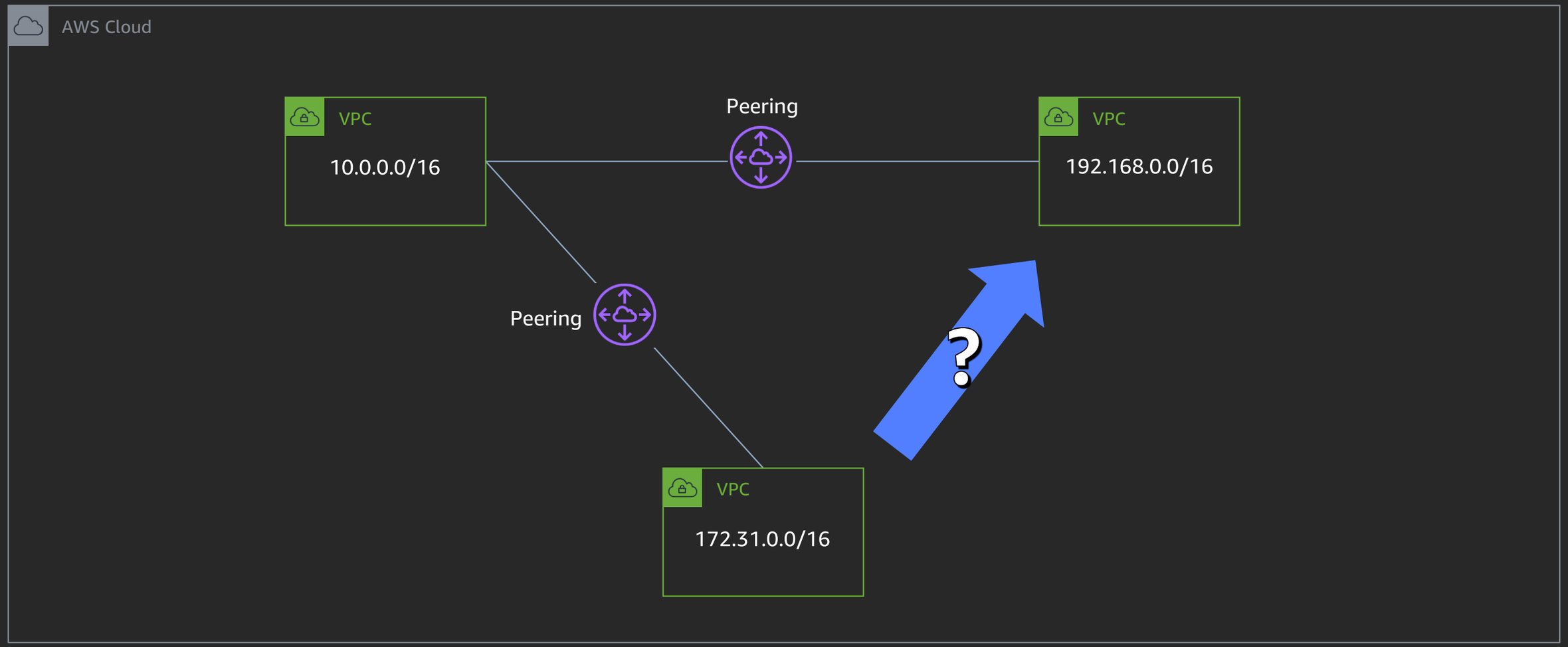
Destination	Target	Status	Propagated
192.168.0....	local	active	No
10.0.0.0/16	pcx-060b1121e759a0a3a	active	No
172.31.0.0/16	pcx-068aca78865cee7dc	active	No

VPC peering: Same Region

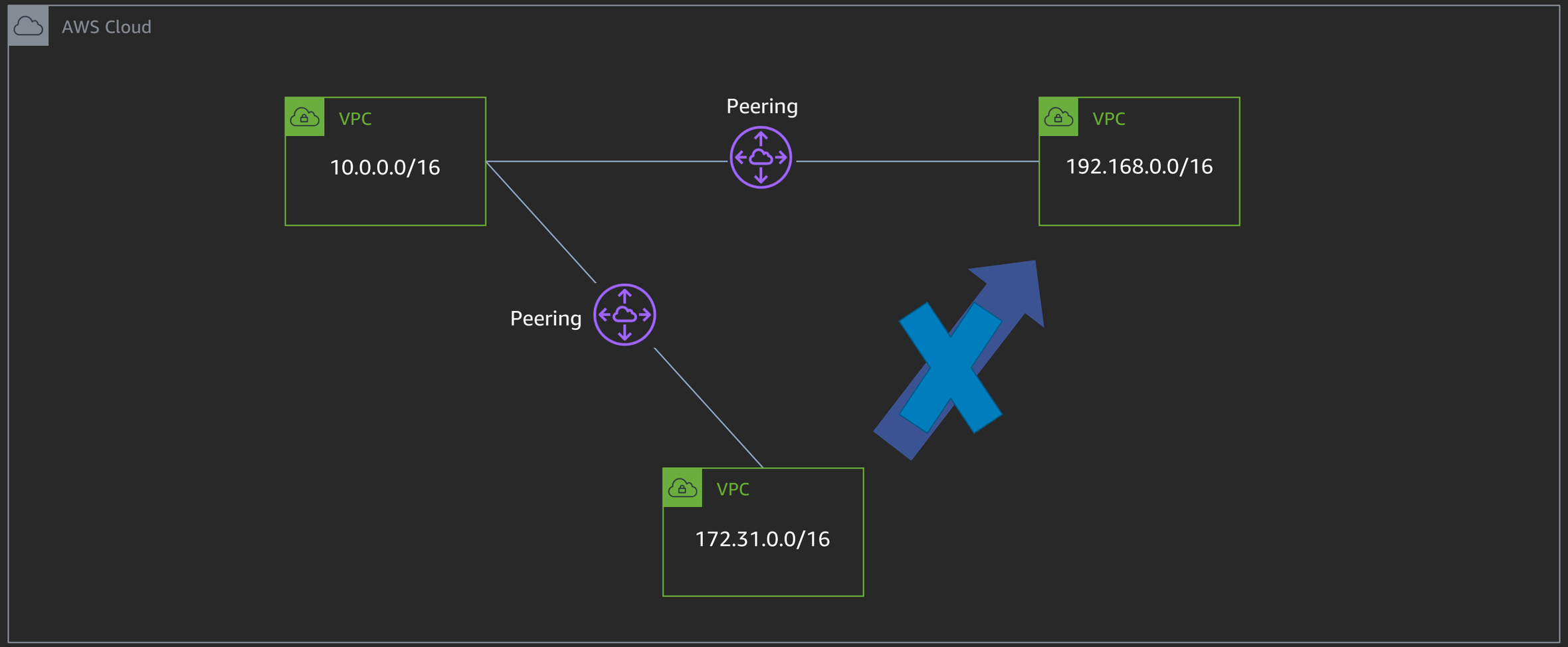


Destination	Target	Status	Propagated
172.31.0.0...	local	active	No
10.0.0.0/16	pcx-018da7d4f1d2564b1	active	No
192.168.0.0/16	pcx-068aca78865cee7dc	active	No

VPC peering: Same Region



VPC peering: Same Region



VPC peering: Different Region

Create Peering Connection

Peering connection name tag Cross-Region Peering

Select a local VPC to peer with

VPC (Requester)* vpc-0af48868ceeb85b11

CIDRs

CIDR	Status	Status Reason
172.31.0.0/16	associated	

Select another VPC to peer with

Account
- (●) My account
- ( ) Another account

Region
- ( ) This region (us-east-1)
- (●) Another Region

US East (Ohio) (us-east-2)

