UNIVERSITY OF ŽILINA
Faculty of Management Science
and Informatics

Presentation 3 - Networking and Content Delivery

- **AWS M5 - Networking and Content Delivery**

aws academy

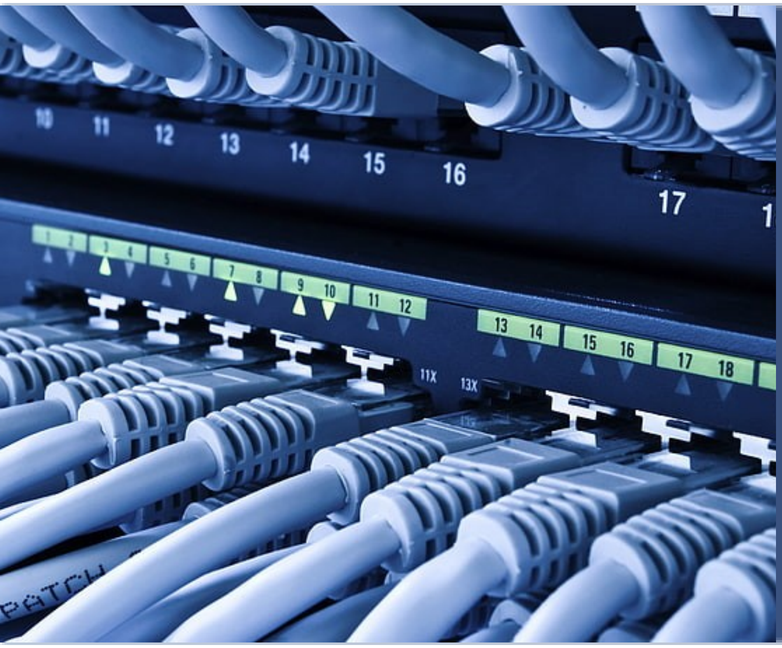
Vytvorené v rámci projektu **KEGA 026TUKE-4/2021**

Outline

- **AWS M5 - Networking and Content Delivery**
  - Networking basics
  - Amazon VPC
  - VPC networking and security
  - Amazon Route 53
  - Amazon CloudFront

Module objectives

- After completing this presentation, you should be able to:
  - Recognize the basics of networking
  - Describe virtual networking in the cloud with Amazon VPC
  - Label a network diagram
  - Design a basic VPC architecture
  - Indicate the steps to build a VPC
  - Identify security groups
  - Create your own VPC and add additional components to it to produce a customized network
  - Identify the fundamentals of Amazon Route 53
  - Recognize the benefits of Amazon CloudFront



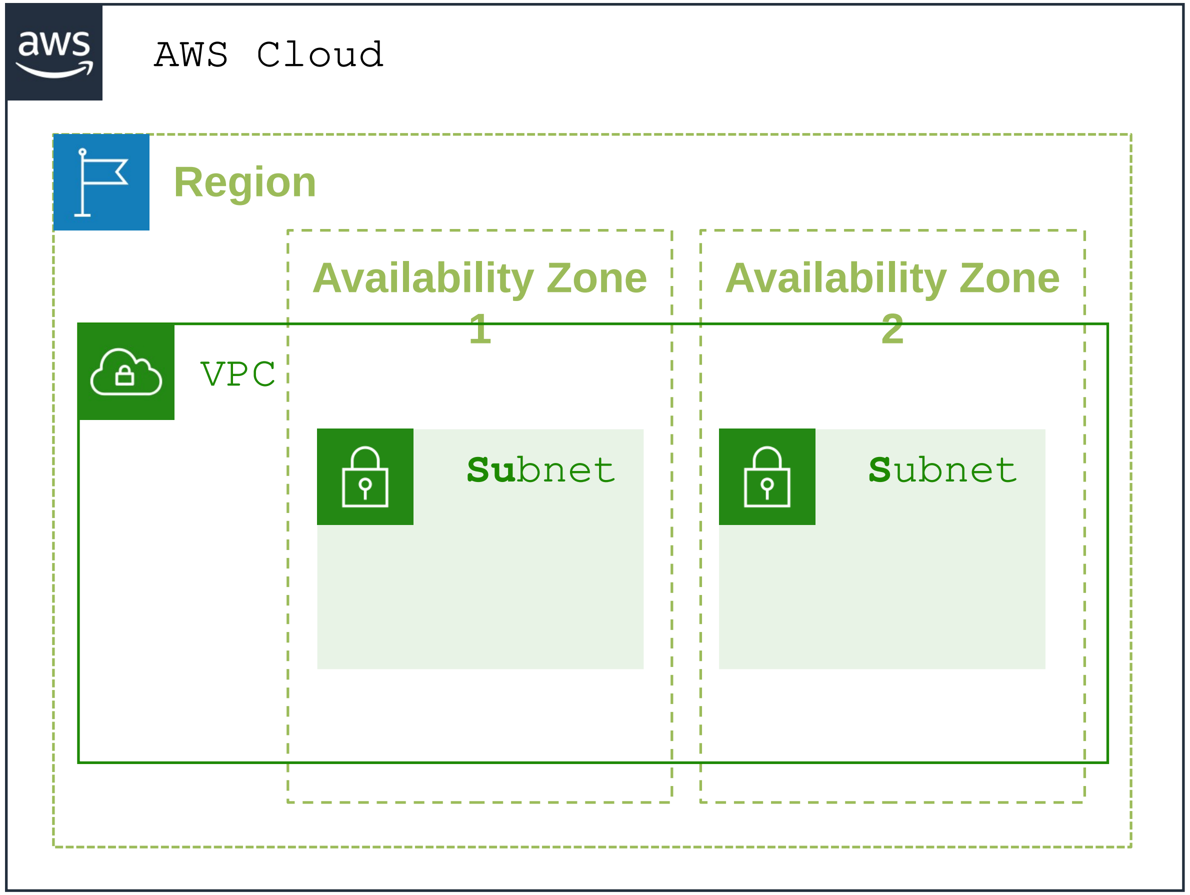
Amazon VPC

Amazon VPC

- **VPC = virtual private cloud =**
  - => logically isolated section of the AWS Cloud
  - Here yours AWS resources are launched in a virtual network that you define
  - Gives control to virtual networking resources, including:
    - Selection of IP address ranges
      - IPv4 and IPv6 are supported
    - Creation of subnets
      - Public/private
    - Configuration of route tables and network gateways
  - Enables to customize the network configuration for your VPC
  - Provides multiple layers of security
    - Security groups
    - Network access control lists (ACL)

VPCs and subnets

- VPCs:
  - **Logically isolated** from other VPCs
  - **Dedicated** to your AWS account
  - Belong to a single **AWS Region** and can span multiple Availability Zones
- Subnets:
  - **Range of IP addresses** that divide a VPC
  - Belong to a single **Availability Zone**
  - Classified as **public** or **private**



VPC and IP addressing

VPC

x.x.x.x/16 or 65,536 addresses (max)
to
x.x.x.x/28 or 16 addresses (min)

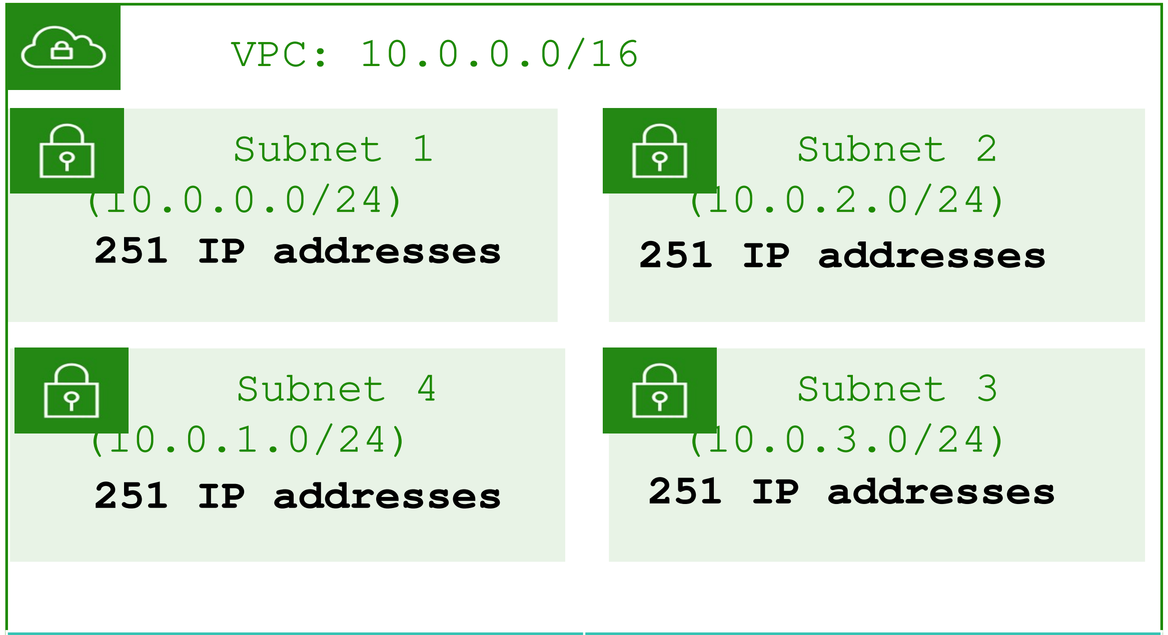
- Each created VPC
  - Must have assigned an IPv4 **CIDR block**
    - **R**ange of **private** IPv4 addresses
    - Required to be able communicate with each other or to outside
  - There is no possible to **change the address range** after the VPC is created
  - The **largest** IPv4 CIDR block size is **/16**.
  - The **smallest** IPv4 CIDR block size is **/28**.
  - IPv6 is also supported
    - with a different block size limit

- VPC
  - Can be a single subnet (the same CIDR)
  - or several subnets (subsets of CIDR)
    - CIDR blocks of subnets **cannot overlap**

Reserved IP addresses

- Each subnet requires own CIDR block
- For each CIDR five addresses are reserved
  - Network address
  - VPC local router (internal communications)
  - Domain Name System (DNS) resolution
  - Future use
  - Network broadcast address

- VPC example:
  - IPv4 CIDR block of 10.0.0.0/16 is assigned
    - Provides 65,536 total IP addresses
  - Has four equal-sized subnets
    - Fire addresses reserved
    - => only 251 IP addresses are available for each subnet
  - Every instance in VPC gets a private IP address
  - Public must be requested



IP Addresses for CIDR block 10.0.0.0/24	Reserved for
10.0.0.0	Network address
10.0.0.1	Internal communication
10.0.0.2	Domain Name System (DNS) resolution
10.0.0.3	Future use
10.0.0.255	Network broadcast address

Public and private IP address types

Public IPv4 address

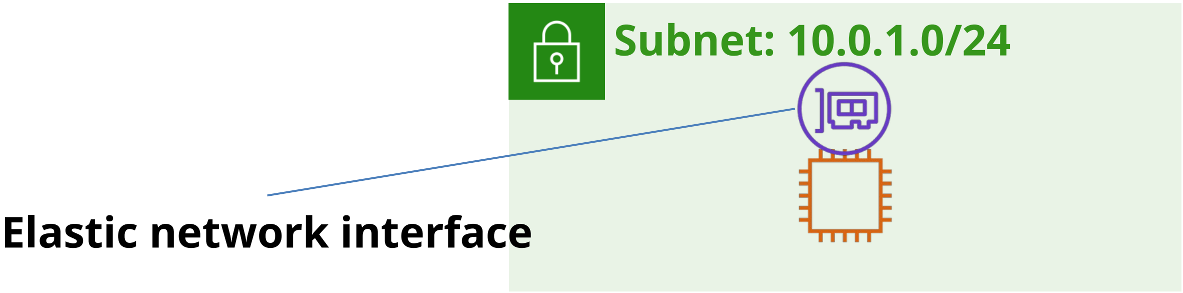
- Manually assigned through an Elastic IP address
- Automatically assigned through the auto-assign public IP address settings at the subnet level

Elastic IP address

- IS a static and public IPv4 address
- Associated with an AWS account
- Can be allocated and remapped anytime
- Additional costs might apply

Elastic network interface

- An elastic network interface is a **virtual network interface**:
  - Can be attached to an instance.
  - Can be detached from the instance, and attach to another instance to redirect network traffic.
- Its **attributes follow** when it is reattached to a new instance.
- Each instance in VPC
  - has a **default network interface** => has assigned a private IPv4 address from the IPv4 address range of your VPC.



