Live Stream – 9 AM GMT (Starting Soon)

BeSA

Become a Solutions Architect

Week 8

01-Apr-2023

BeSA

Become a Solutions Architect

Week 8 - Agenda

1. Technical Track (45 Min)
2. Networking Track (40 Min)
3. Hands On (30 Min)

BeSA
Become a Solutions Architect

What is AWS Auto Scaling?

AWS Auto Scaling

AWS Auto Scaling **monitors** your applications and **automatically adjusts** capacity to maintain steady, predictable performance at the lowest possible cost. Using AWS Auto Scaling, it's easy to setup application scaling for multiple resources across multiple services in minutes.

Amazon EC2 instances

Amazon ECS Tasks

Amazon DynamoDB tables and indexes

Amazon Aurora Replicas

Amazon EC2 spot fleet

What is Amazon EC2 Auto Scaling?

Amazon EC2 Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application.

Auto Scaling components

Groups

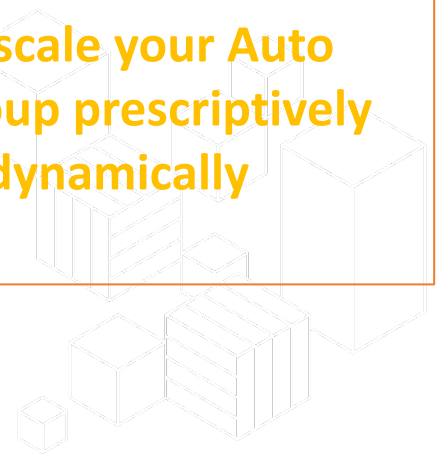
Logical managed collections of EC2 instances that can scale to meet demand

Configuration templates

Templates used to launch EC2 instances during scaling events

Scaling Options

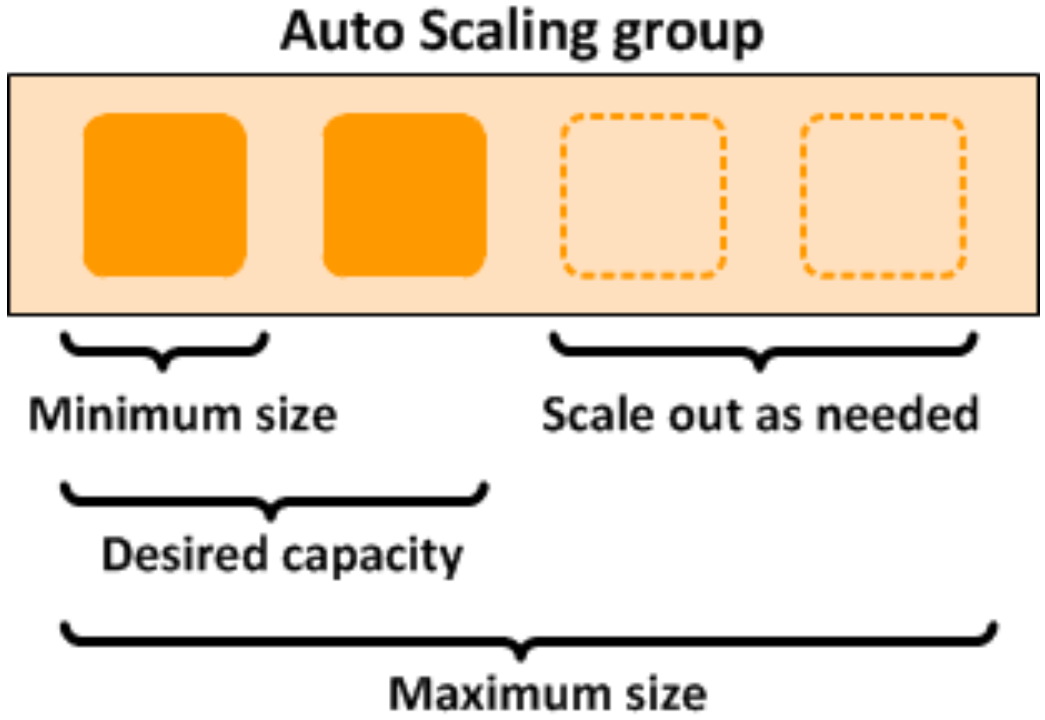
Ways to scale your Auto Scaling group prescriptively and dynamically



"There are no additional fees with Amazon EC2 Auto Scaling"

What is Amazon EC2 Auto Scaling?

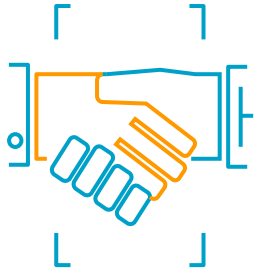
Your EC2 instances are organized into *groups* so that they can be treated as a logical unit for the purposes of scaling and management. When you create a group, you can specify its minimum, maximum, and, desired number of EC2 instances.



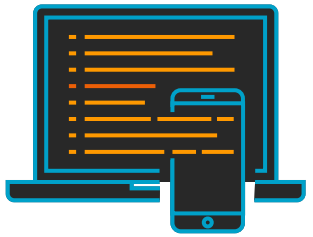
Launch Templates

Increased productivity

Simplified permissions



Governance & best practices



Consistent experience

Launch Templates are supported in EC2 Auto Scaling groups, EC2 Fleet, and AWS Batch.

Eg: Launch Templates

EC2 > Launch templates > BesaLT

BesaLT (lt-040247a777dc9973d)

Actions | Delete template

Launch template details

Launch template ID	Launch template name	Default version	Owner
lt-040247a777dc9973d	BesaLT	1	arn:aws:sts::158985497076:assumed-role/Admin/pateria-Isengard

Details | Versions | Template tags

Launch template version details

Actions | Delete template version

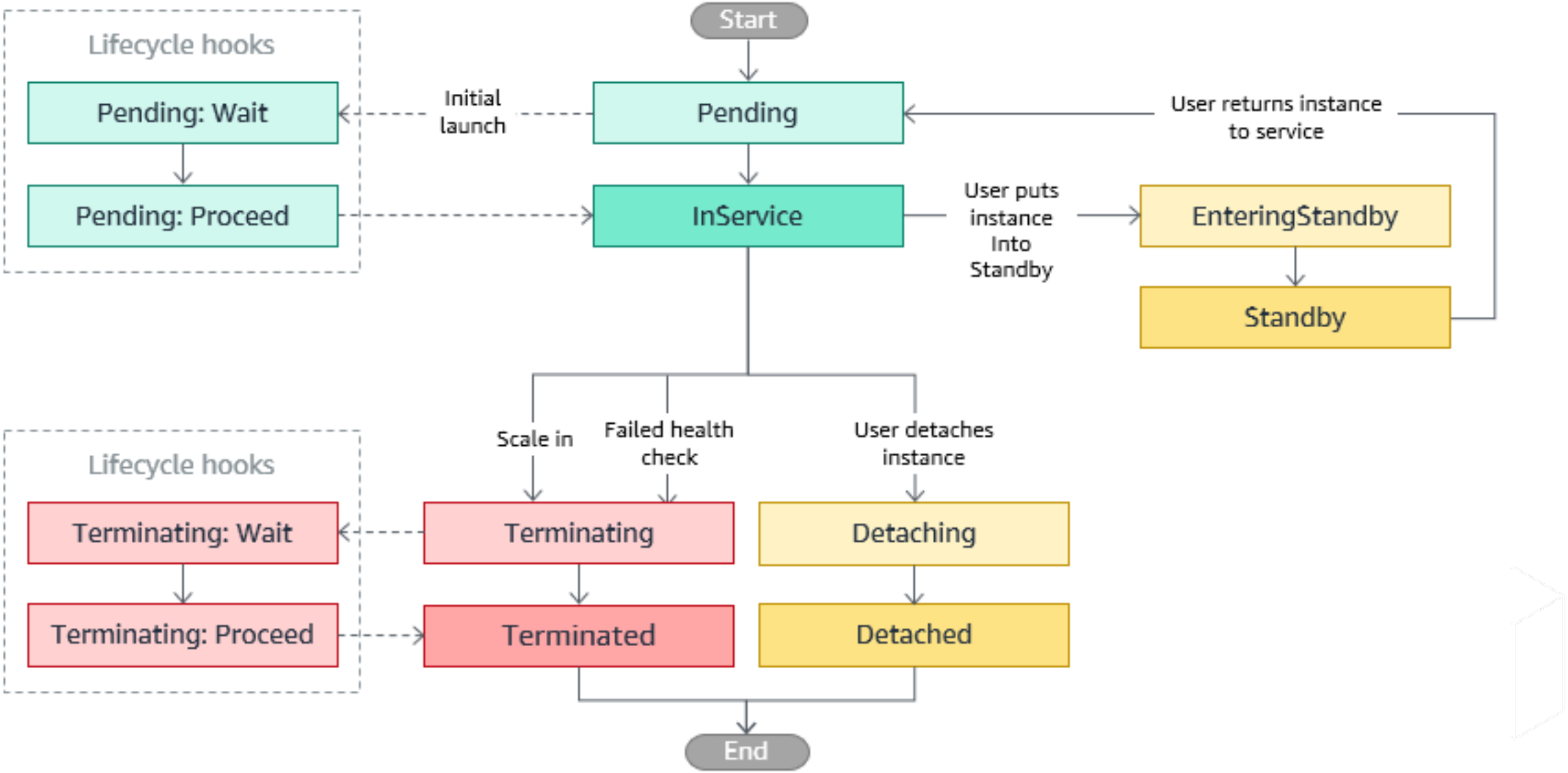
Version	Description	Date created	Created by
1 (Default)	Launch Template for Besa	2023-03-30T17:21:07.000Z	arn:aws:sts::158985497076:assumed-role/Admin/pateria-Isengard

Instance details | Storage | Resource tags | Network interfaces | Advanced details

AMI ID	Instance type	Availability Zone	Key pair name
ami-00c39f71452c08778	t2.micro	-	VG

Security groups	Security group IDs
-	sg-092d3b7aae3fea5e1

Instance Lifecycle

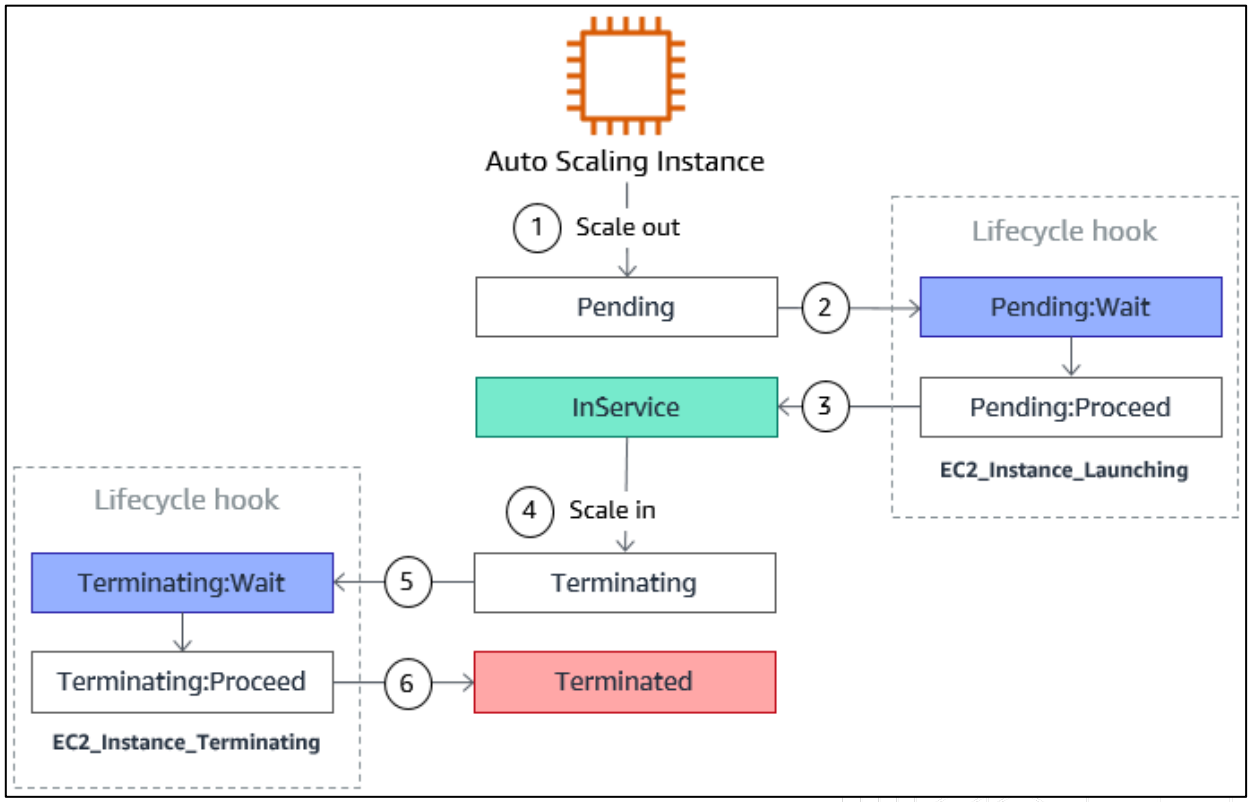


Lifecycle Hooks

Example :

