AWS re:Invent

NET318-R

Building highly available applications using AWS Global Accelerator

James Wenzel
Sr Partner SA, Networking
Amazon Web Services, Inc.

Harvo Jones
Principal SDE
Amazon Web Services, Inc.

AWS re:Invent

aws

Agenda

- Building on AWS
  - Challenges of the internet
  - Creating and using an accelerator
  - Traffic control
- The power of AWS Global Accelerator
  - Security, availability, performance
- Building with AWS Global Accelerator

Key features we'll learn

Single entry point with global static anycast IPs

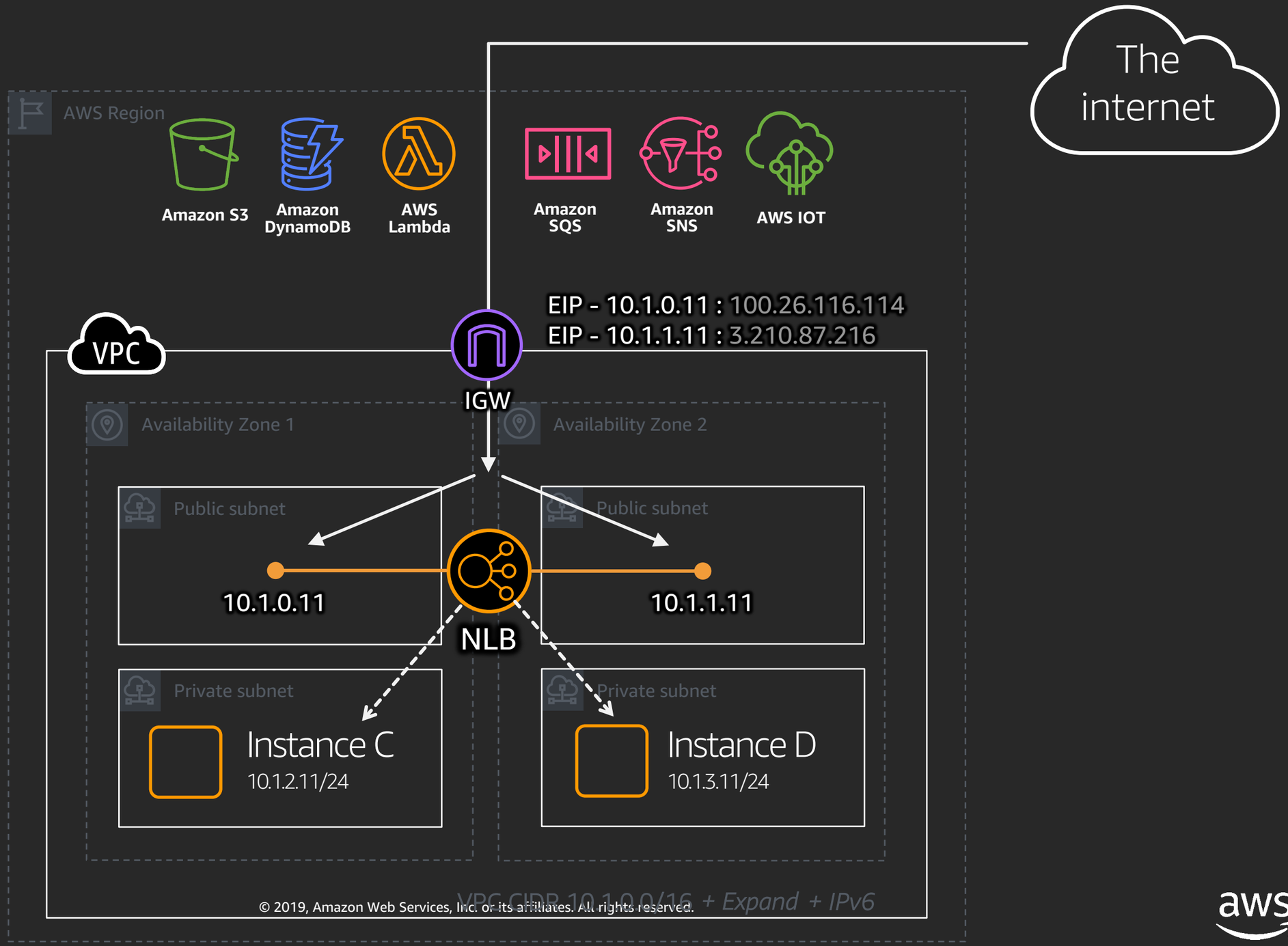
Intelligent distribution of TCP & UDP traffic

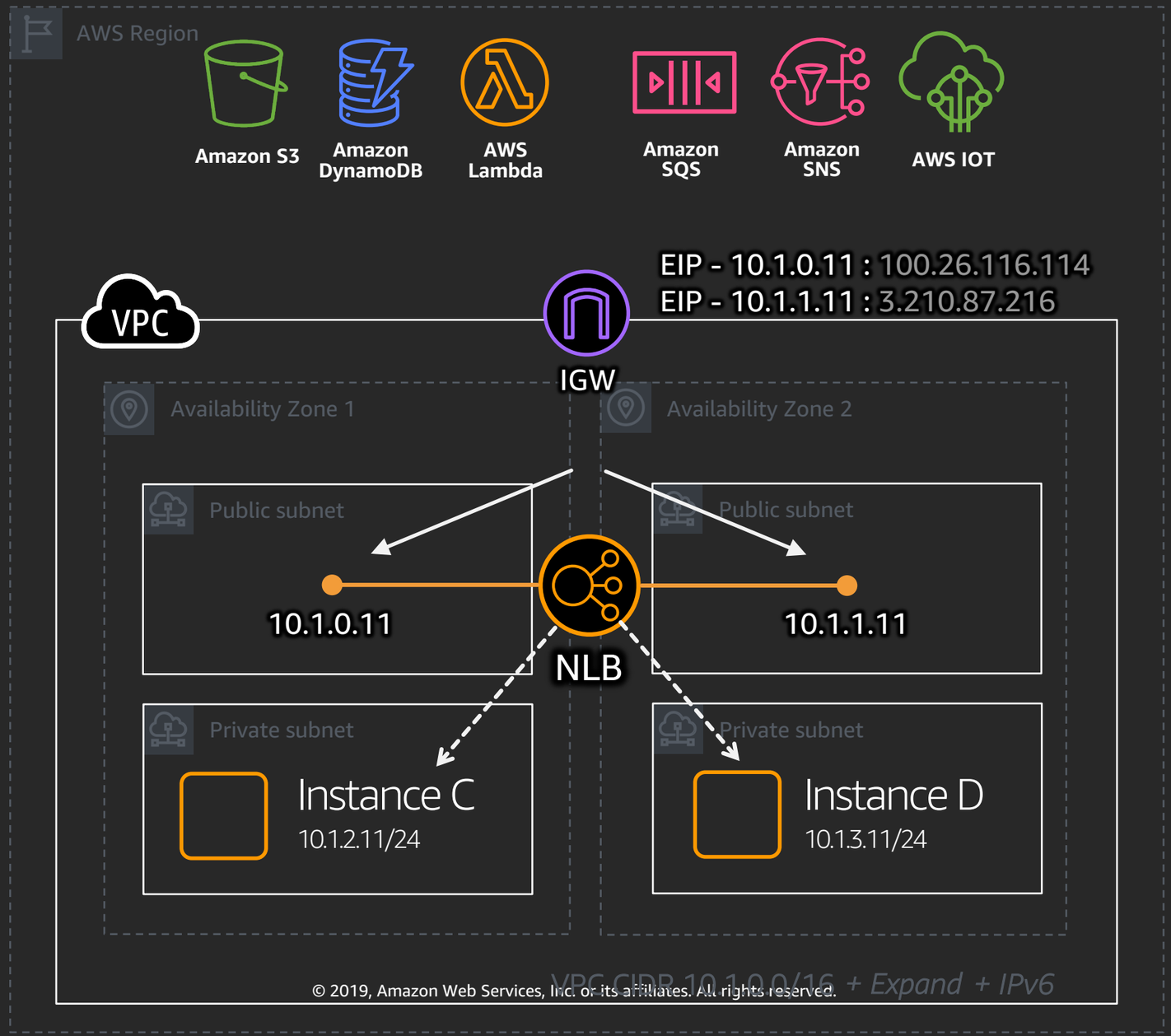
Target EC2 instances and Elastic Load Balancers (ALBs & NLBs) Preserve client IPs

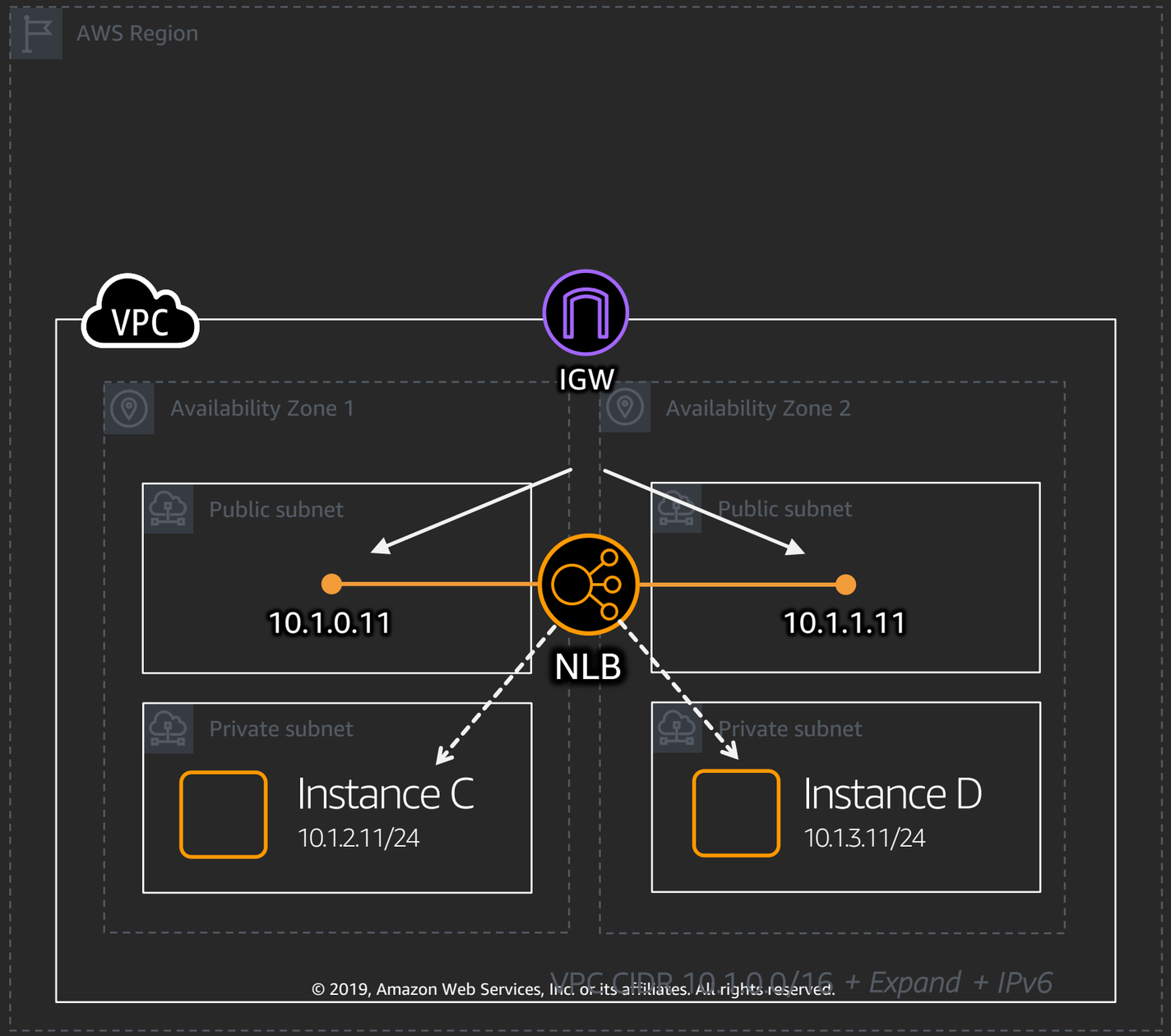
Accelerate both single and multi-region workloads

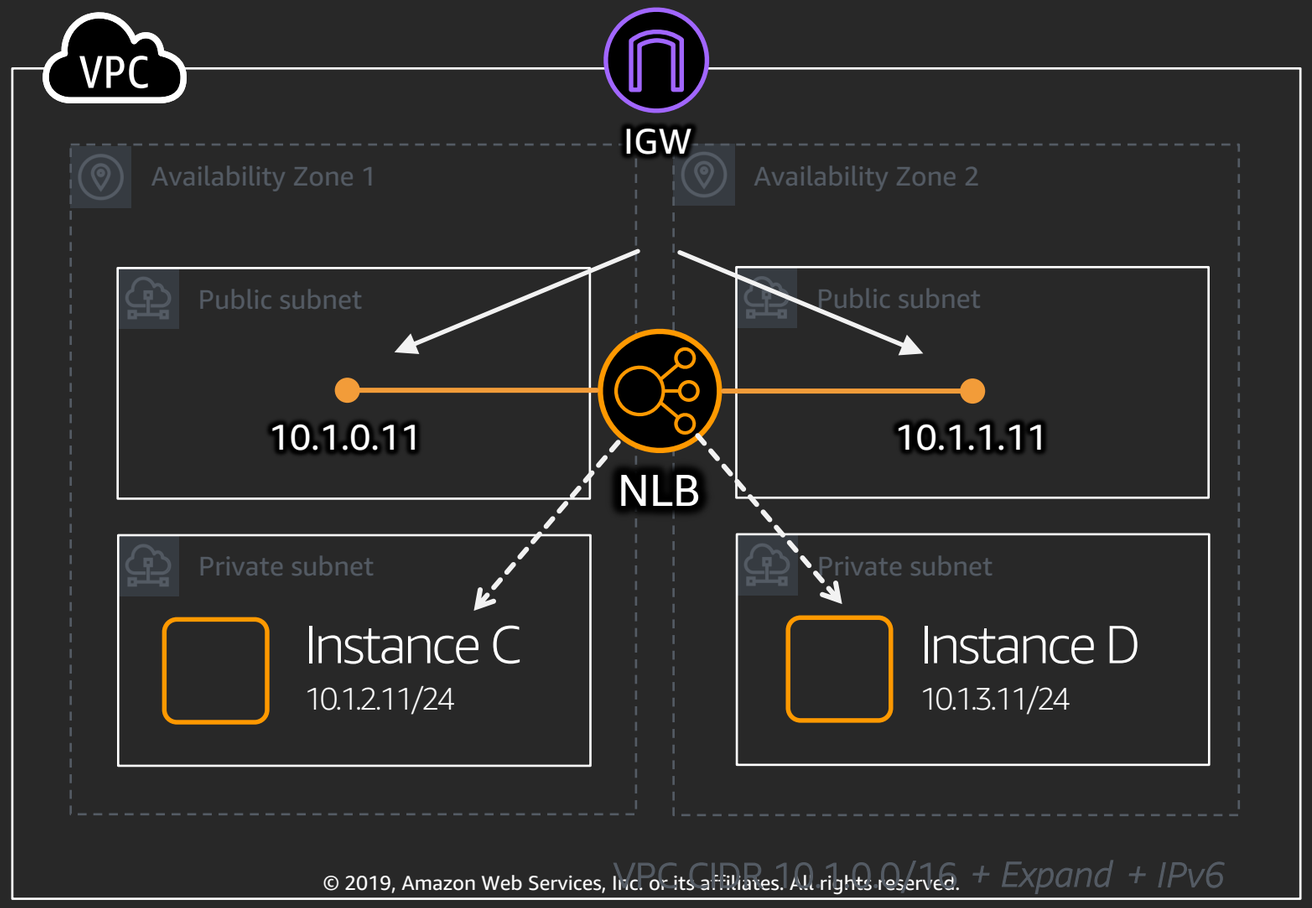
All traffic traverses the backbone* and is protected from DDoS attacks