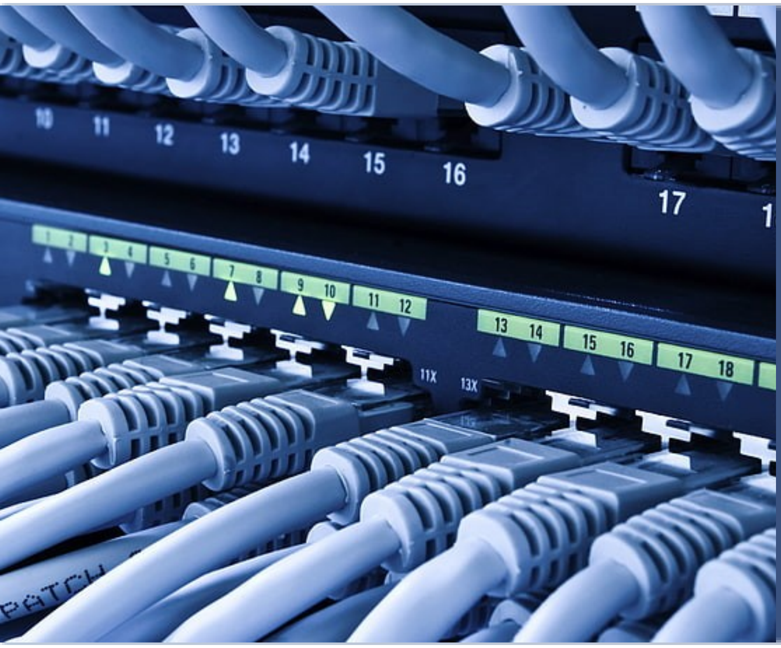
Route tables and routes

- Route table
  - Contains a set of rules (or routes)
    - Directs network traffic from your subnet
  - Each route => specifies a destination and a target
  - Contains built in **local route** for communication within the VPC (By default)
    - Cannot be deleted
  - Allows add additional routes to the table
- Each VPC subnet must be associated with a route table
  - most one at a time
- Main route table
  - Route table automatically assigned to your VPC
  - controls the routing for all subnets that are not explicitly associated with any other route table

Main (Default) Route Table

Destination	**Target**
10.0.0.0/16	local

VPC CIDR block



VPC networking

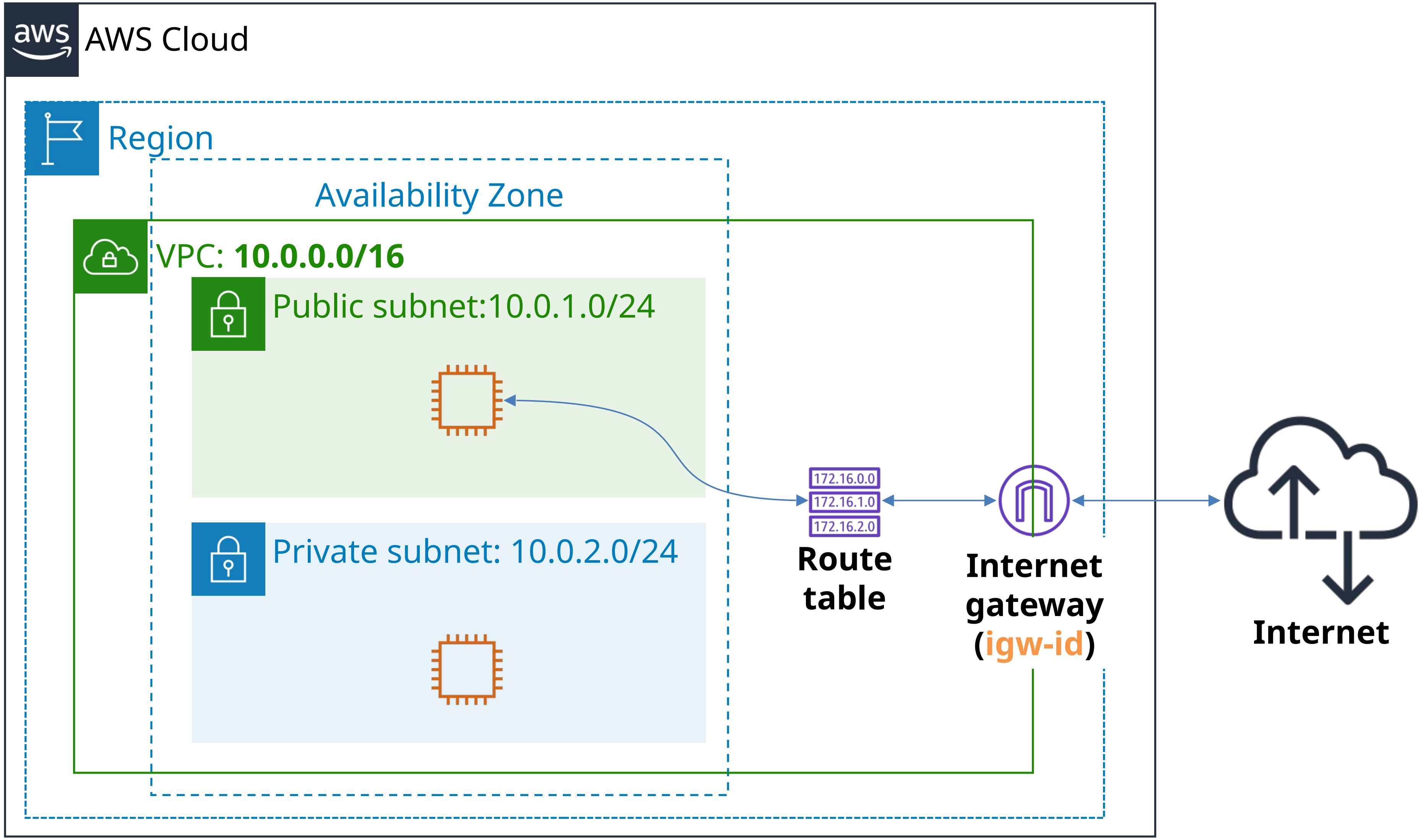
VPC networking

- Solves the question of inter networking / inter connecting
- There are several:
  - Internet gateway
  - NAT gateway
  - VPC endpoint
  - VPC peering
  - VPC sharing
  - AWS Site-to-Site VPN
  - AWS Direct Connect
  - AWS Transit Gateway
- AWS VPC Wizard => simplifies implementation

Internet gateway

- Scalable, redundant, and highly available VPC component
- Allows communication between instances in VPC
  - and the internet
- Two purposes
  - Provides target in VPC route table for internet-routable traffic (default route)
    - Connect Subnet to public net
  - Performs NAT
- How to make a subnet *PUBLIC*
  - Attach a VPC with GW
  - Add a default *route to route non-local traffic*

Internet gateway



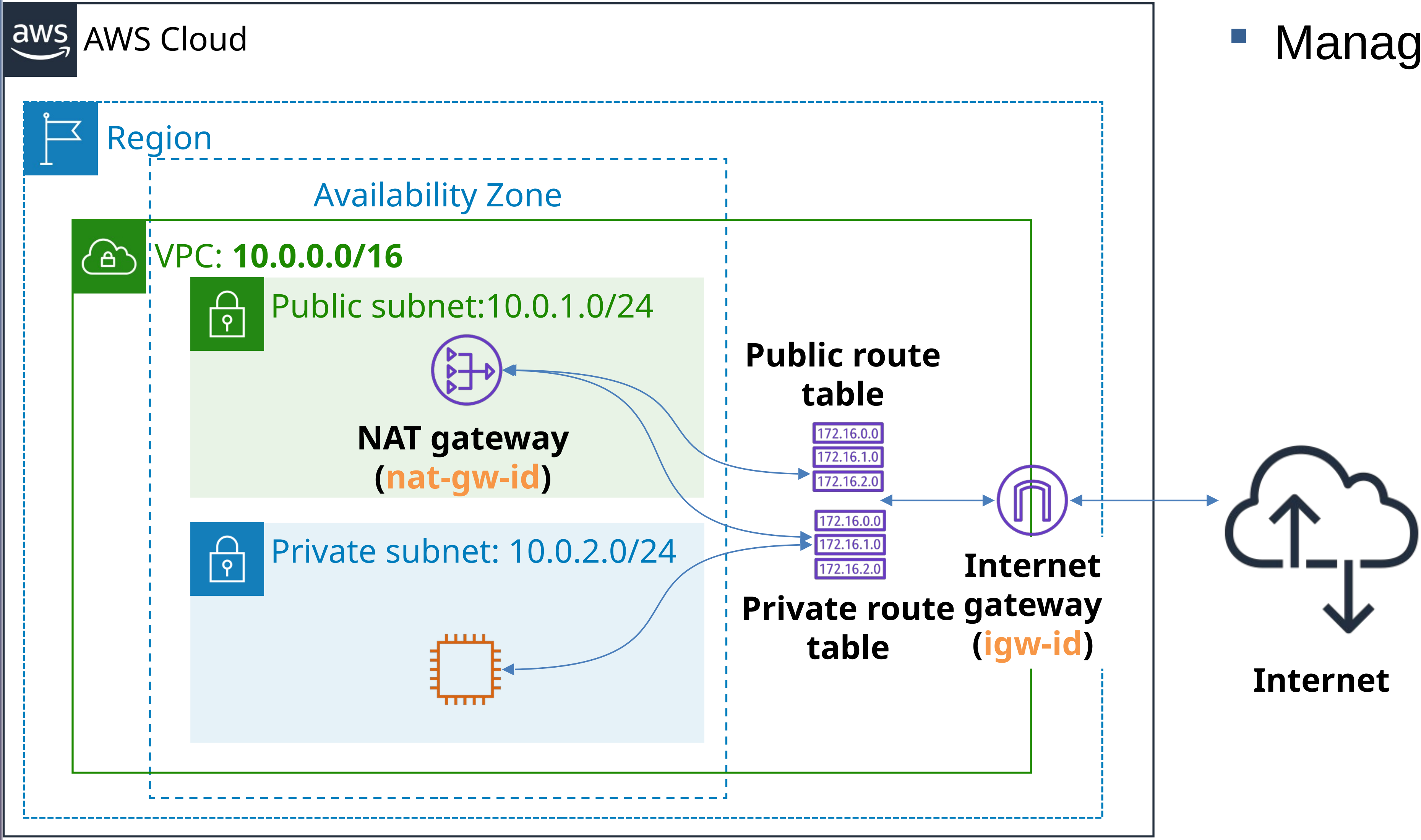
Public Subnet Route Table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-id

Network address translation (NAT) gateway

- NAT GW = Managed NAT Service
  - Outside direction
    - enables instances in a private subnet to connect to the internet
    - or connects to other AWS services,
  - Inside direction
    - prevents the internet from initiating a connection with VPC instances
  - Provides better availability, higher bandwidth, and less administrative effort
  - AWS recommend instead of a NAT instance
- NAT GW deployment
  - Exist in Public subnet
    - Must be specified to which public subnet belongs
  - Must be associated with specific Elastic IP address
  - + route table has to be updated

Network address translation (NAT) gateway



- Managet NAT service

Public Subnet Route Table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-id