Scale Out – When an instance is in a wait state it can use a script to install software, configure services, or download assets

Scale In – When an instance is in a wait state an event can be used to check-point/download logs from off-instance



Scaling Options

Manual Scaling

Change the size of Auto Scaling Group manually by updating the desired capacity

Predictive Scaling

To increase the number of EC2 instances in your Auto Scaling group in advance of daily and weekly patterns

Scheduled Scaling

To set up your own scaling schedule according to predictable load changes.

Dynamic Scaling

Scales the capacity of your Auto Scaling group as traffic changes occur

Simple Scaling

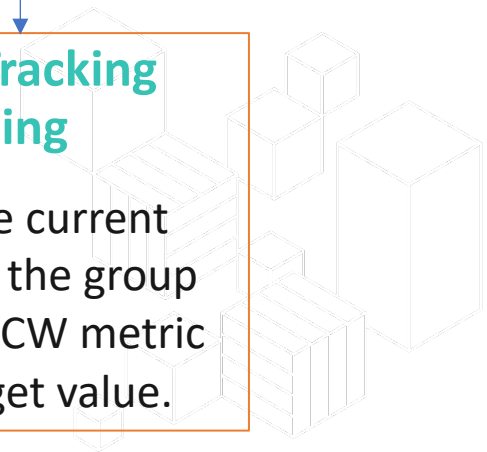
Increase and decrease the current capacity of the group based on a single scaling adjustment

Step Scaling

Automatically scale the number of instances dynamically based on the size of the alarm breach

Target Tracking Scaling

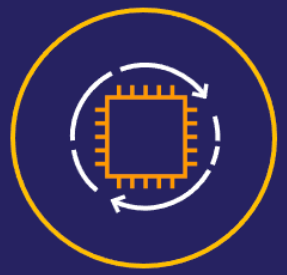
Adjust the current capacity of the group based on a CW metric and a target value.



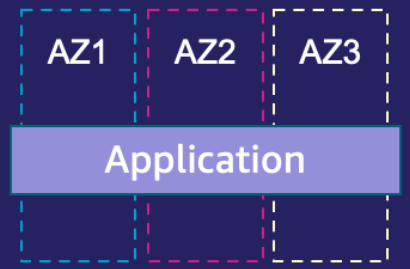
Benefits of EC2 Auto Scaling

Improve Fault Tolerance

- Instance Lifecycle Management
- Across Availability Zones



Lifecycle Management



Fault Tolerant Application

Increase Application Availability

- Match capacity to demand
- Proactive provisioning

Predictive Scaling

Lower Costs

- Scale across instance types, purchase options, architectures
- Reduce over-provisioning