* Except within the People's Republic of China

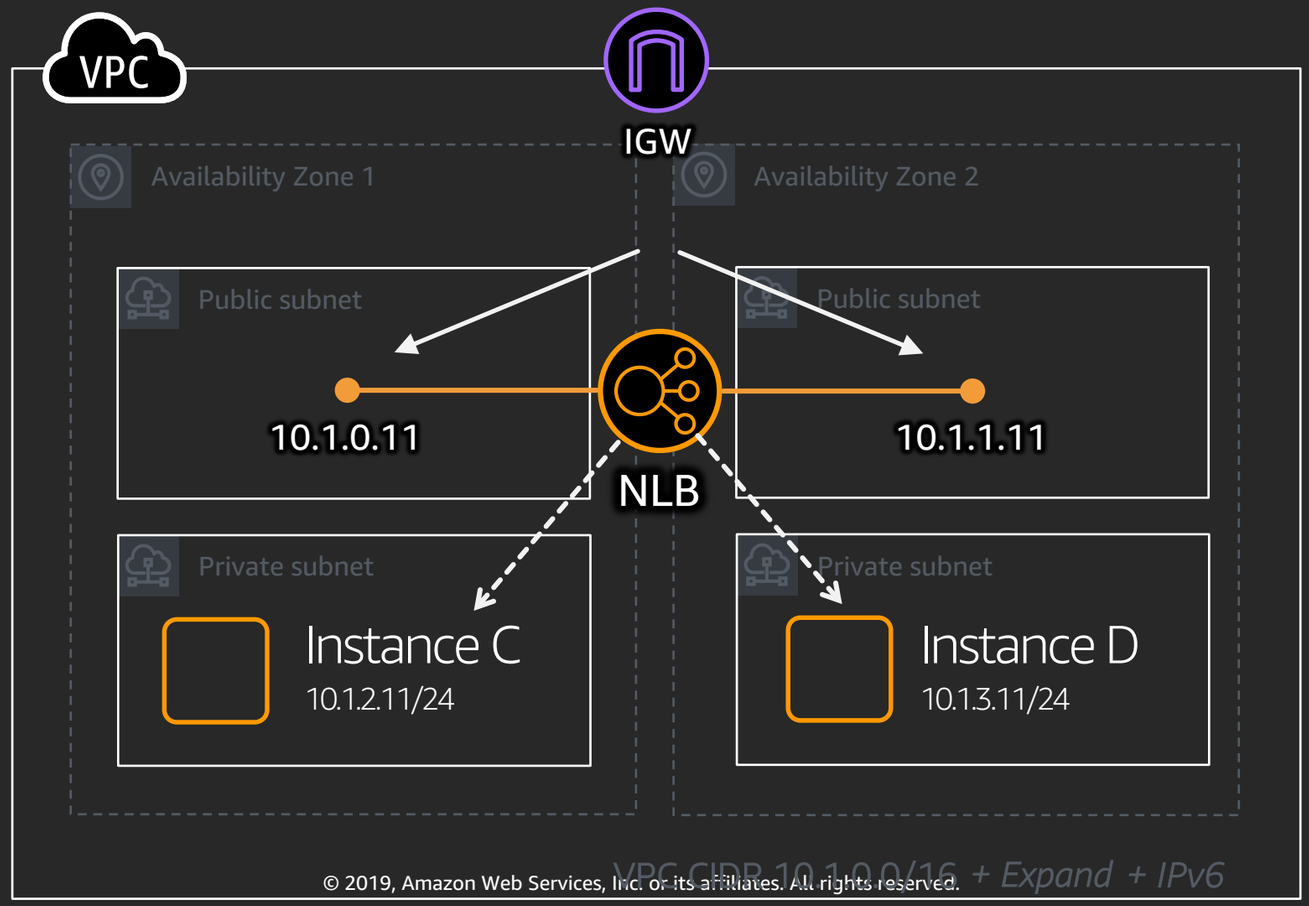


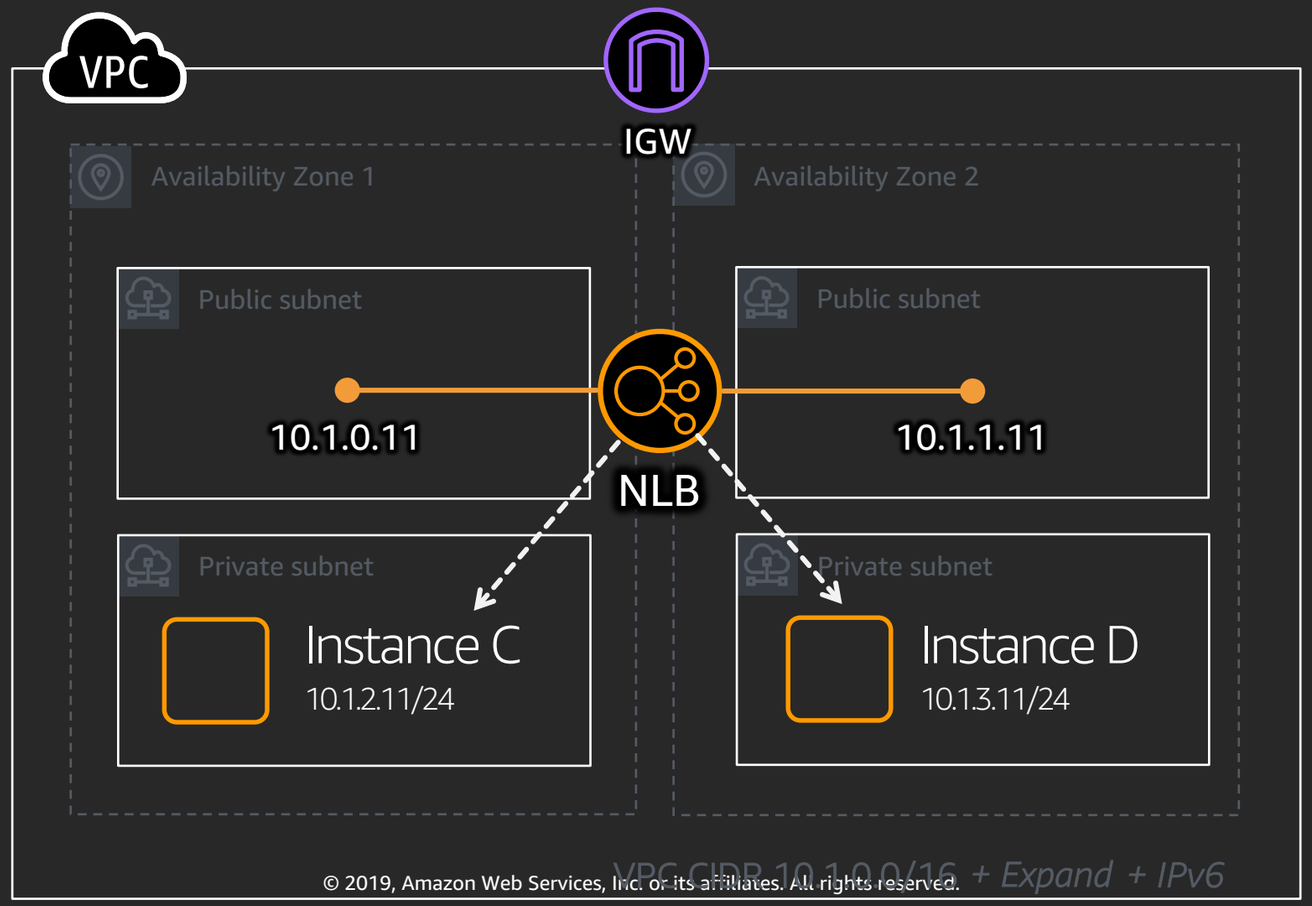






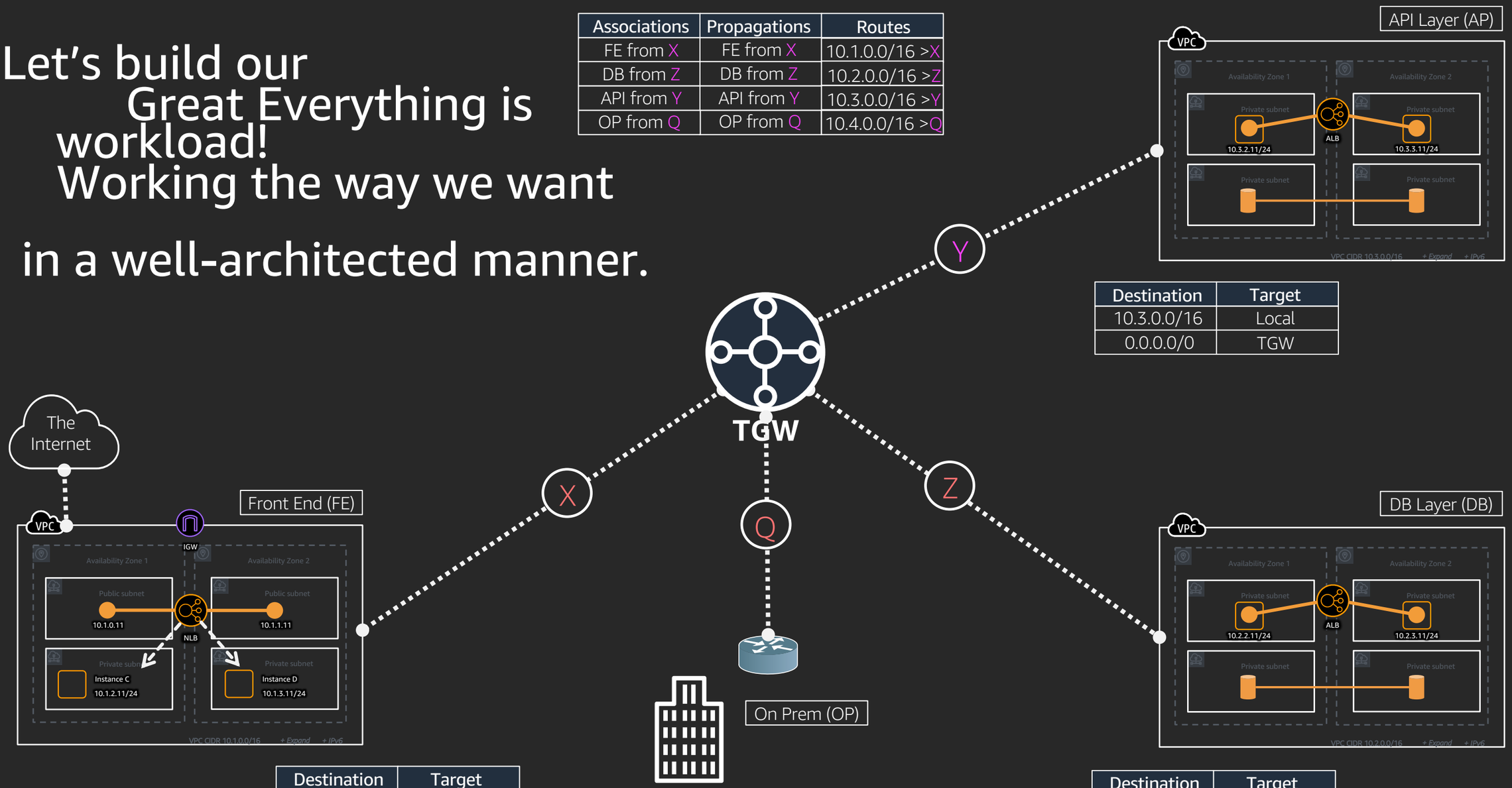
Let’s start with a common work load…





Let's build our Great Everything is workload! Working the way we want in a well-architected manner.

Associations	Propagations	Routes
FE from X	FE from X	10.1.0.0/16 >X
DB from Z	DB from Z	10.2.0.0/16 >Z
API from Y	API from Y	10.3.0.0/16 >Y
OP from Q	OP from Q	10.4.0.0/16 >Q



Destination	Target
10.3.0.0/16	Local
0.0.0.0/0	TGW

Destination	Target
10.1.0.0/16	Local
0.0.0.0/0	IGW
10.0.0.0/8	TGW

Destination	Target
10.2.0.0/16	Local
0.0.0.0/0	TGW

Now we open up our workload to the world

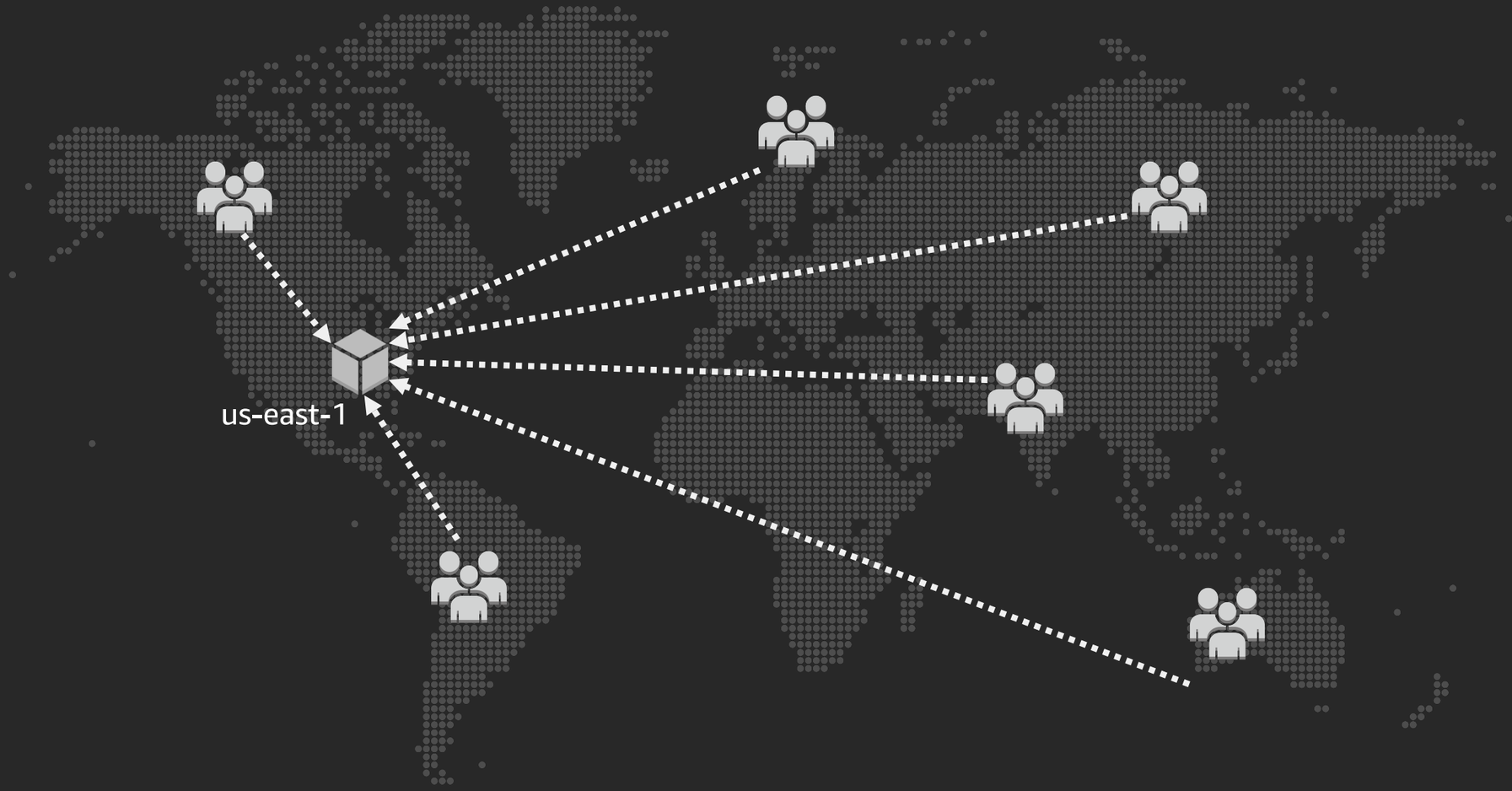
AWS re:Invent

aws

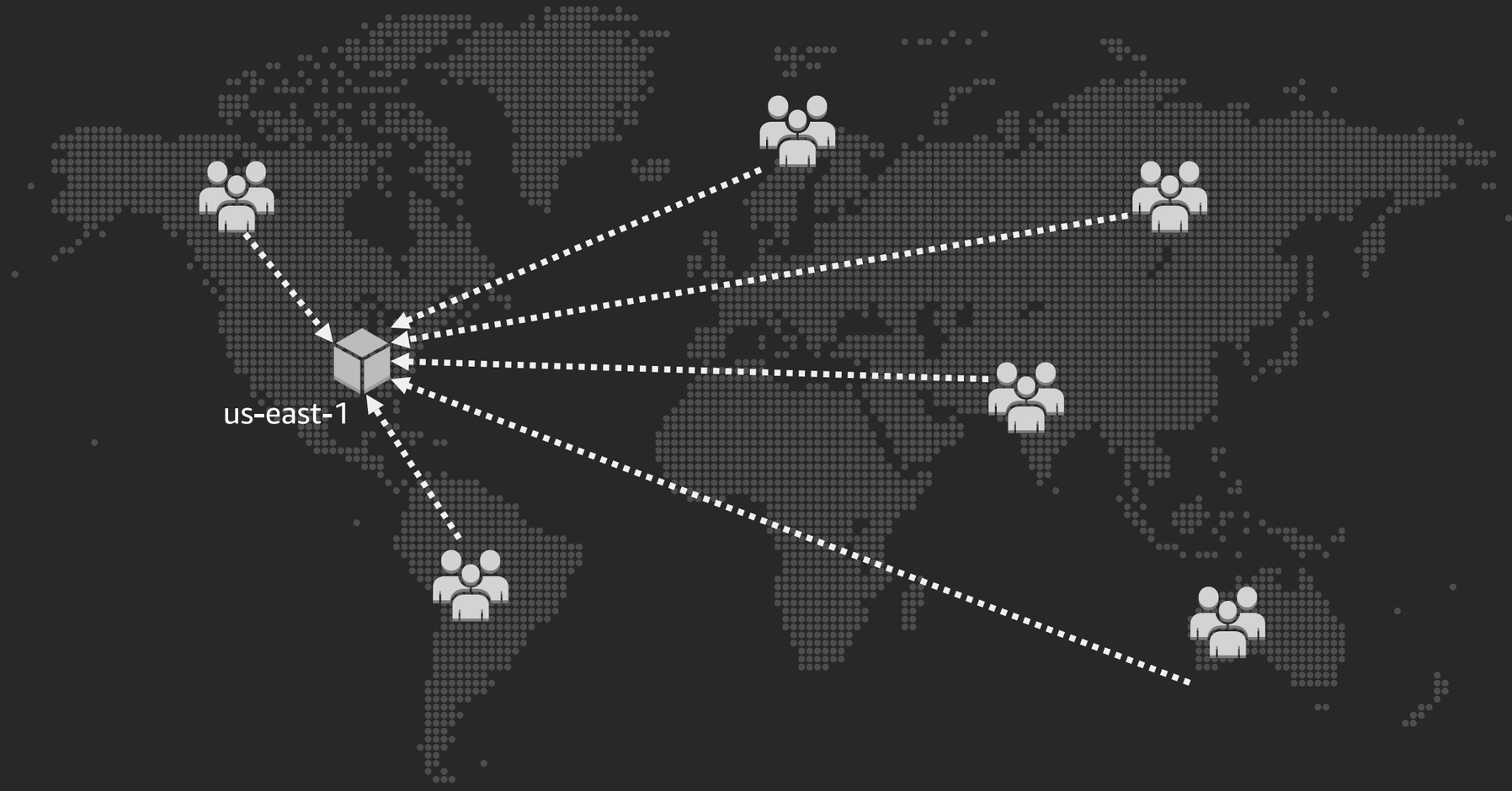
Now we open up our workload to the world

Because we are on the internet, it's accessible from everywhere.

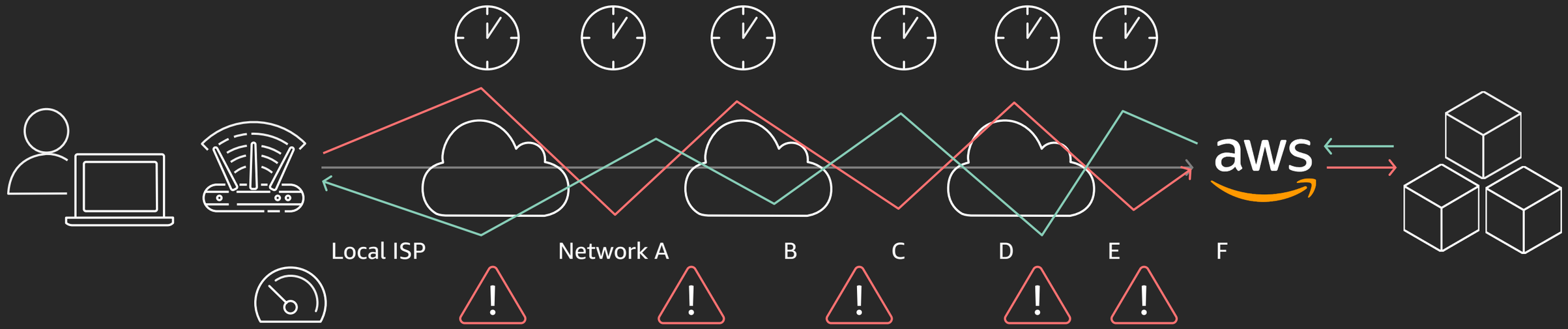
Because we are on the internet, it's accessible from everywhere.



Not all of our customers will have the same experience due to internet weather…



Internet weather



Accessing your application is not this straightforward
It can take many networks to reach the application
Paths to and from the application may differ
Each hop impacts performance and can introduce risk

There are a few ways to handle this…

Amazon Route 53

Our global DNS solution

with routing capabilities

Amazon CloudFront

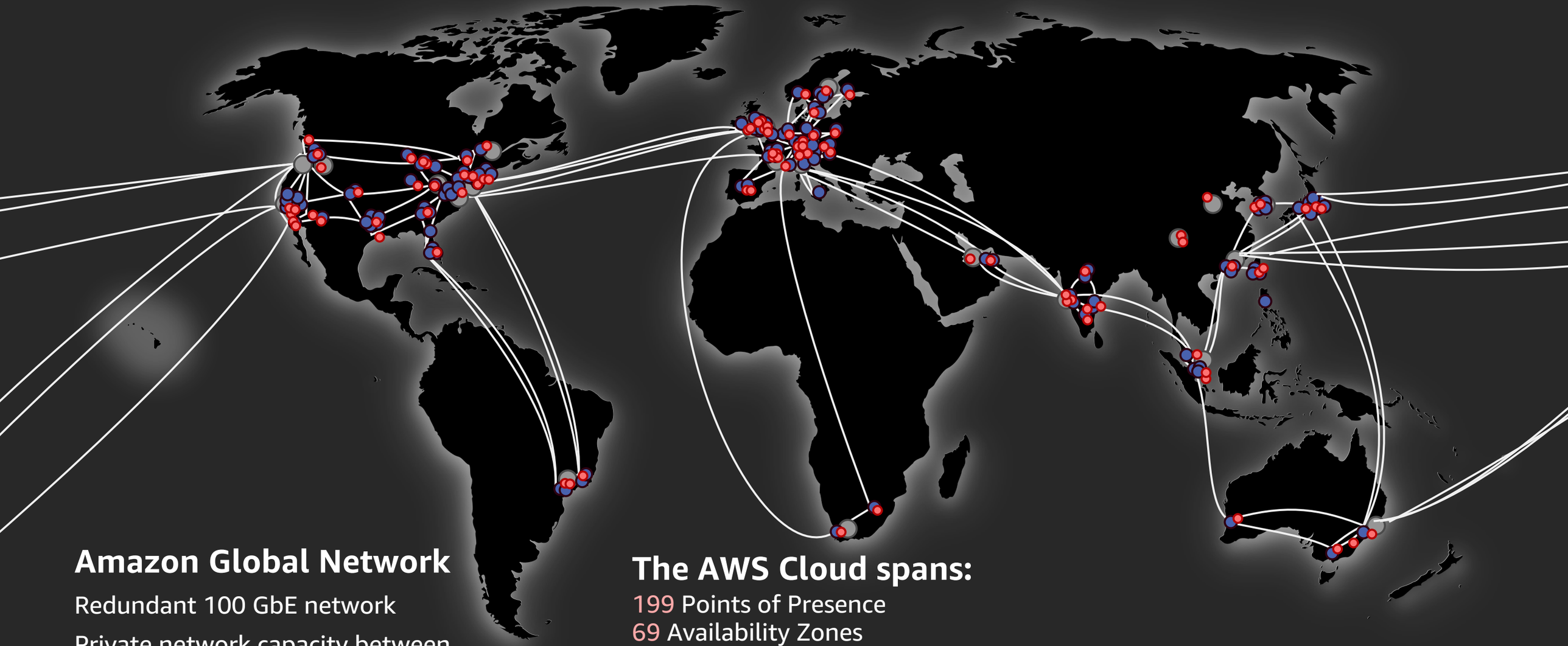
Our HTTP/S CDN solution

with WAF and Edge Compute

AWS Global Accelerator

Accelerators that direct TCP & UDP traffic to optimal endpoints over the AWS global network

What makes a global accelerator ?



Amazon Global Network

Redundant 100 GbE network

Private network capacity between all AWS Regions, except China

The AWS Cloud spans:

199 Points of Presence
69 Availability Zones
22 Geographic Regions around the world*

*With announced plans for 13 more Availability Zones and four more AWS Regions in Cape Town, Jakarta, Milan, and Spain.

AWS re:Invent

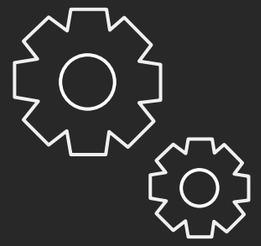
aws

What does having a global backbone do for our customers?

Security

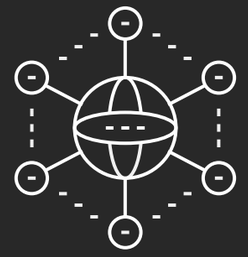
Traffic traverses our infrastructure rather than the internet

*End-to-end backbone encryption



Reliable performance

Controlling paths customer traffic traverses



Availability

Controlling scaling and redundancy

Connecting closer to customers

Avoiding internet hot spots or suboptimal external connectivity

What does this look like for our workloads with AGA?

This lets us reduce jitter and latency

Traffic enters the AWS global network at edge locations

Leverages the Global AWS network

Resulting in improved performance

Accessibility with a globally accessible Anycast IP set

What is Anycast?

Anycast is a network addressing and routing methodology
in which a single destination address has
multiple routing paths to N+ endpoint destinations.

Anycast helps us

Whitelisting only 2 IPs

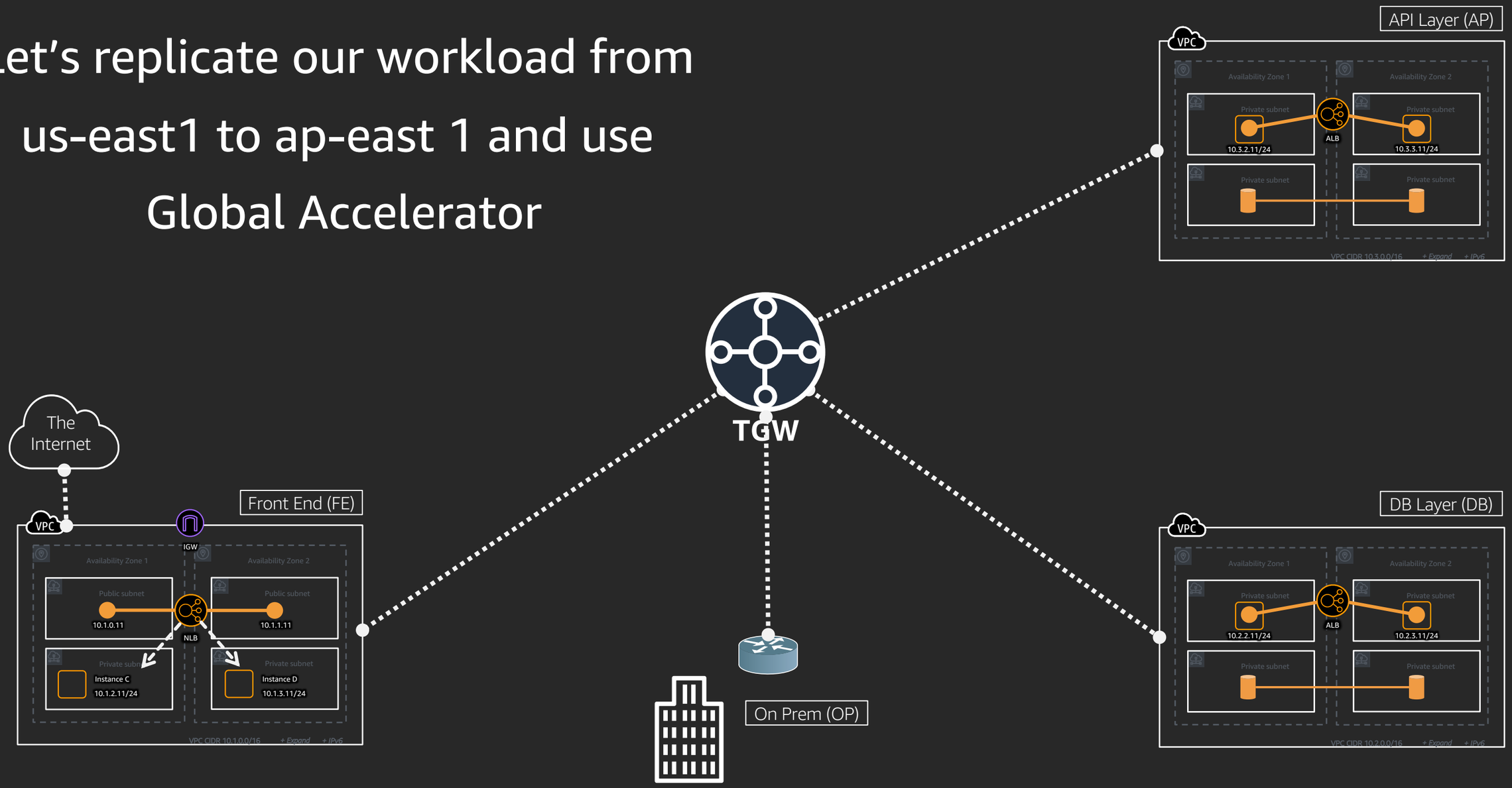
Reduce global address complexity

Source obfuscation

(your endpoints can be private)