VPC (Accepter)* vpc-0c05afa3bd855bd6a

* Required

Cancel | Create Peering Connection

VPC peering: Different account

Create Peering Connection

Peering connection name tag Cross-Account, Cross-Region

Select a local VPC to peer with

VPC (Requester)* vpc-0af48868ceeb85b11

CIDRs

CIDR	Status	Status Reason
172.31.0.0/16	associated	

Select another VPC to peer with

Account ○ My account ◉ Another account

Account ID* 25[illegible]97

Region ○ This region (us-east-1) ◉ Another Region

US East (Ohio) (us-east-2)

VPC (Accepter)* vpc-0a027a281b67d50c4

* Required

Cancel | Create Peering Connection

VPC peering: Things to know

Can reference security groups from the peer VPC in the same Region

Can enable DNS hostname resolution to return private IP addresses

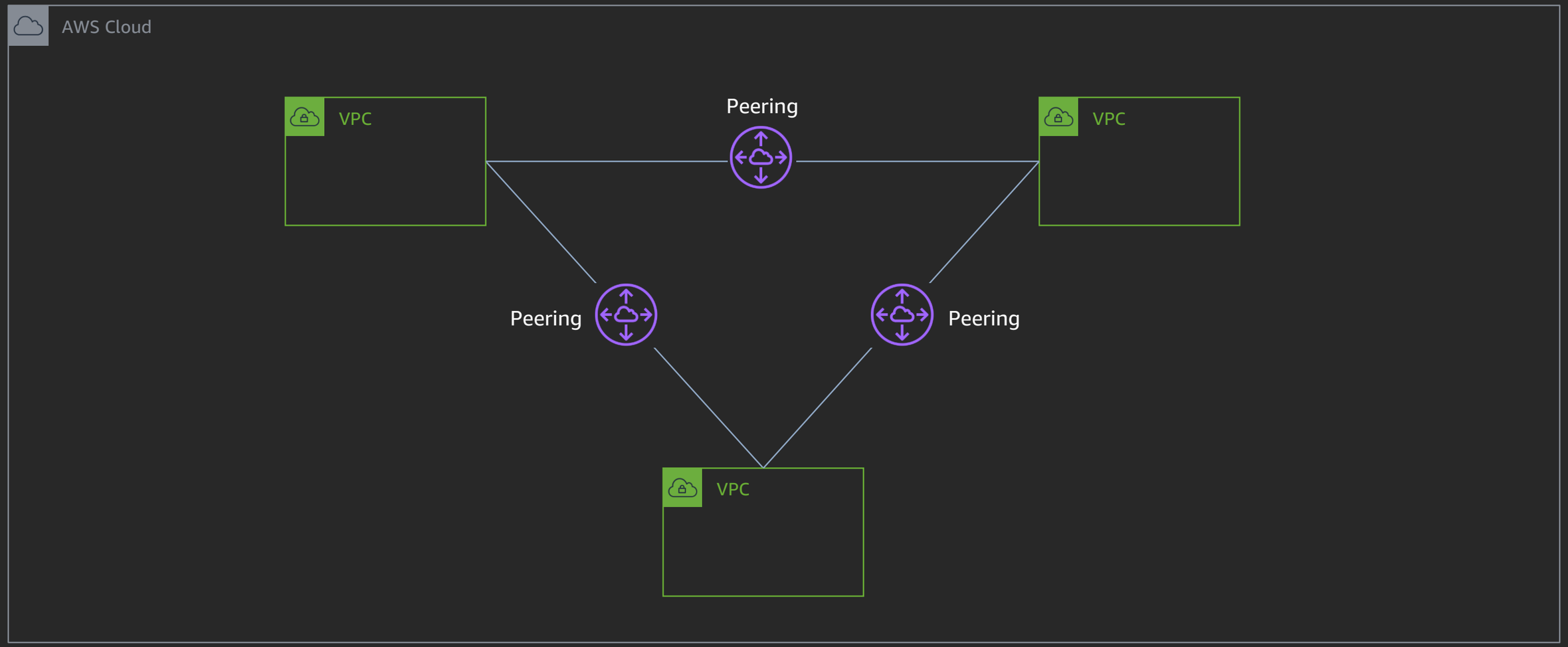
Can peer for both IPv4 & IPv6 addresses

Cannot have overlapping IP addresses

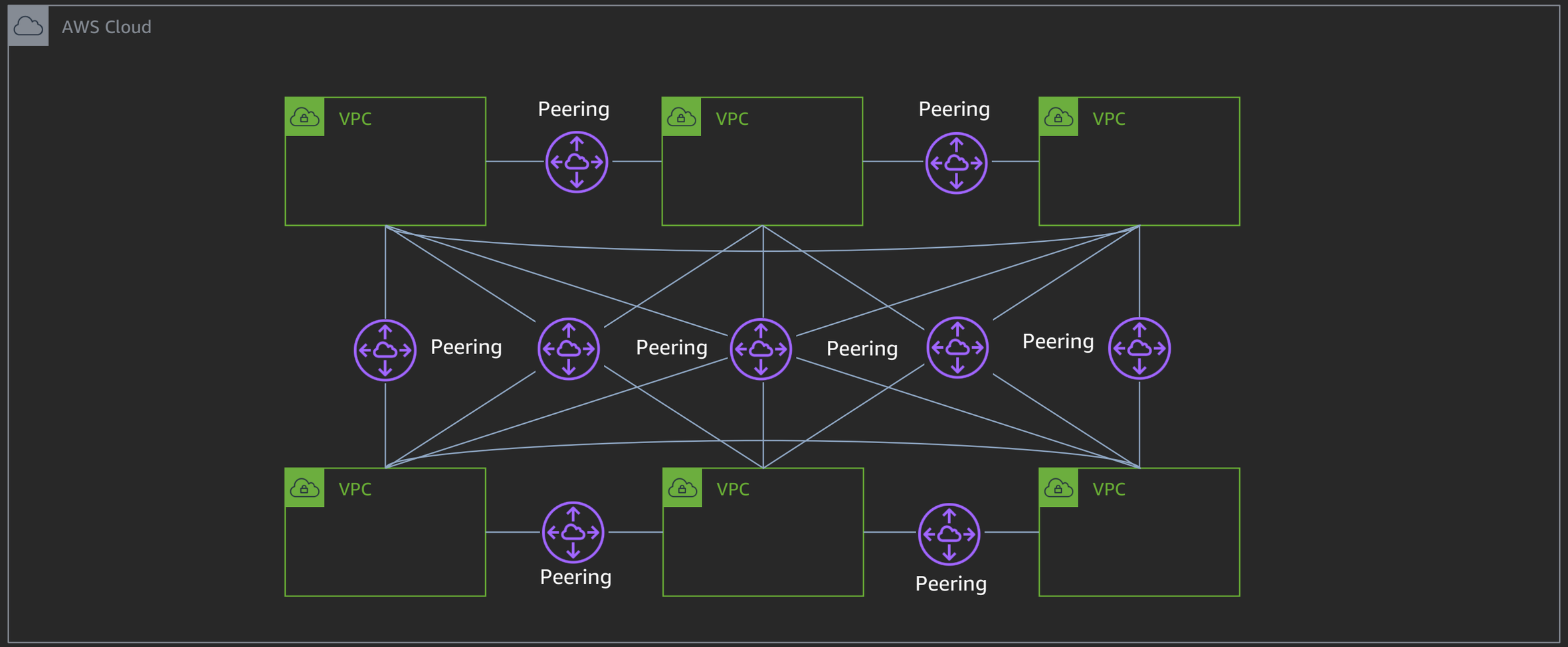
Cannot have multiple peers between the same pair of VPCs

