DevKTOps

DevKTOps

ARCHITECTING ON aws

Module 6: Network Layer II

Module Overeiw

- Connecting Networks
- VPC Endpoints

Virtual Private Gateway (VGW)

Enable you to establish private connections (VPNs) between an Amazon VPC and another network.

- AWS supports internet protocol security(IPSec) VPN connections.
- A VGW is the VPN concentrator on the Amazon side of the VPN connection. You create a VGW and attach it to the VPC from which you want to create the VPN connection.

AWS Site-to-Site VPN

AWS Site-to-Site is a highly available solution that enables you to securely connect your on-premises network or branch offices to your VPC.

- Use IPSec to create encrypted virtual private network tunnels.
- Provides two encrypted tunnels per VPN connection
- Charged per VPN connection hourly

Routing in S2S

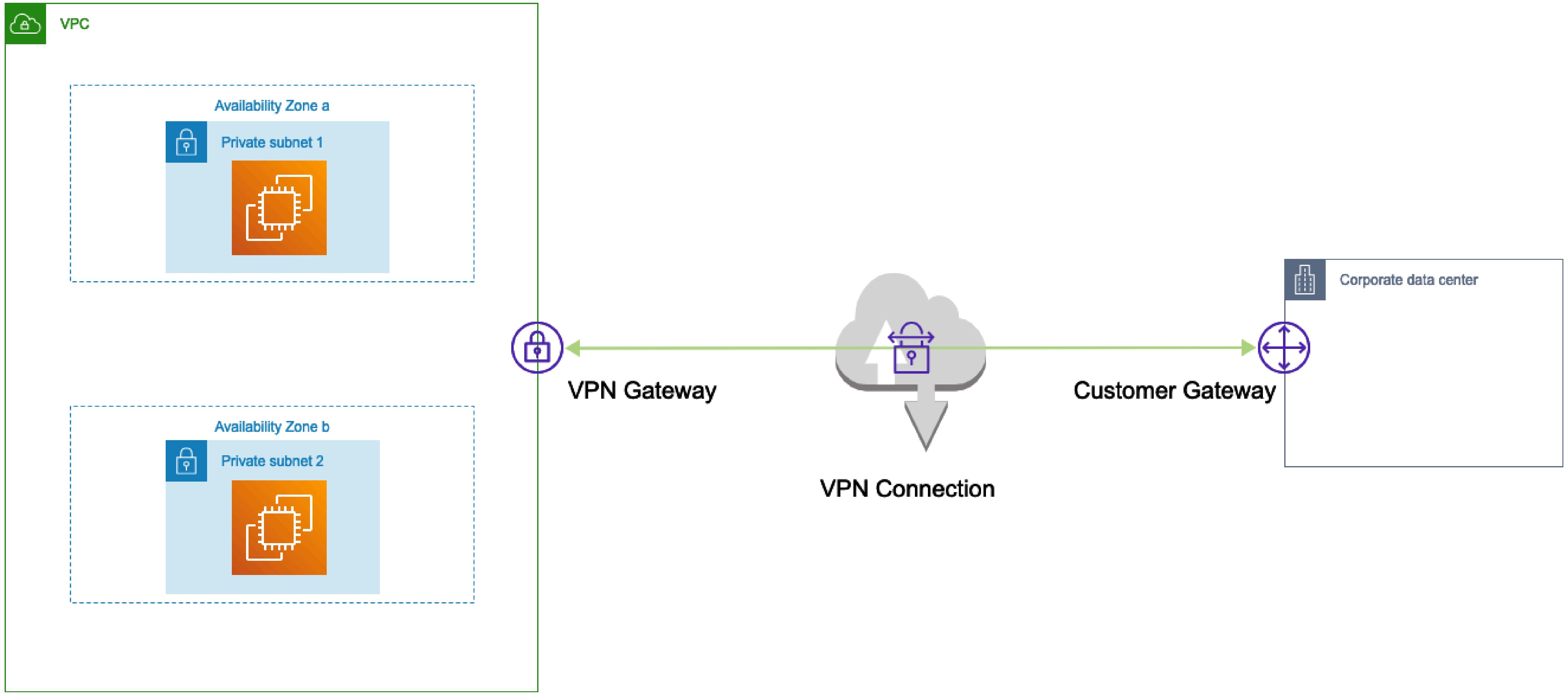
Static Routing

- Requires you to specify all routes (IP prefixes)
- Specify static routing if your customer gateway device does not support BGP