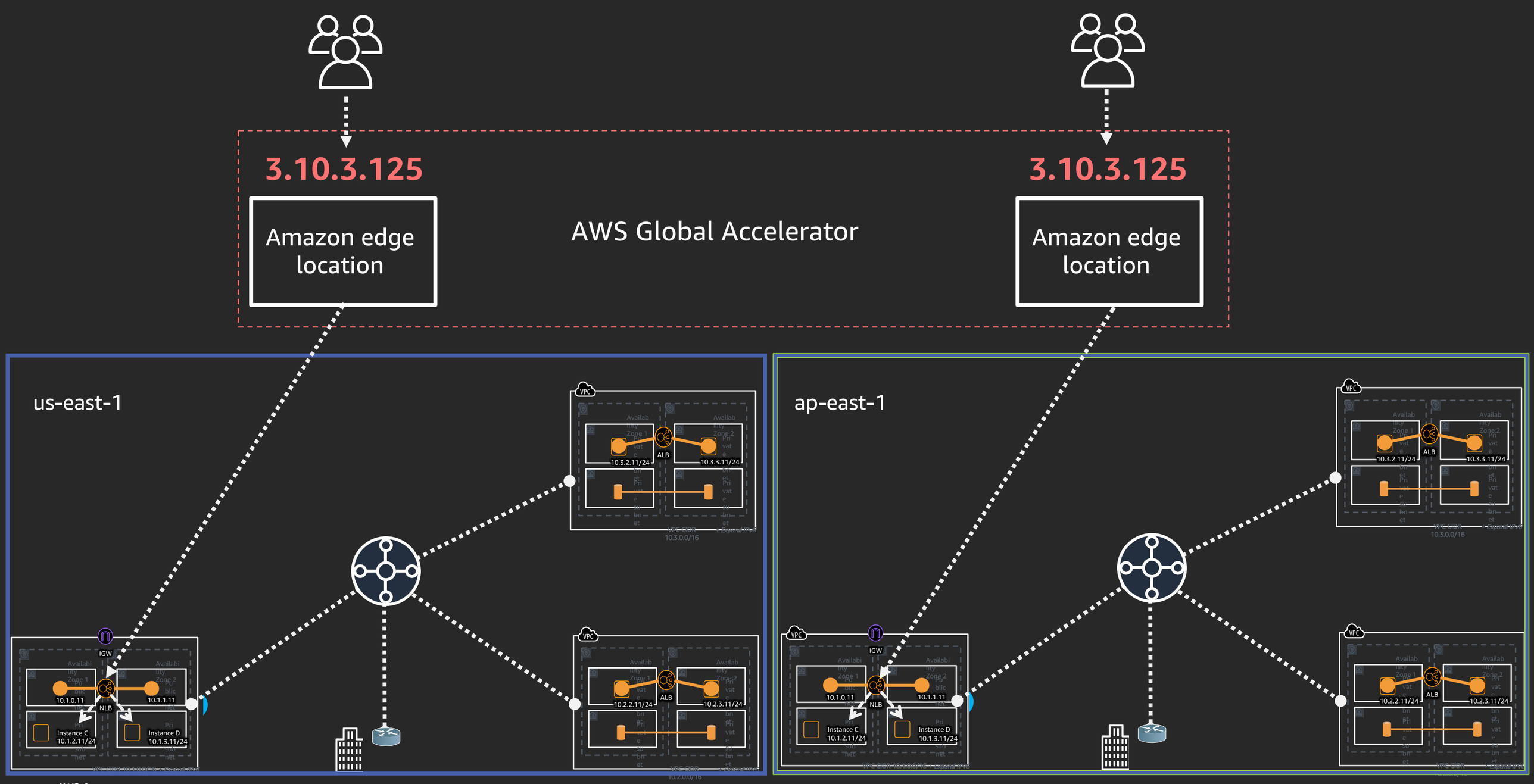
Gives us a global presence

Let’s replicate our workload from us-east1 to ap-east 1 and use Global Accelerator



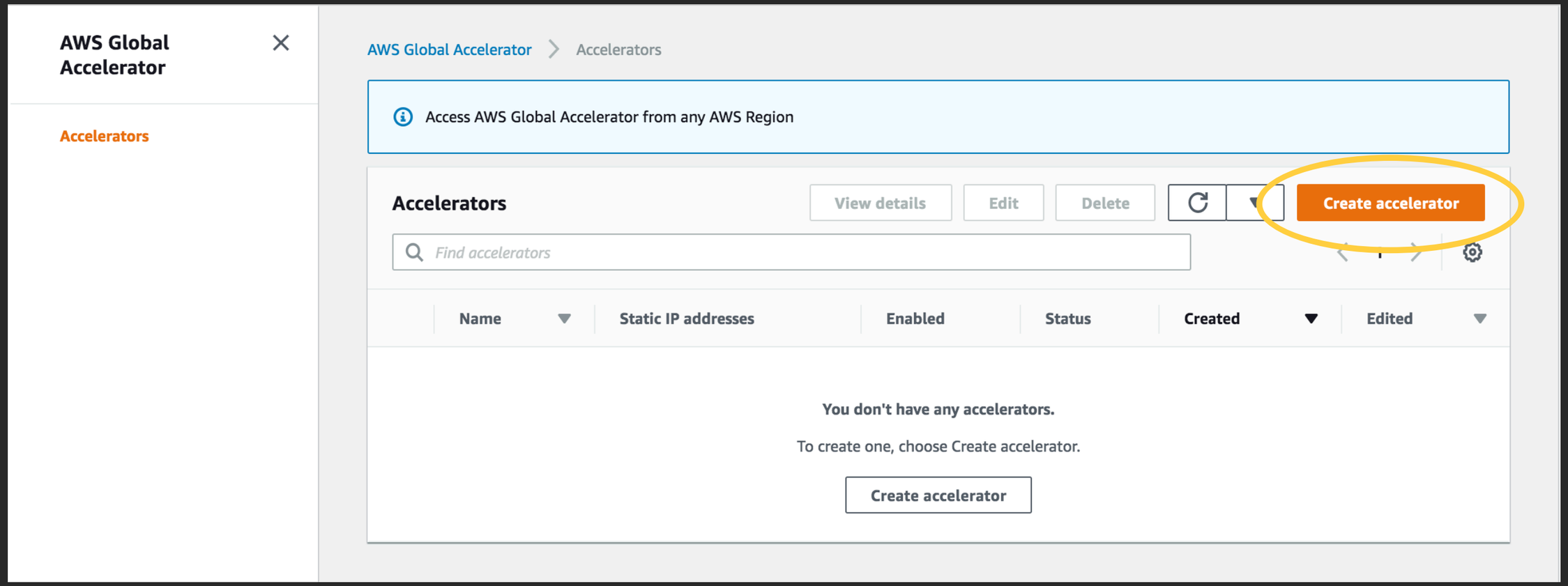
AWS re:Invent

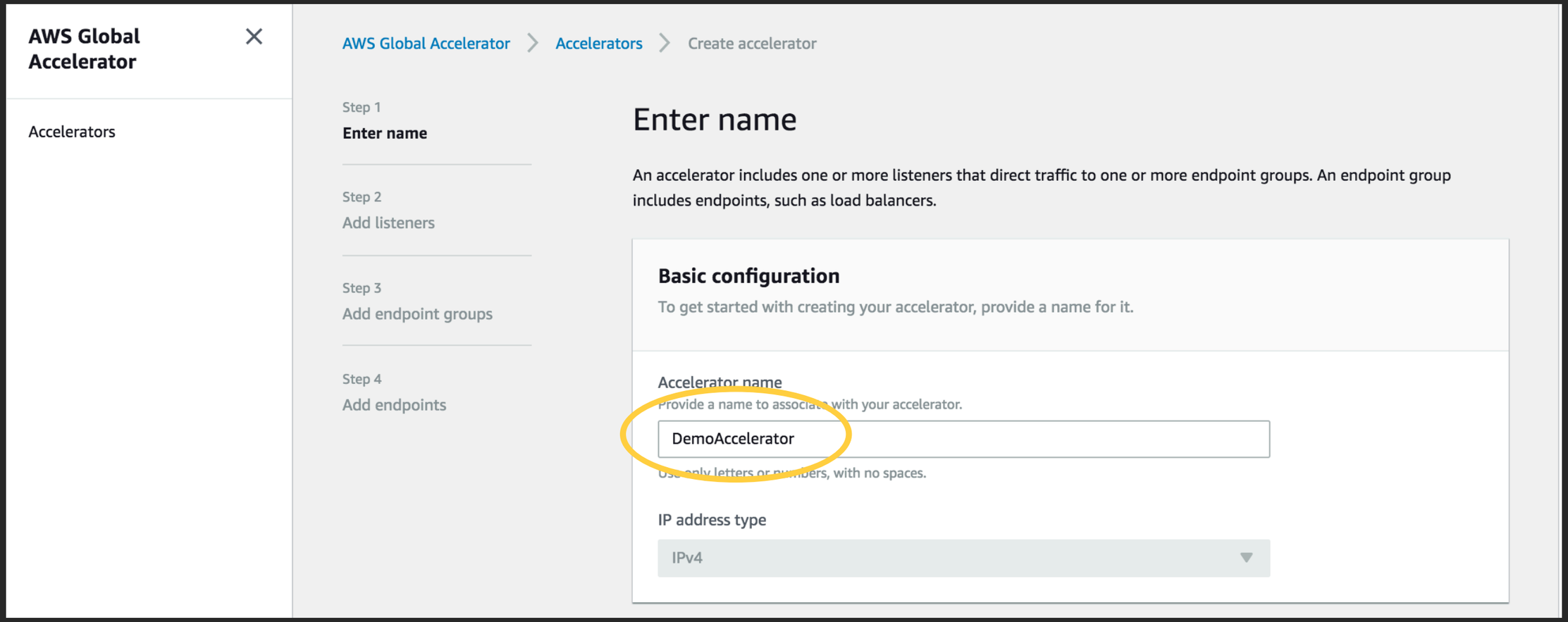
aws



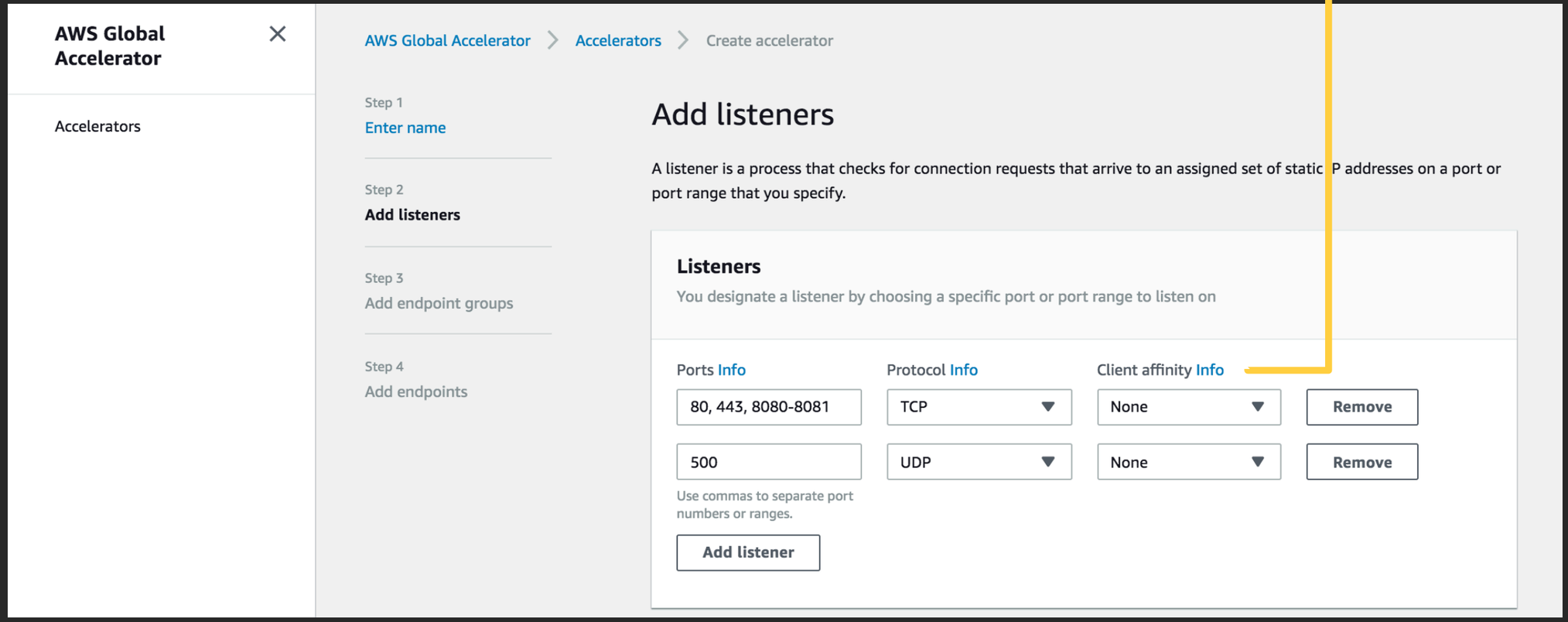
Creating your accelerator

Navigate to Global Accelerator in the console in the networking section

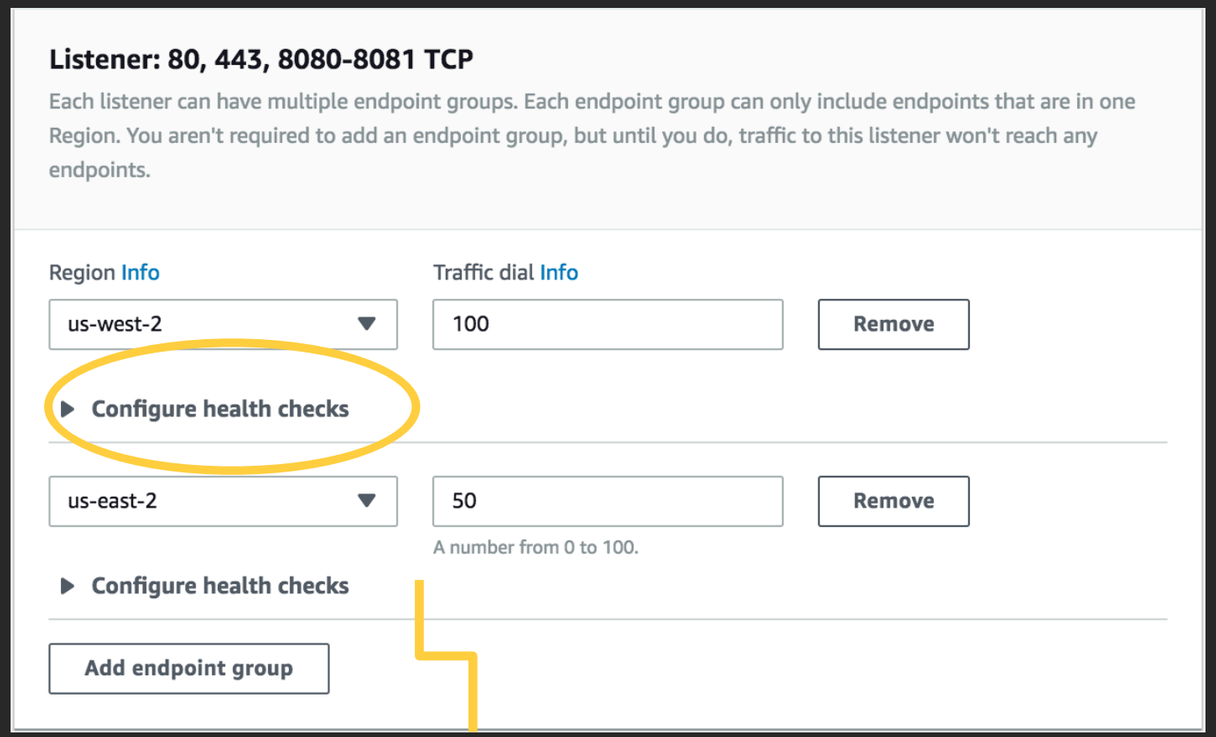


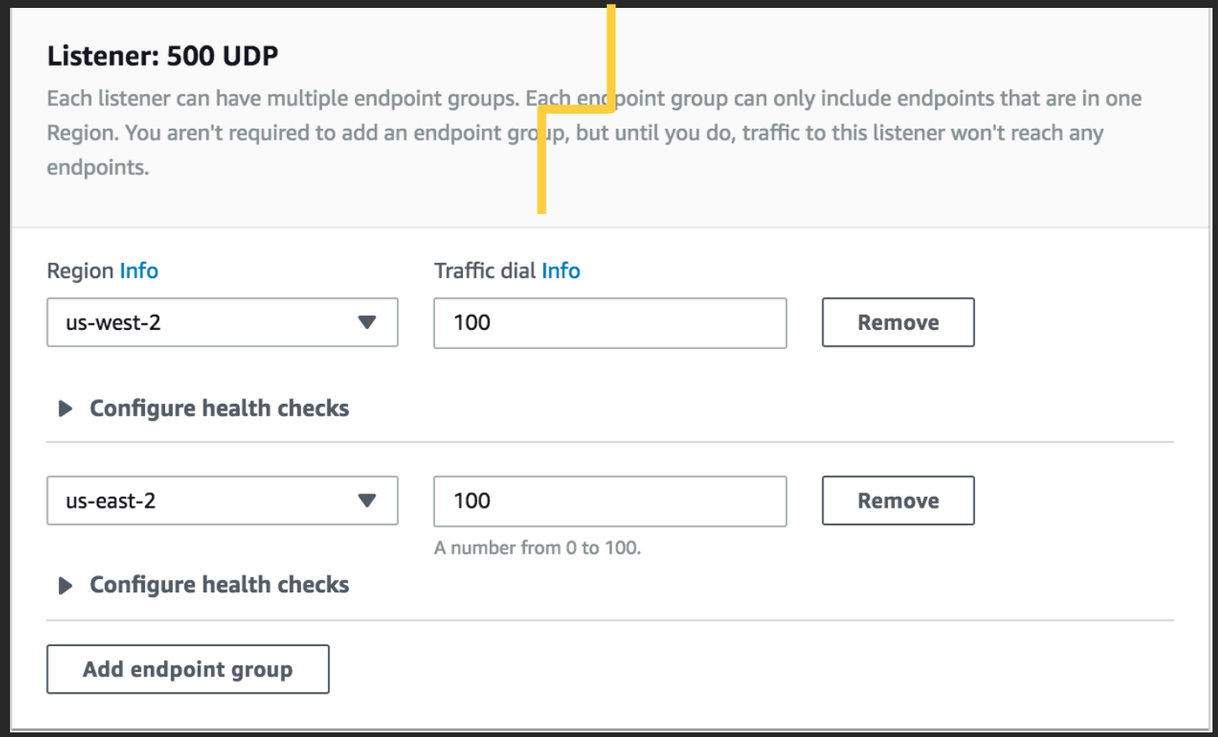


Choose SourceIP if you have stateful applications requiring affinity



Traffic dial is from 0 – 100% of traffic you want to send to this workload





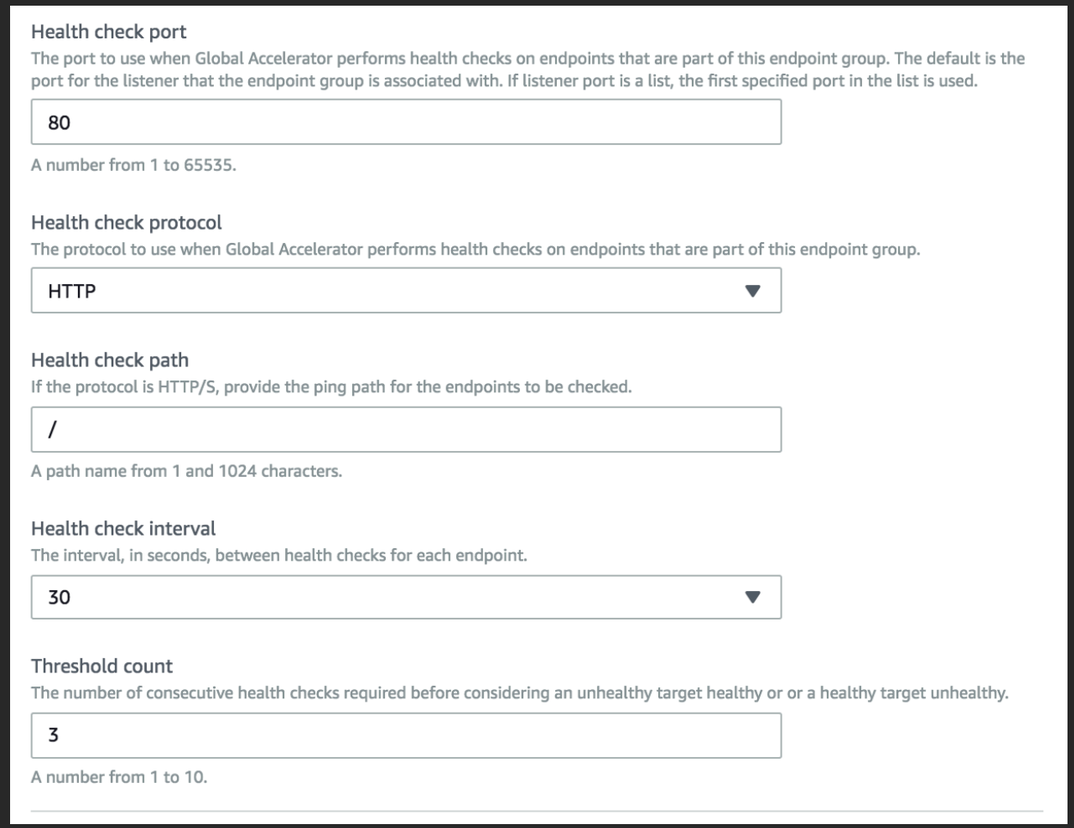
Create an endpoint group for every AWS Region where your application is hosted.