Cannot use jumbo frames across inter-Region VPC peering

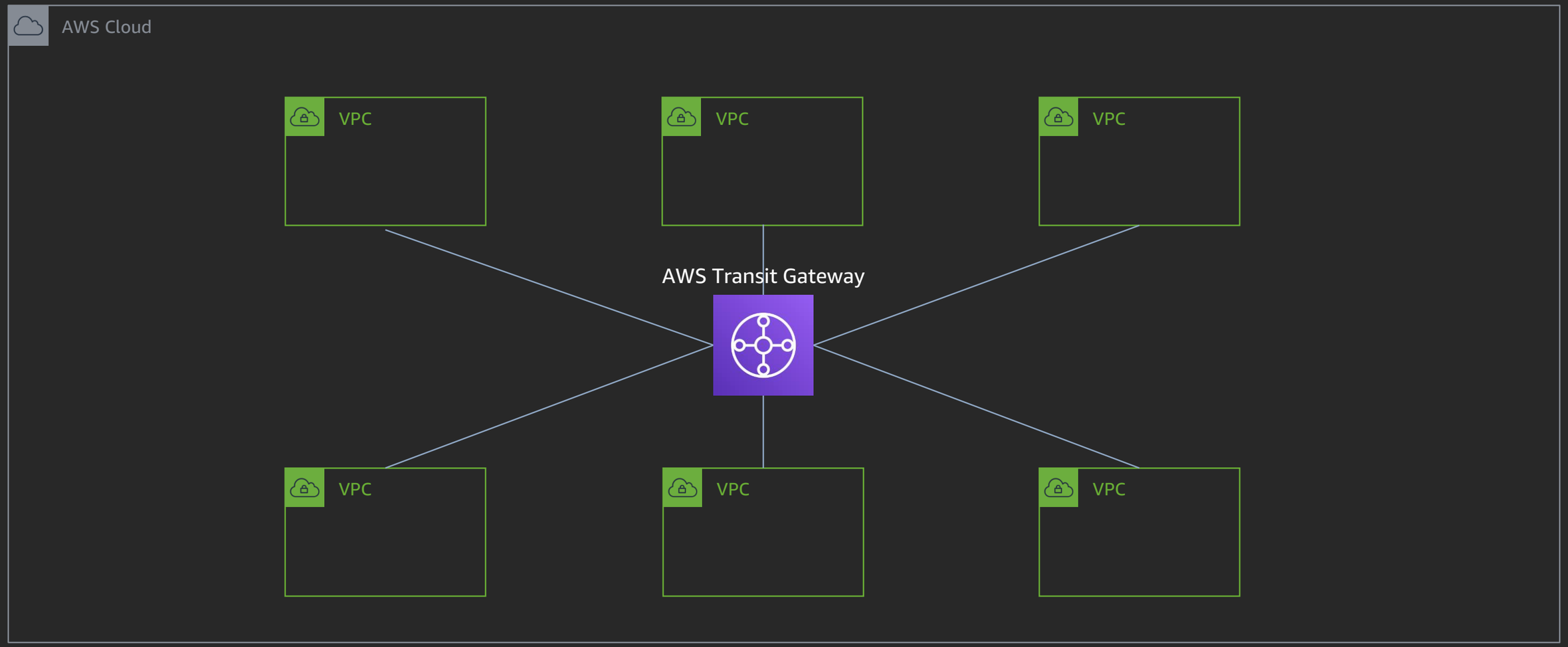
Interconnecting VPCs at scale: VPC peering



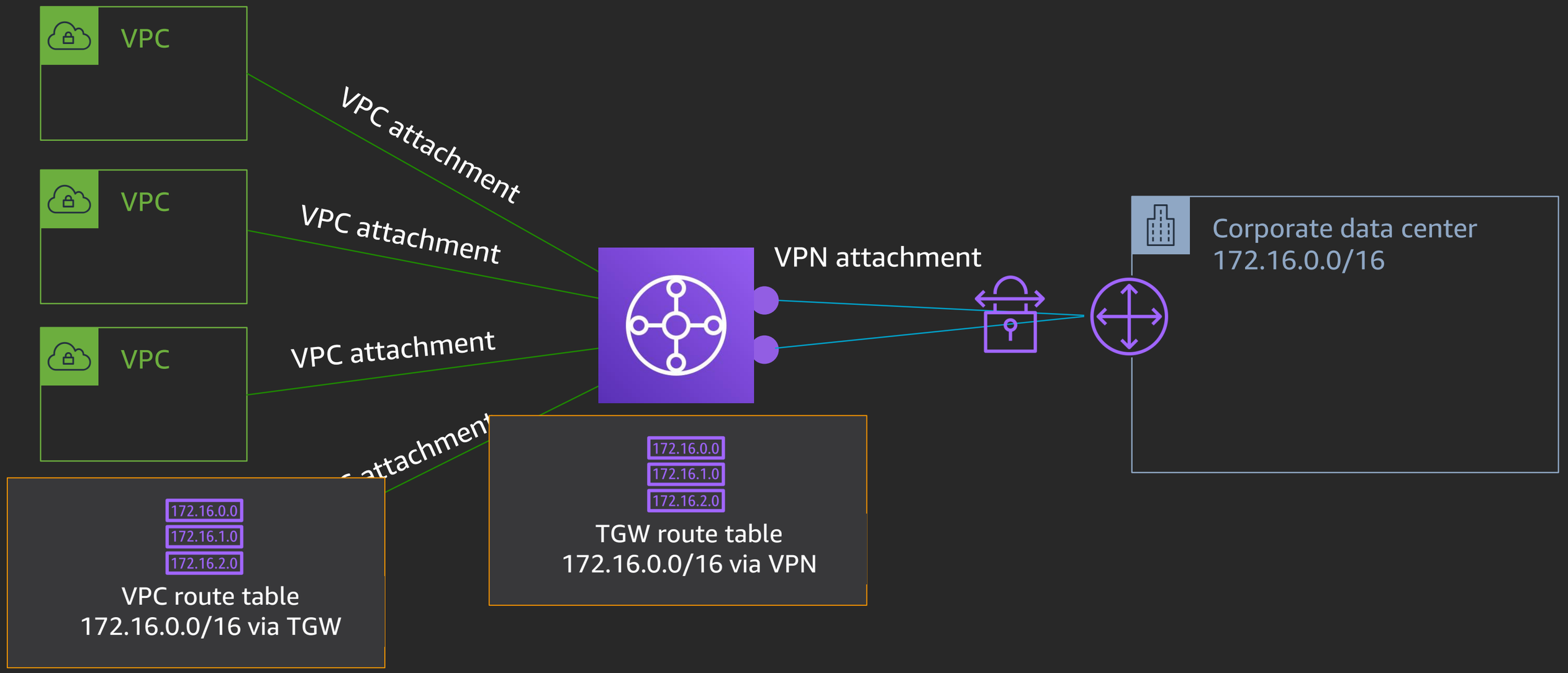
Interconnecting VPCs at scale: VPC peering