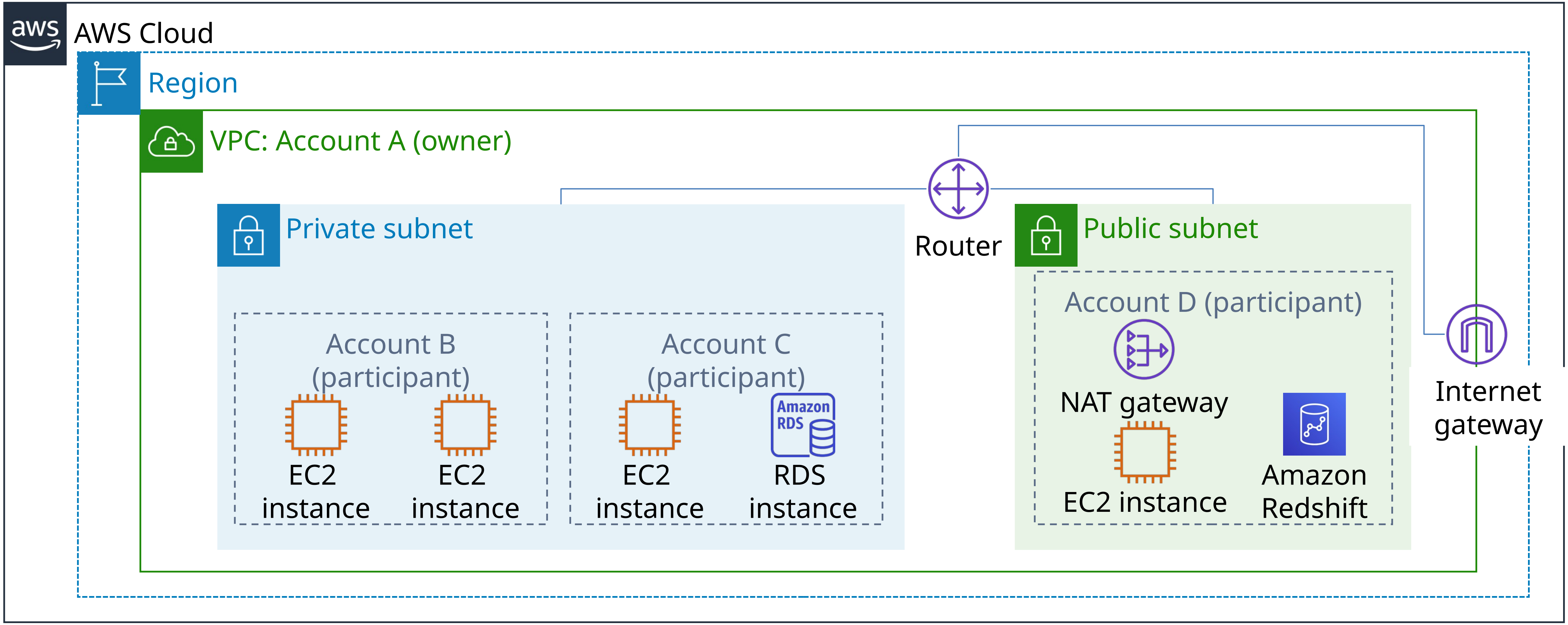
Private Subnet Route Table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	nat-gw-id

VPC sharing

- VPC Sharing model
  - the VPC owner (the account) shares one or more subnets with other AWS accounts (participants) that belong to the same organization in AWS Organizations
  - enables multiple AWS accounts to create their application resources into shared, centrally managed VPCs
    - - such as Amazon EC2 instances, Amazon Relational Database Service (Amazon RDS) databases, Amazon Redshift clusters, and AWS Lambda functions
  - Enables to create fewer, larger, centrally managed VPC
    - Suitable for highly interconnected application
- VPC sharing offers several benefits:
  - Separation of duties – Centrally controlled VPC structure, routing, IP address allocation
  - Ownership – Application owners continue to own resources, accounts, and security groups
  - Security groups – VPC sharing participants can reference the security group IDs of each other
  - Efficiencies – Higher density in subnets, efficient use of VPNs and AWS Direct Connect
  - No hard limits – Hard limits can be avoided—for example, 50 virtual interfaces per AWS Direct Connect connection through simplified network architecture
  - Optimized costs – Costs can be optimized through the reuse of NAT gateways, VPC interface endpoints, and intra-Availability Zone traffic

VPC sharing

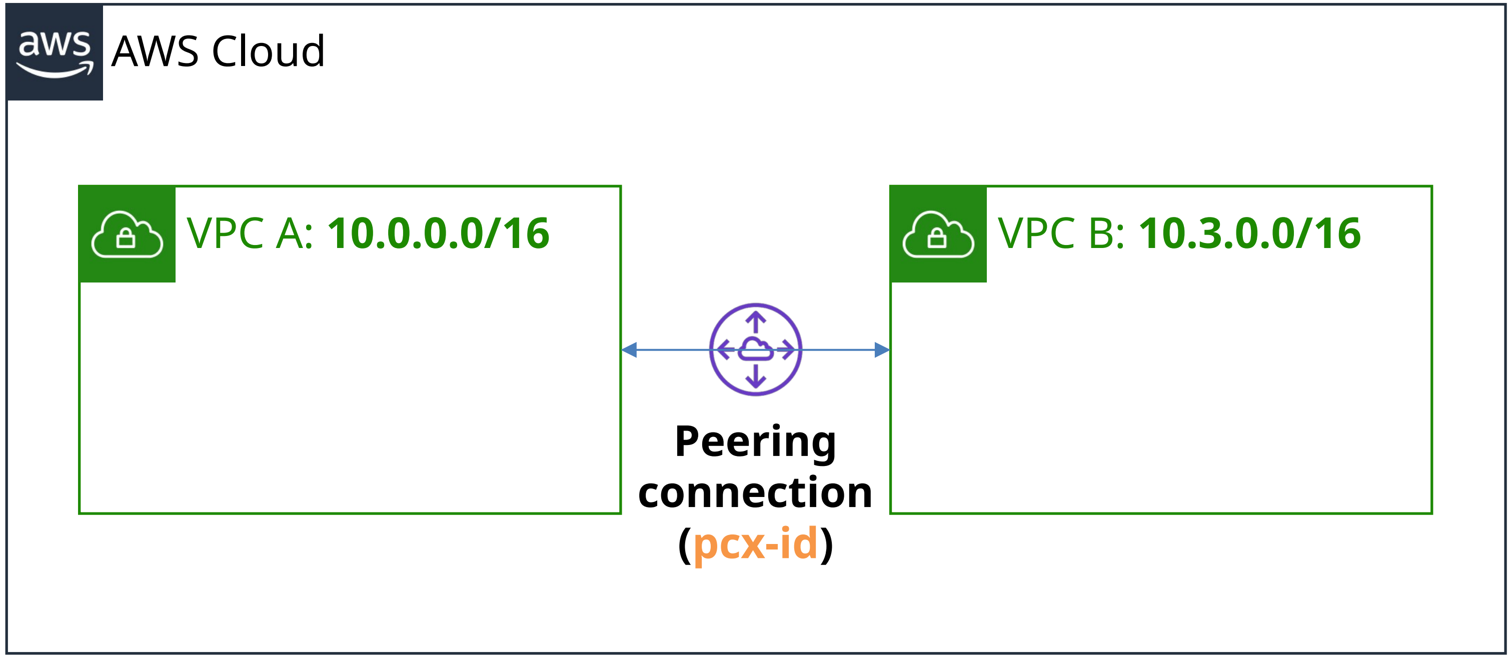


- Enables customers to share subnets with other AWS accounts in the same organization in AWS Organizations

VPC peering

- VPC peering connection is a network connection
  - Allows route traffic between VPC privately
    - To route traffic => requires route table modification
- Supported connections
  - between two VPCs in your own AWS account
  - between AWS accounts,
  - or between AWS Regions.
- Restrictions:
  - IP spaces cannot overlap.
  - Transitive peering is not supported.
  - You can only have one peering resource between the same two VPCs.

VPC peering



Route Table for VPC A

Destination	Target
10.0.0.0/16	local
10.3.0.0/16	pcx-id

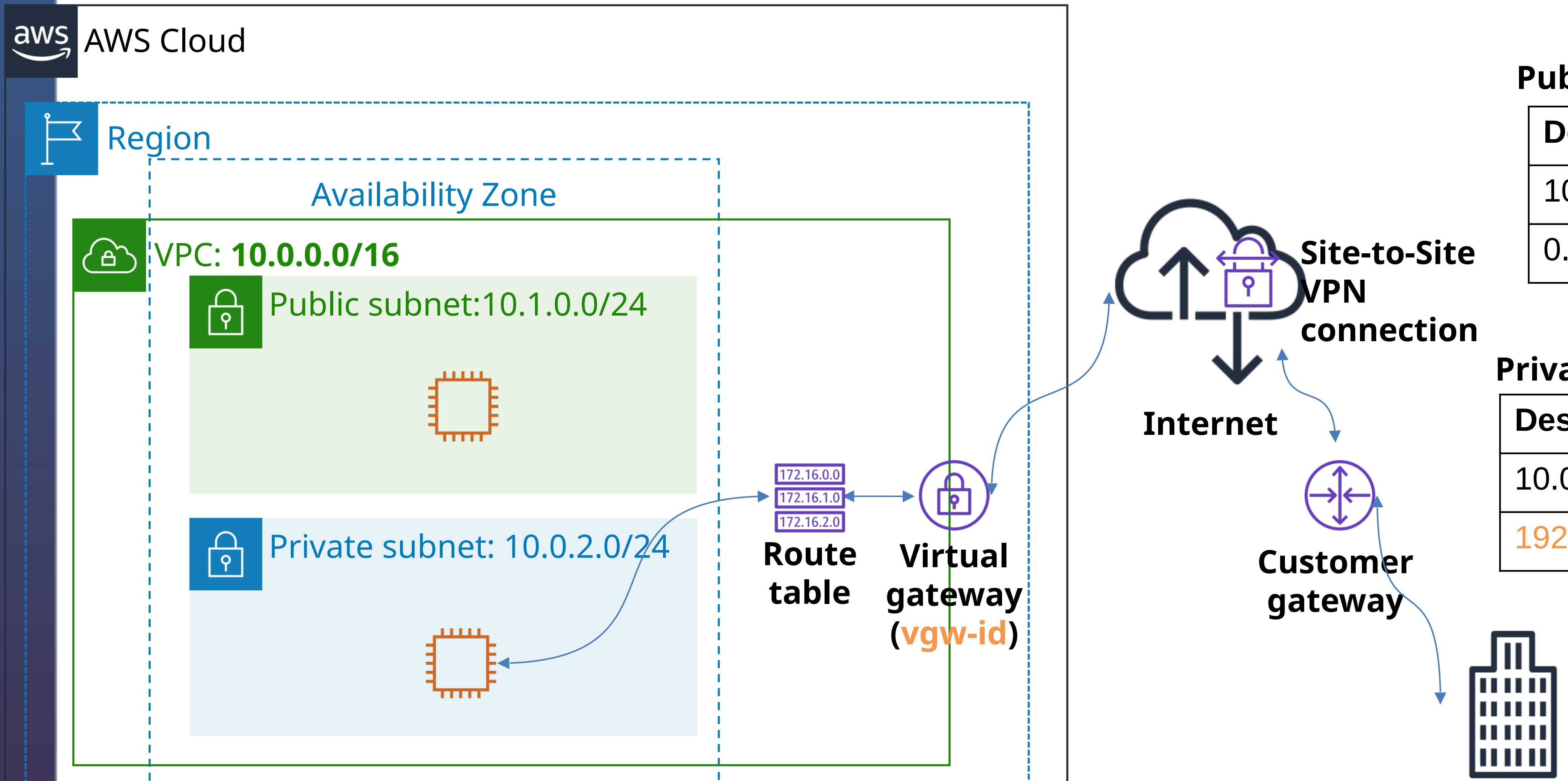
Route Table for VPC B

Destination	Target
10.3.0.0/16	local
10.0.0.0/16	pcx-id

AWS Site-to-Site VPN – connection to a remote network

- Default VPC behavior
  - instances into a VPC cannot communicate with a remote network.
- Solution => create a virtual private network or VPN connection)
  1. Create a new ***Virtual gateway device*** (called a virtual private network (VPN) gateway) and attach it to your VPC.
  2. Define the configuration of the VPN device or ***the customer gateway***.
     1. Customer gateway = an AWS resource that provides information to AWS about your VPN device.
  3. Create a **custom route table** to point corporate data center-bound traffic to the VPN gateway. You also must update security group rules.
  4. Establish an AWS Site-to-Site VPN (Site-to-Site VPN) connection to **link the two systems together.**
  5. **Configure routing** to pass traffic through the connection.