Health checks are for Elastic IPs and EC2 instances

ELB health checks are inherited from the load balancer

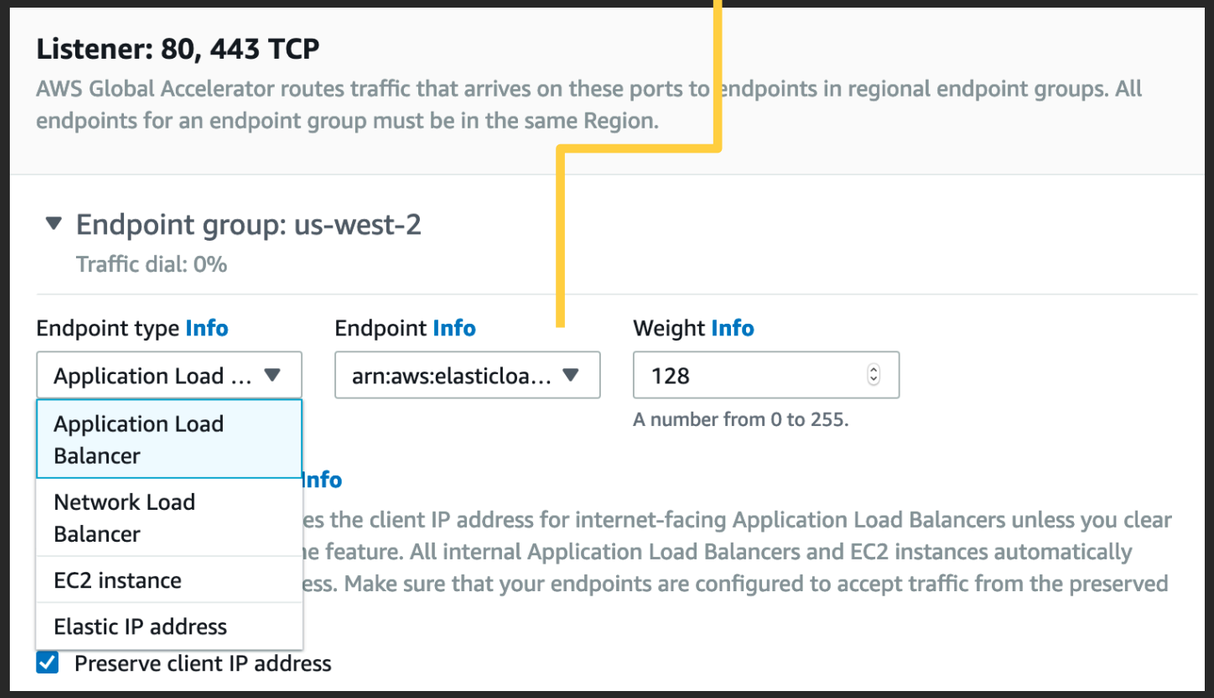
Health checks for EIPs can be TCP, HTTP or HTTPS

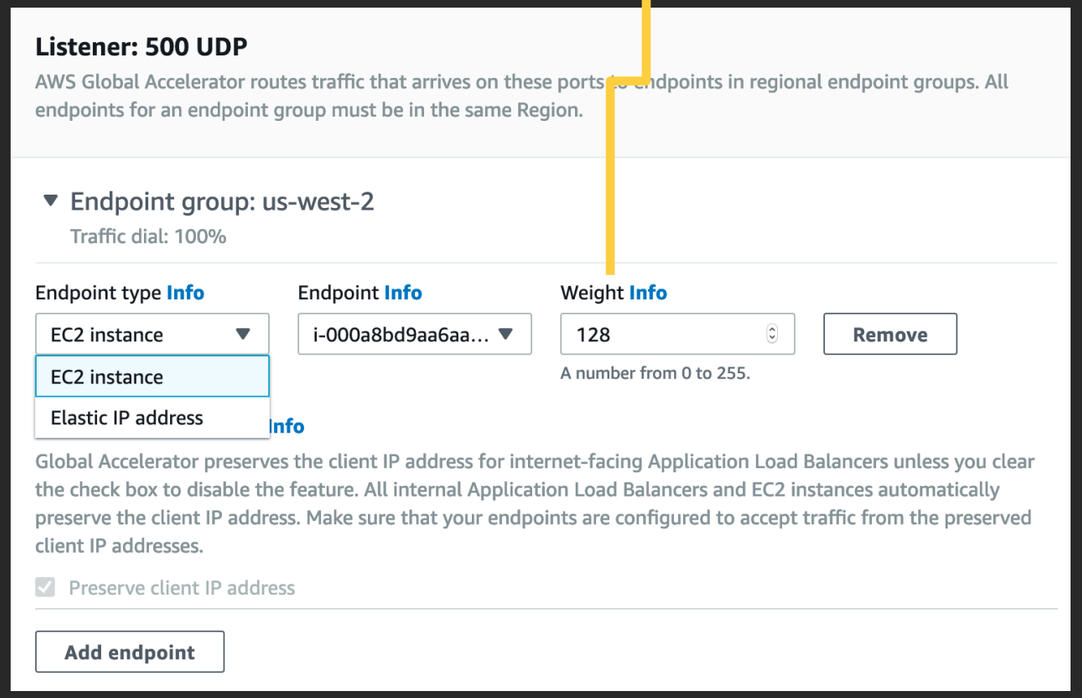
Remember UDP Health Checks are not Supported at this time.



Available resource ARNs will appear in the endpoint info dropdown

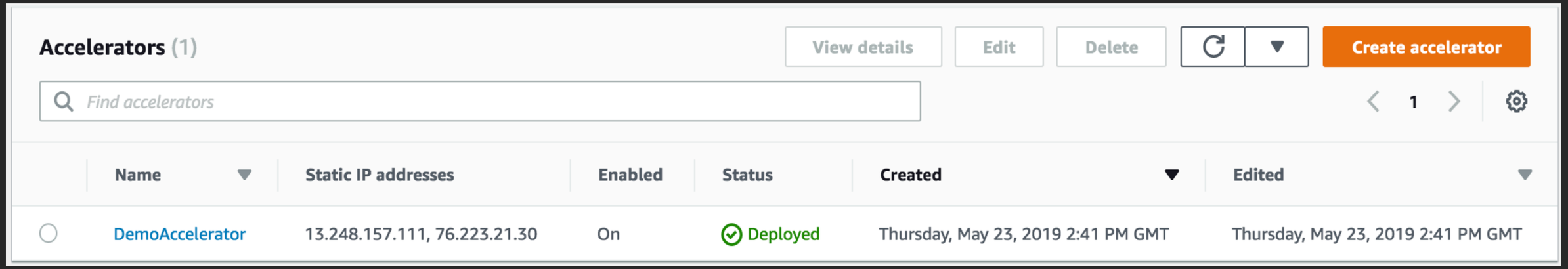
Weight lets you portion your traffic to a resource in the endpoint group





All of the Region's endpoint options will show in the dropdown

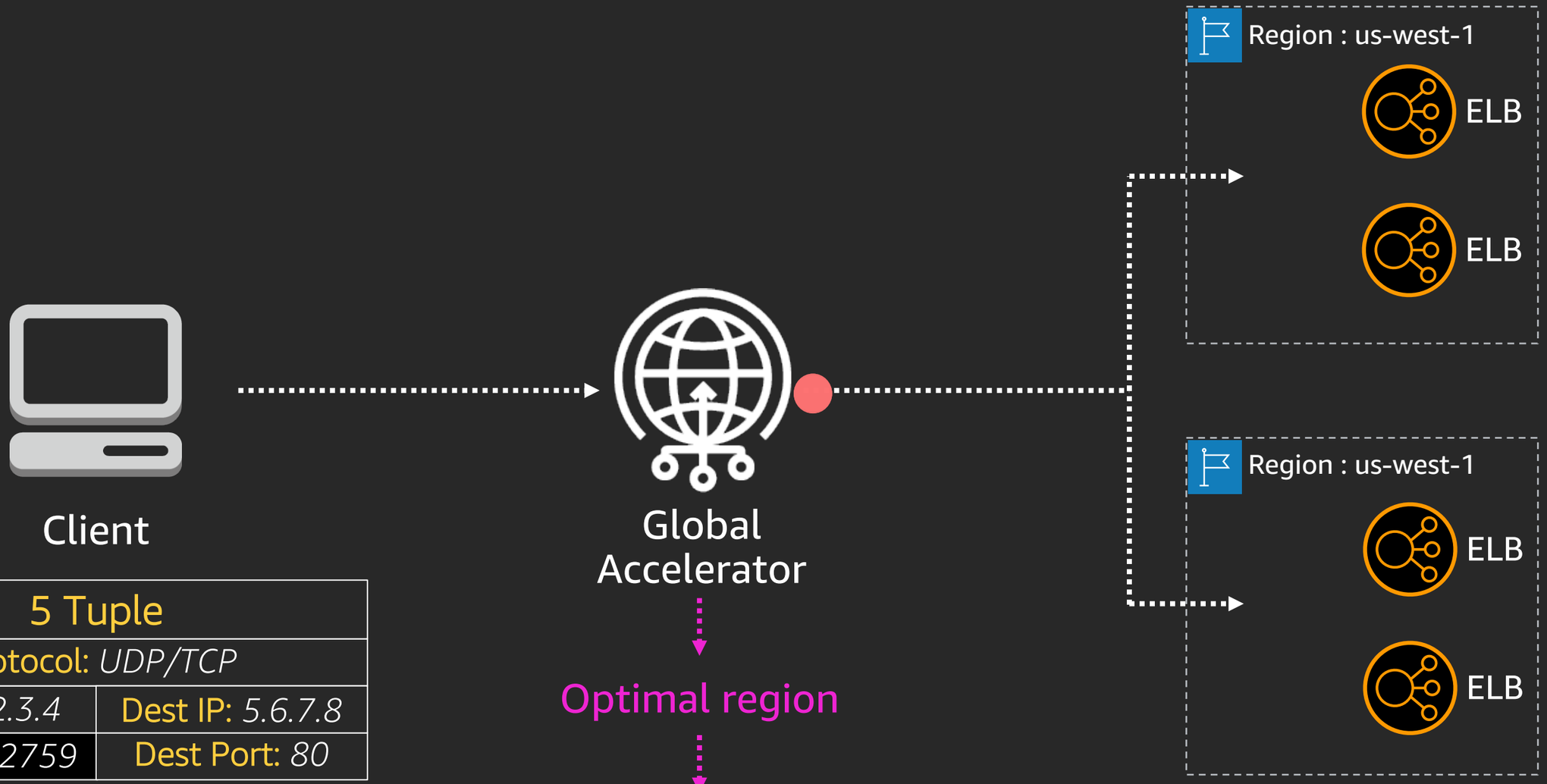
Now the Global Accelerator is up and deployed.



Allocated a set of IP addresses. Each is in an isolated network zone. If one zone fails, the other zone takes over.

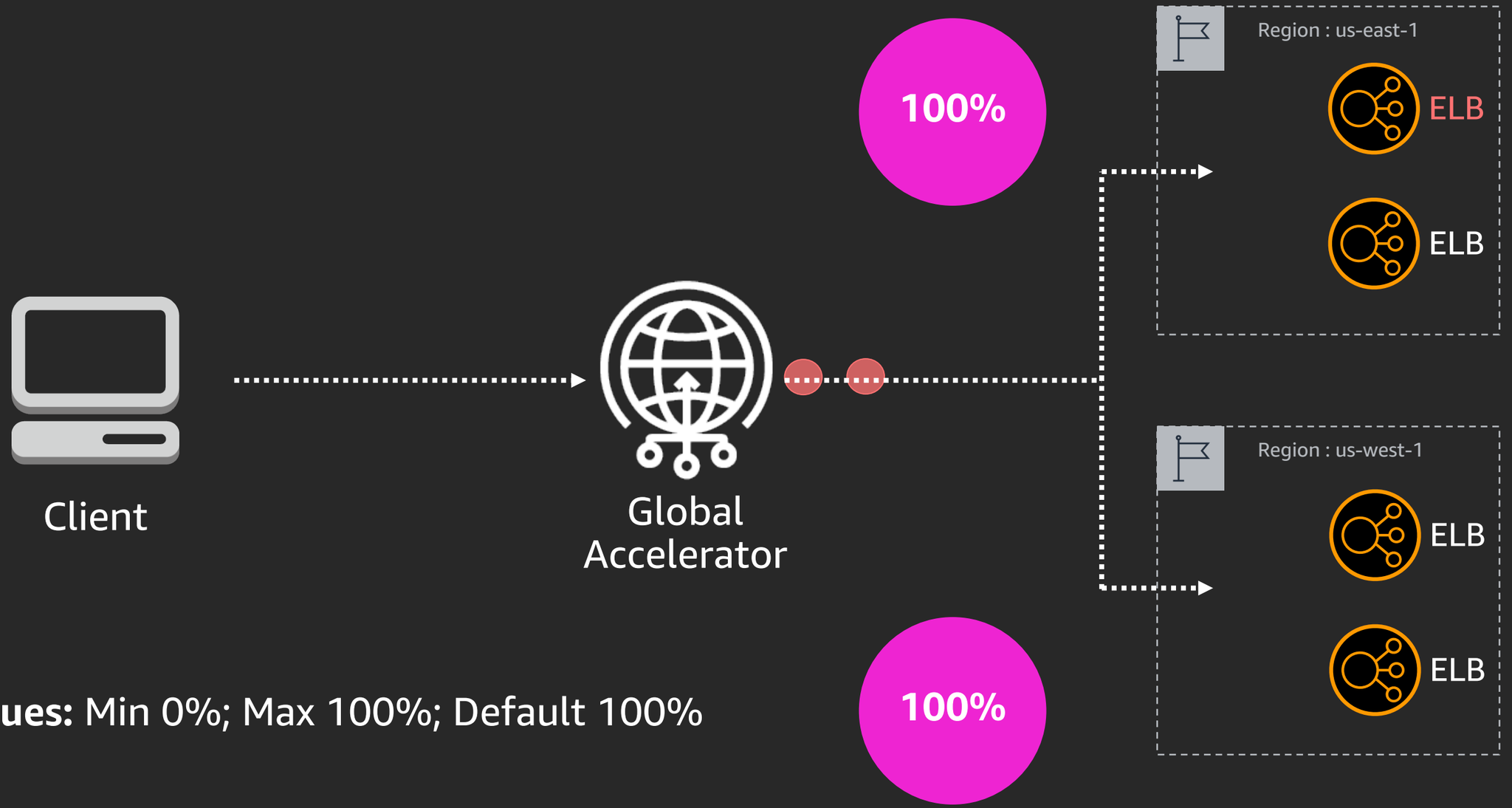
Traffic control

Easy traffic control – Optimal endpoint selection



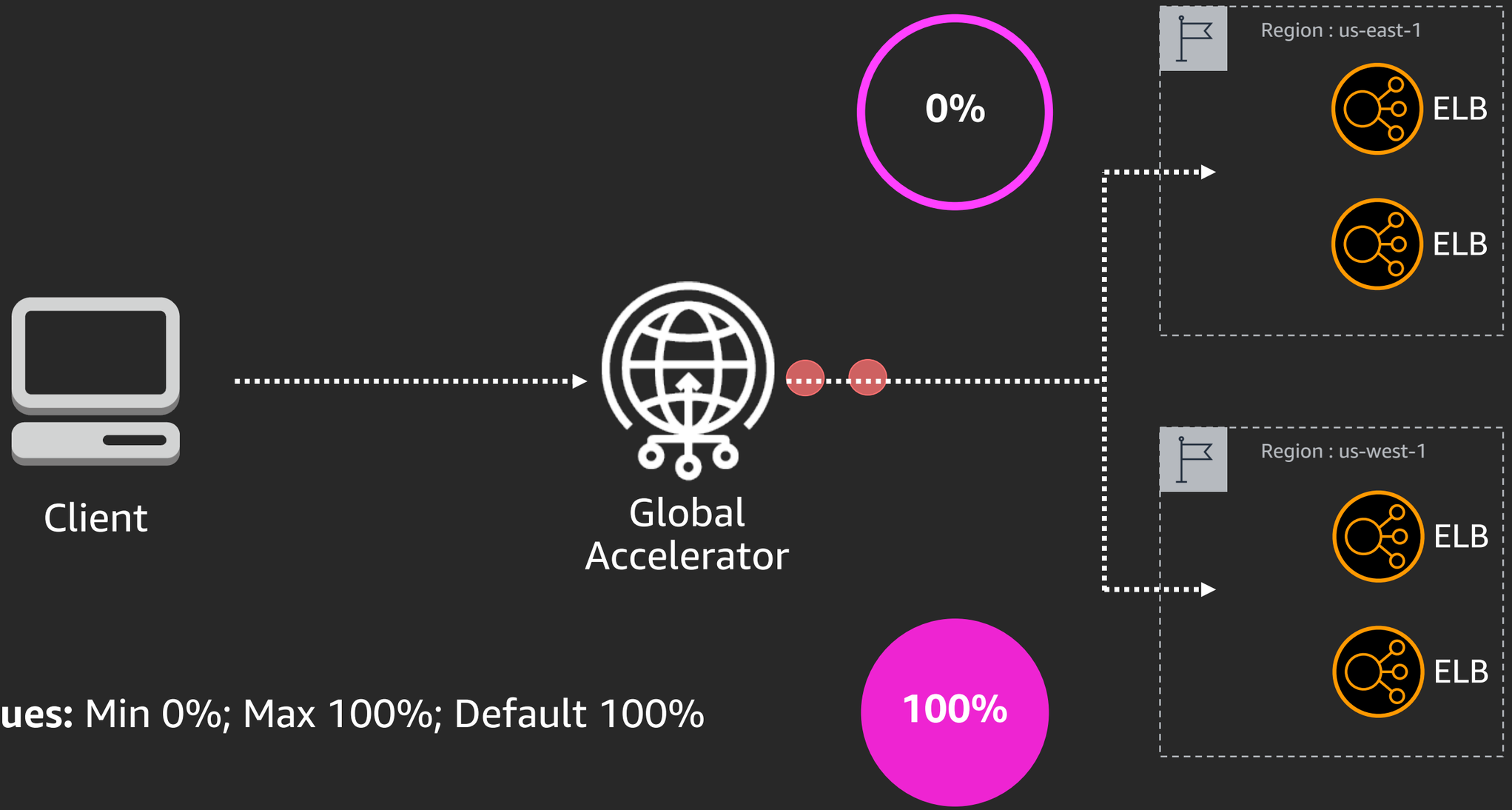
5 Tuple	
Protocol: *UDP/TCP*	
Src IP: *1.2.3.4*	Dest IP: *5.6.7.8*
Src Port: *32759*	Dest Port: *80*

Easy traffic control – Regional traffic dials



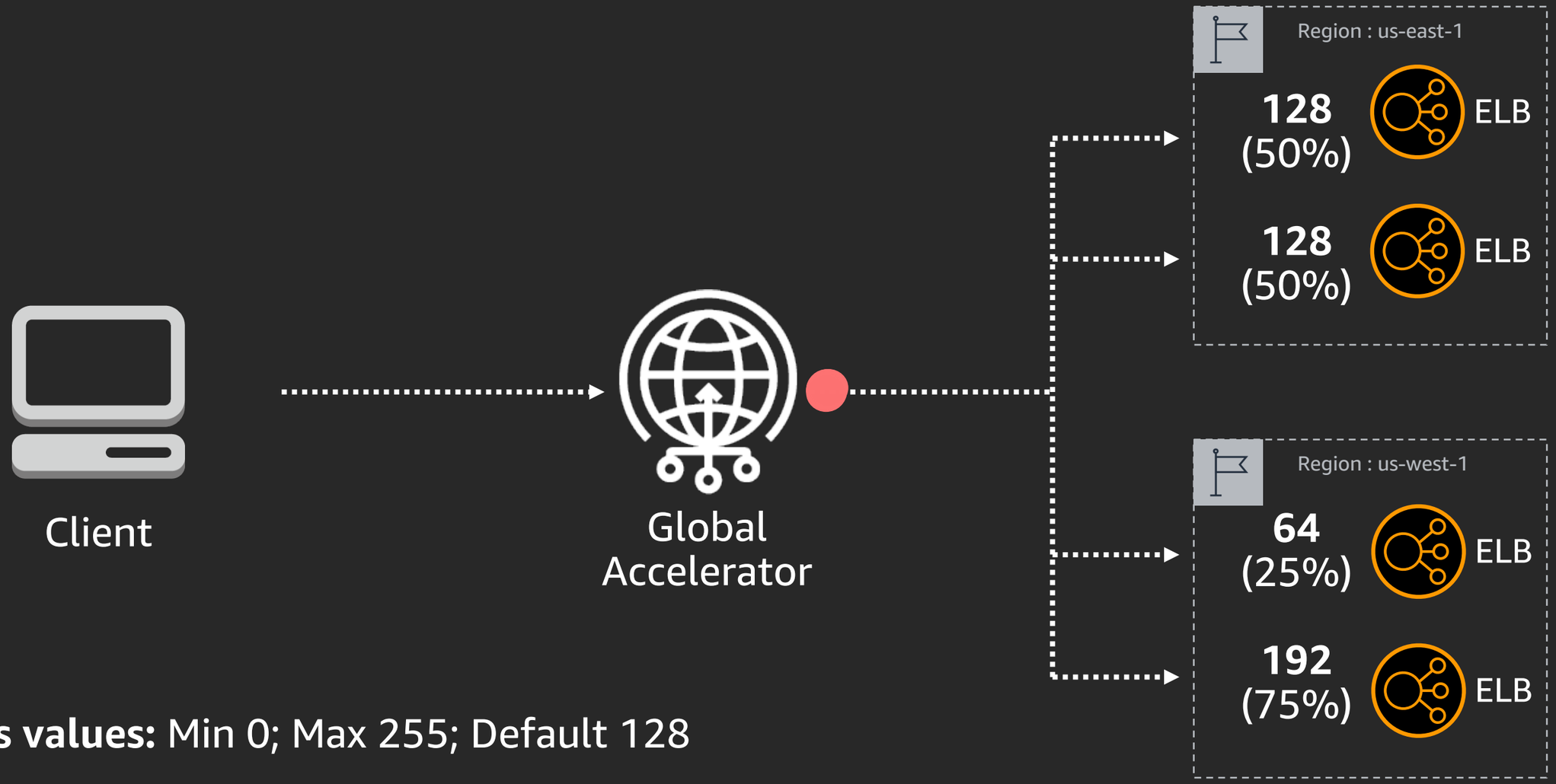
Dial values: Min 0%; Max 100%; Default 100%

Easy traffic control – Regional traffic dials



Dial values: Min 0%; Max 100%; Default 100%

Easy traffic control – Endpoint weights



Weights values: Min 0; Max 255; Default 128

Easy traffic control – Client affinity

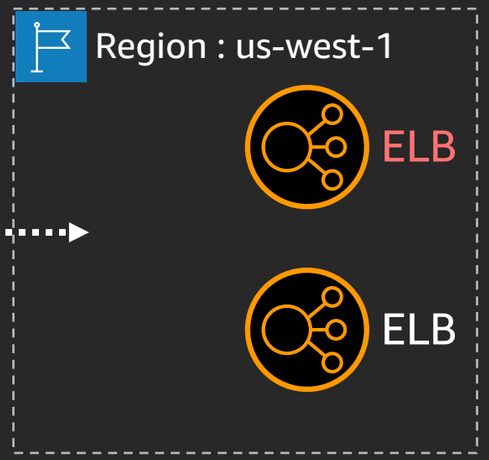
Client

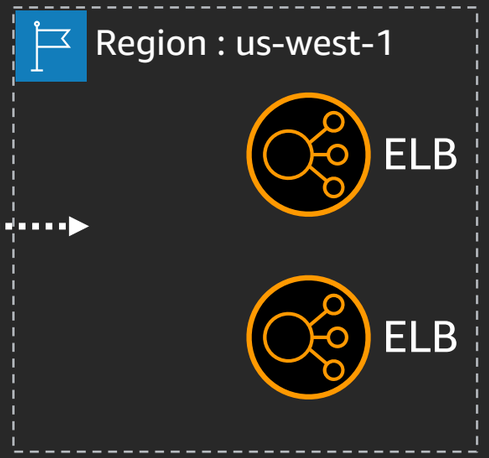
2 Tuple	
Protocol: *UDP/TCP*	
Src IP: *1.2.3.4*	Dest IP: *5.6.7.8*
~~Src Port: *32759*~~	~~Dest Port: *80*~~