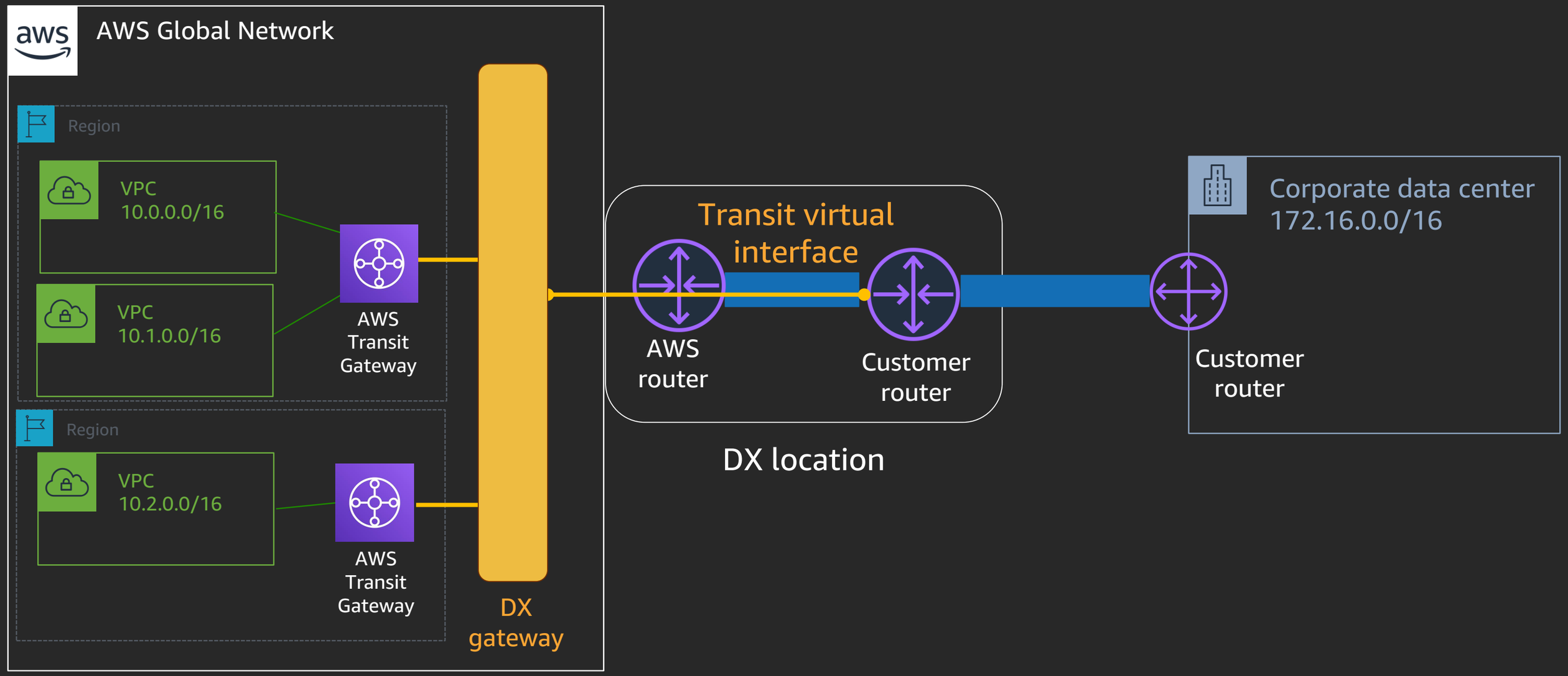
Multiple VPCs access model: AWS Transit Gateway



AWS Transit Gateway with AWS site-to-site VPN

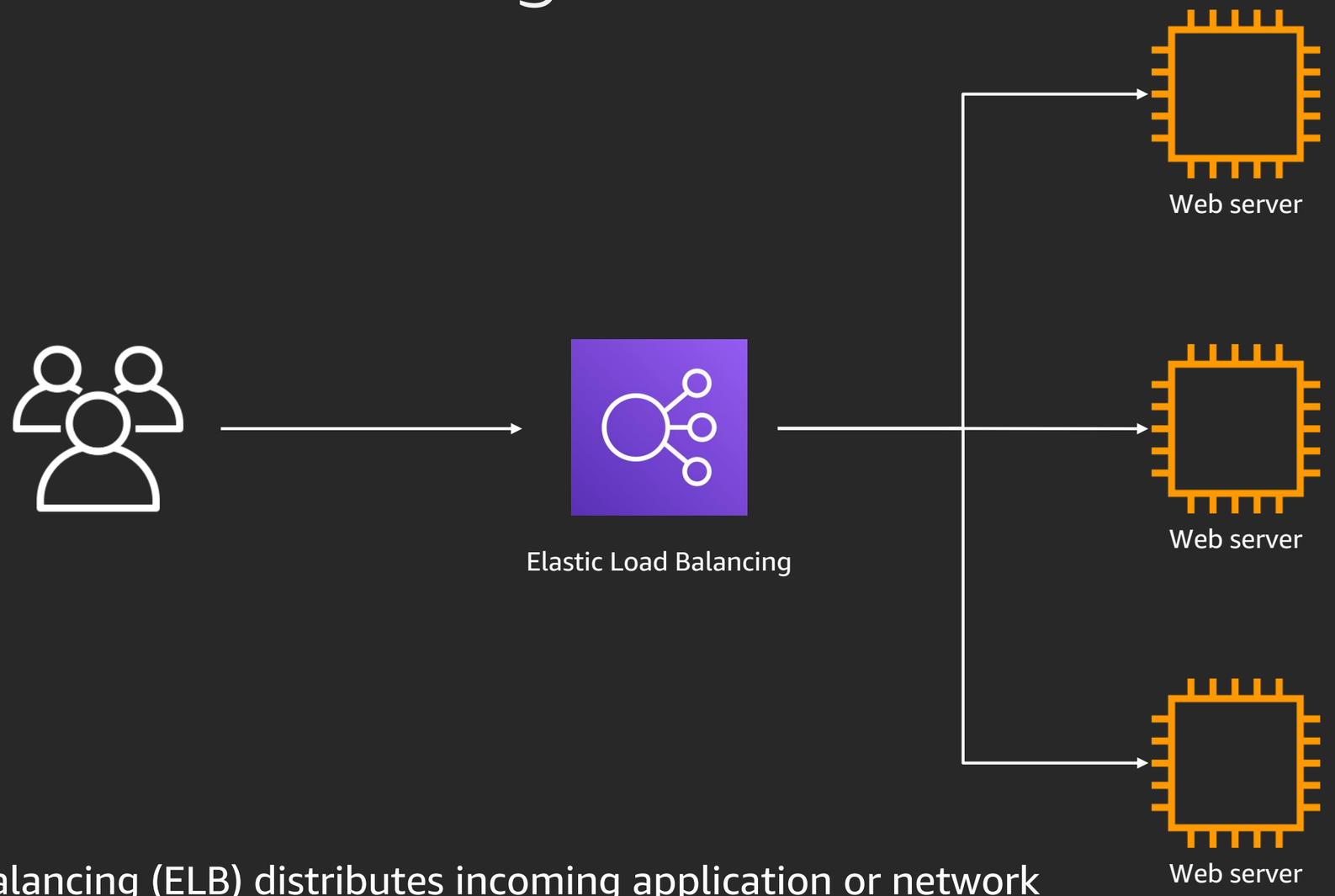


AWS Transit Gateway with AWS Direct Connect gateway



High availability & scale

Elastic Load Balancing



Elastic Load Balancing (ELB) distributes incoming application or network traffic across multiple targets, such as Amazon EC2 instances, containers, Lambda functions, and IP addresses, in multiple Availability Zones