AWS Site-to-Site VPN – connection to a remote network



Corporate data center: 192.168.10.0/24

Public subnet route table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-id

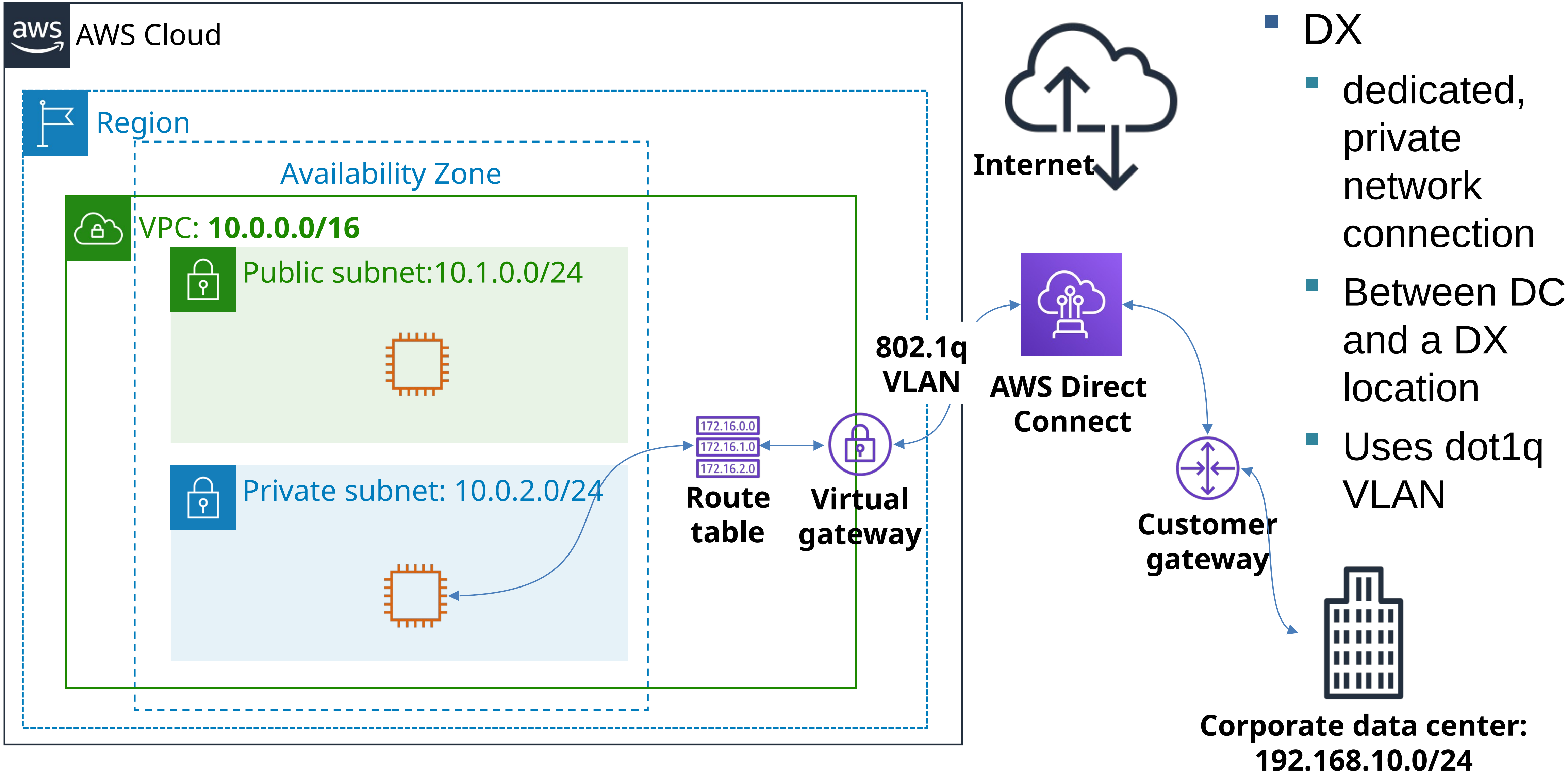
Private subnet route table

Destination	Target
10.0.0.0/16	local
192.168.10.0/24	vgw-id

AWS Direct Connect (DX)

- If a customer Data center resides far away from AWS region
  - Problem of poor net performance
- Solution = Direct connect (DX)
  - Solves the question of network performance
  - enables to establish a dedicated, private network connection between customer on-premise network and one of the DX locations
    - Benefits
      - reduce your network costs,
      - increase bandwidth throughput,
      - provide a more consistent network experience than internet-based connections.
  - Uses open standard 802.1q virtual local area networks (VLANs).

AWS Direct Connect (DX)

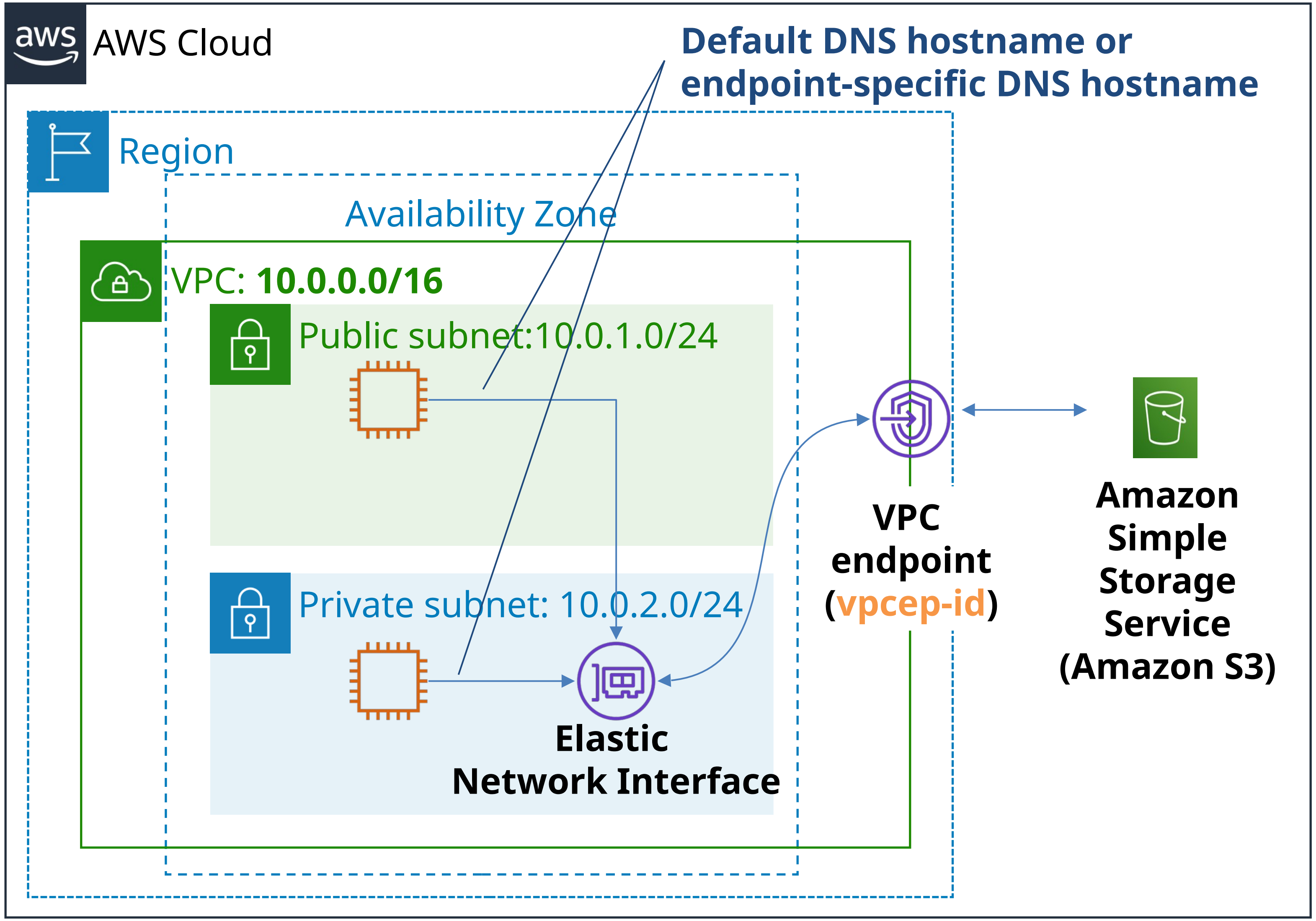


- DX
  - dedicated, private network connection
  - Between DC and a DX location
  - Uses dot1q VLAN

VPC endpoints

- A *VPC endpoint*
  - a virtual device that enables to privately connect VPC to supported AWS services and VPC endpoint services that are powered by AWS PrivateLink.
    - Connection does not require an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection.
    - Instances in VPC do not require public IP addresses to communicate
    - Traffic does not leave the Amazon network
- Two types of endpoints:
  - Interface endpoints
    - Connects to services powered by AWS PrivateLink
    - i.e. connects Service consumer (AWS user) to Service provider
  - Gateway endpoints
    - Amazon S3 and Amazon DynamoDB

VPC endpoints


AWS Cloud
Default DNS hostname or endpoint-specific DNS hostname
Region
Availability Zone
VPC: 10.0.0.0/16
Public subnet:10.0.1.0/24
Private subnet: 10.0.2.0/24
Elastic Network Interface
VPC endpoint (vpcep-id)
Amazon Simple Storage Service (Amazon S3)

Public Subnet Route Table

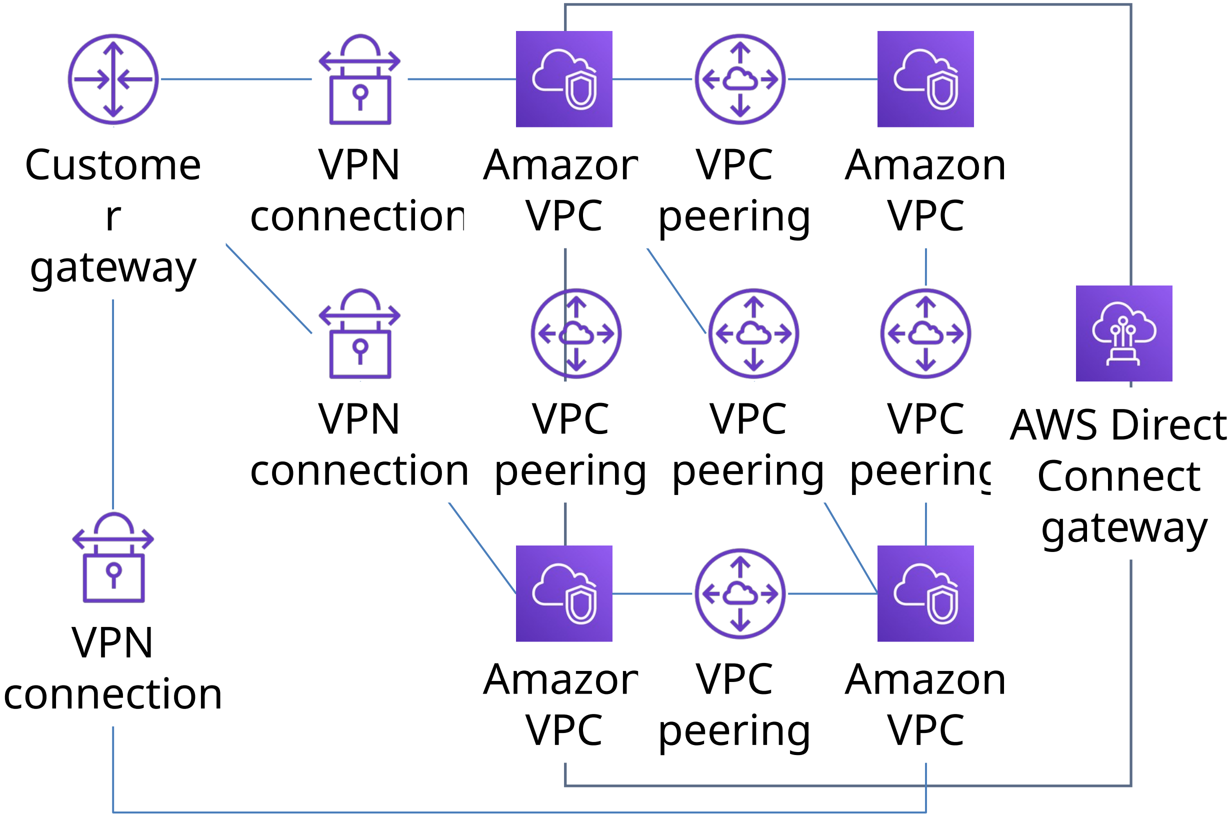
Destination	Target
10.0.0.0/16	local
Amazon S3 ID	vpcep-id