Global Accelerator

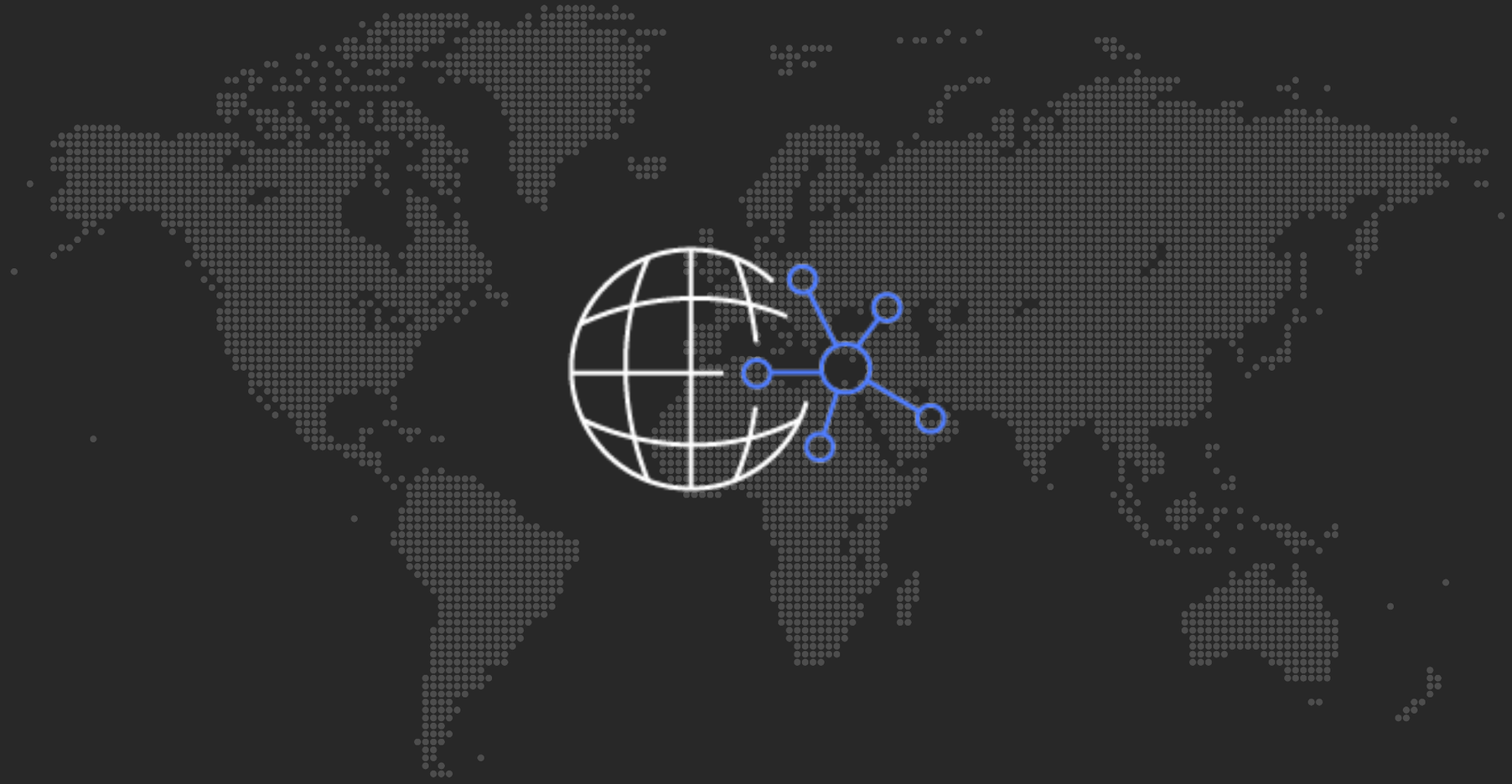
Optimal region

Hash (2 tuple)

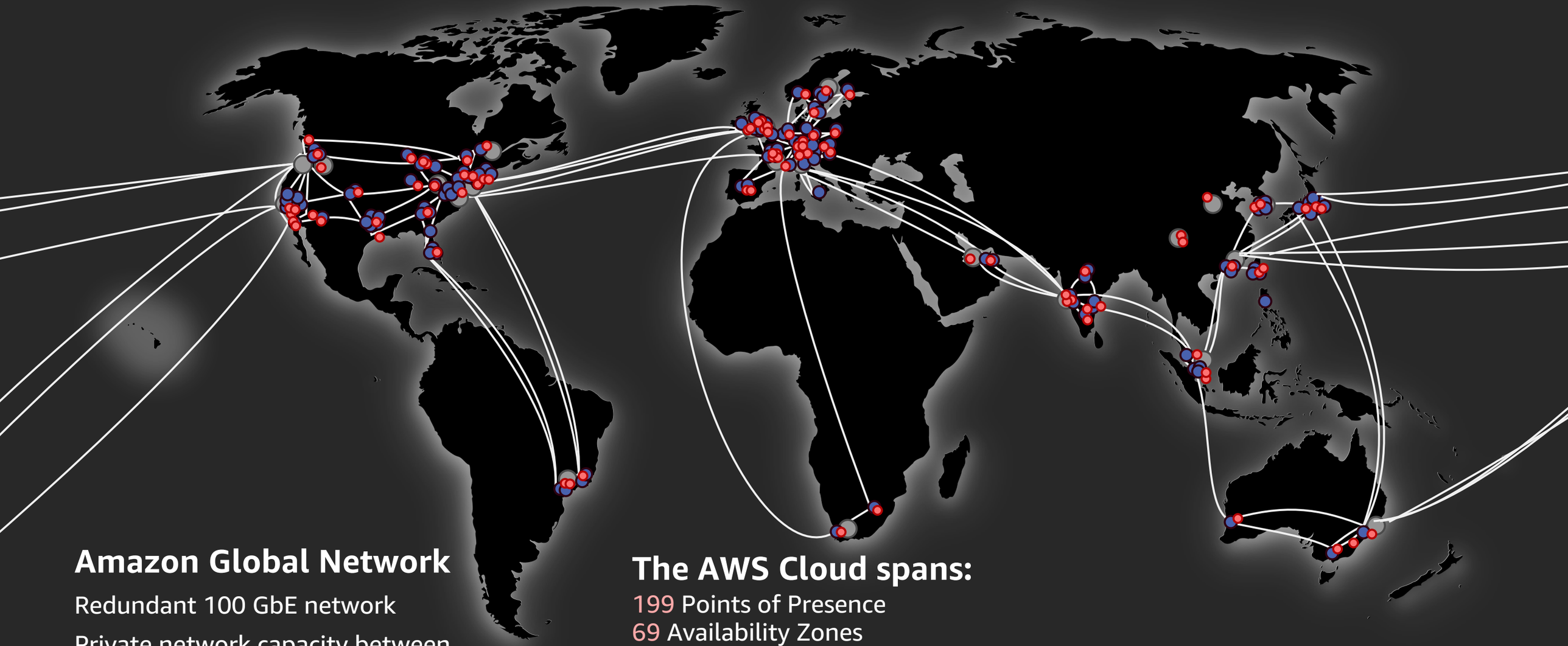




Propagation worldwide takes about 30 seconds



Security



Amazon Global Network

Redundant 100 GbE network

Private network capacity between all AWS Regions, except China

The AWS Cloud spans:

199 Points of Presence
69 Availability Zones
22 Geographic AWS Regions around the world*

*With announced plans for 13 more Availability Zones and four more Regions in Cape Town, Jakarta, Milan, and Spain.

AWS re:Invent

aws

Security

Data
encryption key

- Network level encryption

Security

Data
encryption key

Amazon VPC

- Network level encryption
- Security groups

Security

Data
encryption key

Amazon VPC

AWS WAF

- Network level encryption
- Security groups
- AWS WAF on ALB

Security

Data
encryption key

Amazon VPC

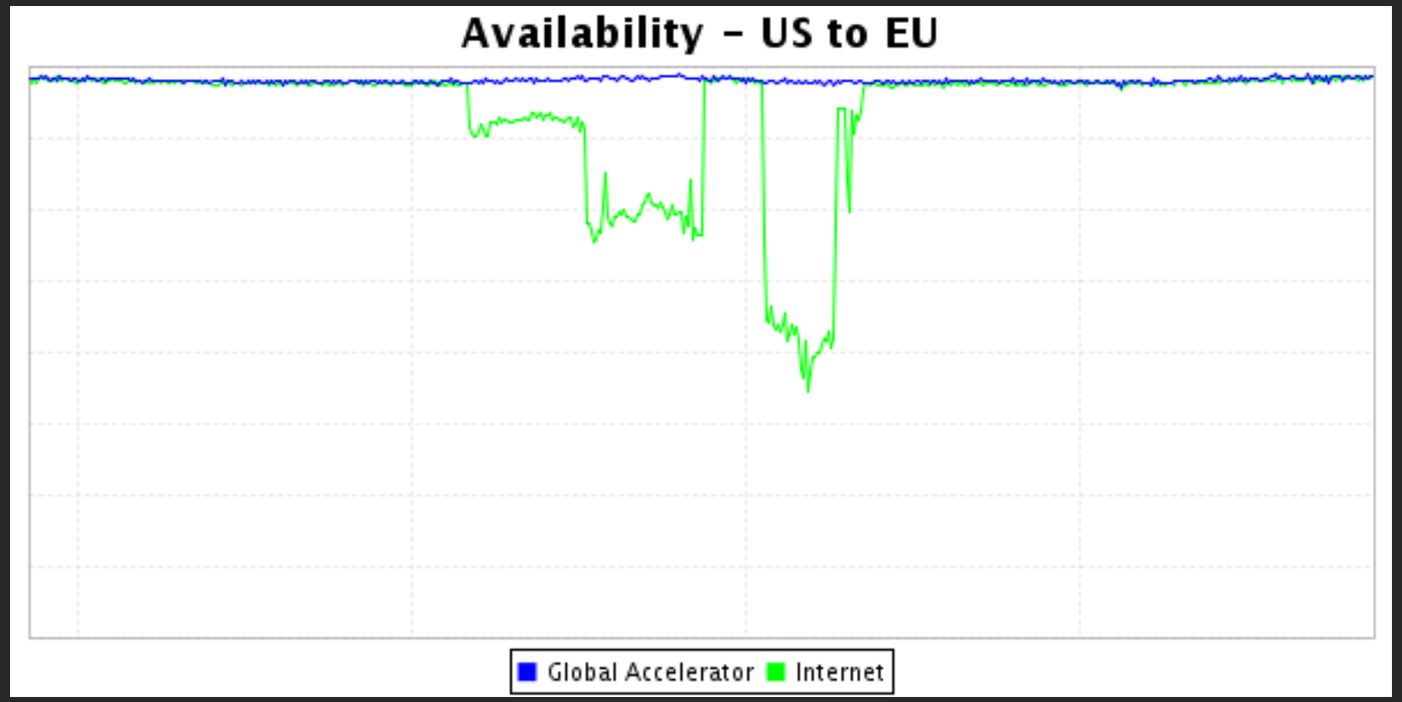
AWS WAF

Amazon EC2

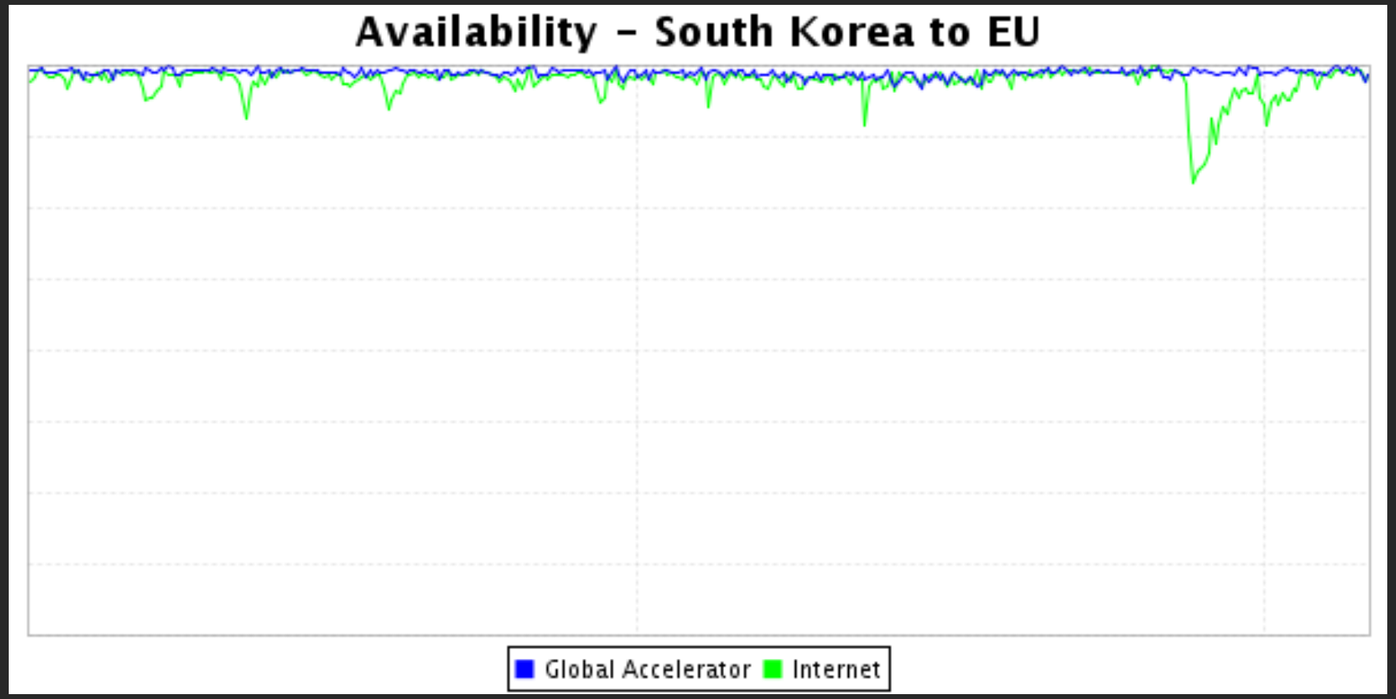
- Network level encryption
- Security groups
- AWS WAF on ALB
- Internal EC2 instances and ALBs