Elastic Load Balancing: Options

Application Load Balancer

- IPv4, dual stack, front-end
- Layer 7
- HTTP, HTTPS
- Host-, path-based routing
- Integrated authentication
- Supported targets
  - EC2 instances
  - Containers
  - AWS Lambda
  - Private IP addresses

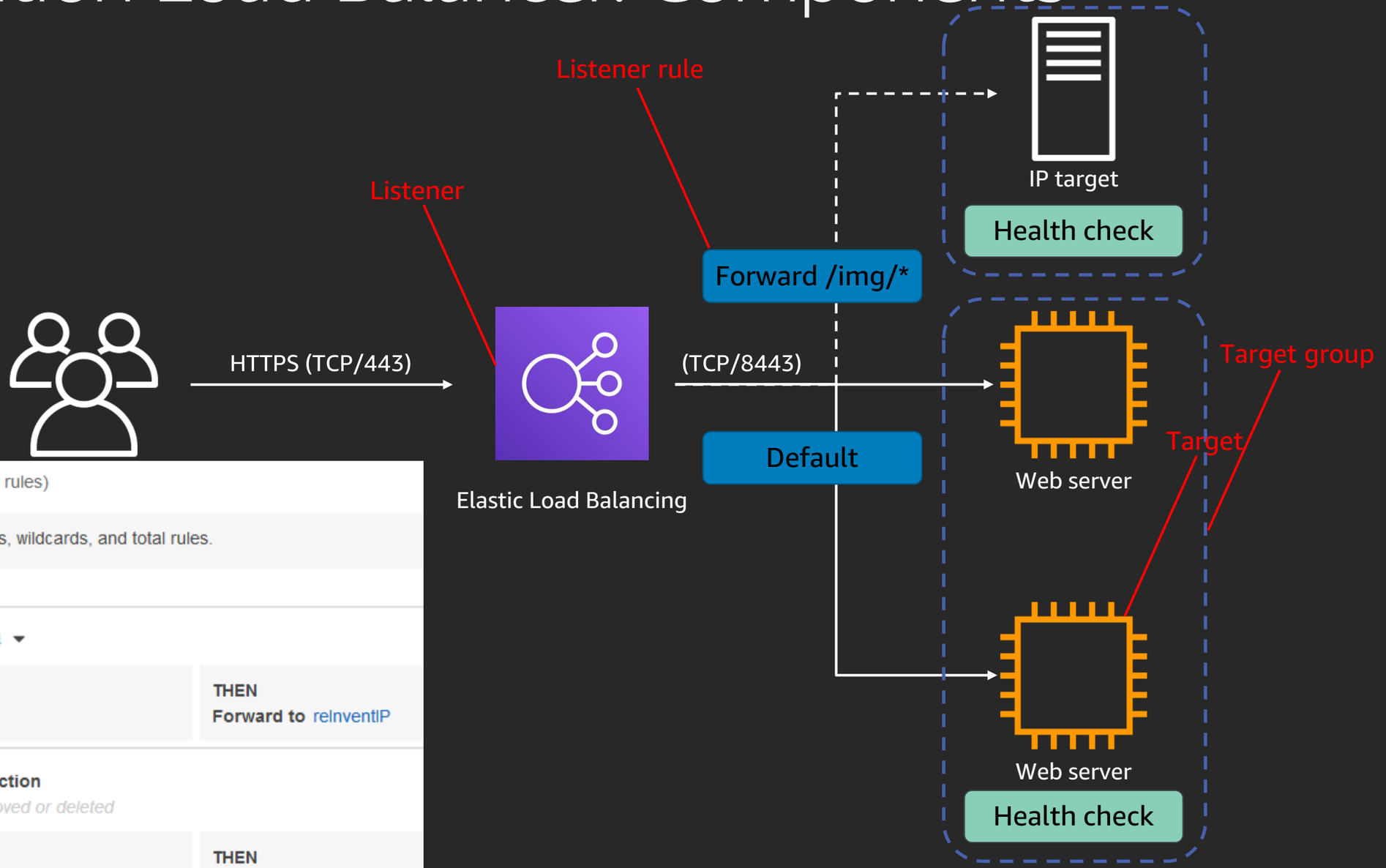
Network Load Balancer

- IPv4
- Layer 4
- TCP, UDP, TLS
- Supported targets
  - EC2 instances
  - Containers
  - Private IP addresses

Classic Load Balancer

- IPv4, dual stack front-end
- Layer 4/7
- HTTP, HTTPS, TCP, TLS
- Supported targets
  - EC2 instances

Application Load Balancer: Components



reInvent | HTTPS:443 (2 rules)

Rule limits for condition values, wildcards, and total rules.

1 arn...a7d9dc36eace9ce4

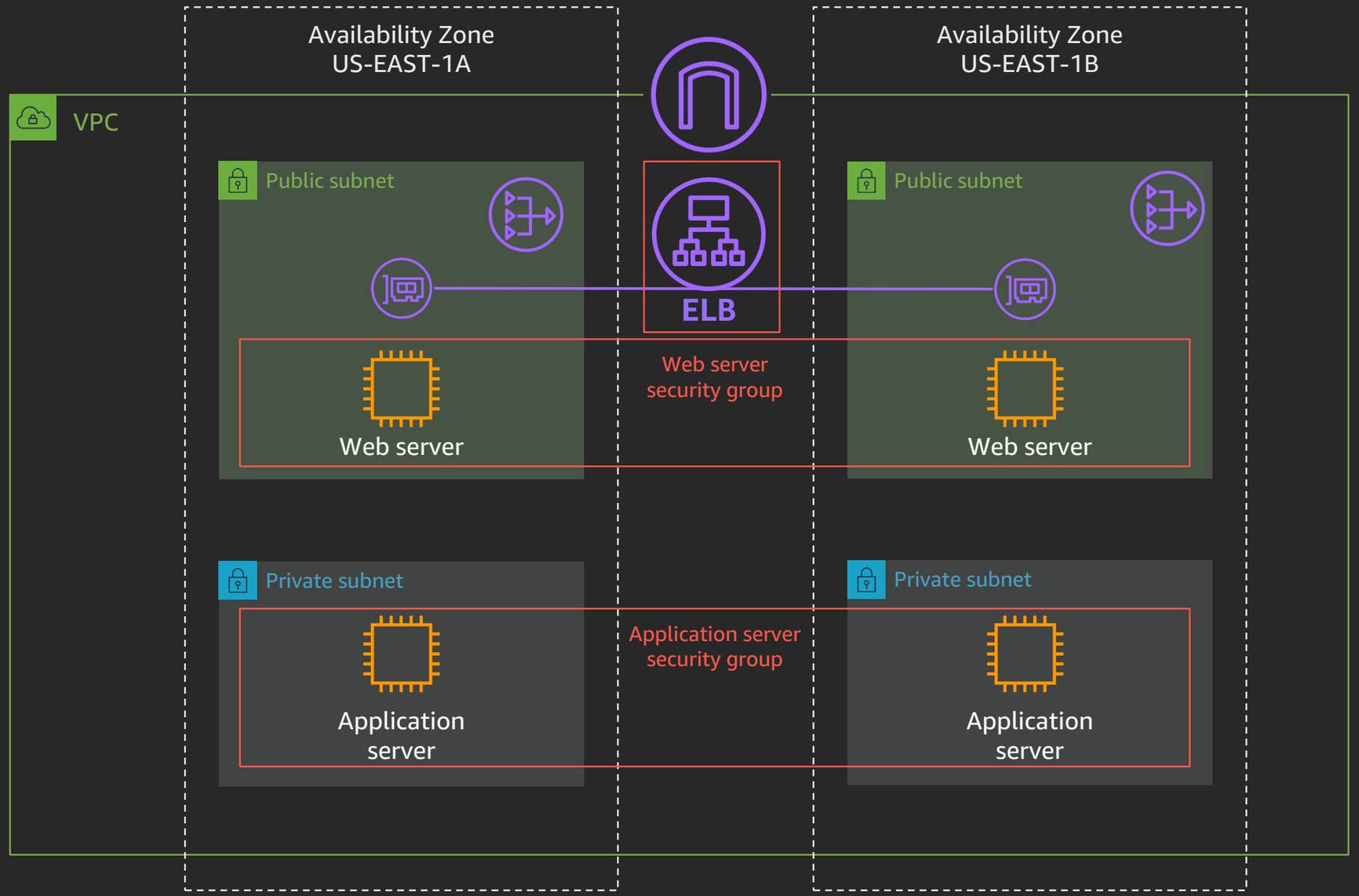
IF	THEN
Path is /img/*	Forward to reInventIP