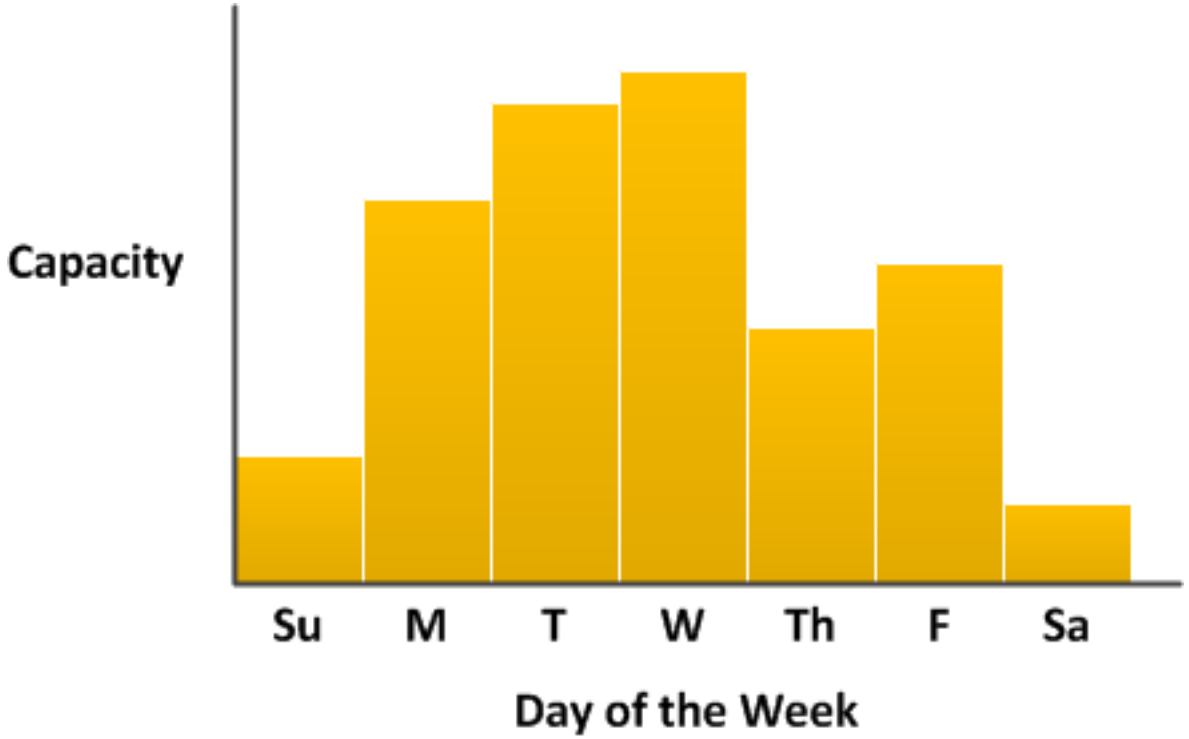
EC2 Spot Instances

Savings Plan

AWS Graviton

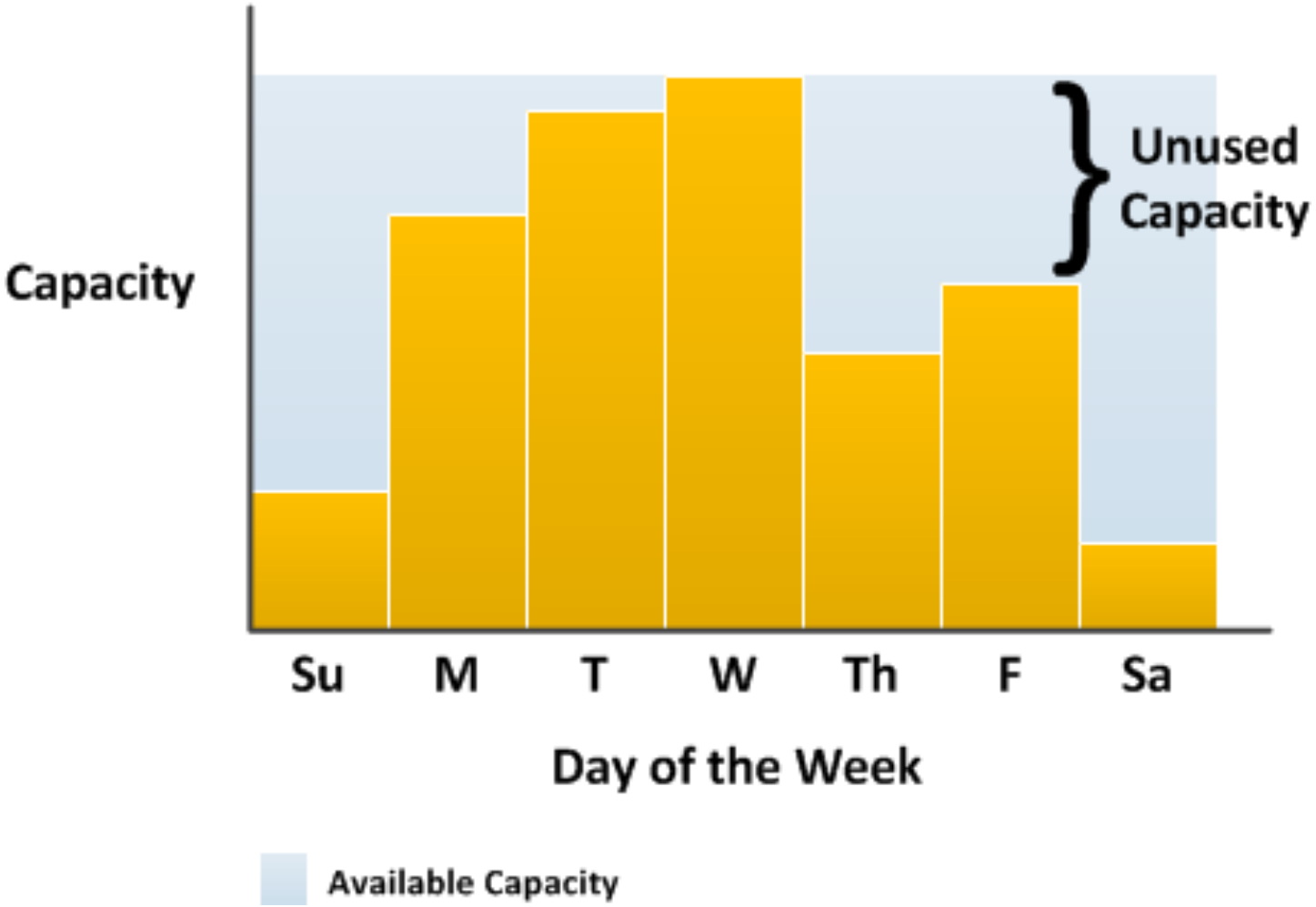
Example

Example: Cover variable demand



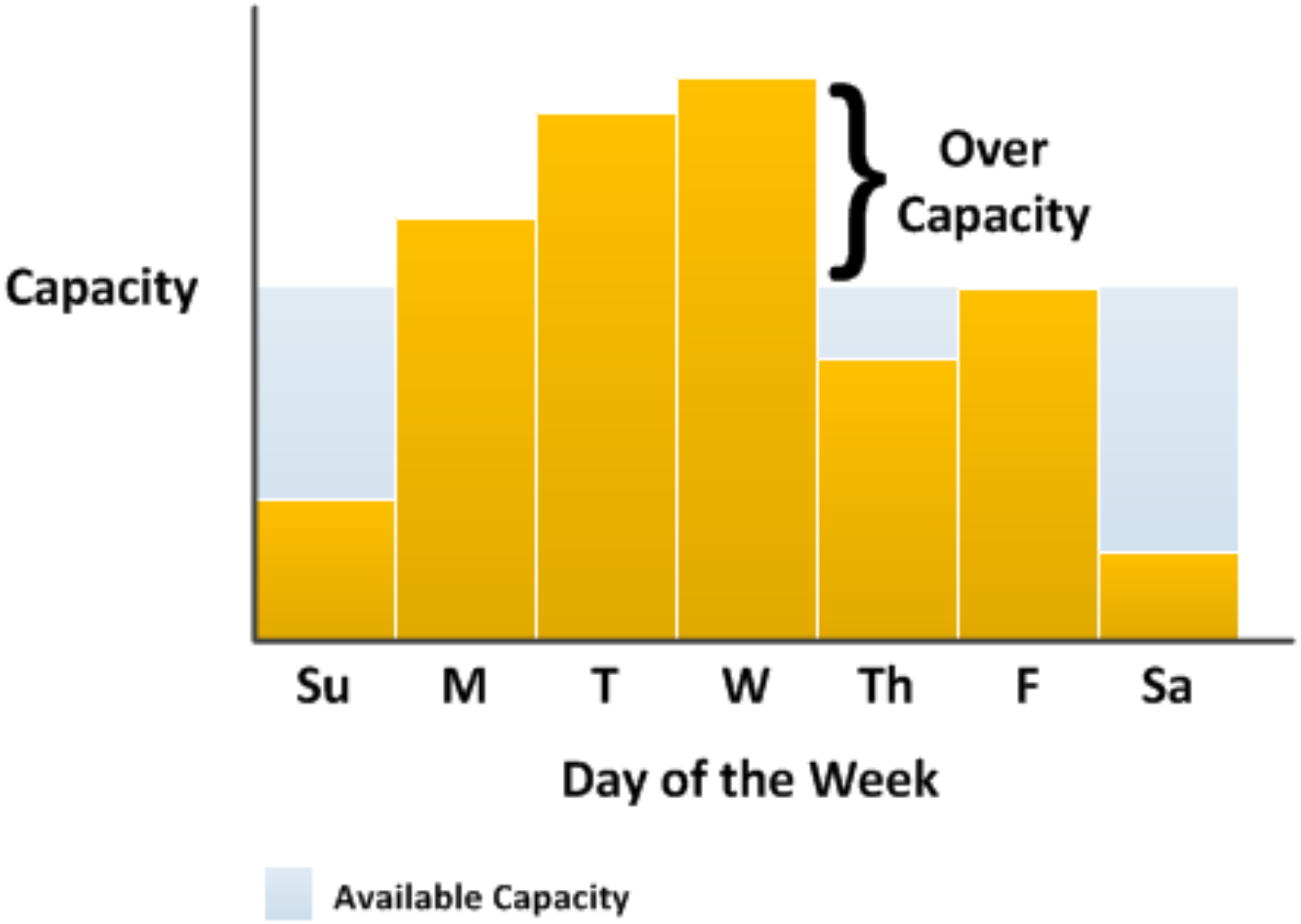
Usage Time	Demand
Start of the Week	Low Demand
Middle of the Week	High Demand
End of the Week	Low Demand

Traditional Way – Option 1



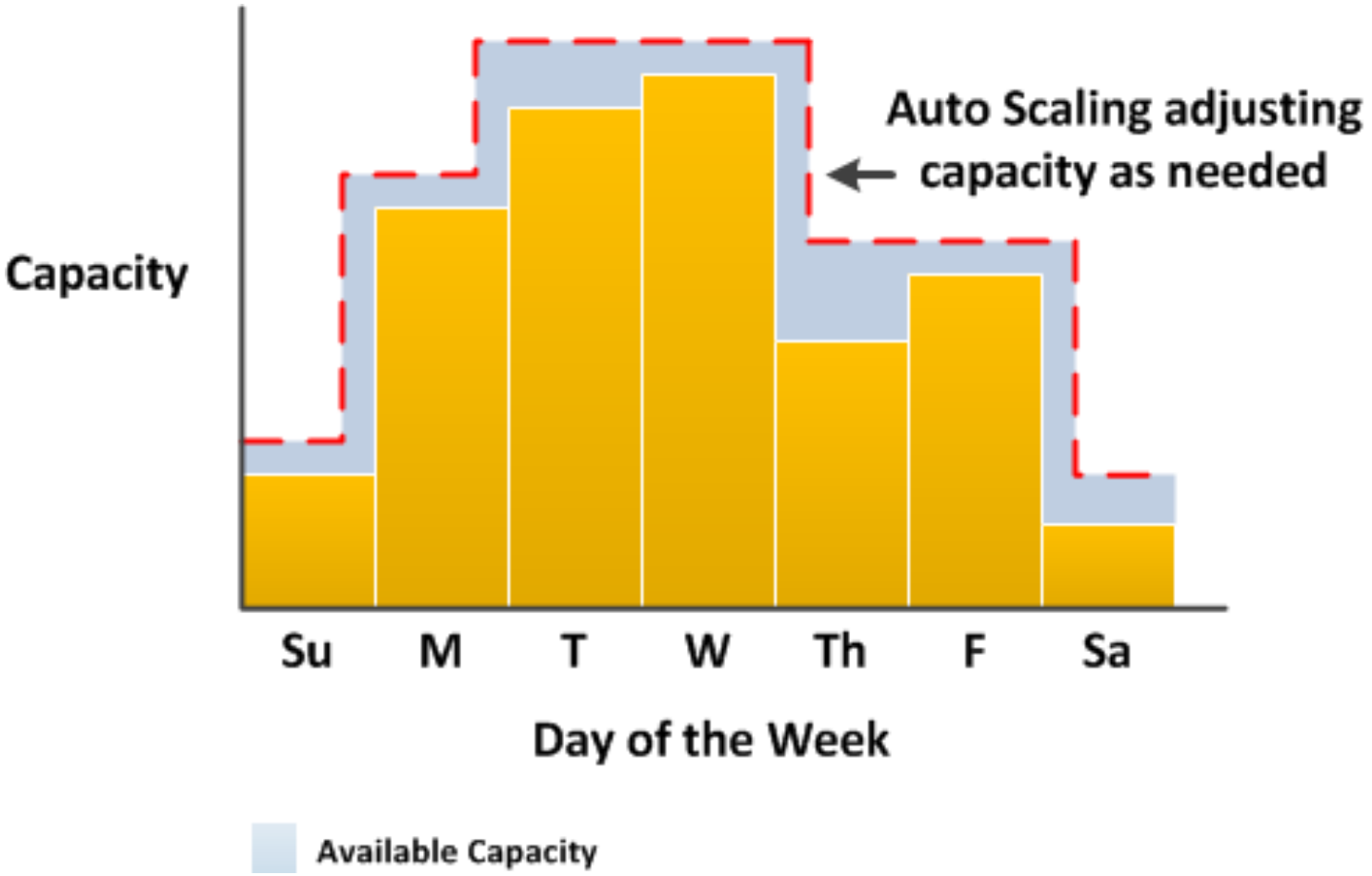
[Plan for the “PEAK” capacity]

Traditional Way – Option 2



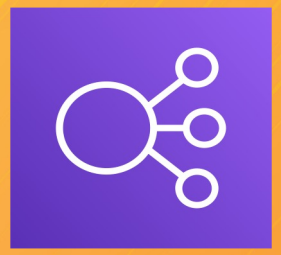
[Plan for the “AVERAGE” capacity]

AWS Way ☺



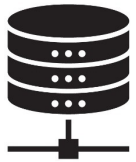
[Adjust to the capacity]

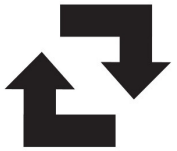
Elastic Load Balancing

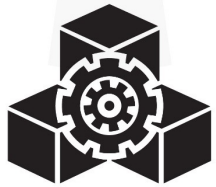


Elastic Load Balancing

A managed load balancing service to act as a front door and distribute incoming traffic across multiple EC2 instances, containers, IP addresses, and Lambda functions.