AWS Transit Gateway

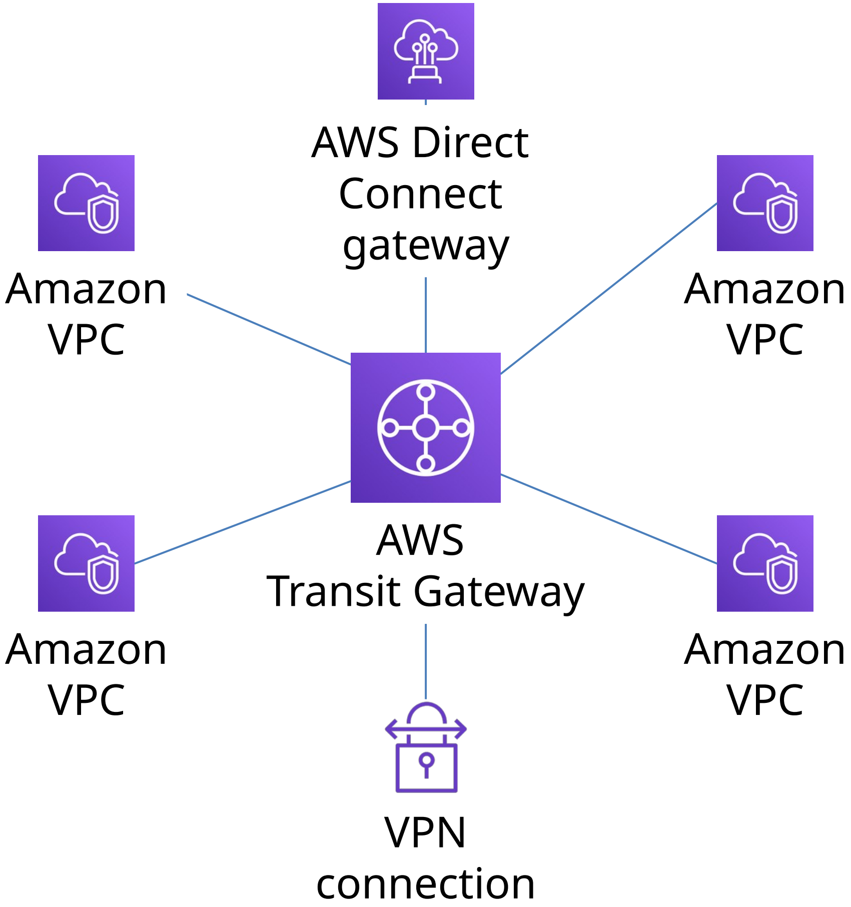
- AWS Transit Gateway
  - Hub and Spoke approach, that simplifies VPC networking model
    - Acts as a hub that controls how traffic is routed among all the connected networks, which act like spokes.
  - Allows to create and manage a single connection
    - from the central gateway into each VPC, on-premises data center, or remote office across your network.

AWS Transit Gateway usage

From this…

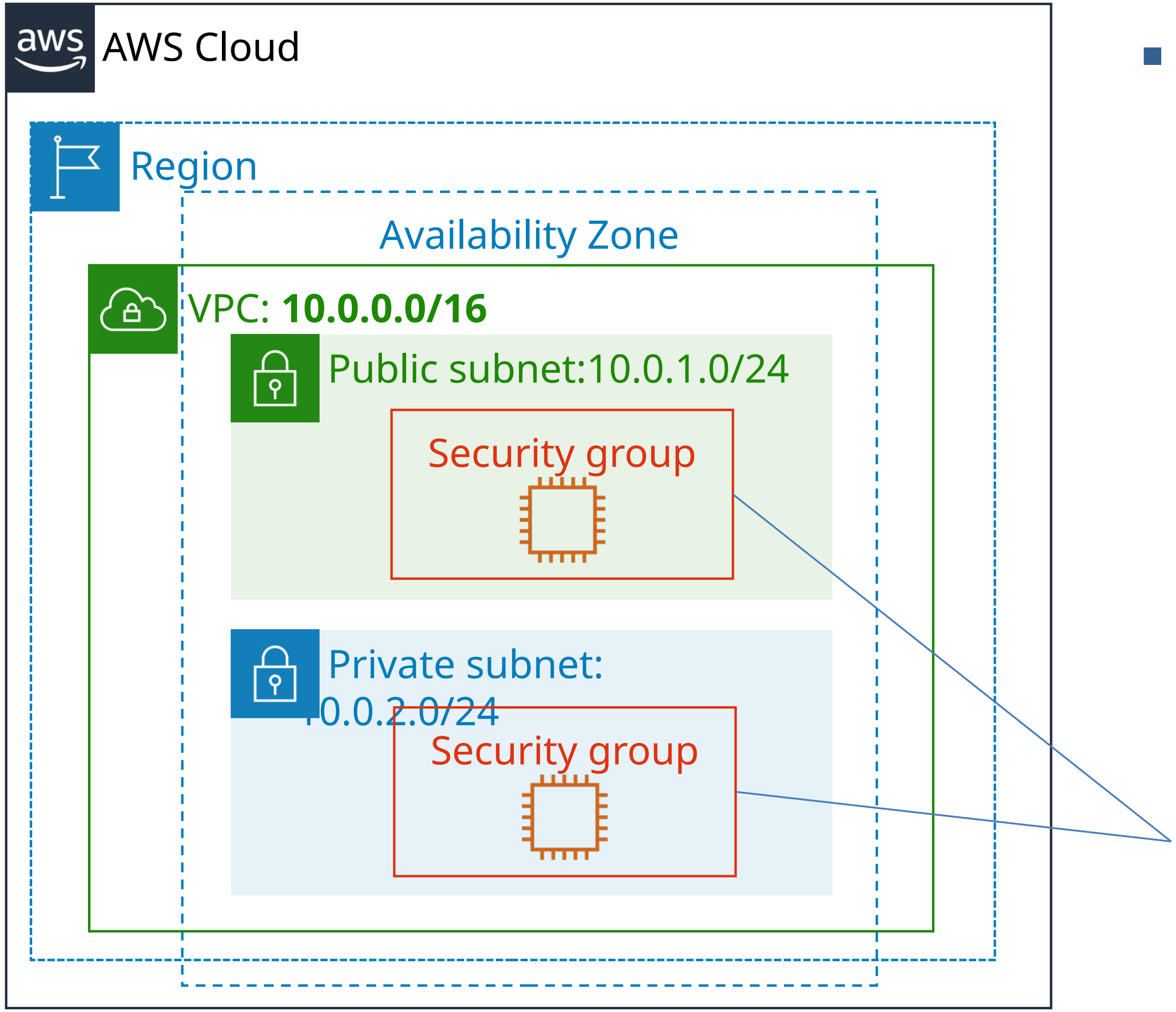


To this…



VPC security

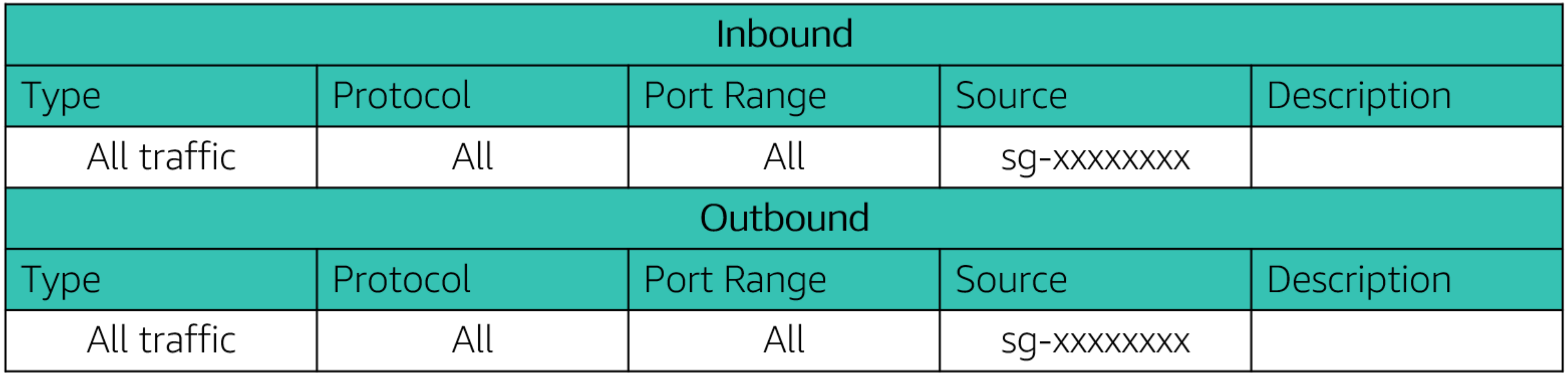
Security groups



- Security groups = Virtual Firewall for a instance
  - Controls Inbound/Outbound traffic
  - Work at the instance level
    - Not for the whole subnet
  - Each instance can have assigned different sets of security group

Security groups act at the instance level

Security groups rules

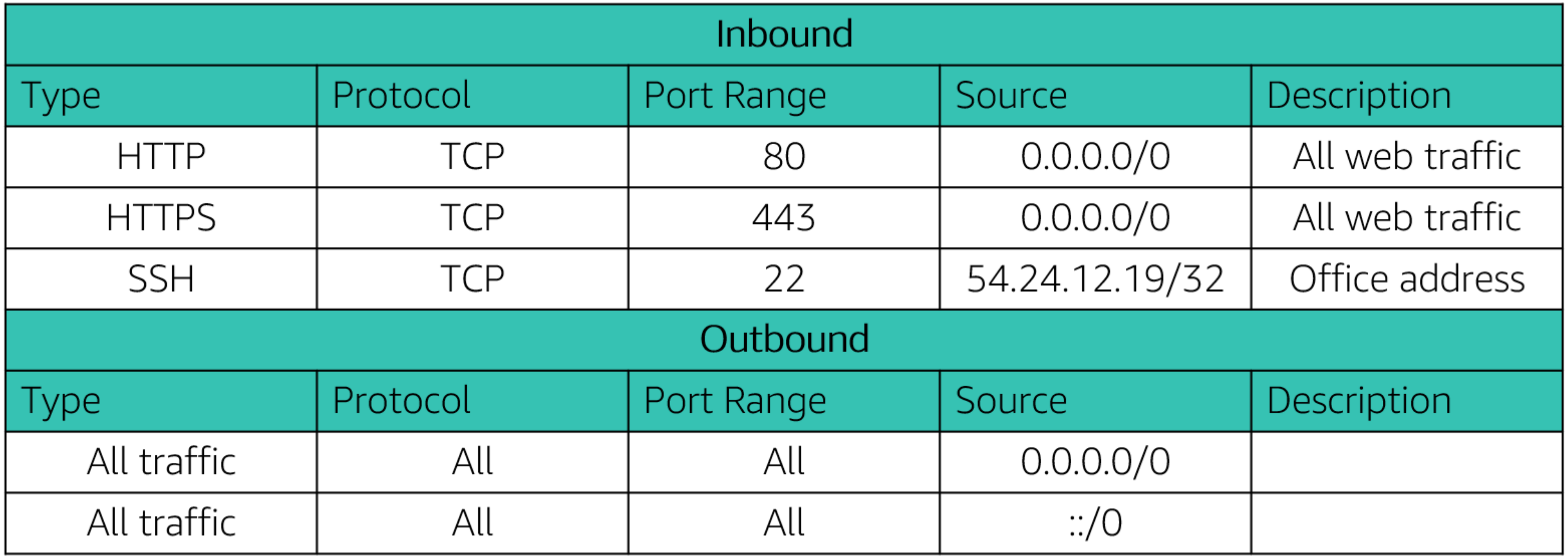


<table>
<tr><th colspan="5">Inbound</th></tr>
<tr><th>Type</th><th>Protocol</th><th>Port Range</th><th>Source</th><th>Description</th></tr>
<tr><td>All traffic</td><td>All</td><td>All</td><td>sg-xxxxxxxx</td><td></td></tr>
<tr><th colspan="5">Outbound</th></tr>
<tr><th>Type</th><th>Protocol</th><th>Port Range</th><th>Source</th><th>Description</th></tr>
<tr><td>All traffic</td><td>All</td><td>All</td><td>sg-xxxxxxxx</td><td></td></tr>
</table>