last **HTTPS 443: default action**

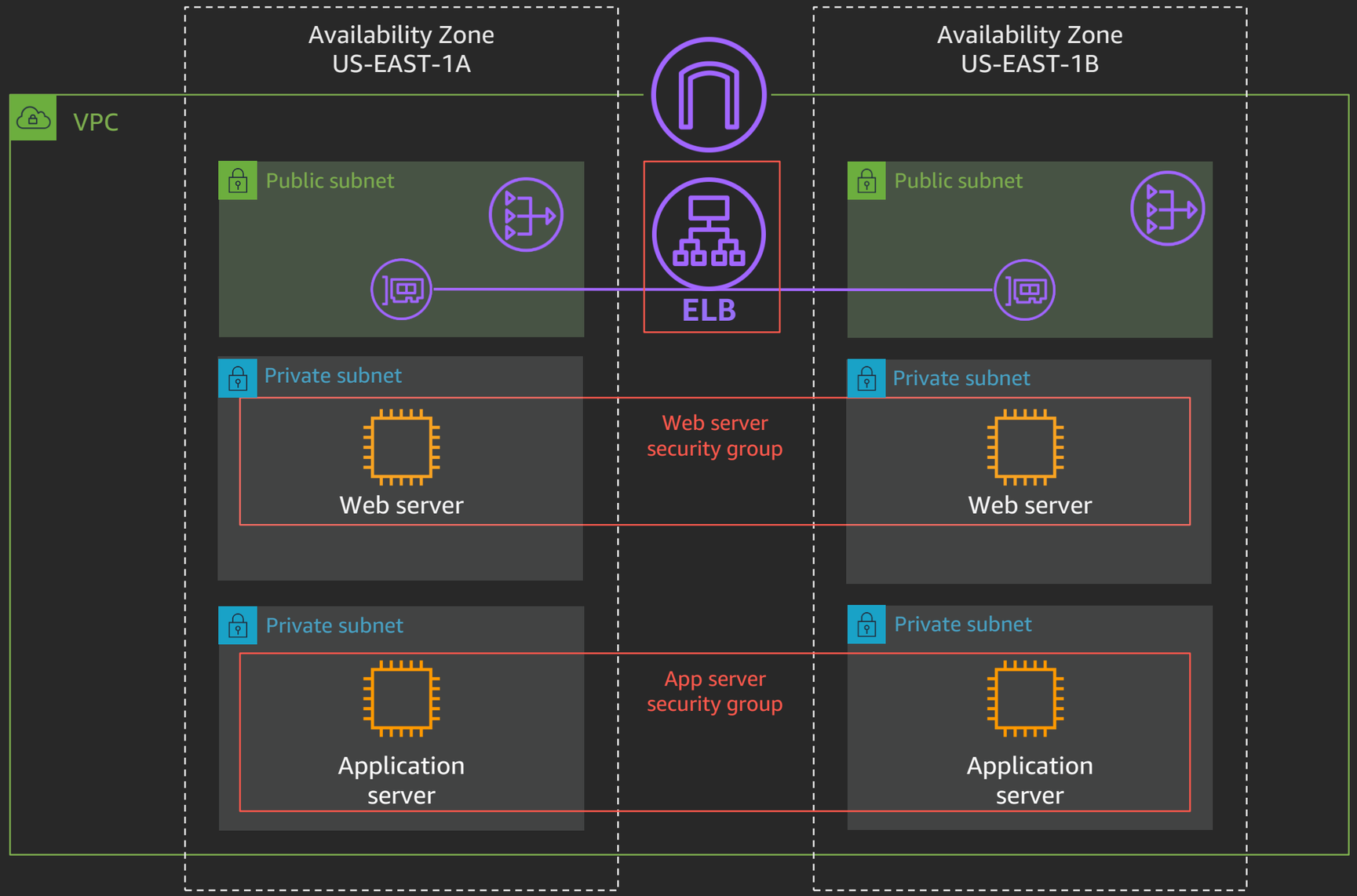
This rule cannot be moved or deleted

IF	THEN
Requests otherwise not routed	Forward to reInventEC2

Example web application



Example web application – Final



APN **Cloud Management Tools** Competency Partners

vmware®

Visit the Partner Discovery Zone to meet the partner and view the full list of APN Competency Partners

How can I distribute content ?

Amazon CloudFront

CloudFront is the AWS content delivery network

It securely delivers data, videos, applications, and APIs to customers globally with low latency and high transfer speeds

CloudFront is integrated with AWS; physical locations are directly connected to the AWS Global Cloud Infrastructure and other AWS services

It features a global network of >200 points of presence (PoPs)

What benefits does CloudFront deliver?

- Built-in security & DDoS protection
- Massive scale
- Performance-based request routing
- Connection optimization
- Dedicated, private AWS backbone
- Multi-tiered caching architecture for origin protection and offload
- Lower data transfer costs than regional endpoints

Amazon CloudFront: >200 PoPs

North America

Added 50% more PoPs in last year

Launched in 2 new cities: Houston and Salt Lake City

EMEA

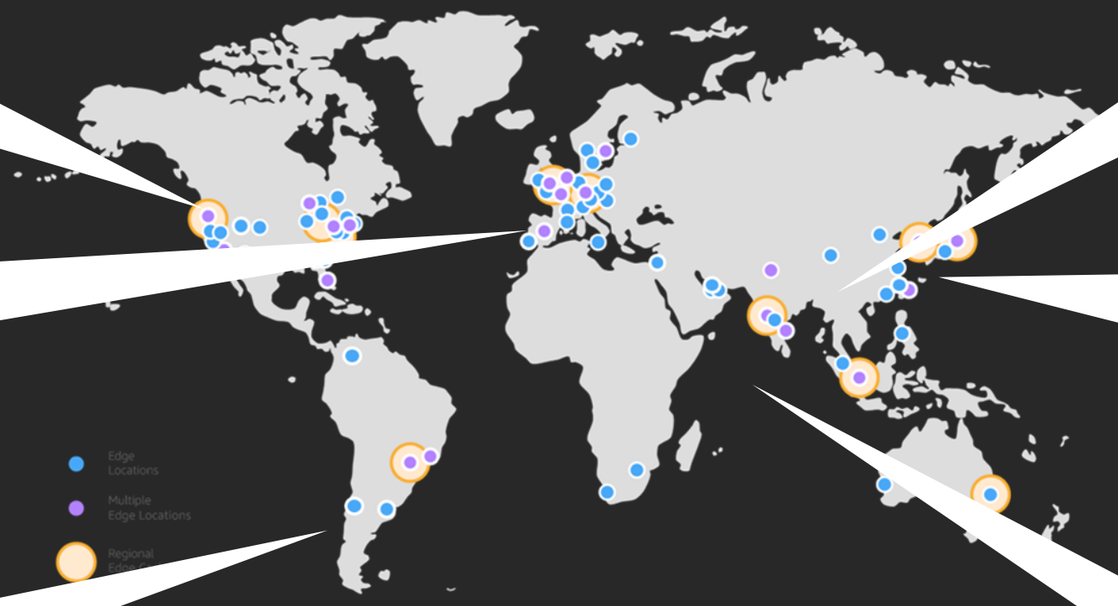
4 new countries: Bahrain, Belgium, Portugal, Israel