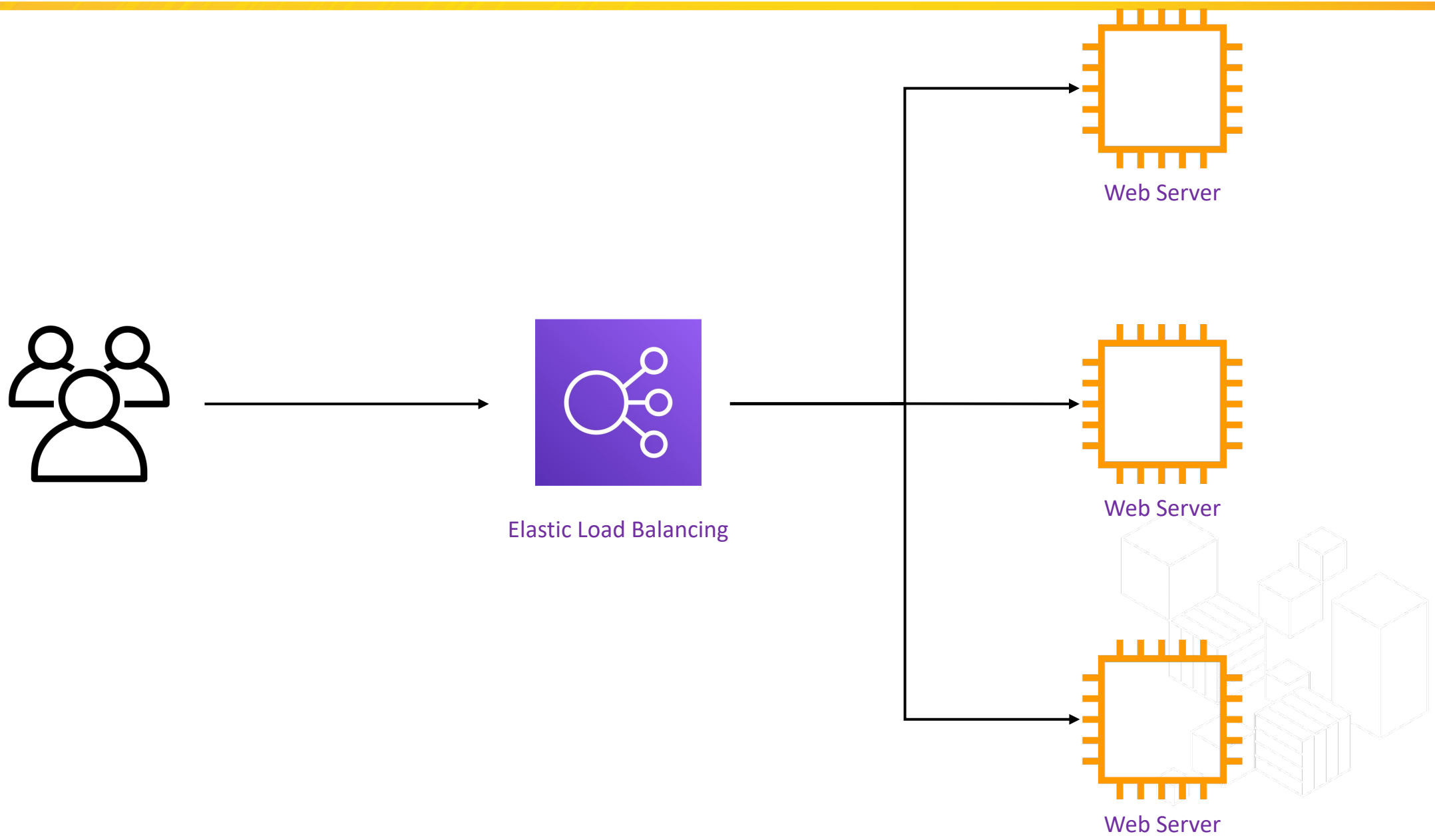





“Single server – single point of failure , limited availability “

Front door approaach



Load balancer benefits

High Availability

Distributes workloads across multiple compute resources, such as virtual servers

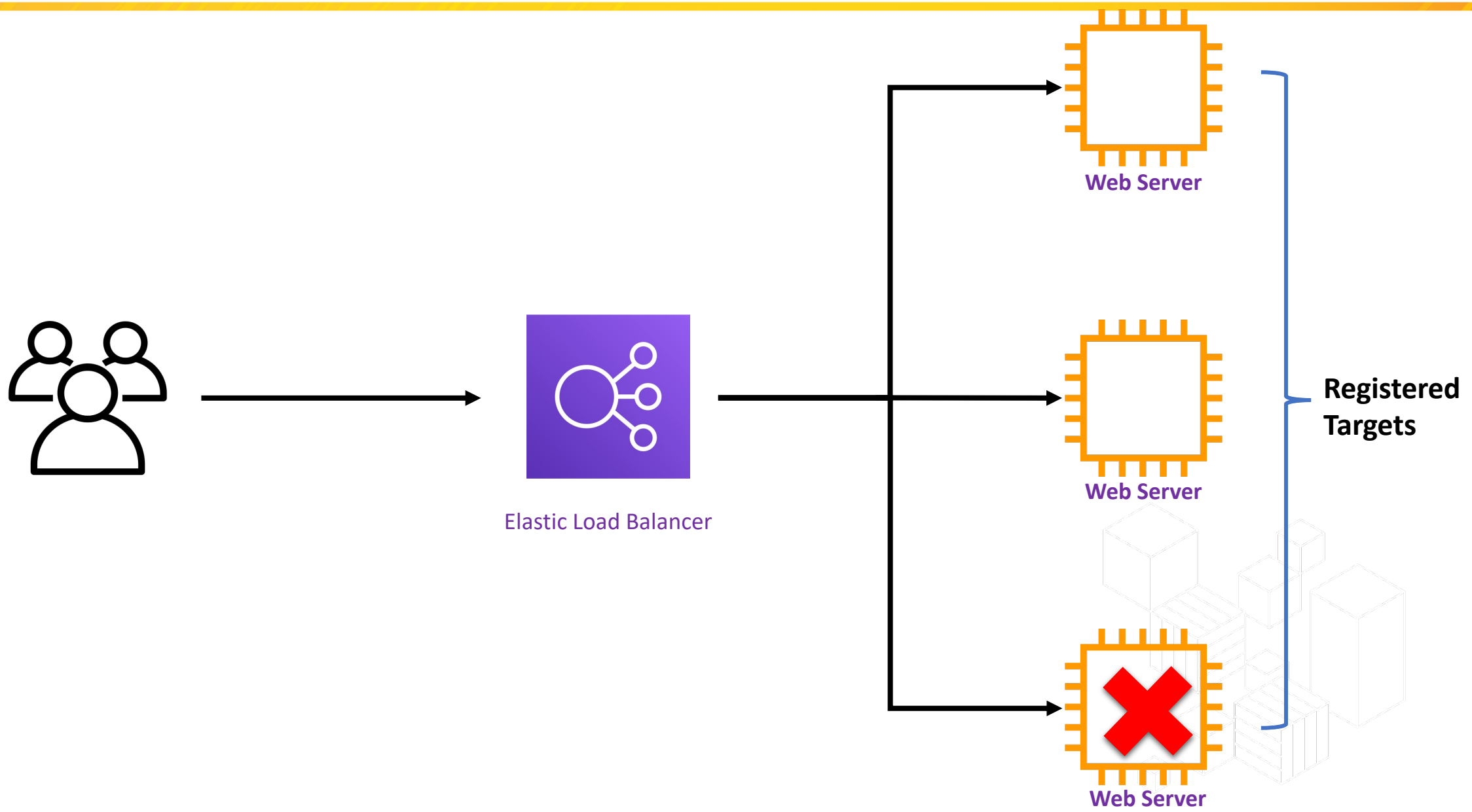
Customer Experience

You can add and remove compute resources from your load balancer as per need, without disrupting the overall flow of requests

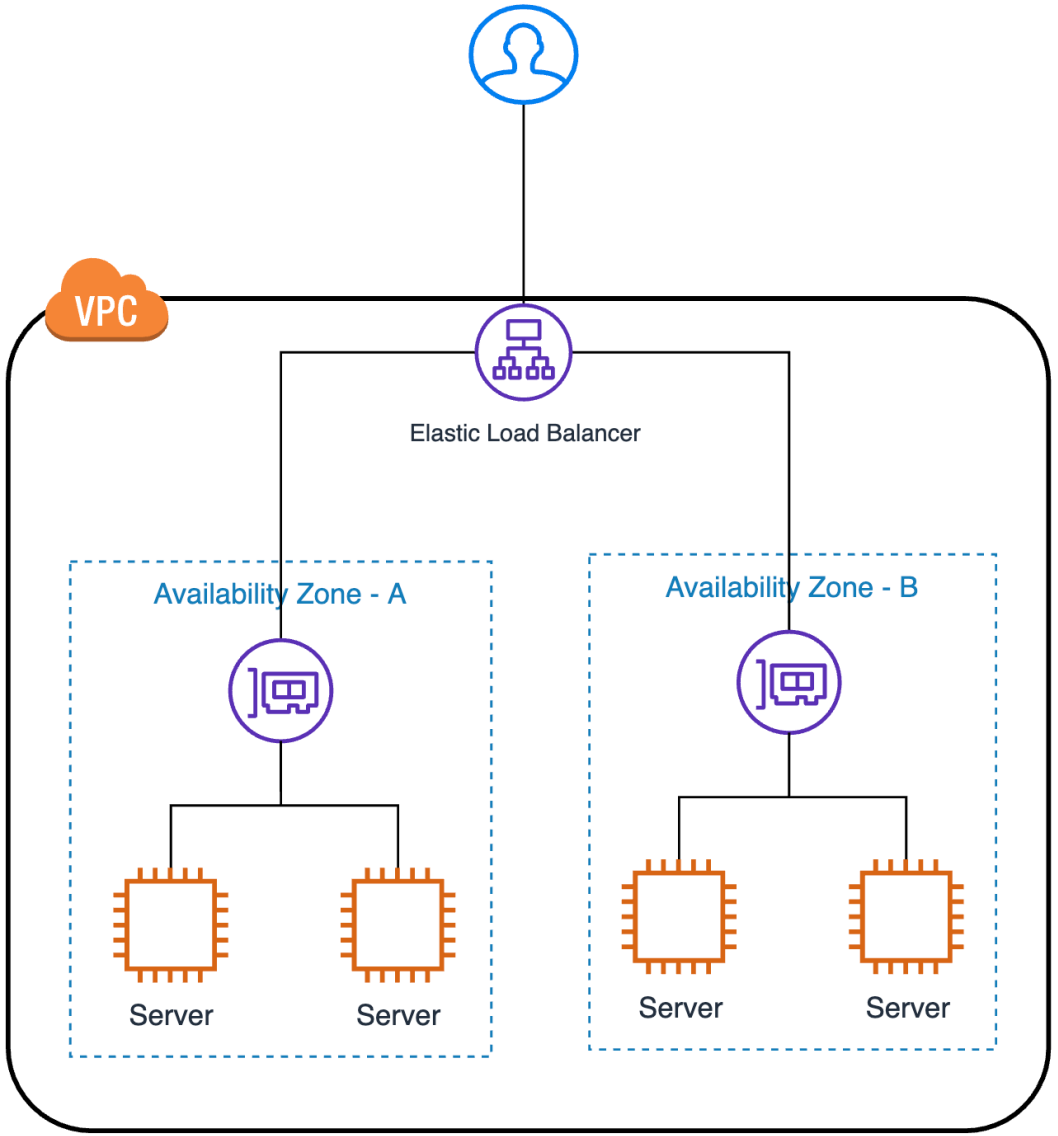
Fault Tolerance

You can configure health checks, which monitor the health of the compute resources, so that the load balancer sends requests only to the healthy ones.

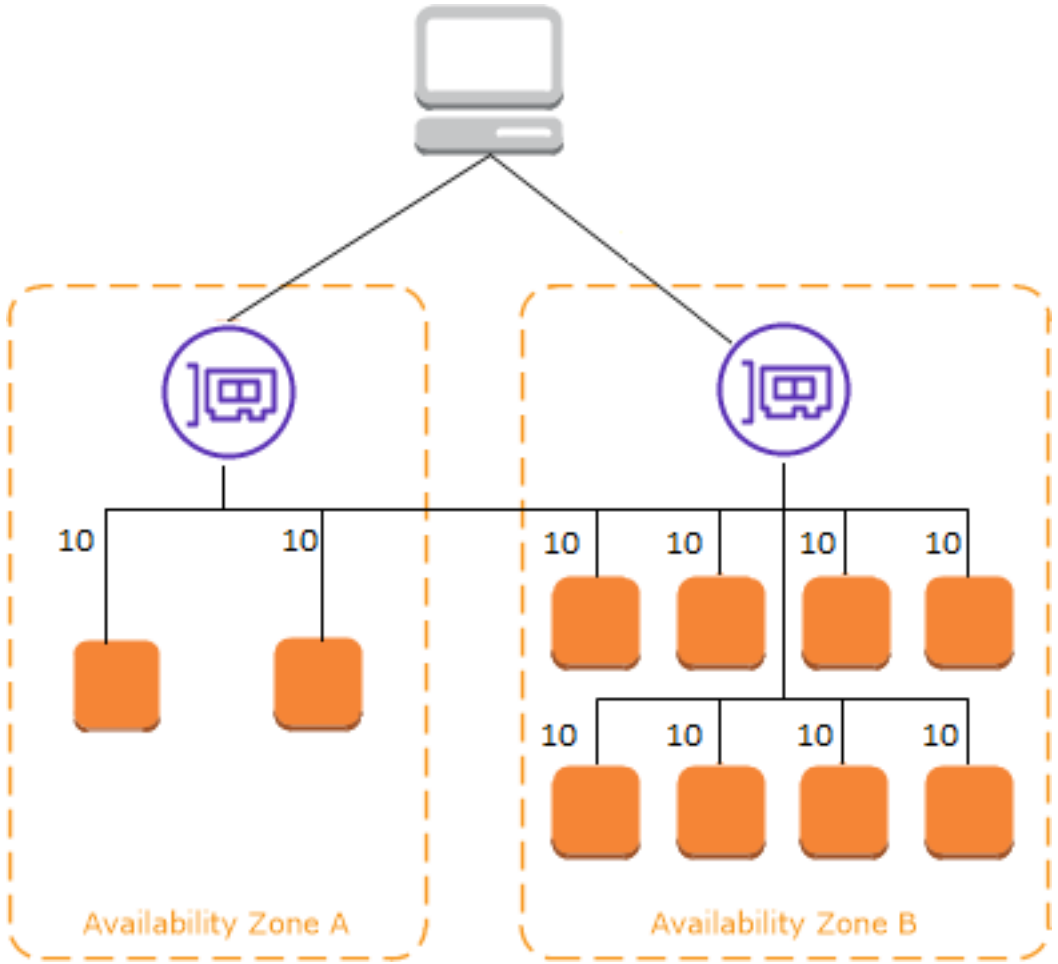
How it actually works ?