Availability

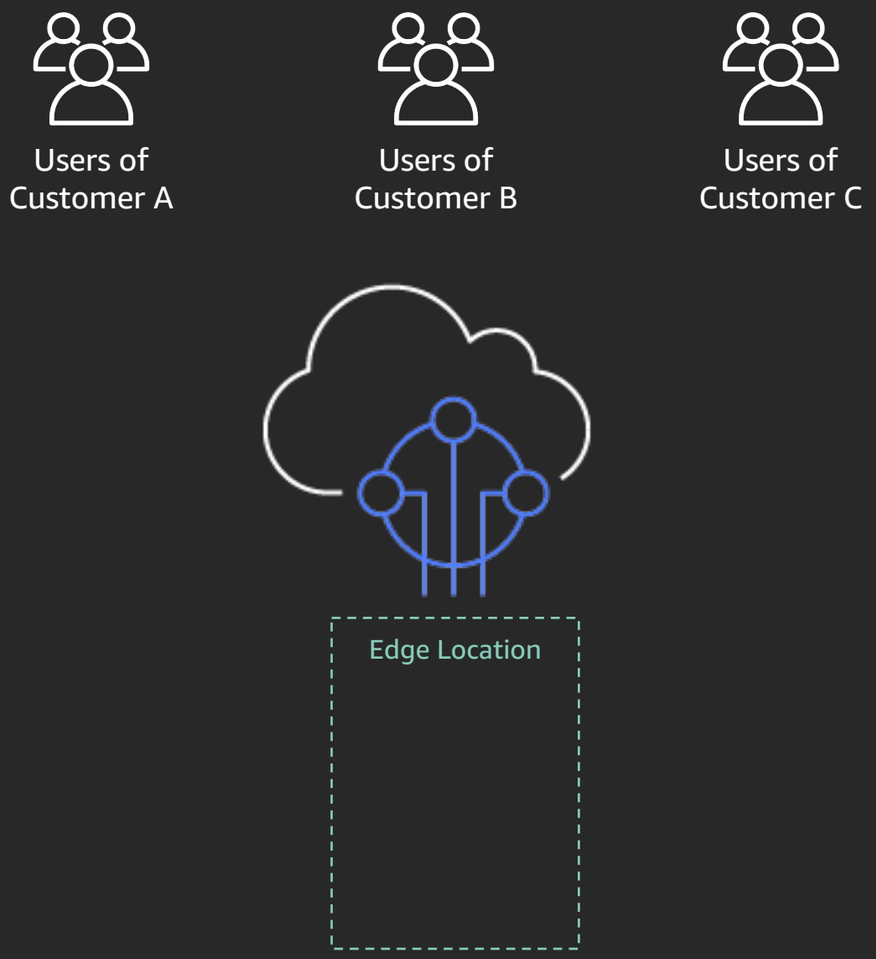
Availability



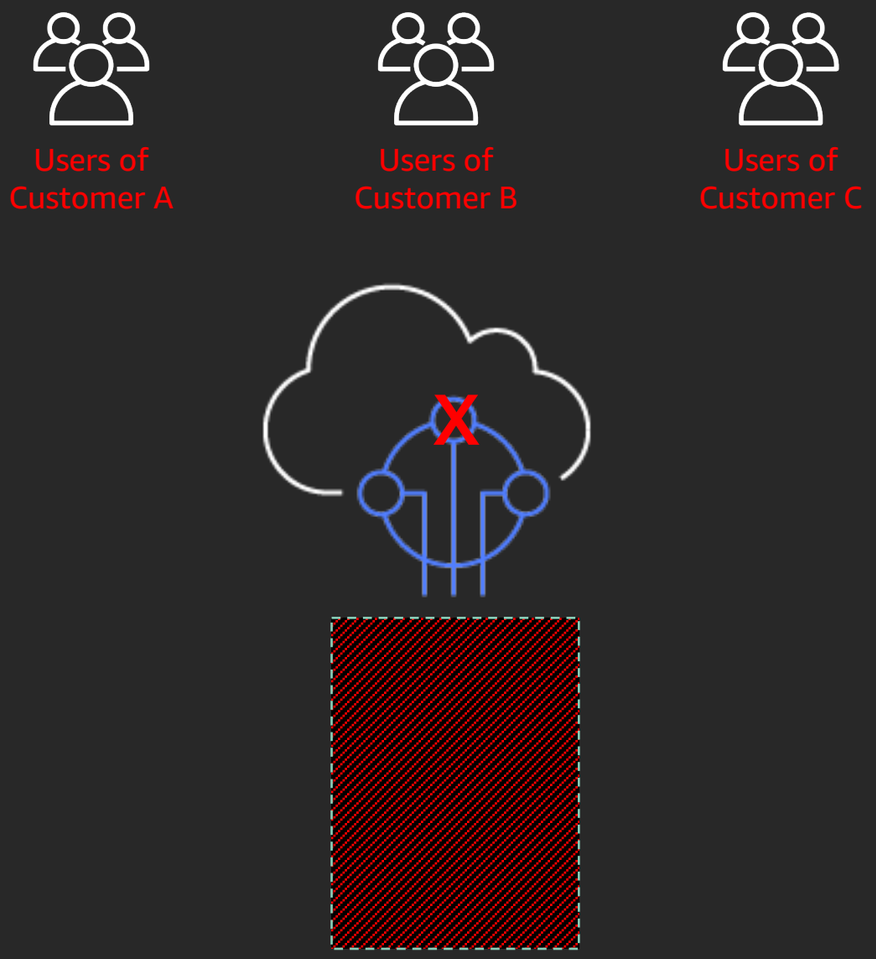
Availability



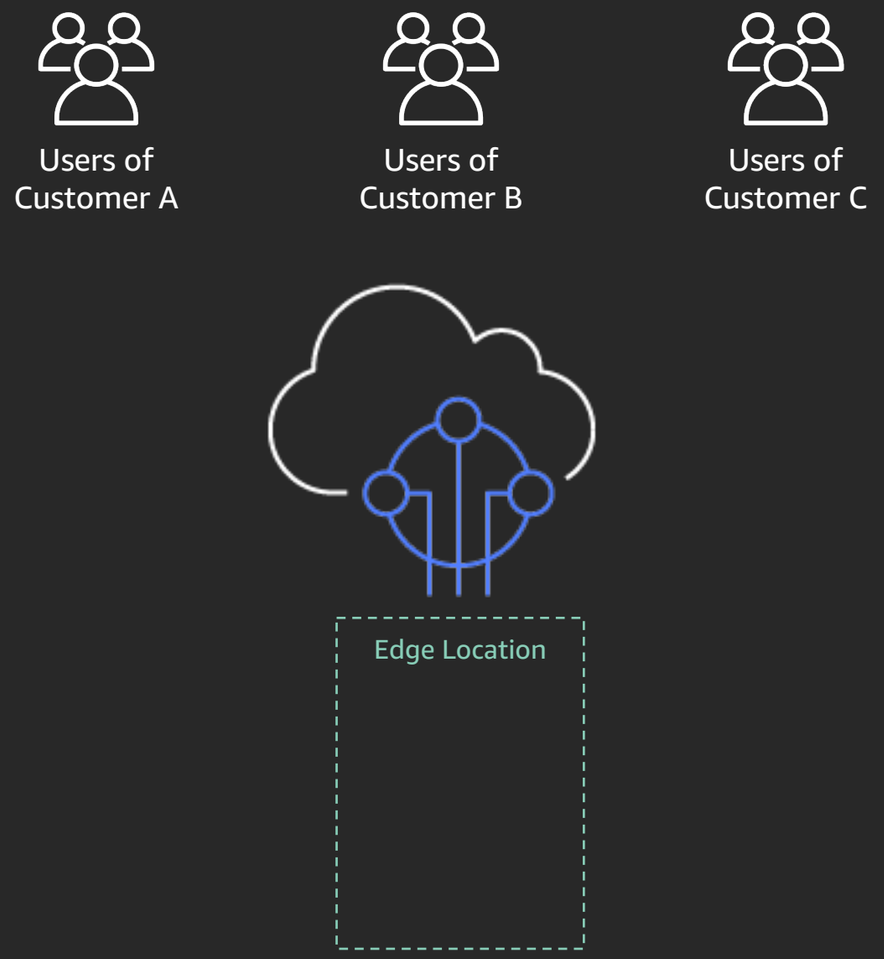
Edge location – BGP announcements



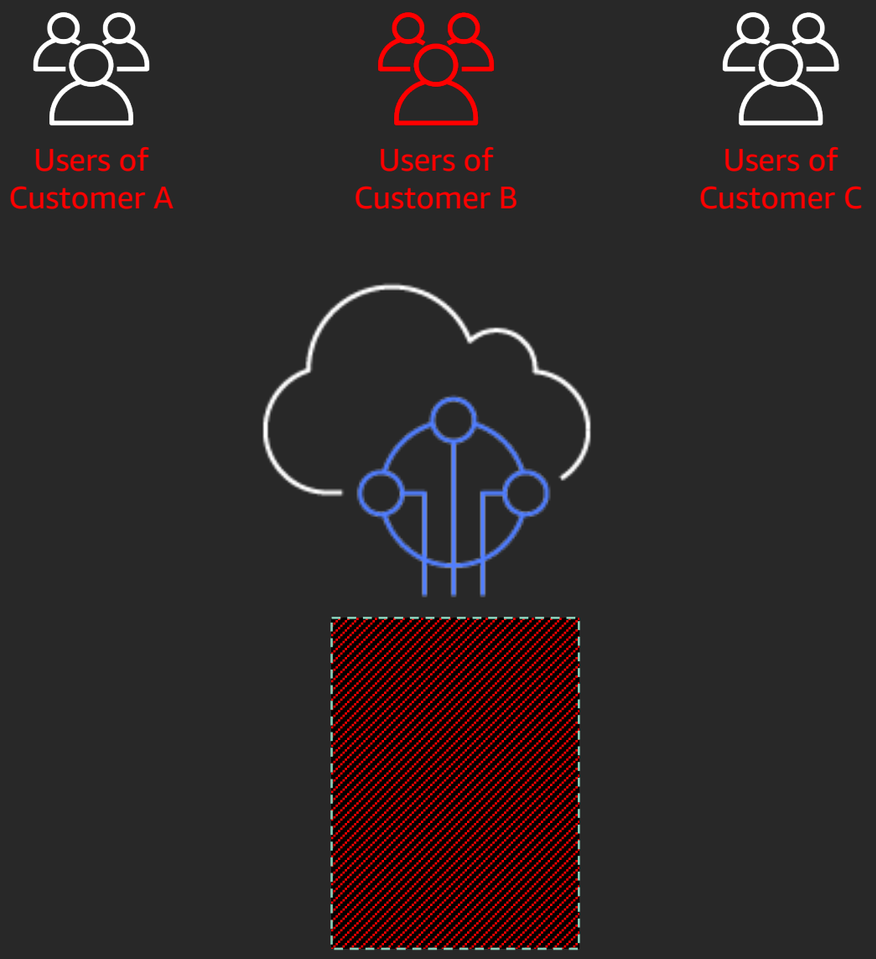
Network events



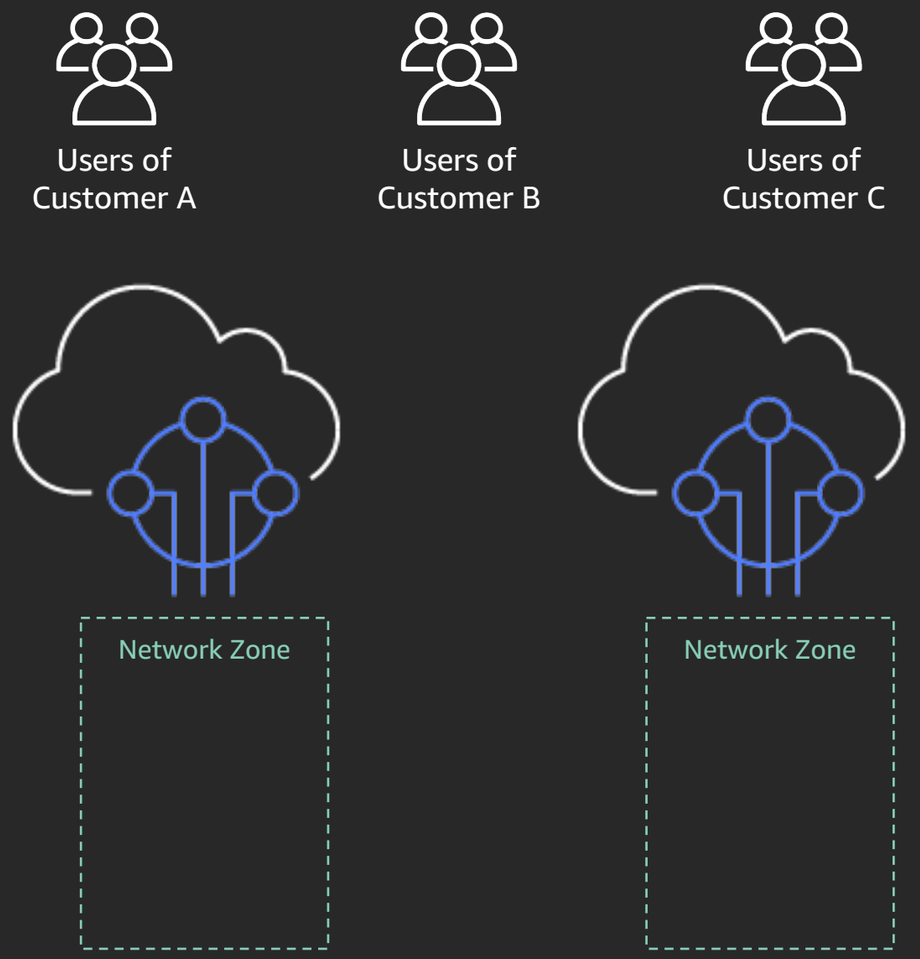
Heavy workloads



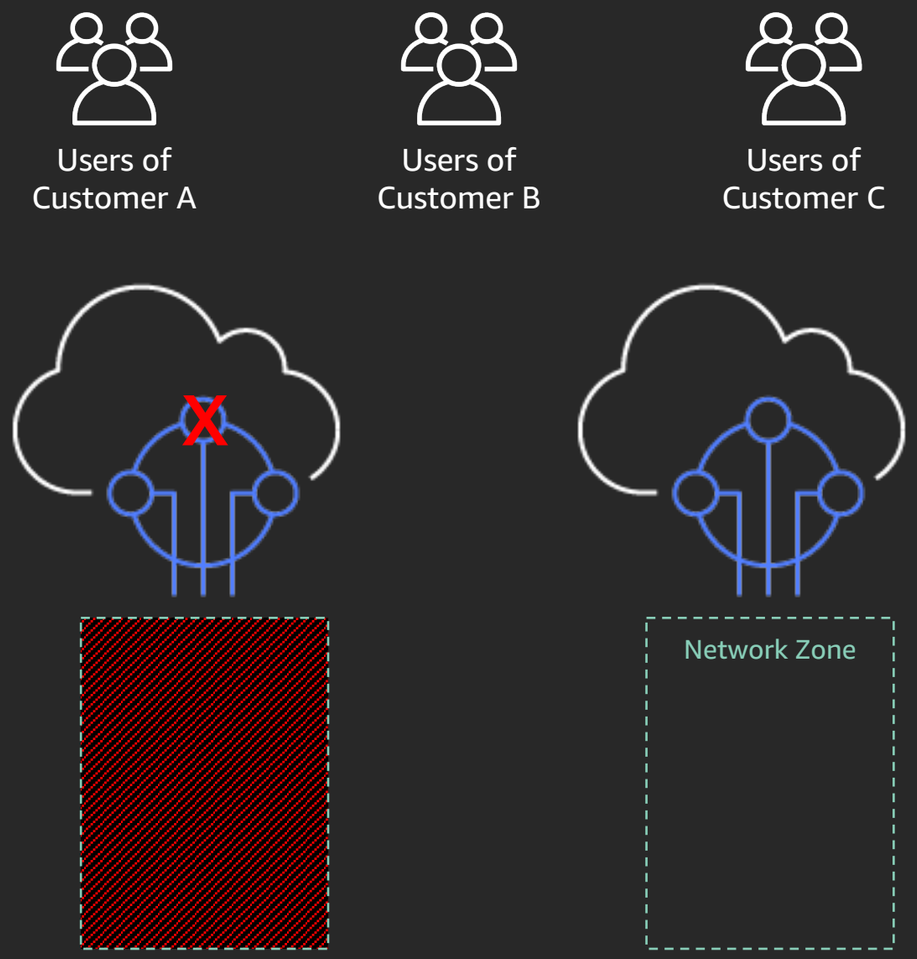
Heavy workloads – noisy neighbors



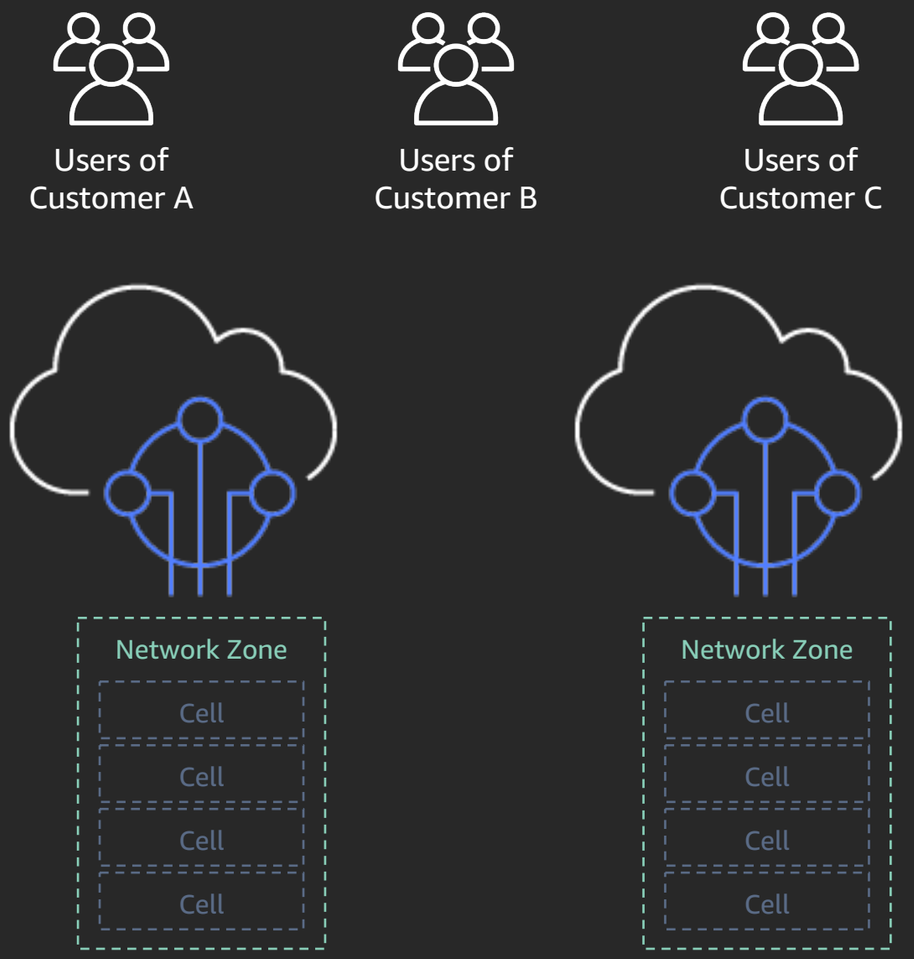
Network zones



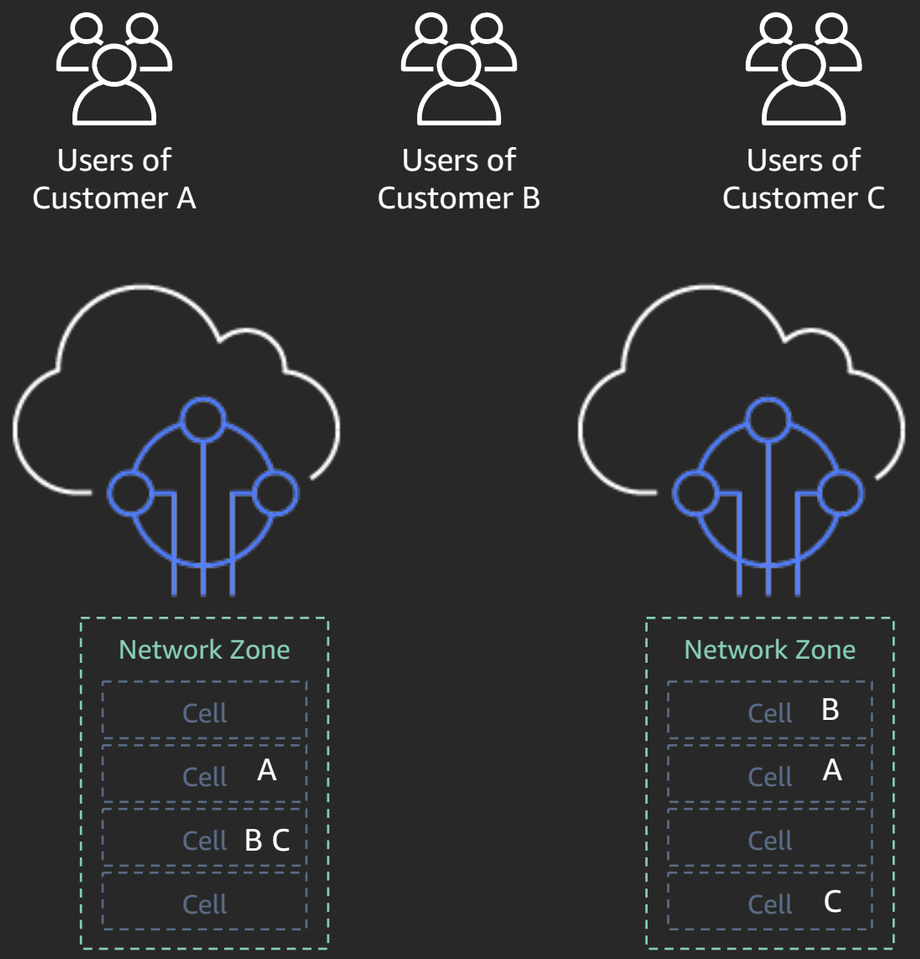
Network event affecting one zone



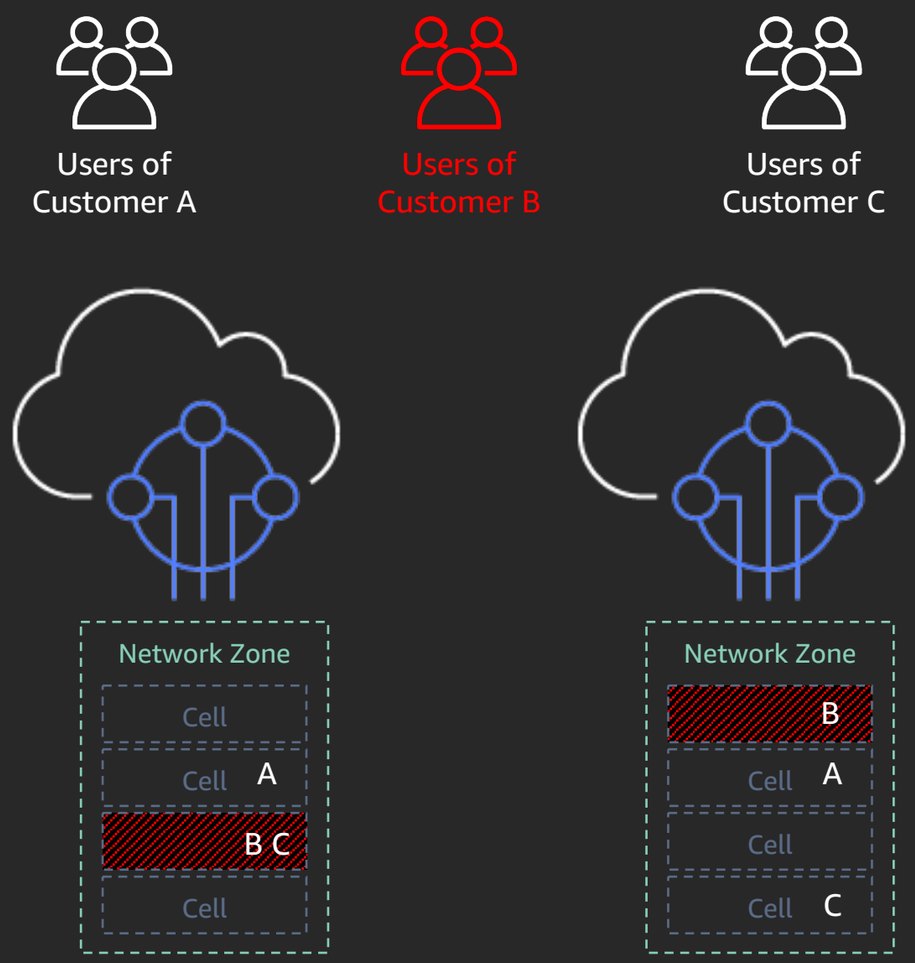
Cells per zone



Cells per zone



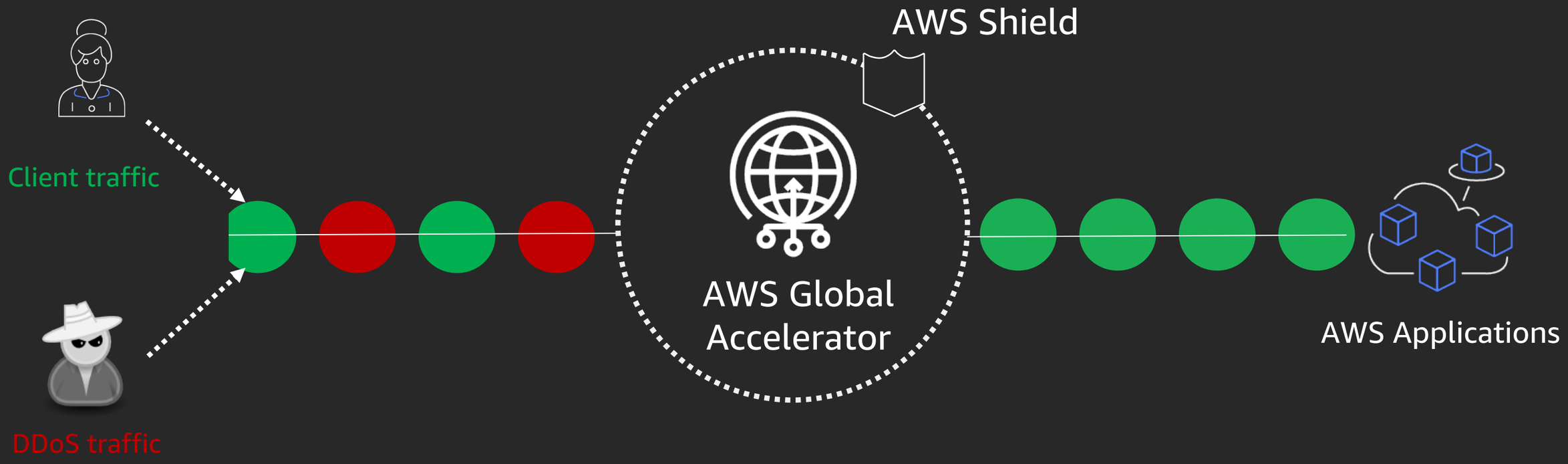
Noisy neighbor affecting one cell per zone



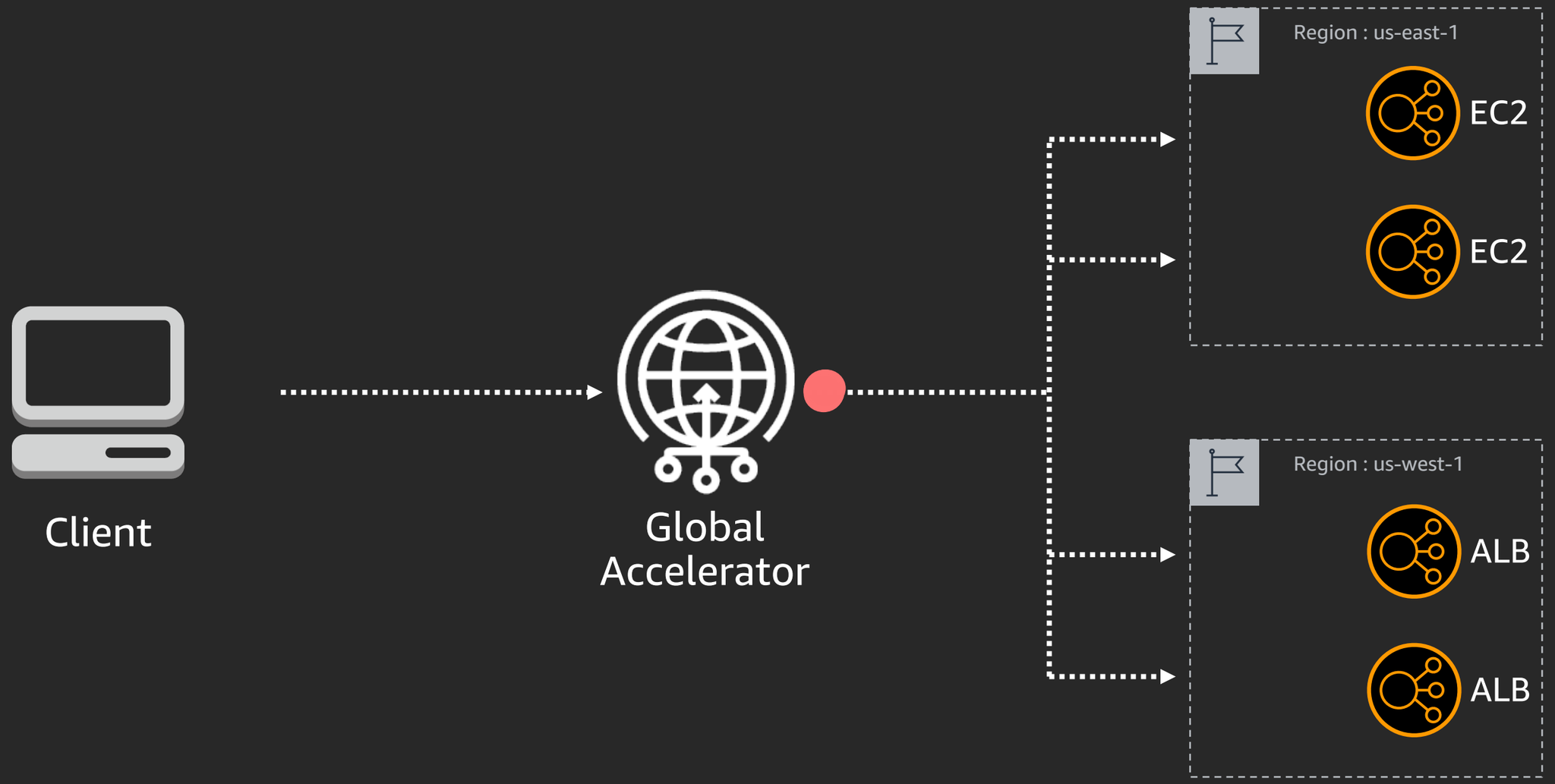
Blast radius reduction

	1 Network Zone	2 Network Zones	4 Network Zones
2 Cell Per NZ	$1/2^1 = 50\%$	$1/2^2 = 25\%$	$1/2^4 = 6.25\%$
4 Cells Per NZ	$1/4^1 = 25\%$	$1/4^2 = 6.25\%$	$1/4^4 = 0.4\%$
8 Cells Per NZ	$1/4^1 = 12.5\%$	$1/8^2 = 1.56\%$	$1/8^4 = 0.02\%$

AWS Shield will safeguard against L3/L4 DDoS attacks



Availability across failures



Network Address Translation

	Source	Destination
IP	1.2.3.4	5.6.7.8
TCP	32456	443
Payload	GET /fun.jpg HTTP/1.1	

Network Address Translation

	Source	Destination
IP	1.2.3.4	5.6.7.8
TCP	32456	443
Payload	GET /fun.jpg HTTP/1.1	

	Source	Destination
IP	9.8.7.6	10.2.3.4
TCP	45678	443
Payload	GET /fun.jpg HTTP/1.1	

Client IP preservation

	Source	Destination
IP	1.2.3.4	5.6.7.8
TCP	32456	443
Payload	GET /fun.jpg HTTP/1.1	

Client IP preservation

	Source	Destination
IP	1.2.3.4	5.6.7.8
TCP	32456	443
Payload	GET /fun.jpg HTTP/1.1	