13 embeds in UK

South America

Added 3 new countries: Argentina, Chile, and Colombia

56% price cut



China

Launched in China with 4 new cities: Beijing, Shanghai, Zhongwei, Shenzhen

Japan

Added 6 PoPs in Tokyo

Currently 17 PoPs in Japan

India / Australia

16 PoPs in India

Added 3 PoPs in Sydney; currently 6 PoPs in Australia

210 PoPs in 79 cities, 37 countries

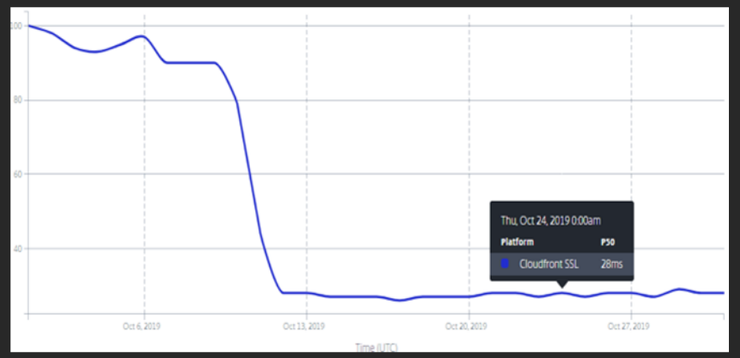
75+ PoPs added in 2019

Latency benefits with PoP launches

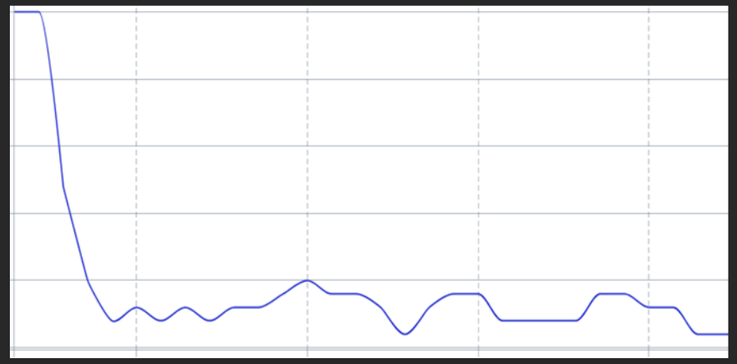
PoP launches ensure connectivity with majority views and redundant AWS backbone

Israel
75% Latency reduction
78 ms → 20 ms

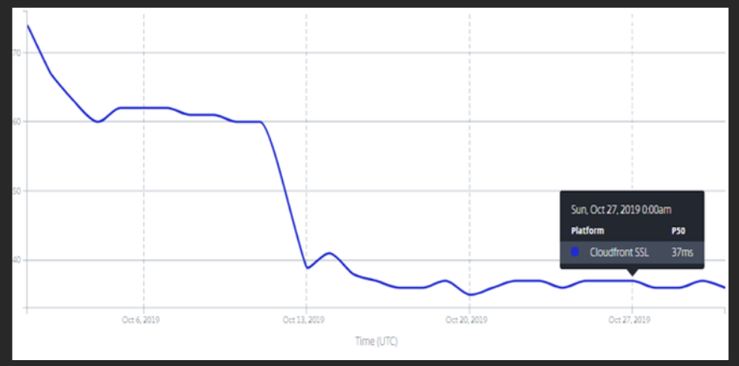
Chile
73% Latency reduction
104 ms → 28 ms



Bahrain:
40% Latency reduction
38 ms → 27 ms



Argentina
55% Latency reduction
79 ms → 35 ms



P50 FBL according to Cedexis

Building blocks of a CloudFront configuration

Distributions

- Unique CloudFront.net domain name to reference objects (abc123.cloudfront.net)
- Custom domains
- Custom TLS configuration
- Enable H2, IPV6 & logging to Amazon S3
- Associate to AWS WAF ACL

Origins

- Any HTTP(S) endpoint
- TCP ports & timeouts
- TLS configuration

Behaviors

- Path condition
- Select origin
- HTTP methods
- Caching and forwarding policy
- Enable object compression
- Configure features (Lambda@Edge triggers, field-level encryption, signed URLs)

Advanced security capabilities

Robust content protection controls & encryption

Access control

- Custom origin protection
  - Header and ACL
- Content protection
  - Signed URL / Cookies
- Content restriction
  - geo blocking
- Amazon S3 origin access identity

Encrypted connections

- Advanced ciphers
  - Certificate manager
- OCSP stapling
  - Session tickets
  - Perfect forward secrecy
- Protocol enforcement
  - Half or full bridge connections
- TLSv1.0, 1.1, 1.2
  - Field-level encryption

Integrations with AWS security services

- AWS WAF
- AWS Shield
- AWS Certificate Manager (ACM)
- AWS Identity and Access Management (IAM)
- AWS Config
- AWS CloudTrail

API acceleration with CloudFront

- TLS termination at edge
- Network optimizations: persistent connections, connection pooling, keep-alive
- AWS private backbone
- Edge DDoS protection

"The performance gains are amazing, positively impacting our app's usage across the globe, especially in Regions further from US EAST 1."

Sample data from a customer test

Region	Avg SSL Negotiation w/o CDN	Avg SSL Negotiation w/ CDN	SSL Negotiation Latency Improvement
India	750 ms	50 ms	~93%
Australia (Sydney)	460 ms	50 ms	~90%
Indonesia	550 ms	70 ms	~87%
Africa (Mauritius)	650 ms	250 ms	~61%

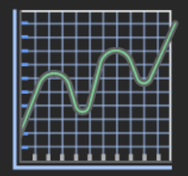
Region	Avg SSL Negotiation w/o CDN	Avg SSL Negotiation w/ CDN	SSL Negotiation Latency Improvement
Brazil	350 ms	50 ms	~81%
US (Los Angeles)	210 ms	60 ms	~71%
US (Denver)	180 ms	70 ms	~61%
Toronto	140 ms	90 ms	~36%

Region	Avg SSL Negotiation w/o CDN	Avg SSL Negotiation w/ CDN	SSL Negotiation Latency Improvement
Berlin	470 ms	50 ms	~89%
Paris	400 ms	70 ms	~82%
Brussels	410 ms	80 ms	~80%
Spain	460 ms	90 ms	~70%
London	280 ms	90 ms	~68%

Lambda@Edge

Lambda@Edge is an extension of AWS Lambda that enables you to run Node.js functions at AWS global edge locations in response to CloudFront events