- Security groups
  - have rules that control inbound and outbound instance traffic.
  - Default security groups
    - deny all inbound traffic => No one can connect
    - allow all outbound traffic.
- Security groups are stateful

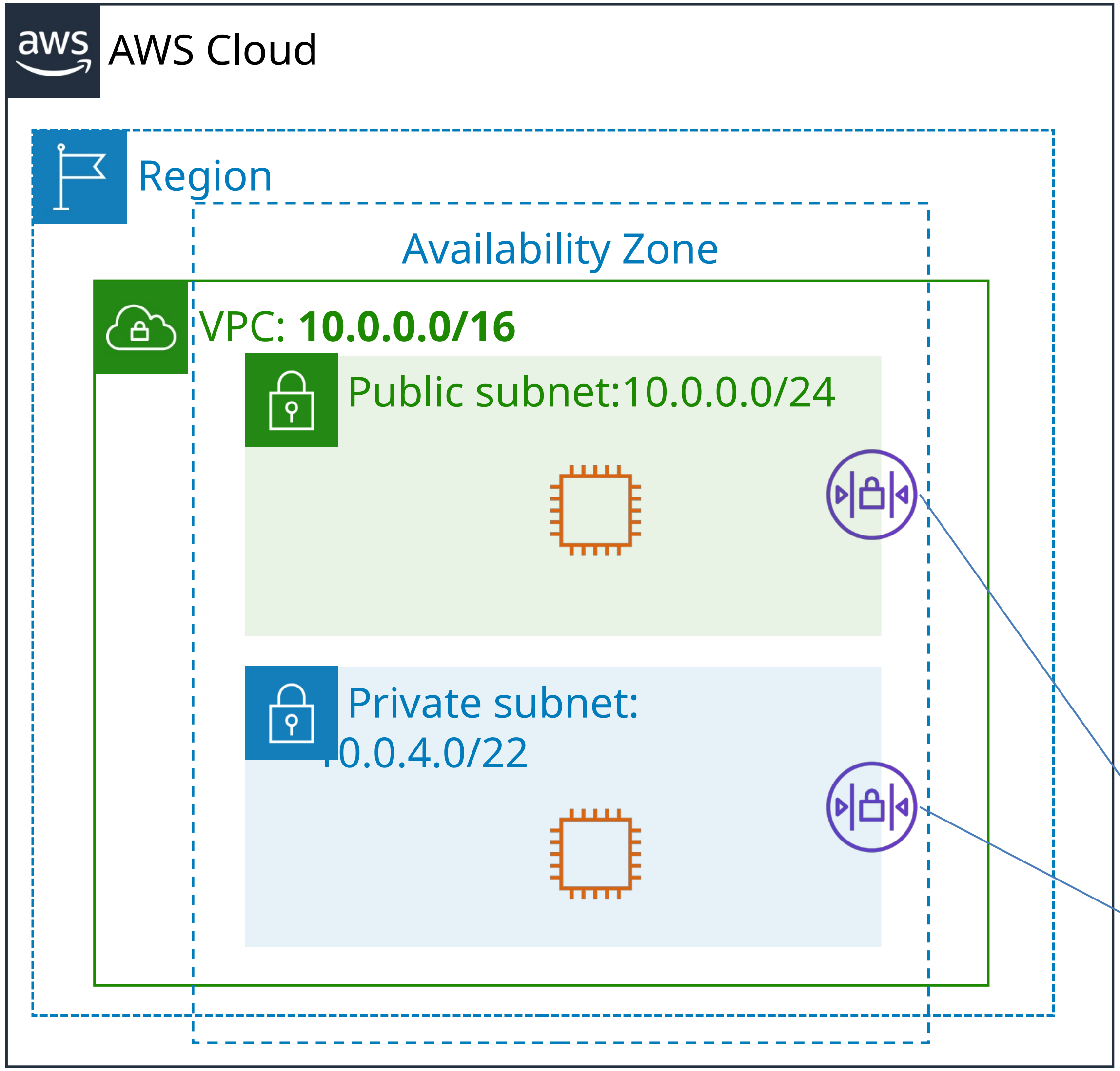
Custom security groups

- Allows to specify
  - allow (permit) rules, but not deny rules.
- All rules are evaluated before the decision to allow traffic.



Inbound				
Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	All web traffic
HTTPS	TCP	443	0.0.0.0/0	All web traffic
SSH	TCP	22	54.24.12.19/32	Office address
Outbound				
Type	Protocol	Port Range	Source	Description
All traffic	All	All	0.0.0.0/0	
All traffic	All	All	::/0	

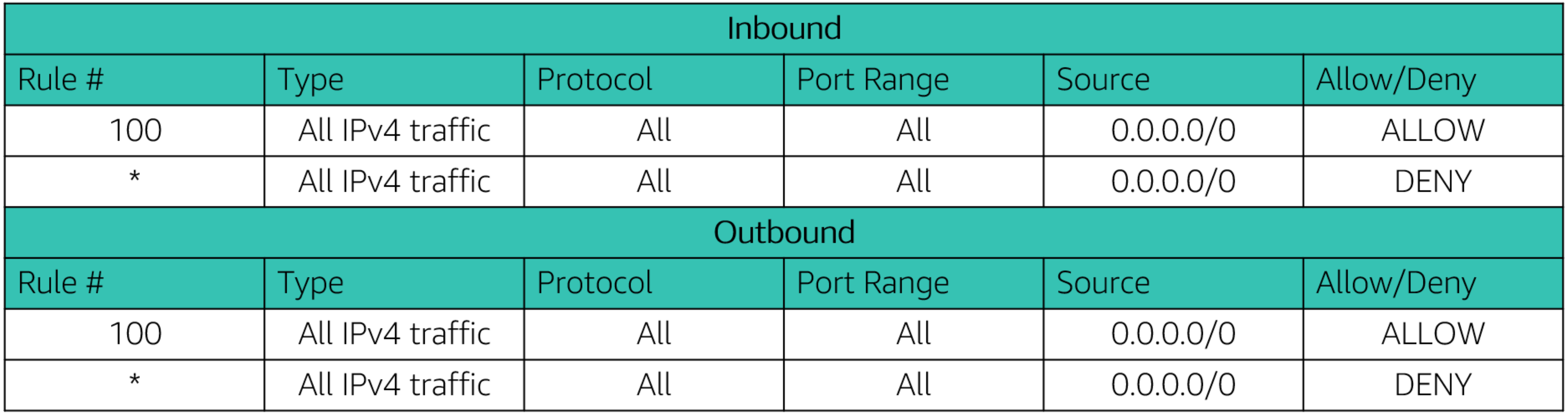
Network access control lists (network ACLs)



- Optional layer of security for VPC
- Like a subnet Firewall
  - Control in/out traffic for subnet/-s
  - Each subnet must be associated with a Net ACL, but the only one
    - Otherwise associated to Default Network ACL

Network ACLs act at the subnet level.

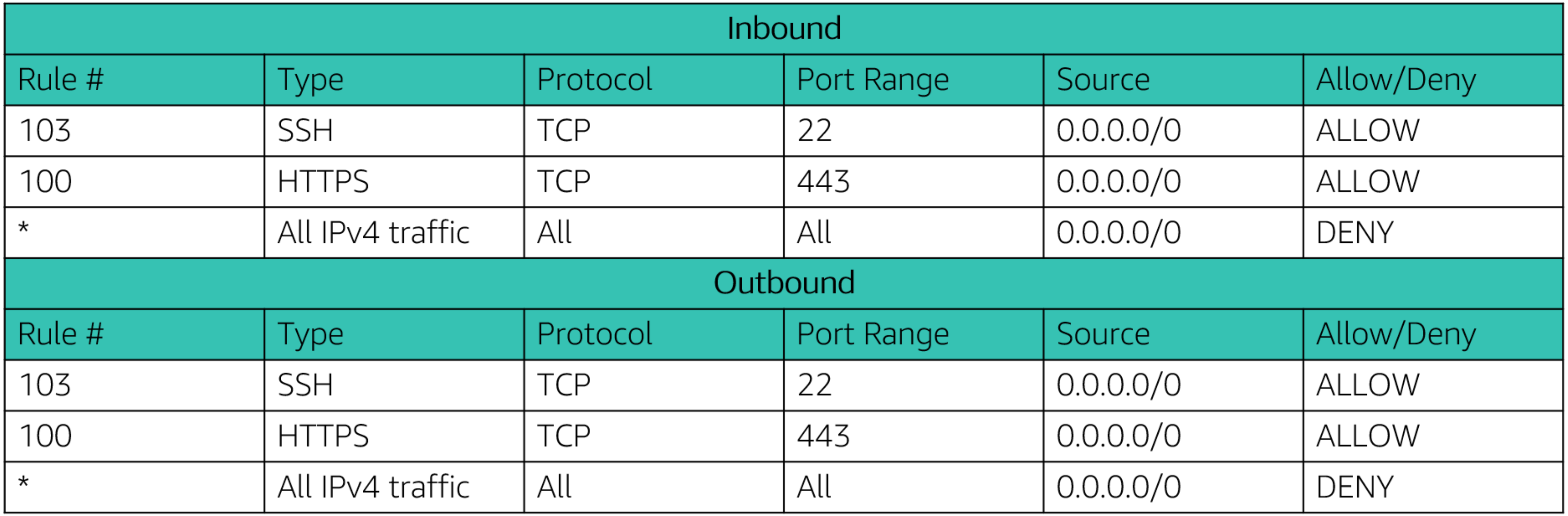
Network ACLs



Inbound					
Rule #	Type	Protocol	Port Range	Source	Allow/Deny
100	All IPv4 traffic	All	All	0.0.0.0/0	ALLOW
*	All IPv4 traffic	All	All	0.0.0.0/0	DENY
Outbound					
Rule #	Type	Protocol	Port Range	Source	Allow/Deny
100	All IPv4 traffic	All	All	0.0.0.0/0	ALLOW
*	All IPv4 traffic	All	All	0.0.0.0/0	DENY

- Network ACL
  - has separate inbound and outbound rules,
  - each rule either allow or deny traffic
  - Is **stateless**
- Default network ACLs
  - Allow all inbound and outbound IPv4 traffic.