Load Balancer and Availability Zones

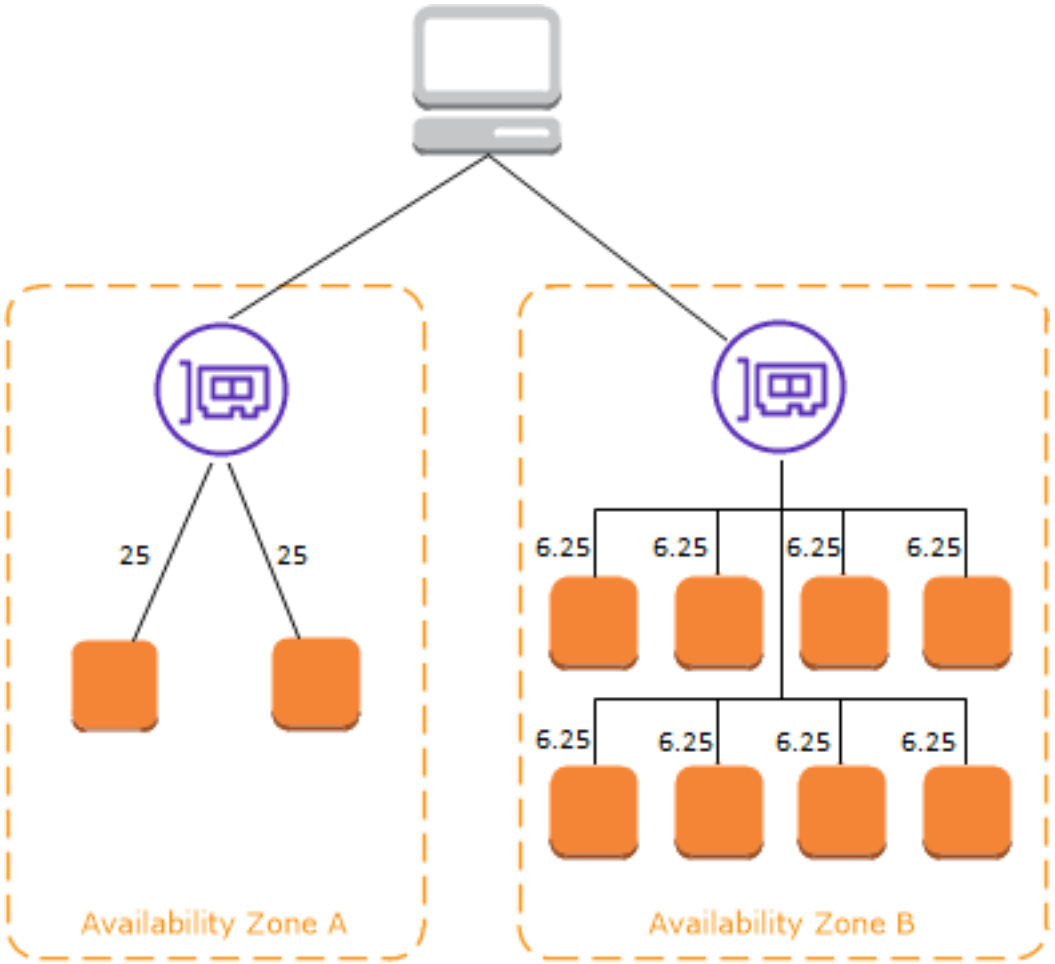


Cross Zone Load Balancing



[Equal distribution of traffic]

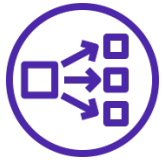
Cross Zone Load Balancing



[Un-equal distribution of traffic]

Types of Load Balancers

Network Load Balancer

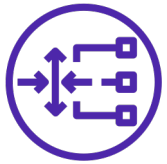


Operates at transport layer (Layer 4), routing connections to targets (EC2 instances, IP addresses, and containers) within Amazon VPC, based on IP protocol data

Application Load Balancer

Operates at the application level (layer 7), routing traffic to targets (EC2 instances, containers, IP addresses, and Lambda functions) based on the content of the request.

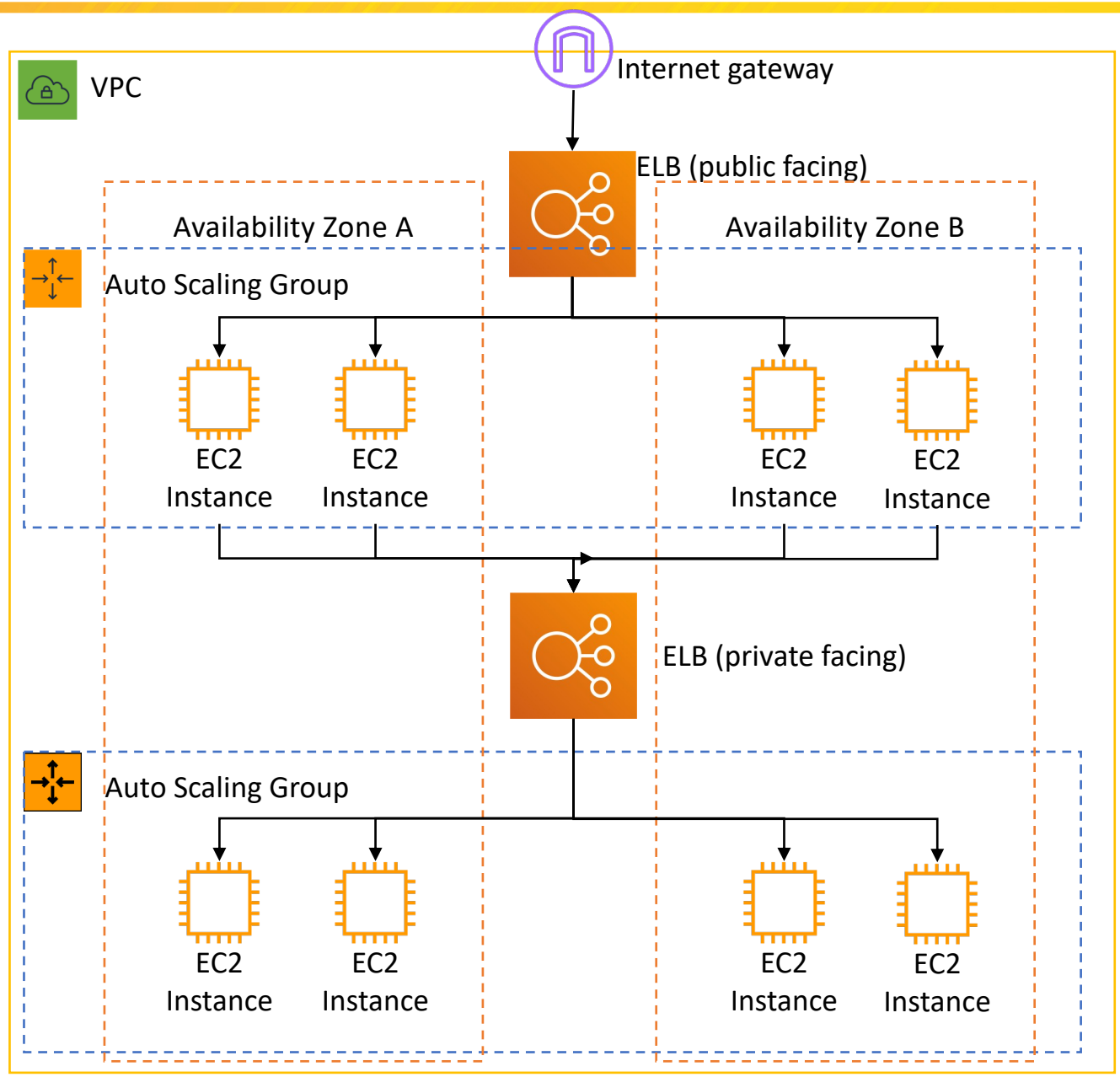
Gateway Load Balancer



Helps you easily deploy, scale, and manage your third-party virtual appliances.

Feature	Application Load Balancer	Network Load Balancer	Gateway Load Balancer
Load Balancer type	Layer 7	Layer 4	Layer 3 Gateway + Layer 4 Load Balancing
Target type	IP, Instance, Lambda	IP, Instance, Application Load Balancer	IP, Instance
Terminates flow/proxy behavior	Yes	Yes	No
Protocol listeners	HTTP, HTTPS, gRPC	TCP, UDP, TLS	IP
Reachable via	VIP	VIP	Route table entry

Horizontal scaling: Elastic Load Balancing



- Distribute traffic to multiple targets

- EC2 instances

- Containers

- IP addresses

- Multiple Availability Zones

- ELB Scales automatically

- Support Auto Scaling Groups

- Automatically (de)register instances to the ELB

Gateway Load Balancer

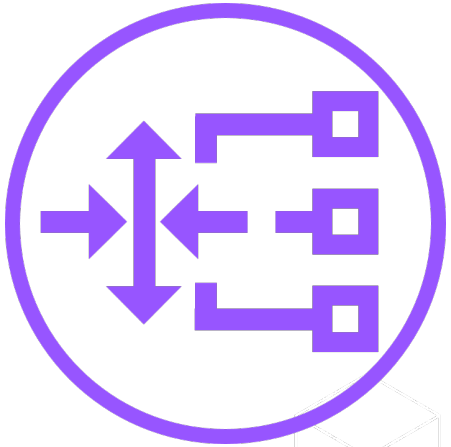
Gateway Load Balancer

Reduce complexity and deploy faster

Elastically scale and reduce costs

Improve appliance availability

Supported by leading appliance vendors

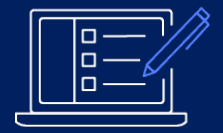


Deploy virtual appliances in an elastic, fault-tolerant, and transparent manner

Network Appliances

Transparent to network traffic

Security, monitoring, analytics, and other use cases

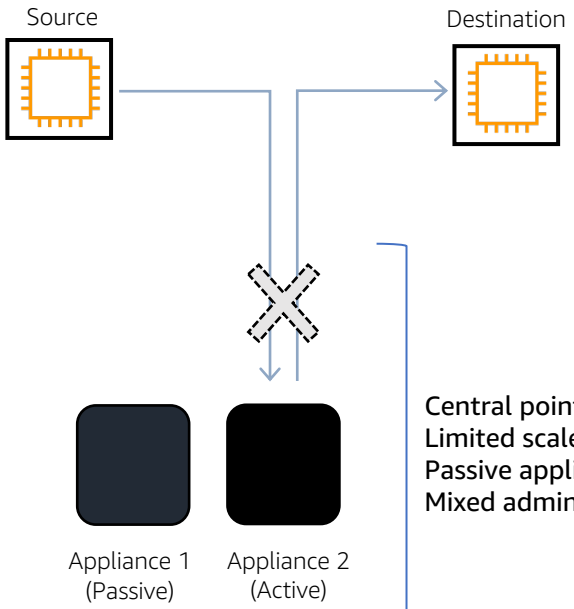


Often required by policy, or due to expertise and investment

Use the same Network Appliances on AWS and Hybrid Environments

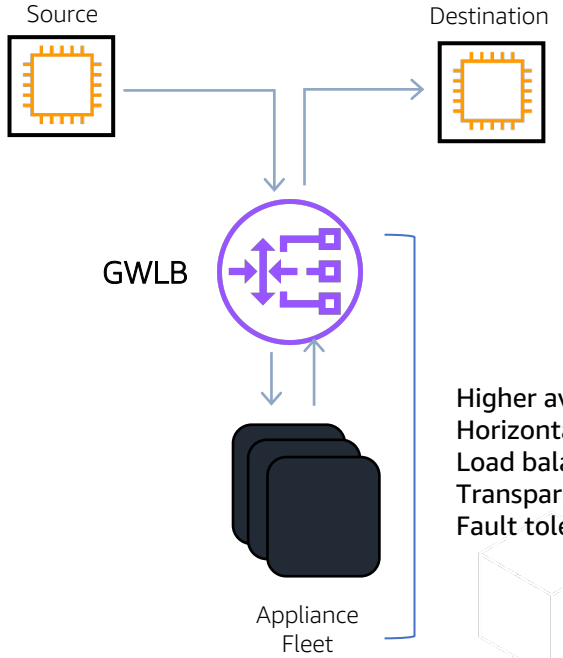
Traditional challenges vs now with Gateway Load Balancer