No servers to manage



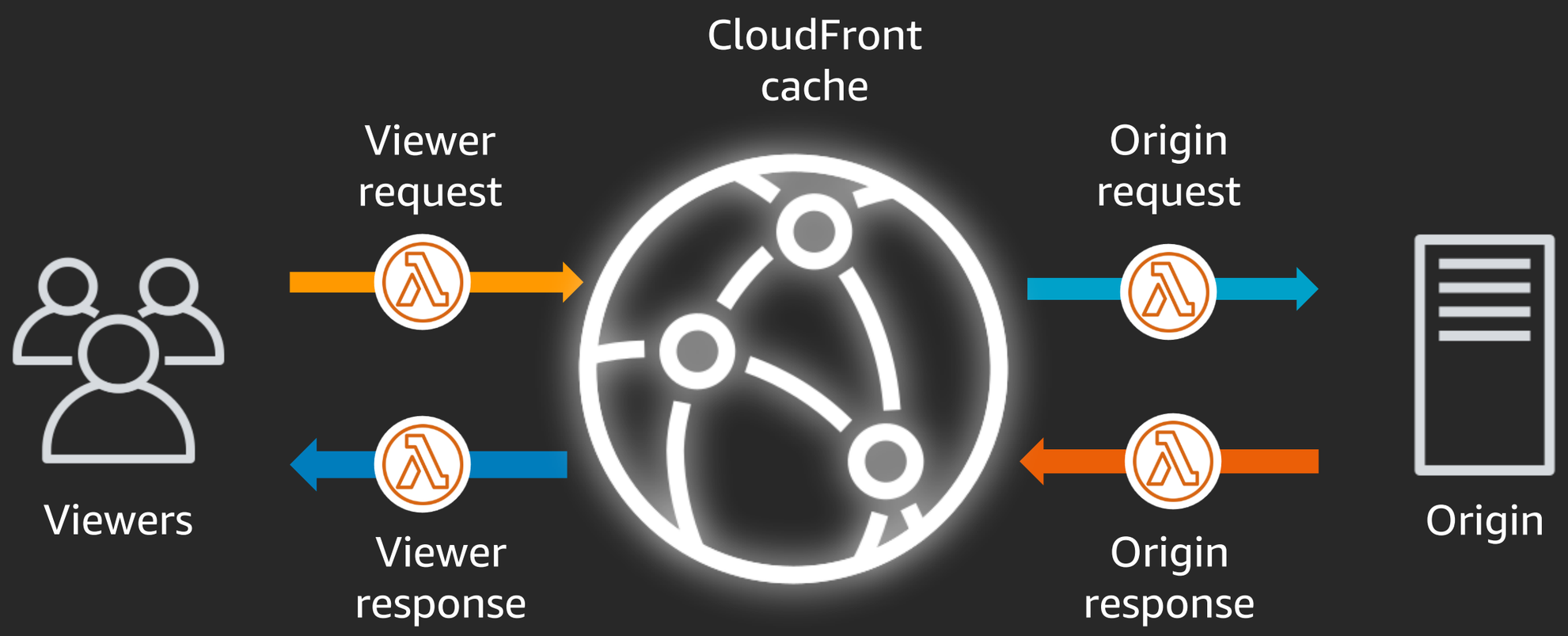
Continuous scaling

Globally distributed

Never pay for idle – no cold servers

- Improve viewer performance
- Reduce origin load / simplify origin architecture

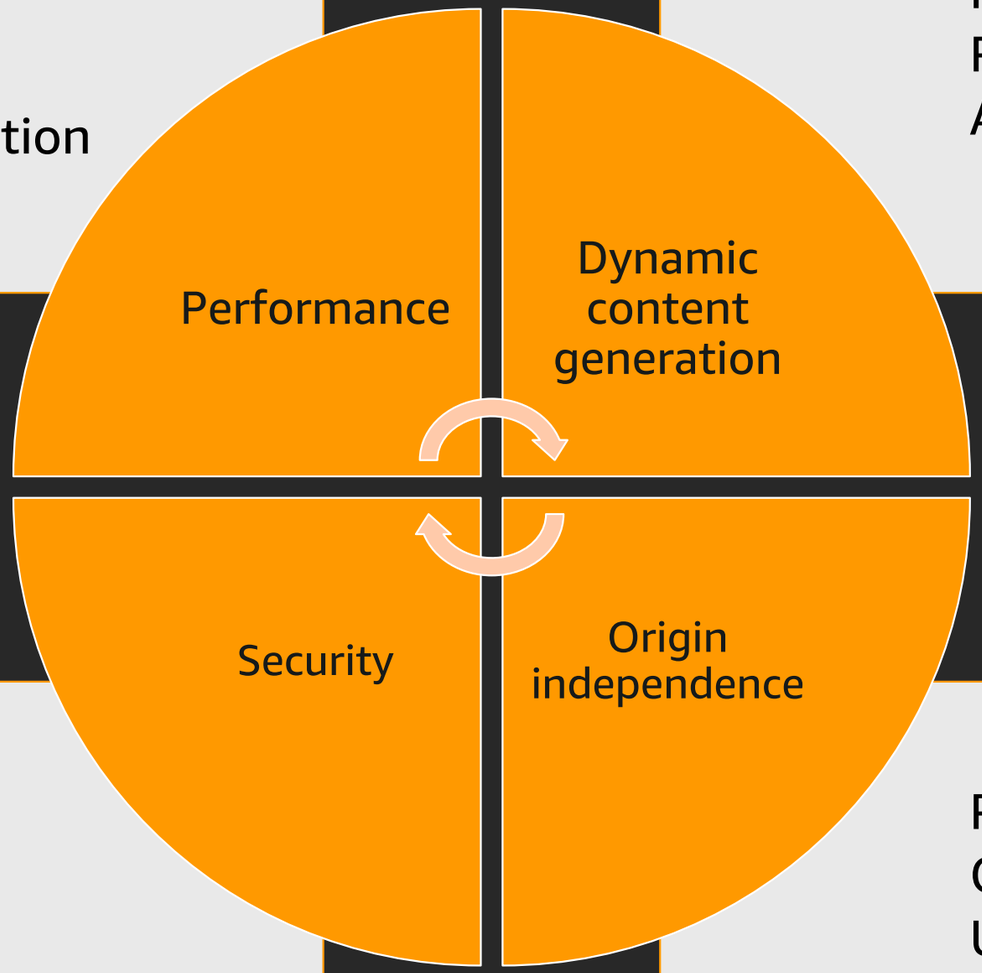
CloudFront and Lambda@Edge



Lambda@Edge use cases

Cache header manipulation
3xx follow redirection
Query string / UA normalization

Resize images
Render pages
A/B testing



Security headers
Token authentication
Sign requests to origin

Pretty URLs
Origin failover
URL rewriting

Amazon CloudFront customers

Media & Entertainment

amazon prime video
hulu
PBS
mlbam
Spotify
snapchat
Discovery COMMUNICATIONS
6 WEB

Gaming

King
Mm2
ROVIO
BANDAI NAMCO Games
CMUNE
SUPERCELL
NEOWIZ GAMES
NAUGHTY DOG
MOVIESTAR PLANET

Financial Services

intuit
intuit mint
nab more than money
HSBC
me Bank BANK FAIRER.
MACQUARIE

Ecommerce, Social Media, Digital Advertising, EdTech

amazon.com
Canon
instacart
TRUECar
evitamins Live Healthy. Save Money.
VIVINO
Allergan

Learn networking with AWS Training and Certification

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

Free digital courses cover topics related to networking and content delivery, including Introduction to Amazon CloudFront and AWS Transit Gateway Networking and Scaling

Validate expertise with the AWS Certified Advanced Networking – Specialty exam

Visit the advanced networking learning path at aws.amazon.com/training/path-advanced-networking

Thank you!

Sebastien Stormacq

@sebsto

aws SUMMIT ONLINE