Custom network ACLs



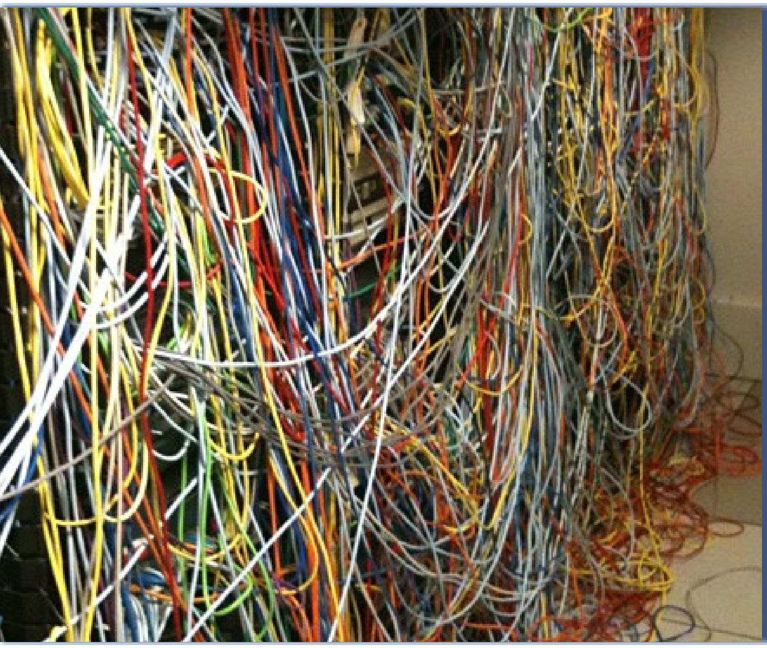
Inbound					
Rule #	Type	Protocol	Port Range	Source	Allow/Deny
103	SSH	TCP	22	0.0.0.0/0	ALLOW
100	HTTPS	TCP	443	0.0.0.0/0	ALLOW
*	All IPv4 traffic	All	All	0.0.0.0/0	DENY
Outbound					
Rule #	Type	Protocol	Port Range	Source	Allow/Deny
103	SSH	TCP	22	0.0.0.0/0	ALLOW
100	HTTPS	TCP	443	0.0.0.0/0	ALLOW
*	All IPv4 traffic	All	All	0.0.0.0/0	DENY

- Custom network ACL
  - Deny all inbound and outbound traffic until you add rules.
  - Allows to specify both allow and deny rules.

- Contains numbered list of rules
  - Rules are evaluated in number order, starting with the lowest number.
  - Max 32,766
  - Add rules in increments of 10/100

Security groups versus network ACLs

Attribute	Security Groups	Network ACLs
Scope	Instance level	Subnet level
Supported Rules	Allow rules only	Allow and deny rules
State	Stateful (return traffic is automatically allowed, regardless of rules)	Stateless (return traffic must be explicitly allowed by rules)
Order of Rules	All rules are evaluated before decision to allow traffic	Rules are evaluated in number order before decision to allow traffic

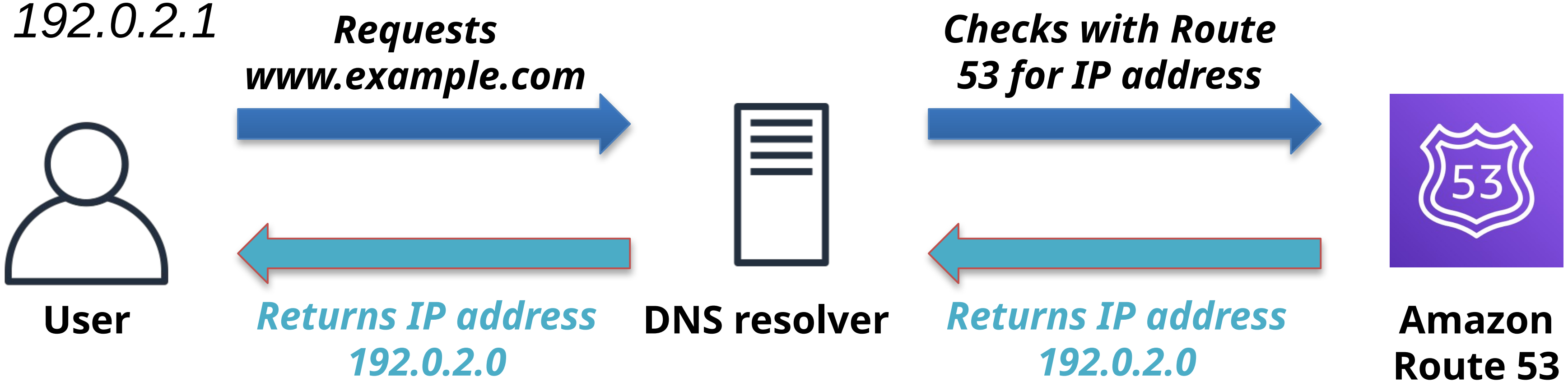


Amazon Route 53

Amazon Route 53

Amazon Route 53

- Highly available and scalable Domain Name System (DNS) web service
  - translates names (like www.example.com) into numeric IP addresses (like *192.0.2.1*



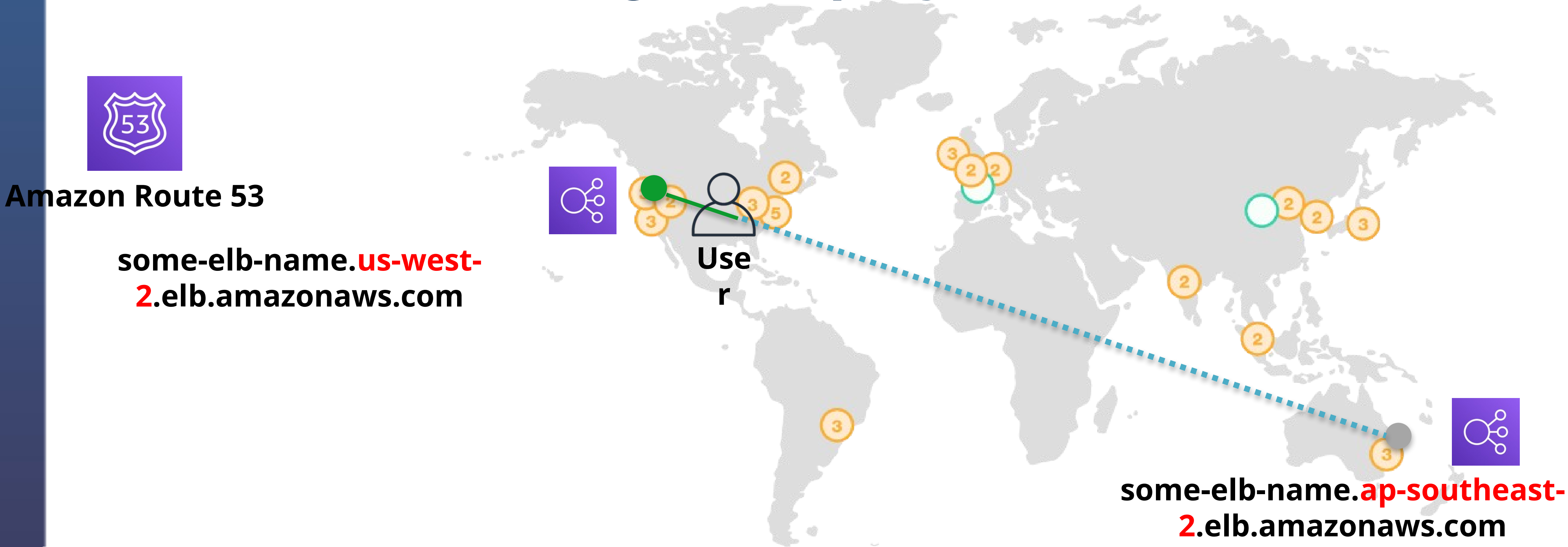
Requests www.example.com
Checks with Route 53 for IP address
User
Returns IP address 192.0.2.0
DNS resolver
Returns IP address 192.0.2.0
Amazon Route 53

- Fully compliant with IPv4 and IPv6
- Connects user requests to infrastructure running in AWS and also outside of AWS
- Used to check the health of your resources
- Enables to register domain names

Amazon Route 53 supported routing policies

- Traffic flow manipulation:
  - Simple routing – Use in single-server environments
  - Weighted round robin routing – Assign weights to resource record sets to specify the frequency
  - Latency routing – Help improve your global applications
  - Geolocation routing – Route traffic based on location of your users
  - Geoproximity routing – Route traffic based on location of your resources
  - Failover routing – Fail over to a backup site if your primary site becomes unreachable
  - Multivalue answer routing – Respond to DNS queries with up to eight healthy records selected at random

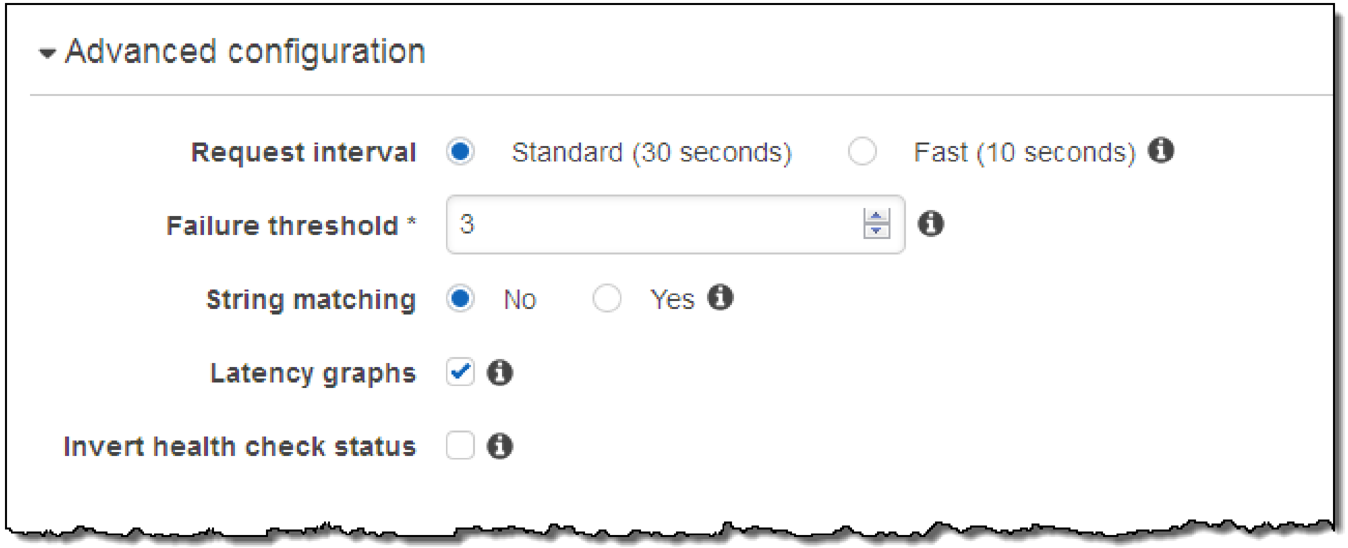
Use case: Multi-region deployment


Amazon Route 53
some-elb-name.us-west-2.elb.amazonaws.com
User
some-elb-name.ap-southeast-2.elb.amazonaws.com
3
2
2
3
2
2
3
5
3
2
2
3
2
2
3
3

Name	Type	Value
example.com	ALIAS	some-elb-name.us-west-2.elb.amazonaws.com
example.com	ALIAS	some-elb-name.ap-southeast-2.elb.amazonaws.com

Amazon Route 53 DNS failover

- Improve the availability of your applications that run on AWS by:
  - Configuring backup and failover scenarios for your own applications
  - Enabling highly available multi-region architectures on AWS
  - Creating health checks



▾ Advanced configuration

Request interval ● Standard (30 seconds) ○ Fast (10 seconds)