Before Gateway Load Balancer

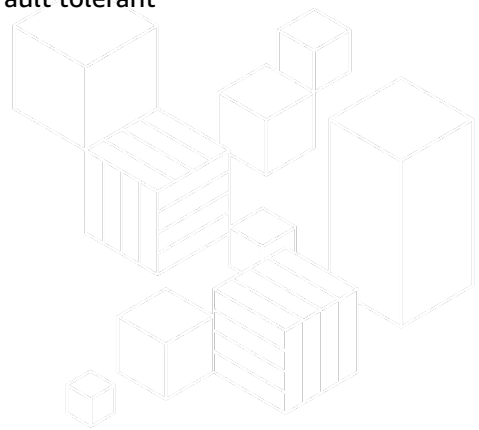


Central point of failure
Limited scale
Passive appliances sit idle
Mixed admin domains

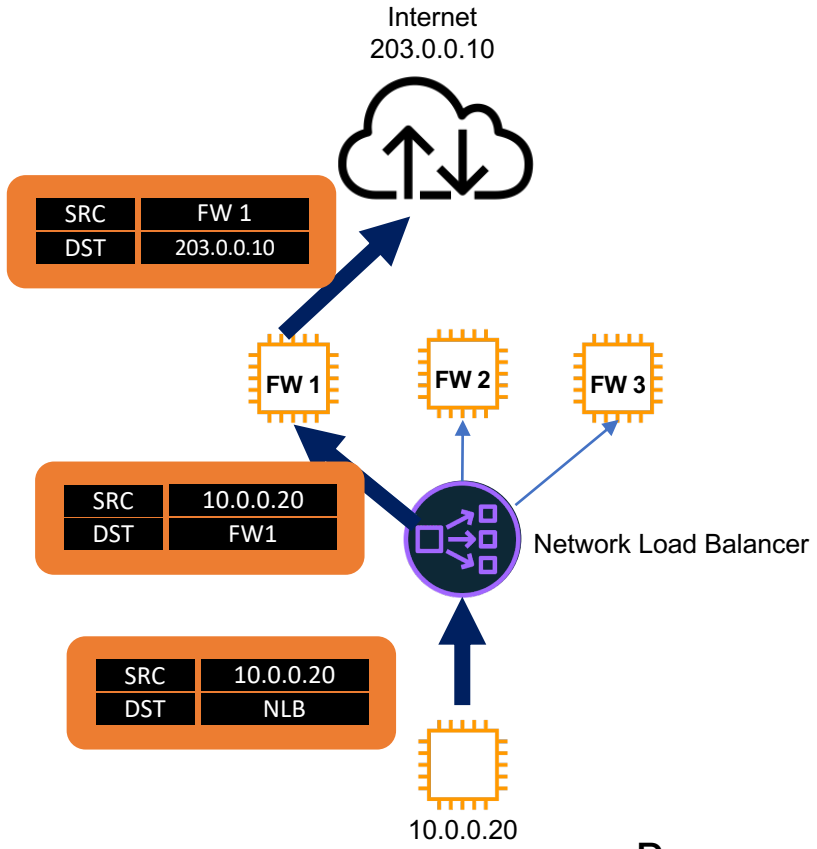
After Gateway Load Balancer



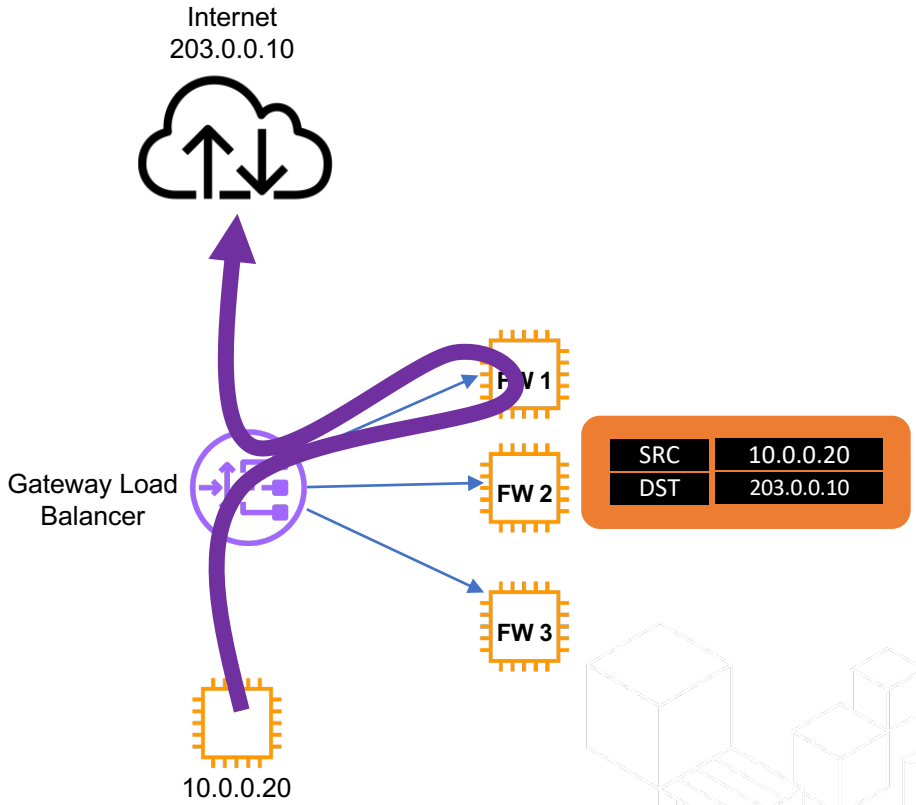
Higher availability
Horizontal scaling
Load balancing
Transparent
Fault tolerant