Encapsulation

IP
Transport
Metadata
IP
TCP
Payload

Client IP preservation

	Source	Destination
IP	1.2.3.4	5.6.7.8
TCP	32456	443
Payload	GET /fun.jpg HTTP/1.1	

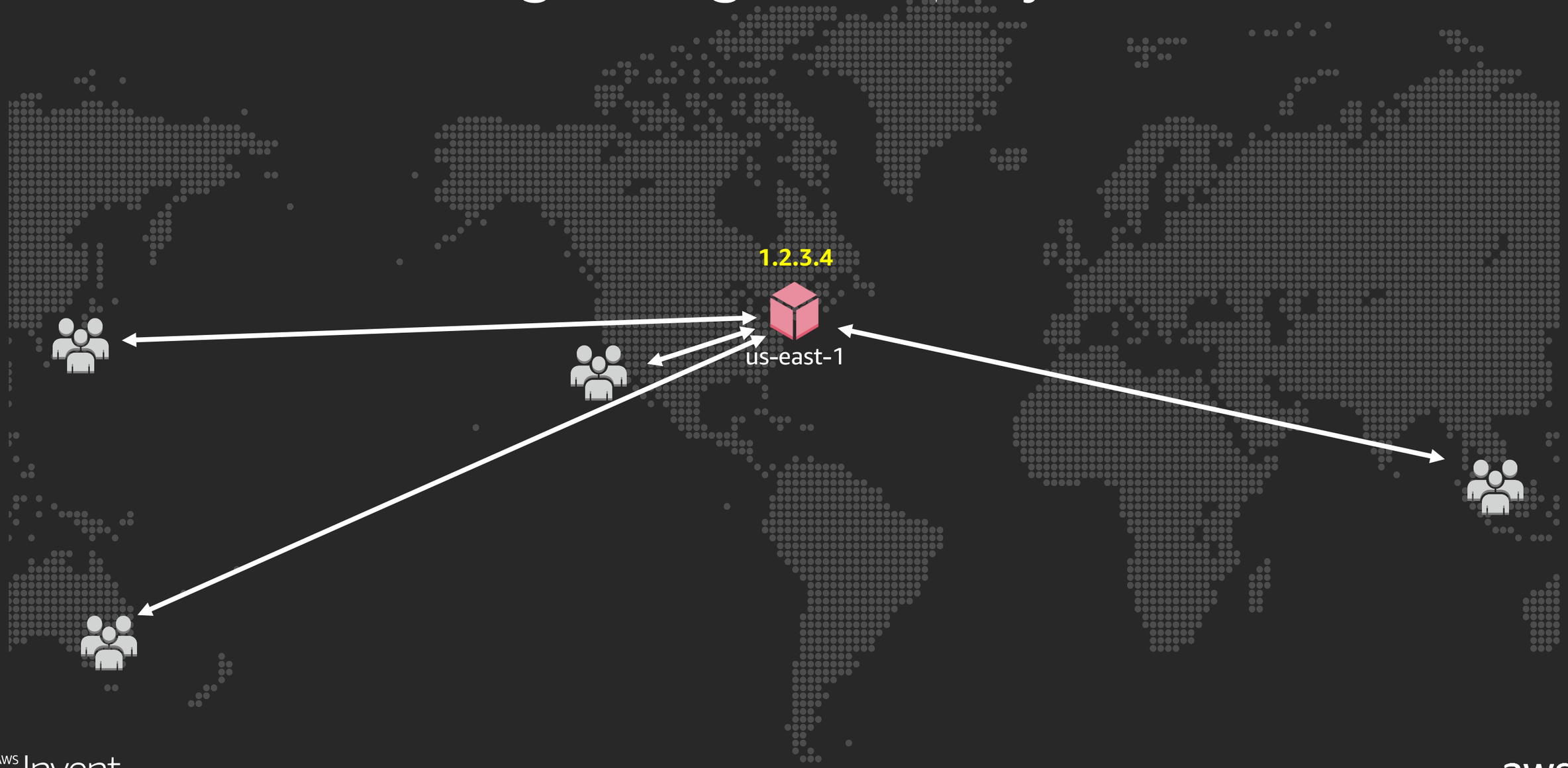
Encapsulation

IP
Transport
Metadata
IP
TCP
Payload

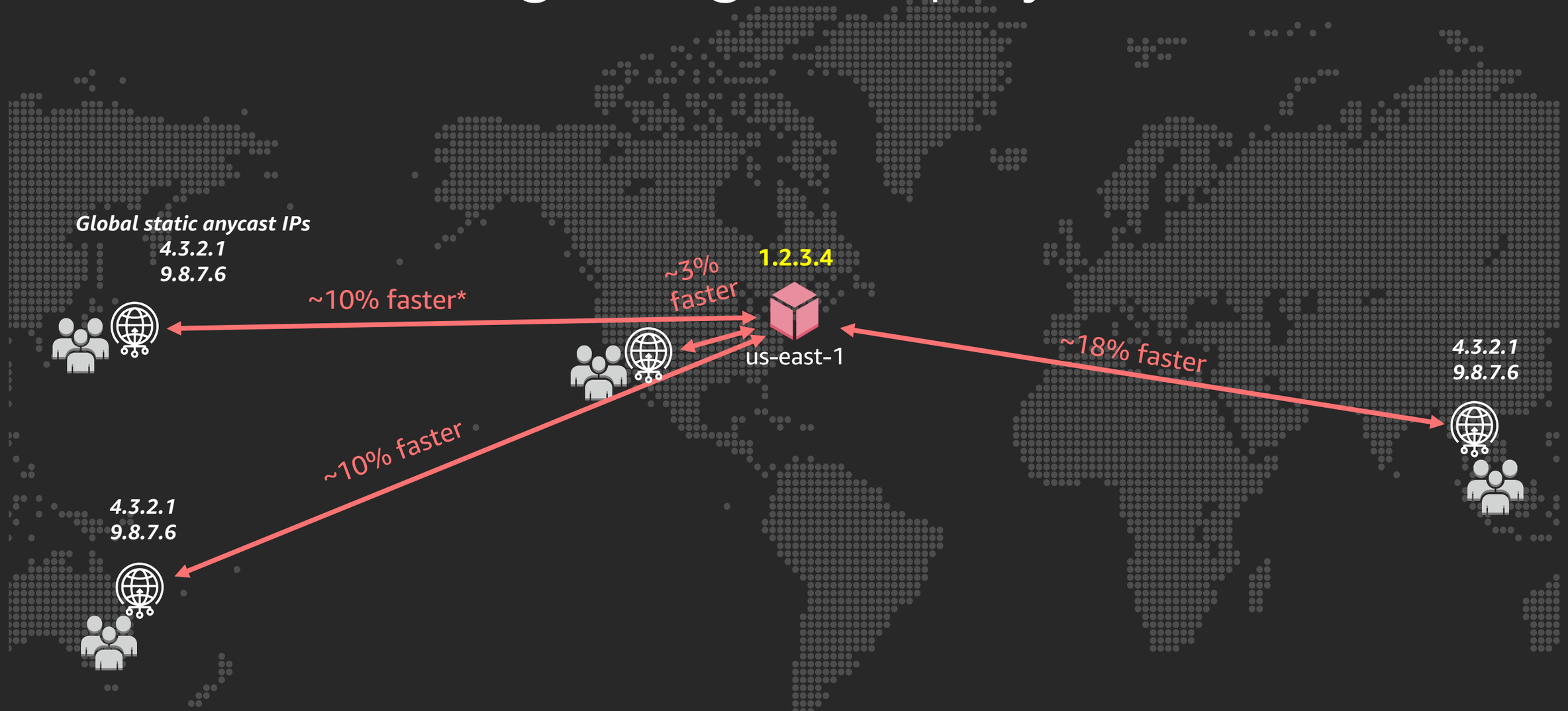
	Source	Destination
IP	1.2.3.4	10.2.3.4
TCP	32456	443
Payload	GET /fun.jpg HTTP/1.1	

Performance

Performance – Single-Region deployments

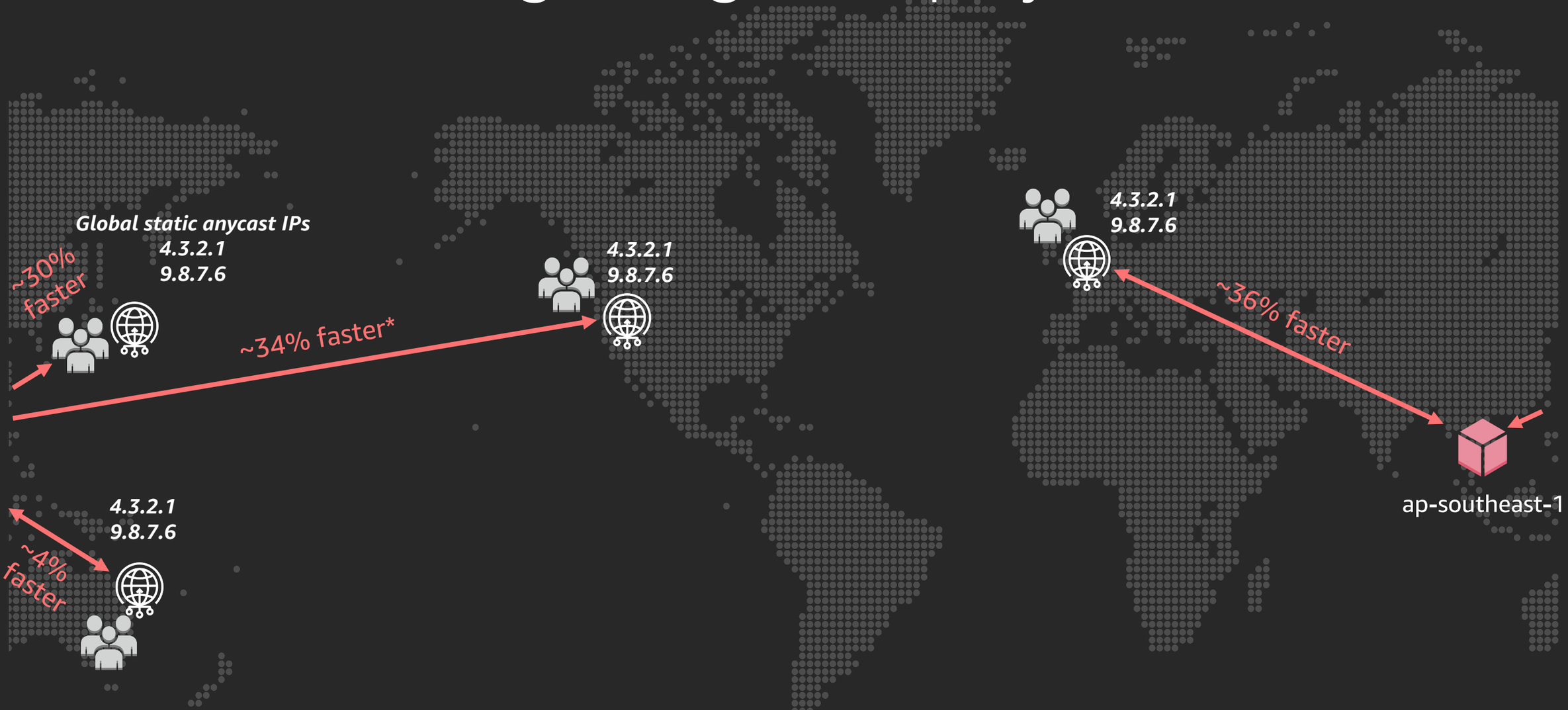


Performance – Single-Region deployments



*p90 First Byte Latency from 3rd party real-user measurements

Performance – Single-Region deployments

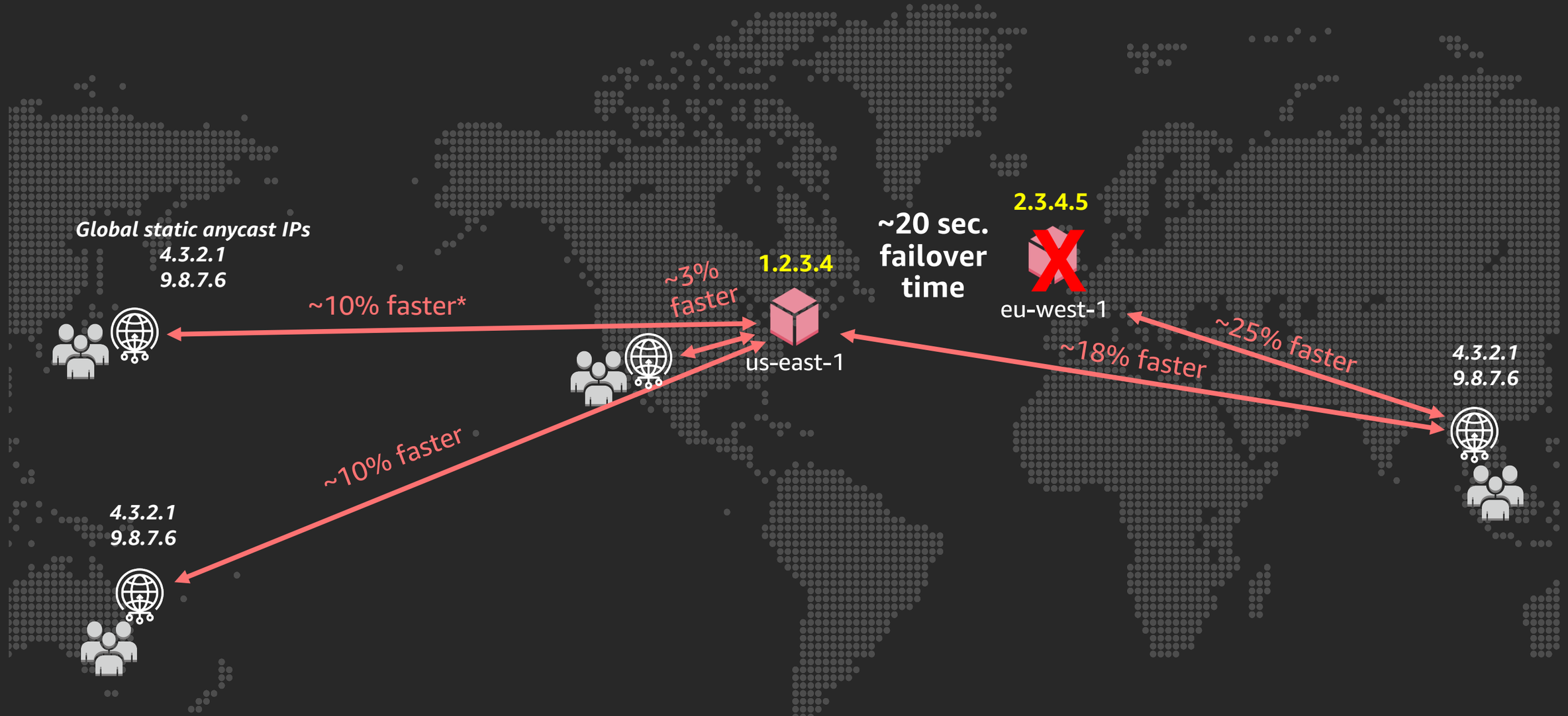


AWS re:Invent

*p90 First Byte Latency from 3rd party real-user measurements

aws

Performance – Plus failover



*p90 First Byte Latency from 3rd party real-user measurements

Building with Global Accelerator

AWS re:Invent

aws

Build an application

AWS Management Console

Developer Tools

CodeStar

CodeCommit

CodeBuild

CodeDeploy

CodePipeline

Cloud9

X-Ray

HTTP front end

AWS CodeCommit

Developer Tools > CodeCommit > Repositories > reysrv

reysrv

reysrv / app / webroot / todo / index.html Info

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>todo</title>
5     <meta name="viewport" content="width=device-width,
```

HTTP front end

AWS CodeCommit

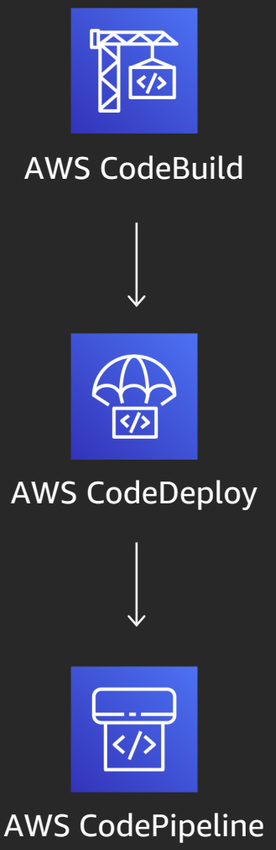
```
250   <table>
251       <tr>
252           <td>Categories:</td>
253           <td><span ng-repeat="viewName in viewNames | orderBy">
254       </tr>
255       <tr>
256           <td align="right">Upcoming:</td>
257           <td>
258               <a href="" ng-click="viewToday()">Today</a>
259               <a href="" ng-click="viewTomorrow()">Tomorrow</a>
260               <a href="" ng-click="viewBacklog()">Backlog</a>
261           </td>
262       </tr>
263   </table>
```

HTTP backend

AWS CodeCommit