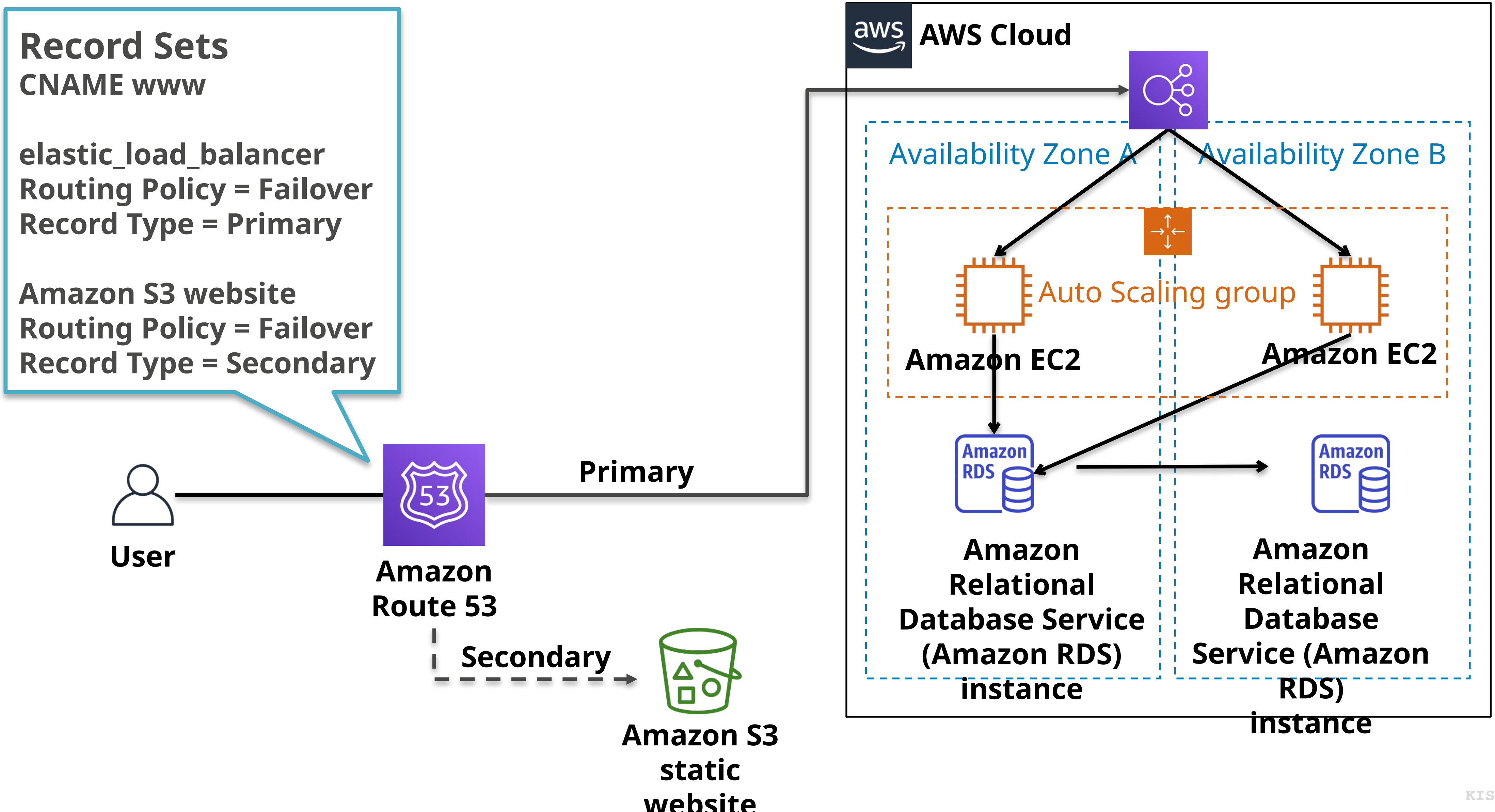
Failure threshold * 3

String matching ● No ○ Yes

Latency graphs ☑

Invert health check status □

DNS failover for a multi-tiered web application

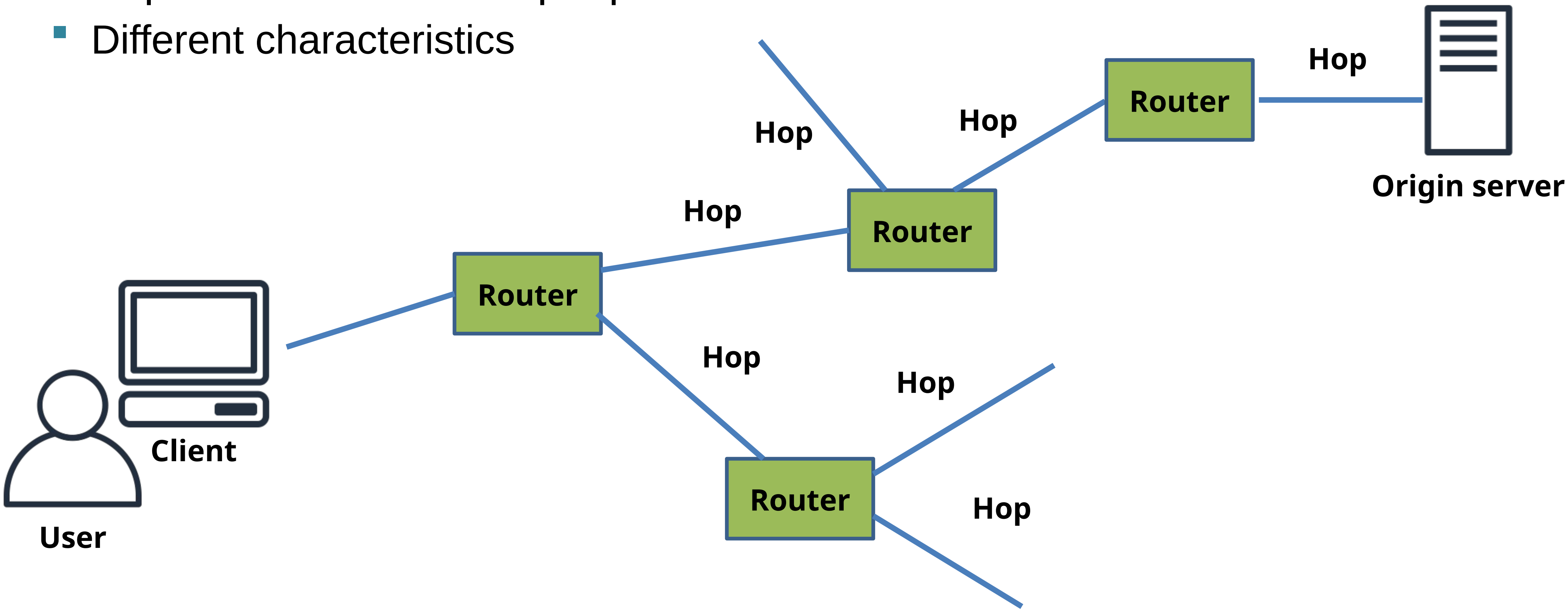


Ak chcete pridať obrázok, kliknite na ikonu

Amazon CloudFront

Content delivery and network latency

- Multiple networks = multiple paths
  - Different characteristics



Content delivery network (CDN) – what is it?

- CDN
  - Is a globally distributed system of caching servers
  - Accelerates delivery of dynamic content
  - Improves application performance and scaling
    - Caches copies of commonly requested files (static content)
    - Delivers a local copy of the requested content from a nearby cache edge or Point of Presence

Amazon CloudFront

Amazon CloudFront

- Fast, global, and secure CDN service
- Global network of edge locations and Regional edge caches
- Self-service model
- Pay-as-you-go pricing

Amazon CloudFront benefits

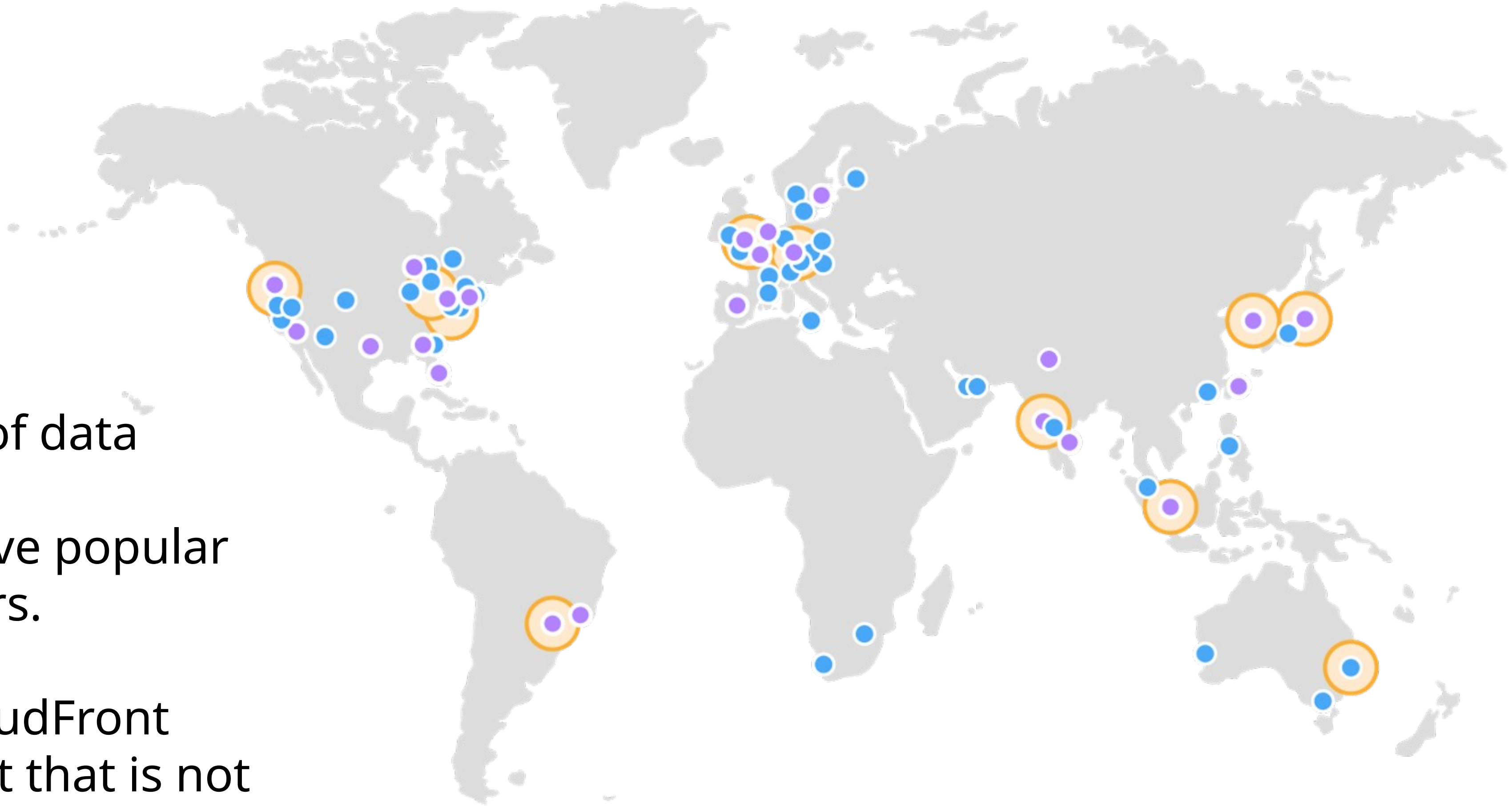
- Fast and global
- Security at the edge
- Highly programmable
- Deeply integrated with AWS
- Cost-effective

Amazon CloudFront infrastructure

● Edge locations

● Multiple edge locations

● Regional edge caches



- **Edge locations –** Network of data centers
that CloudFront uses to serve popular content quickly to customers.

- **Regional edge cache –** CloudFront location that caches content that is not popular enough to stay at an edge location.
It is located between the origin server

Amazon CloudFront pricing

Data transfer out

- Charged for the volume of data transferred out from Amazon CloudFront edge location to the internet or to your origin.

HTTP(S) requests

- Charged for number of HTTP(S) requests.

Invalidation requests

- No additional charge for the first 1,000 paths that are requested for invalidation each month. Thereafter, $0.005 per path that is requested for invalidation.

Dedicated IP custom SSL

- $600 per month for each custom SSL certificate that is associated with one or more CloudFront distributions that use the Dedicated IP version of custom SSL certificate support.

UNIVERSITY OF ŽILINA
Faculty of Management Science
and Informatics

MINISTERSTVO
ŠKOLSTVA, VEDY,
VÝSKUMU A ŠPORTU
SLOVENSKEJ REPUBLIKY

Thank you for your attention.

The content was chapter from AWS Foundations Modules

- AWS M5 - AWS Global Infrastructure Overview

Vytvorené v rámci projektu **KEGA 026TUKE-4/2021**