Traditional challenges vs now with Gateway Load Balancer

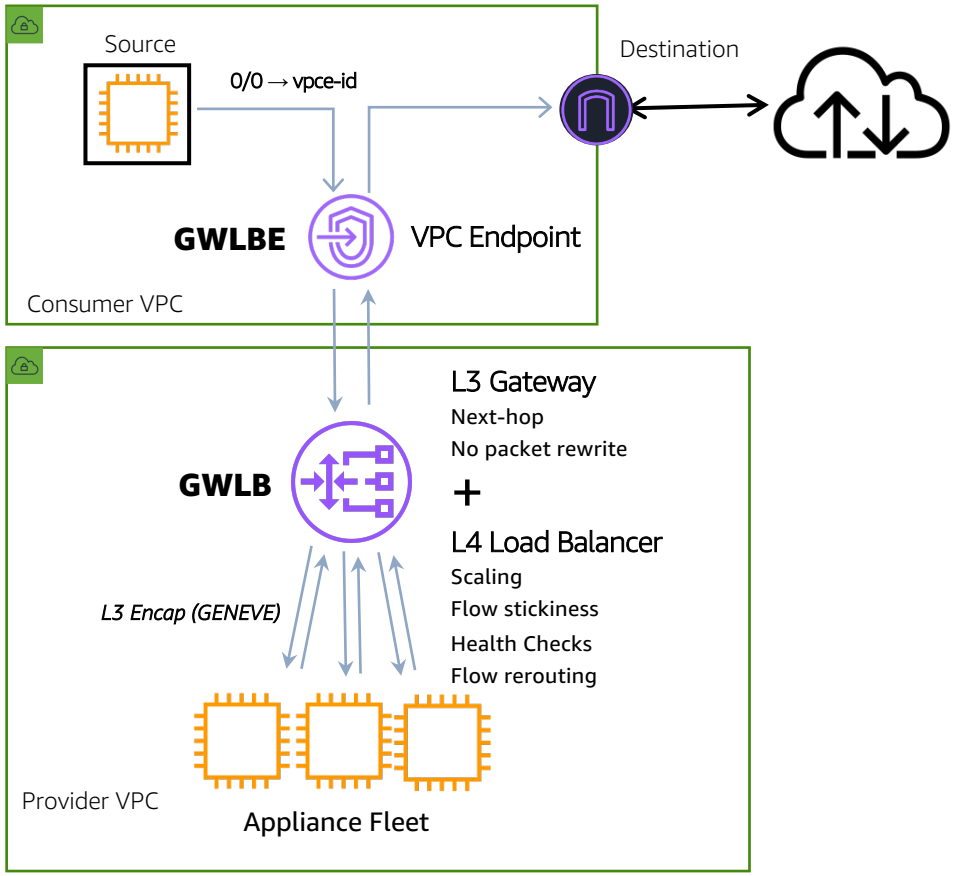


Proxy



BITW

Gateway Load Balancer: At-a-Glance

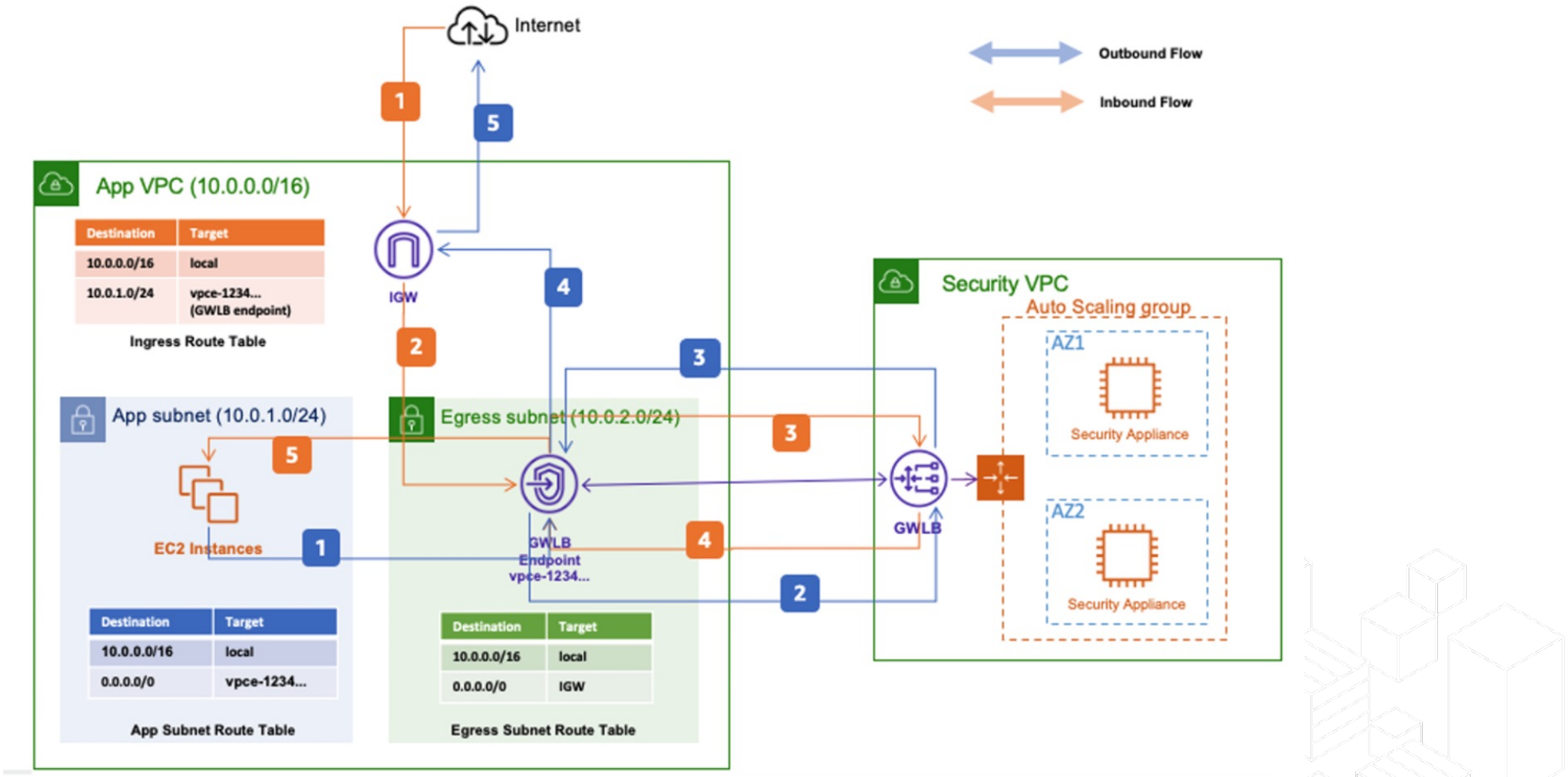


Components

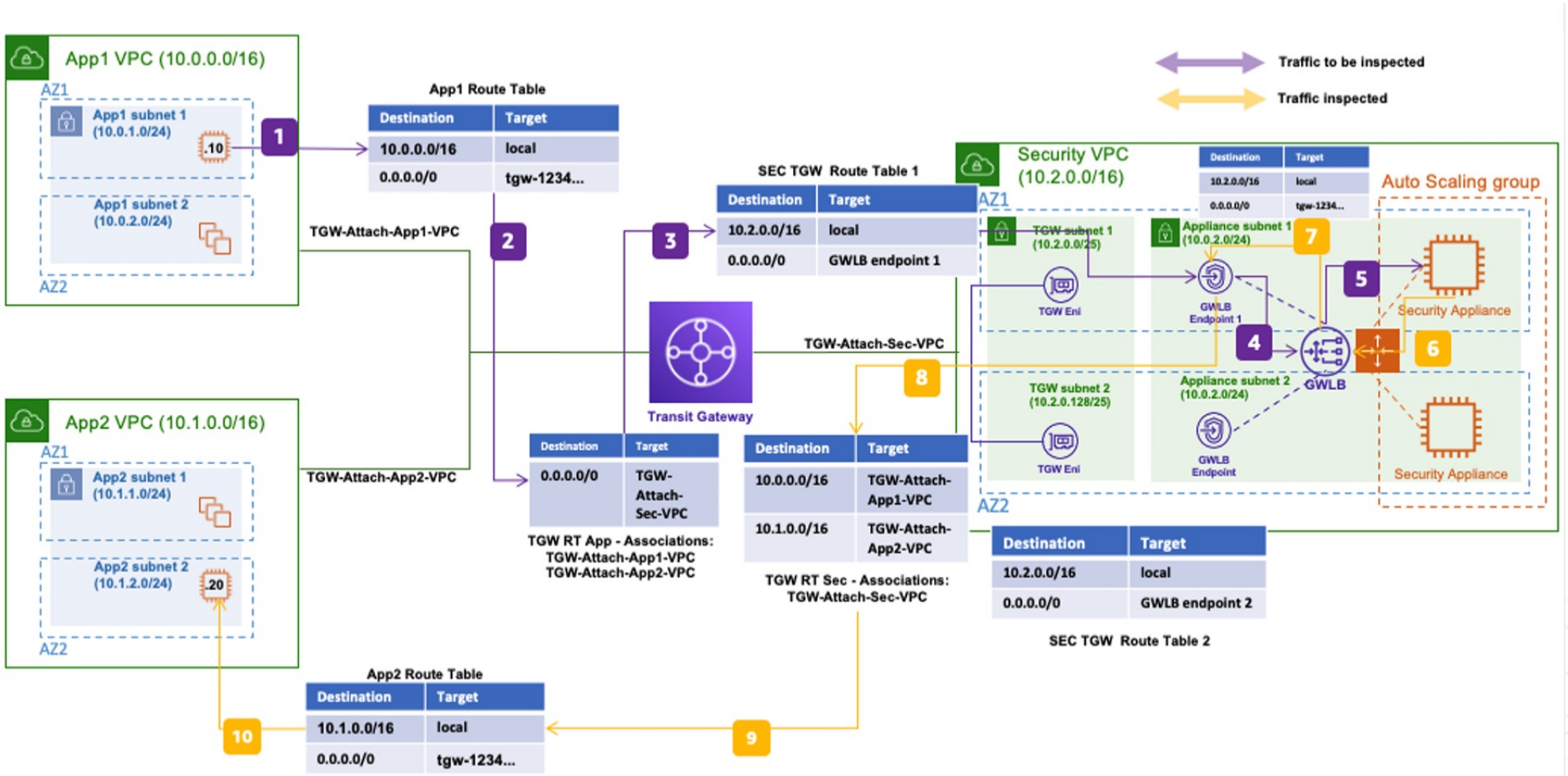
- **Gateway Load Balancer Endpoint (GWLBE)** - A new type of VPC endpoint that can be a next-hop in a VPC route table
- **Gateway Load Balancer (GWLB)** - A new type of load balancer that includes L3 Gateway + L4 Load Balancer capabilities
- Both components powered by AWS Hyperplane

AWS Gateway Load Balancer Partners

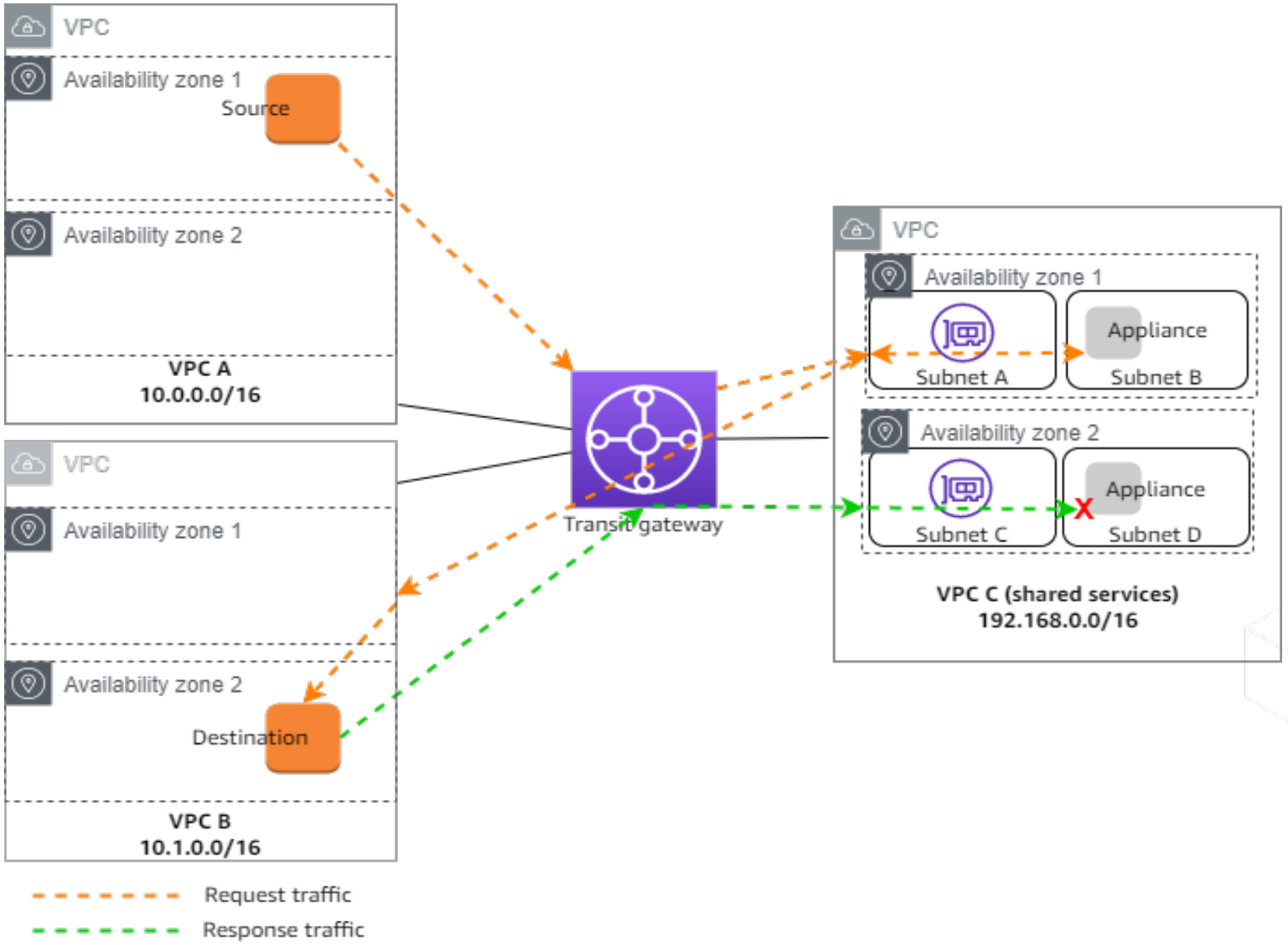
Architecture for Gateway Load Balancer – North/South Inspection



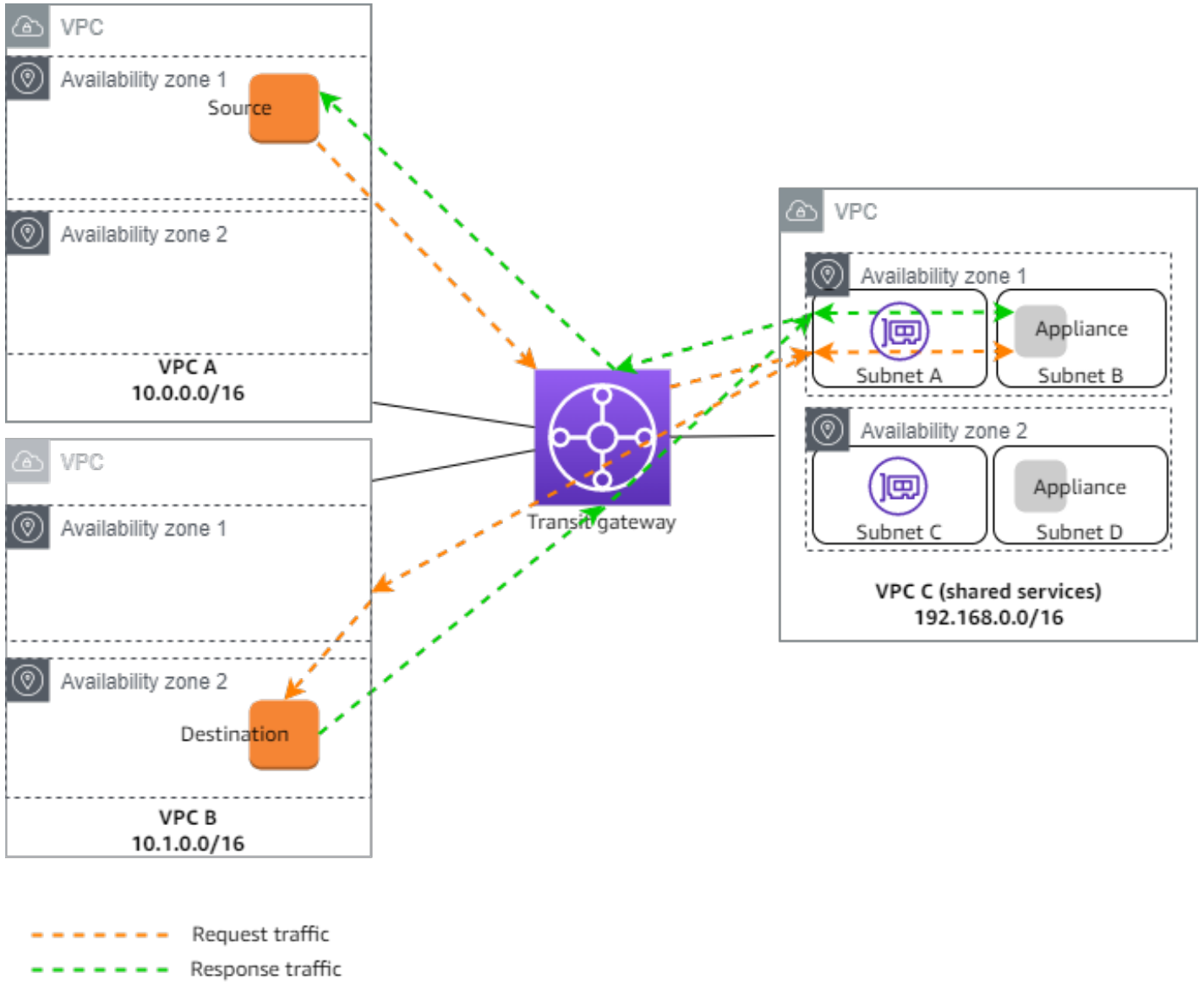
Architecture for Gateway Load Balancer – East/West Inspection