```
690     newVersion := *task.Version + 1
691     params := &dynamodb.UpdateItemInput{
692         TableName: aws.String("Task"),
693         Key: map[string]*dynamodb.AttributeValue{
694             "ViewID": {S: task.View},
695             "TaskID": {S: task.TaskID},
696         },
697         ConditionExpression: aws.String("Version = :version OR attribute_not_exists
698         UpdateExpression:    aws.String("SET Description = :desc, DoDate = :doDate,
699         ExpressionAttributeValues: map[string]*dynamodb.AttributeValue{
700             ":desc":       {S: task.Desc},
701             ":doDate":     {S: task.DoDate},
702             ":version":    {N: aws.String(strconv.Itoa(*task.Version))},
703             ":newVersion": {N: aws.String(strconv.Itoa(newVersion))},
704         },
705     }
```

Deploy worldwide



Production

PDX
AWS CodeDeploy
Succeeded - 41 minutes ago
Details

ICN
AWS CodeDeploy
Succeeded - 41 minutes ago
Details

DUB
AWS CodeDeploy
Succeeded - 41 minutes ago
Details

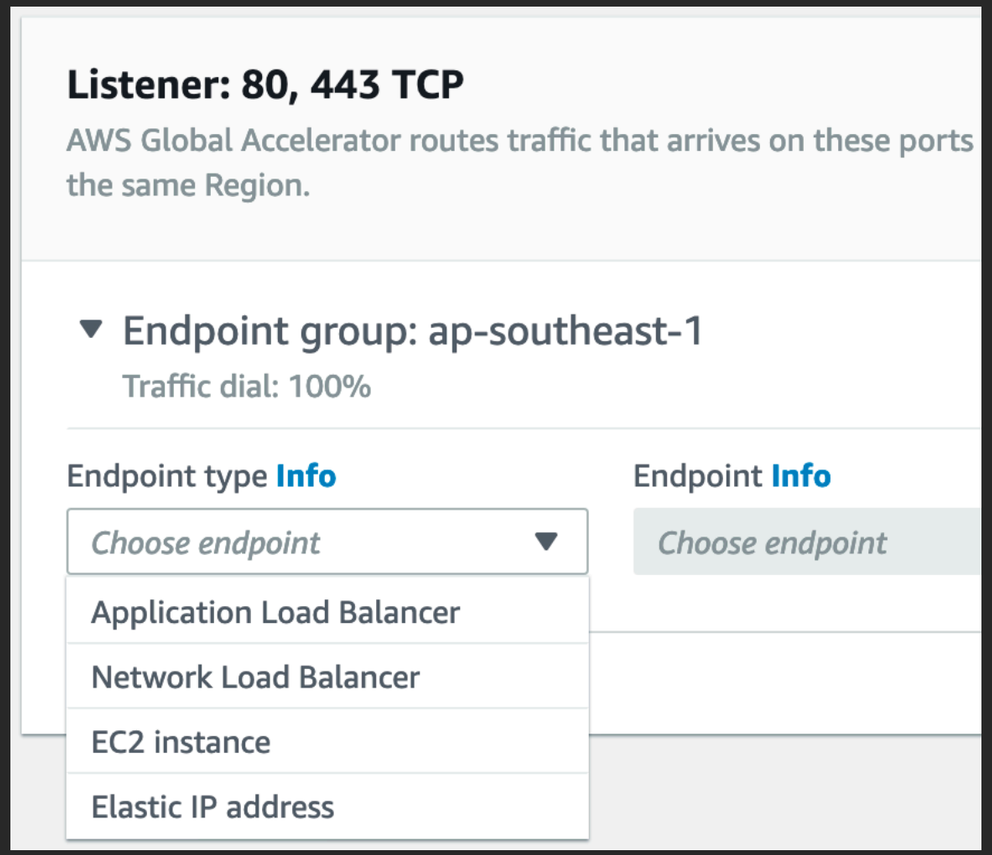
SIN
AWS CodeDeploy
Succeeded - 40 minutes ago
Details

FRA
AWS CodeDeploy
Succeeded - 40 minutes ago
Details

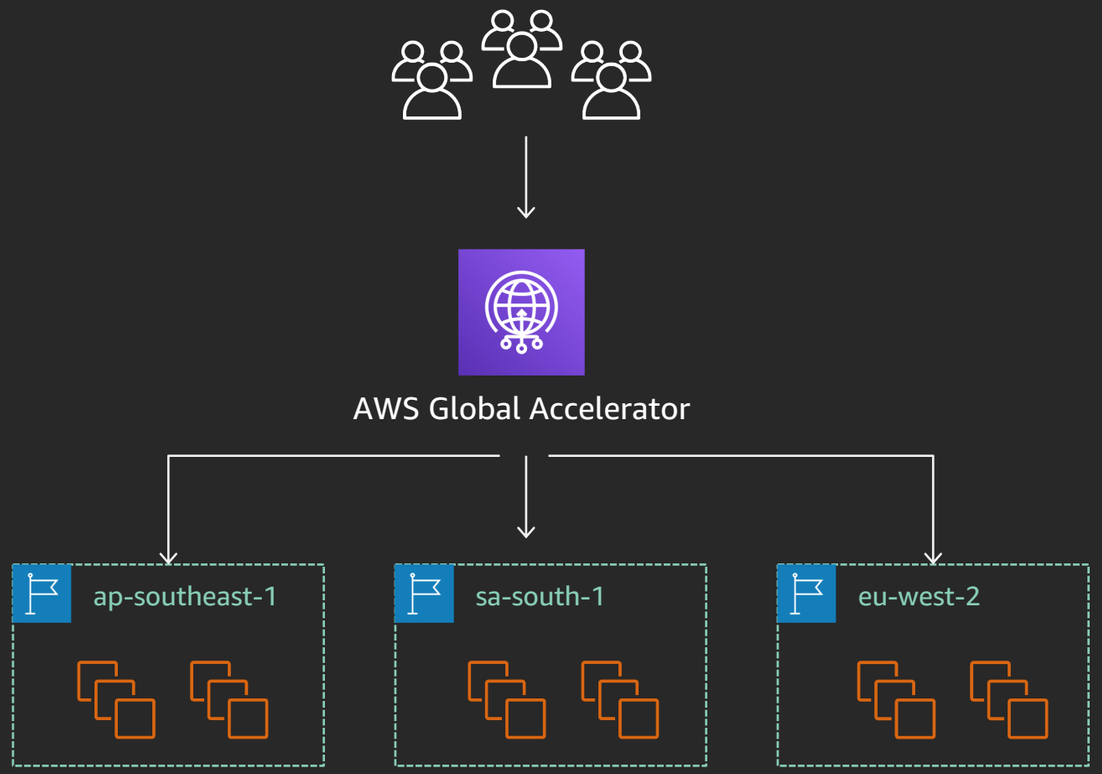
IAD
AWS CodeDeploy
Succeeded - 40 minutes ago
Details

Set up your accelerator

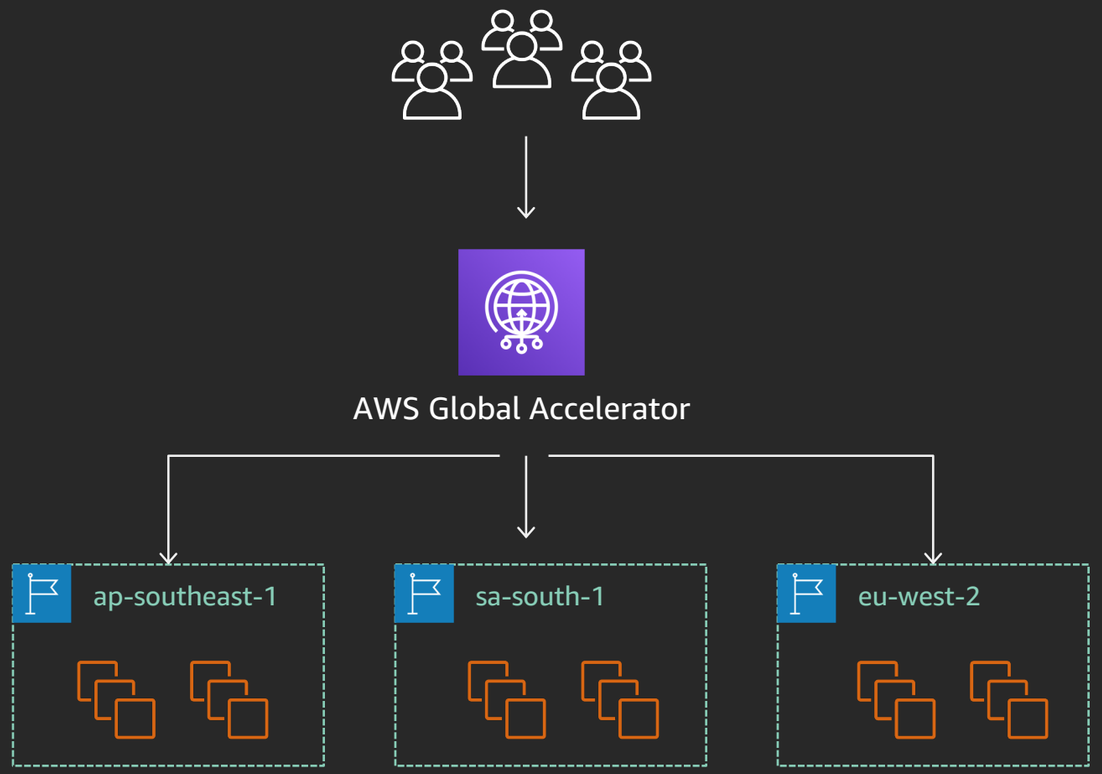
AWS Global Accelerator



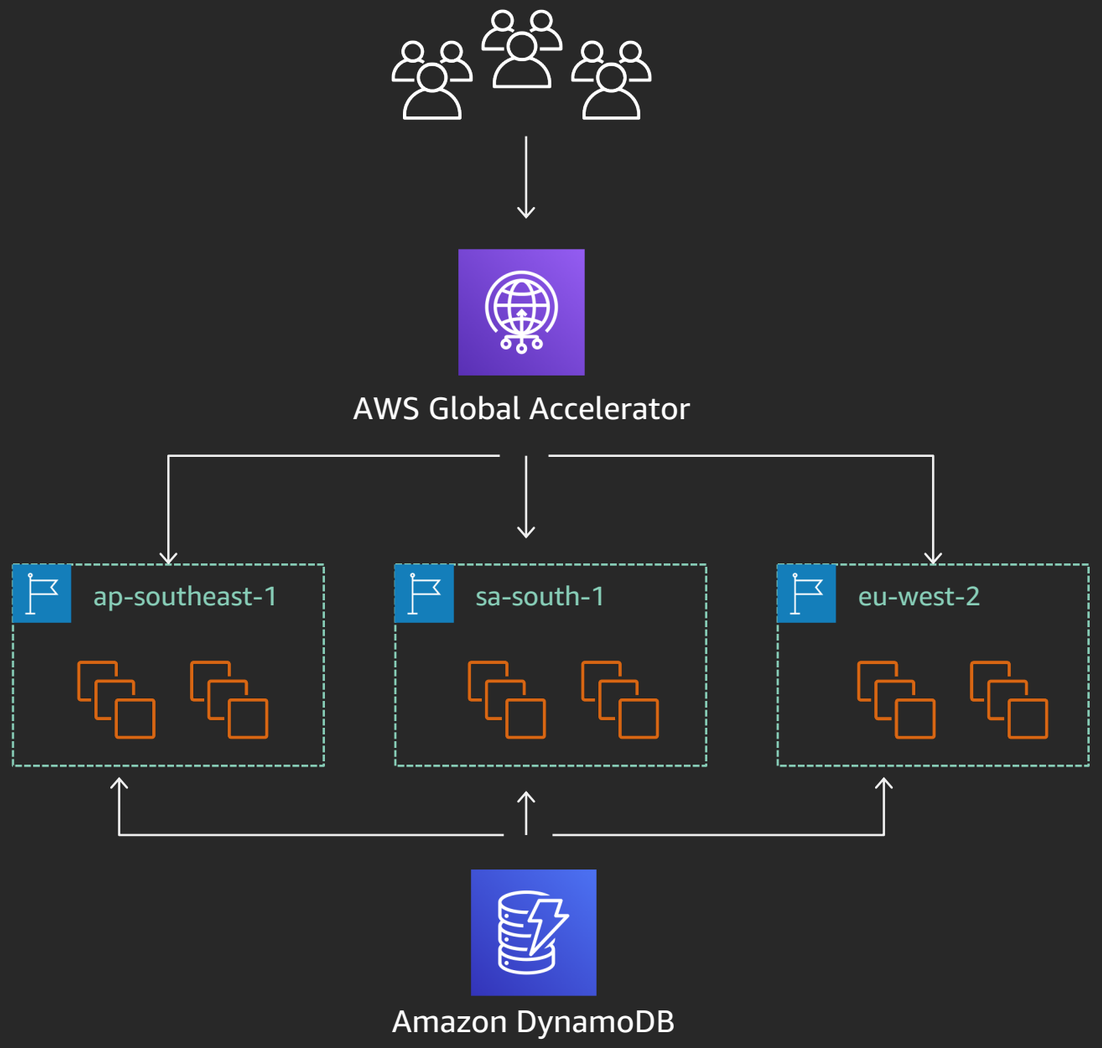
Multi-Region application



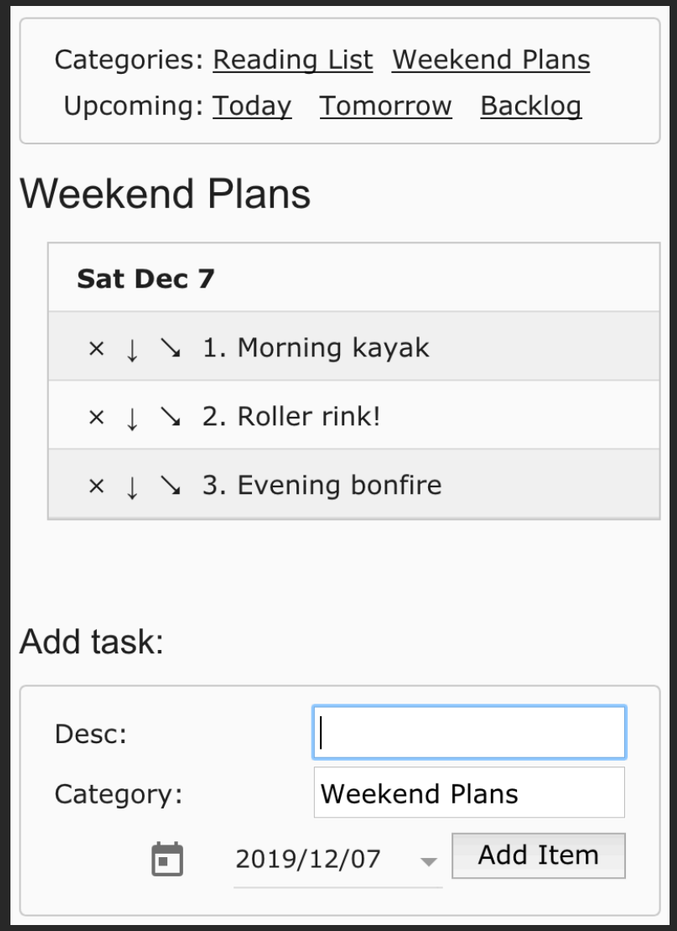
One thing missing…



Distributed state



Multi-Region application



When to use Global Accelerator

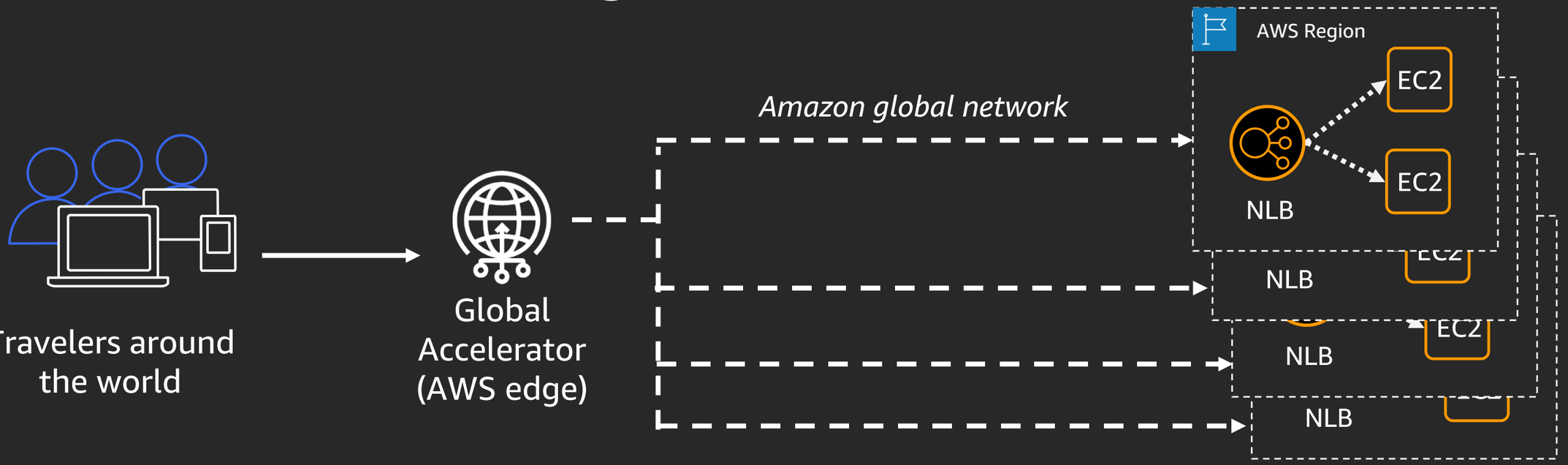
Non-HTTP applications

- Gaming (UDP)
- Real-time video (RTP)
- Voice over IP
- VPN/private connectivity
- DNS hosting (UDP)
- IoT (MQTT)
- File uploads (FTP)
- Push notifications (WebSocket)

HTTP applications

- Blue/green deployment and A+B testing
- Static IPs for IP whitelisting
- Failover resilience for multi-Region applications

Use case: DNS management



"Skyscanner is a leading global travel search site, a place where people are inspired to plan and book direct from millions of travel options at the best prices. AWS Global Accelerator enables us to effortlessly build resilient, multi-region services, allowing us to focus on providing our travelers with the best experience possible."

— Stewart Wallace, Senior Engineer, Skyscanner

Skyscanner

AWS re:Invent

aws

Use case: IP whitelisting + API acceleration

"All our points of sale use AWS Global Accelerator to reach our applications running on AWS with high availability and low latency."

—Leonard Redles, System Architect, ParTech, Inc.

Use case: Live video ingest

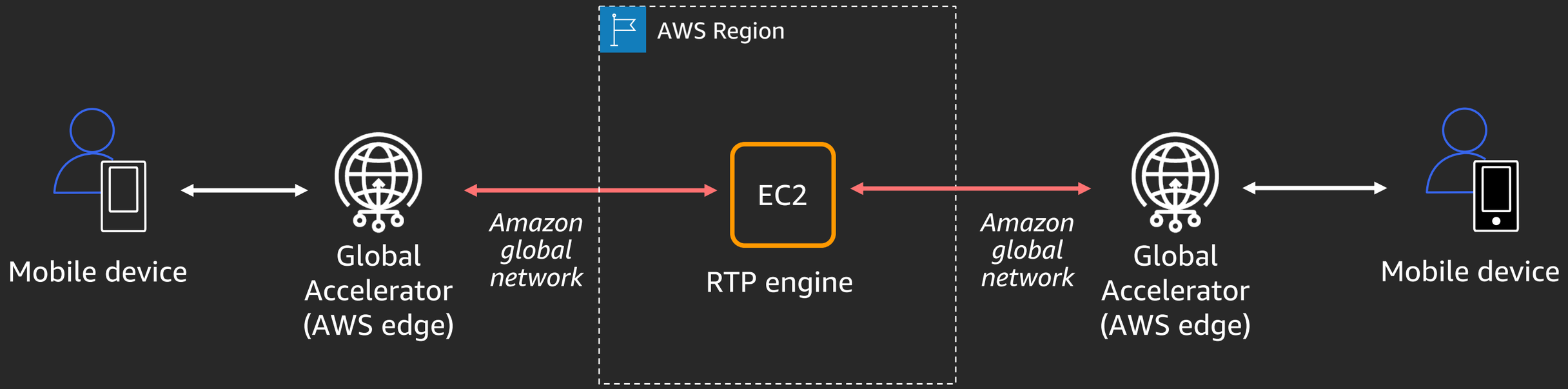
"Our video platform is hosted on AWS and uses Global Accelerator to improve the performance and availability of video ingest for our users all around the world."

flowplayer

—Erik Viklund, VP Dev Ops, Flowplayer

Blog post: https://aws.amazon.com/blogs/networking-and-content-delivery/how-flowplayer-improved-live-video-ingest-with-aws-global-accelerator/

Use case: Voice over IP

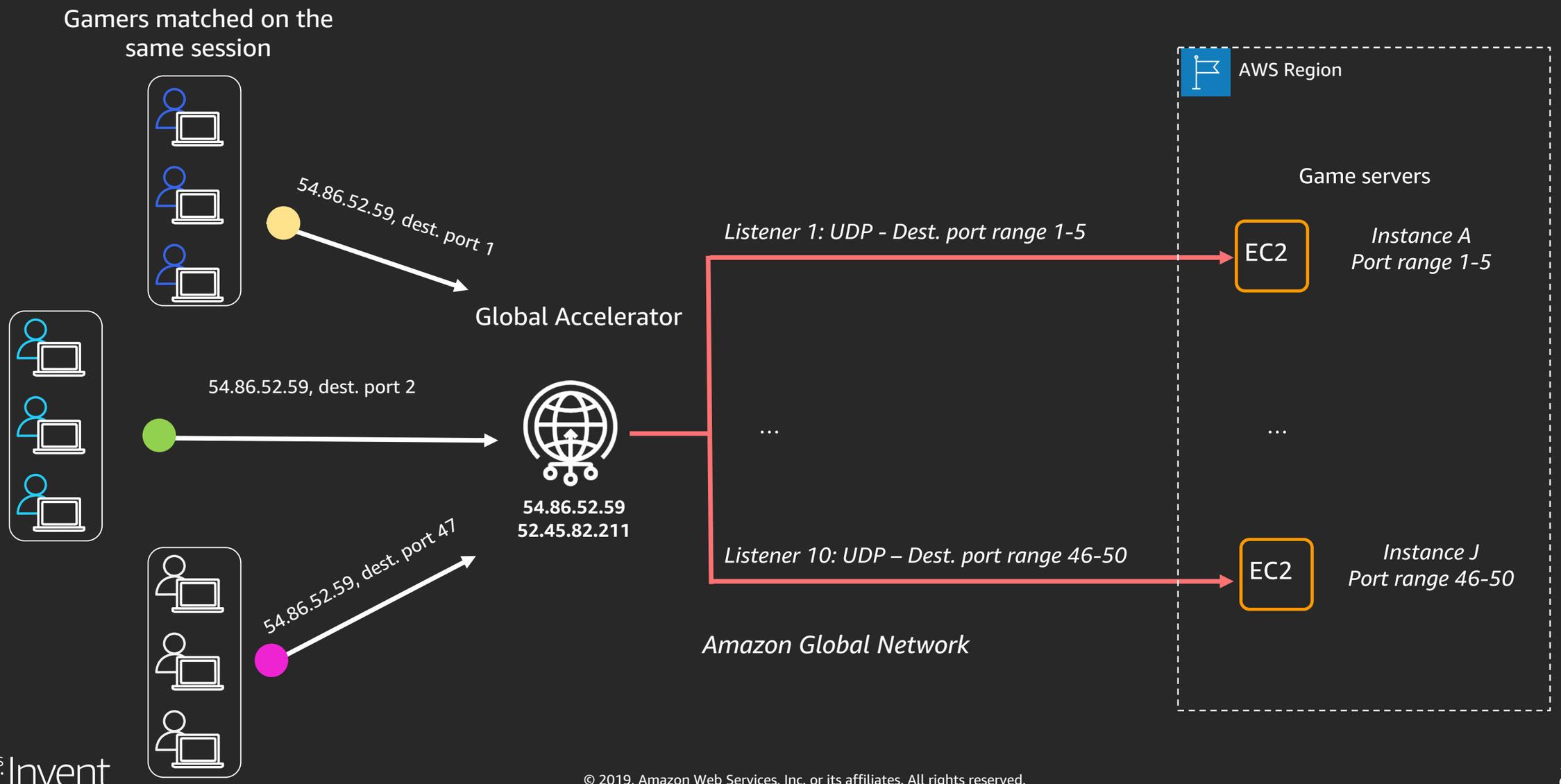


"We use AWS Global Accelerator's static anycast IPs and global network to handle voice over IP (VoIP) traffic, thus ensuring that our customers get the best quality of service."

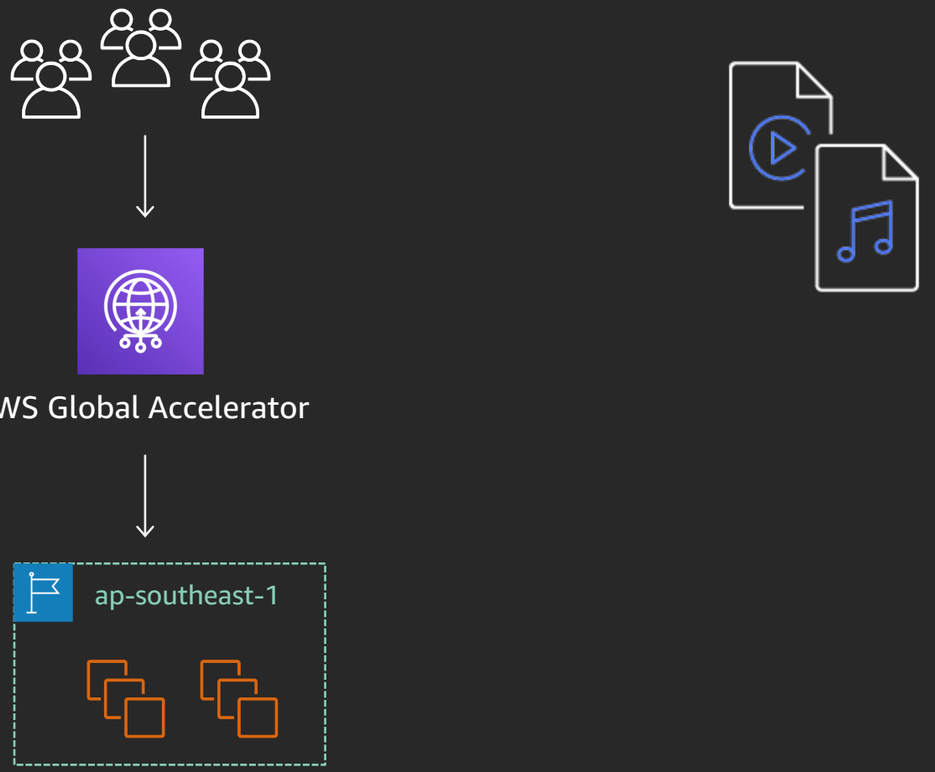
CrazyCall

—Marcin Kowalczyk, AWS Architect, CrazyCall

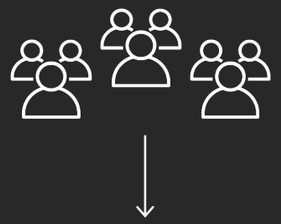
Use case: Multiplayer real-time games with Global Accelerator



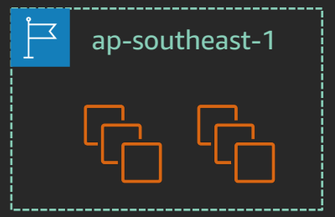
UDP galore

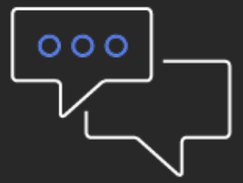


UDP galore

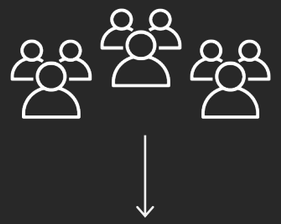


AWS Global Accelerator

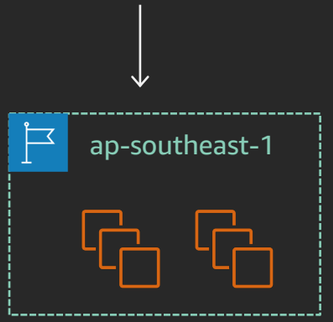


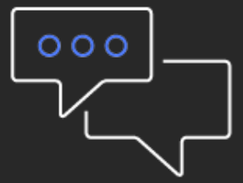


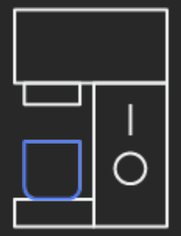
UDP galore



AWS Global Accelerator







Key features of AWS Global Accelerator

Single entry point with global static anycast IPs

Intelligent distribution of TCP & UDP traffic

Target EC2 instances and Elastic Load Balancers (ALBs & NLBs) Preserve client IPs

Accelerate both single and multi-Region workloads

All traffic traverses the backbone* and is protected from DDoS attacks

* except within the People's Republic of China

Thank you!

James Wenzel
Sr Partner SA, Networking
Amazon Web Services, Inc.

Harvo Jones
Principal SDE
Amazon Web Services, Inc.

AWS re:Invent

aws

Please complete the session survey in the mobile app.