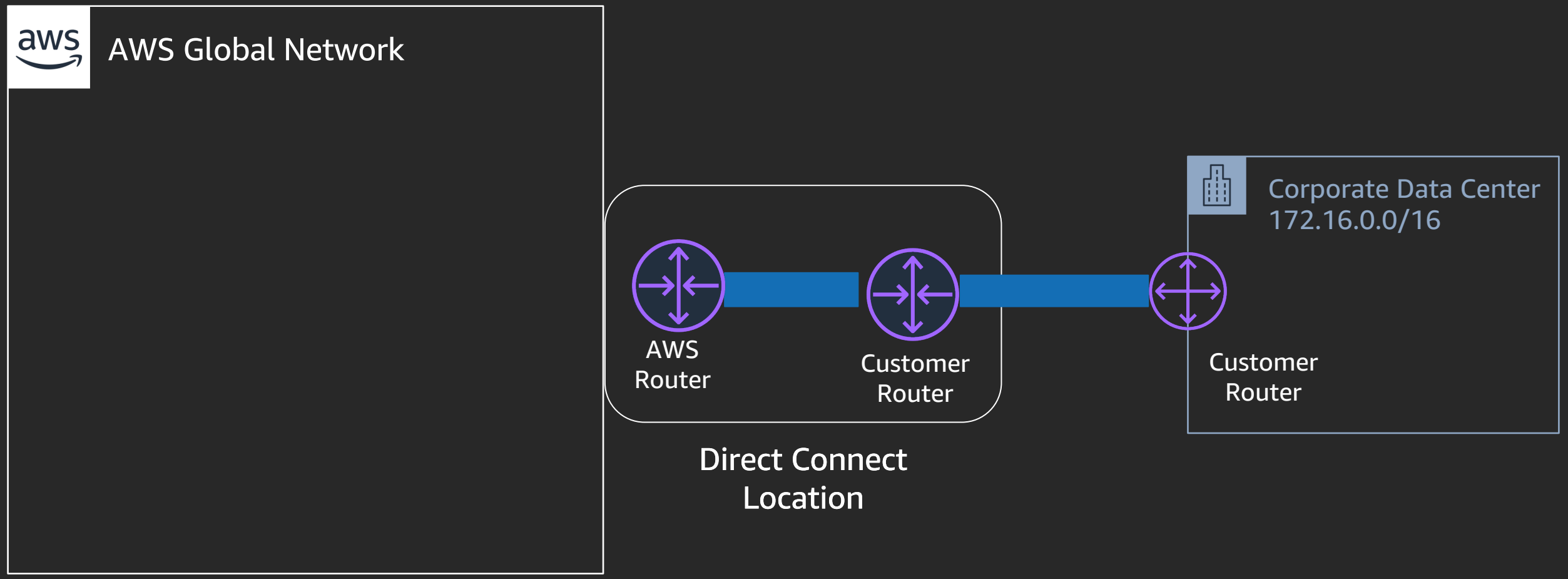
East/West Inspection – Asymmetric routing issue



Transit Gateway Appliance Mode



AWS Direct Connect – physical connection

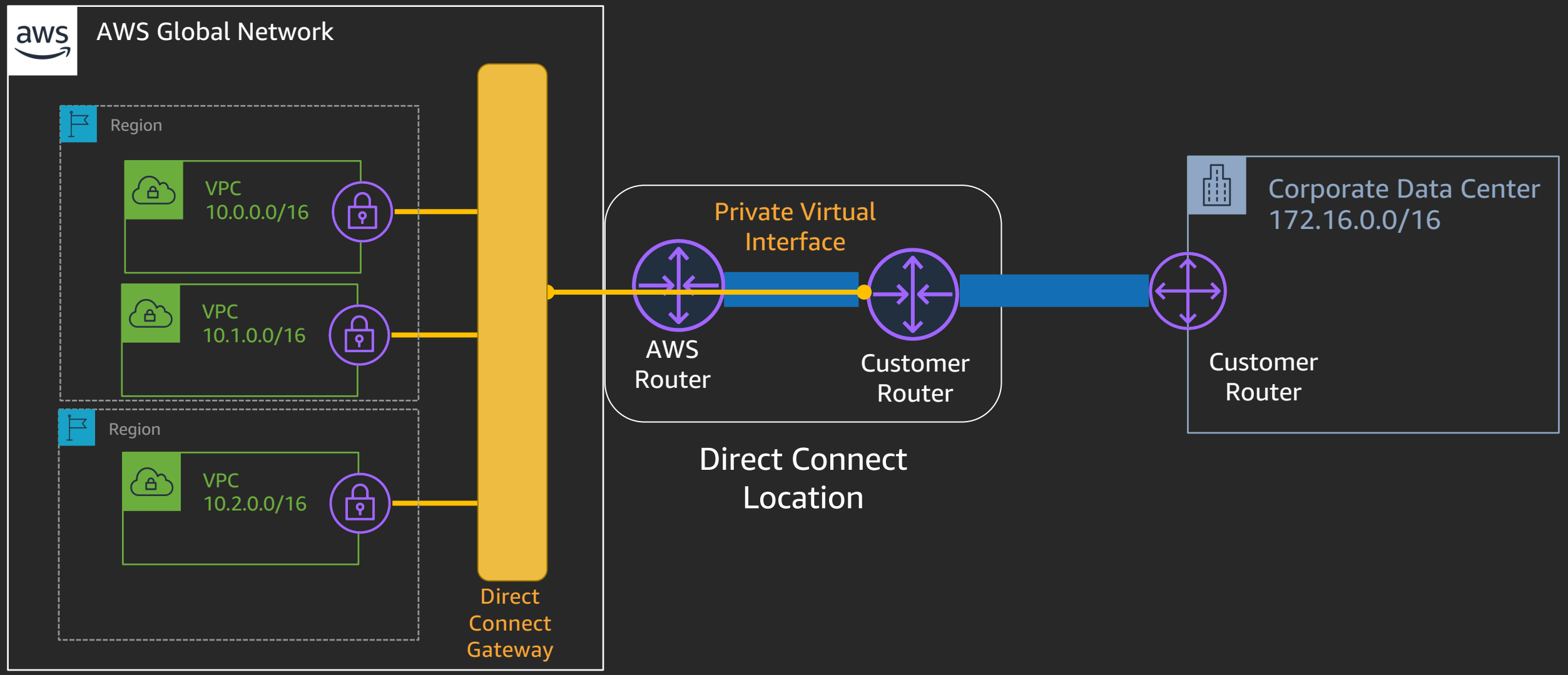


AWS Direct Connect – Interface types

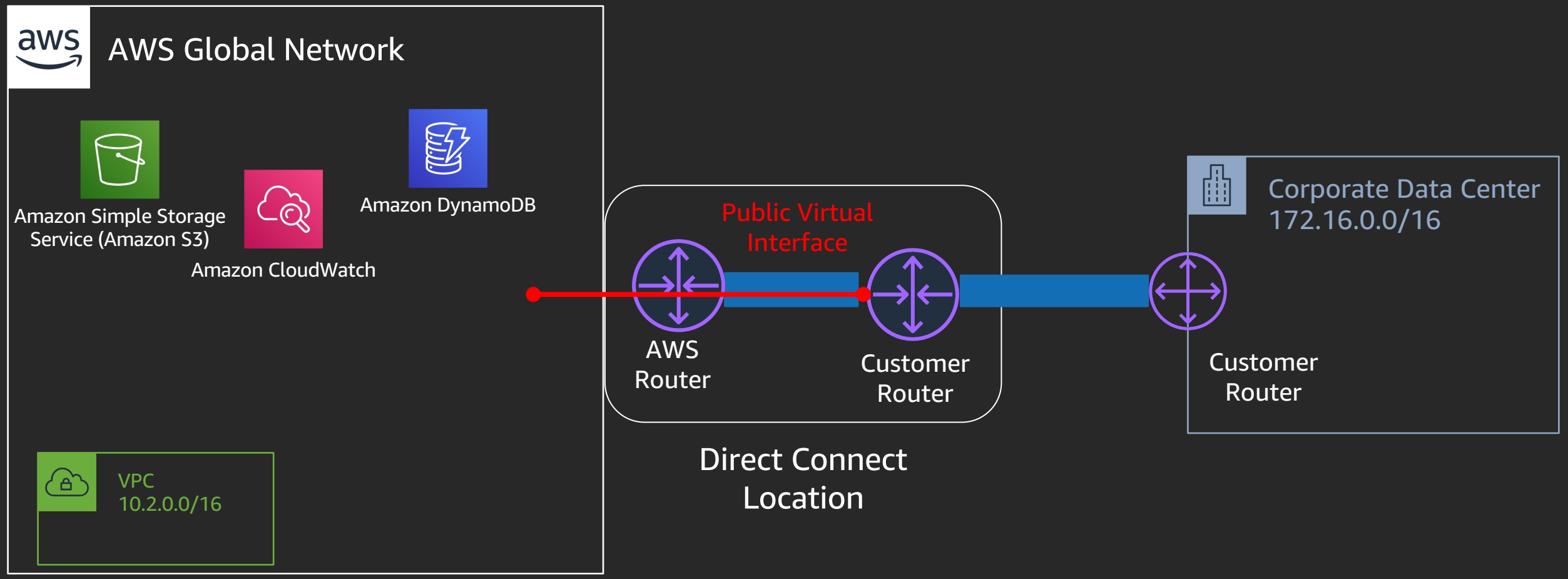
- **Private VIF** – Used to connect to Amazon VPCs using private IP addresses; directly or via Direct Connect gateway
- **Transit VIF** – Used to connect to AWS Transit Gateways via Direct Connect gateway
- **Public VIF** – Used to access all AWS public services using public IP addresses

All Virtual Interfaces are 802.1Q VLANs with BGP peering

AWS Direct Connect gateway – Private VIF



AWS Direct Connect – Public VIF



Time for Hands on