Dynamic Routing

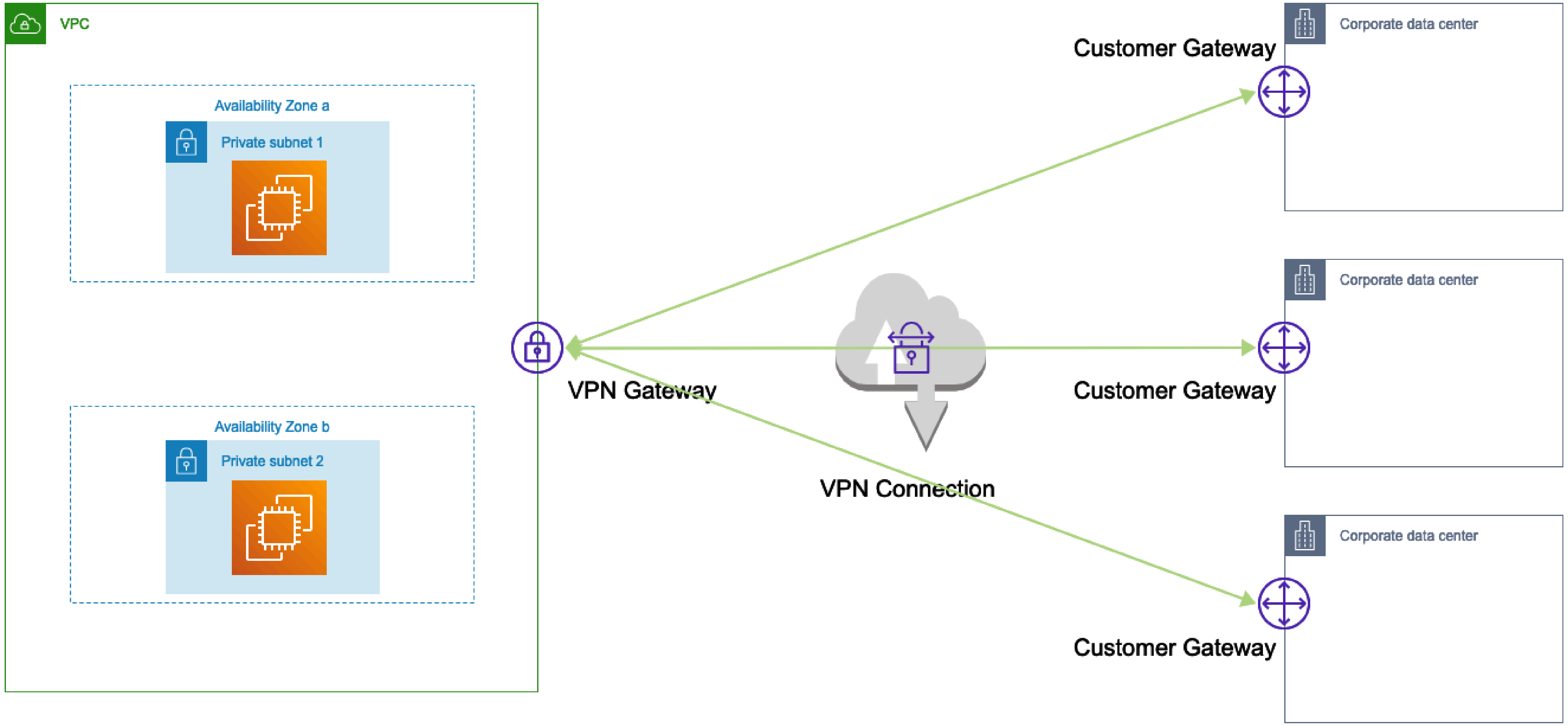
- Uses the Border Gateway Protocol (BGP) to advertise its routes to the virtual private gateway
- Specify dynamic routing if your customer gateway device supports BGP

VPN Connections (Site to Site)



Multi VPN Connections

DevKTOps



AWS Direct Connect (DX)

AWS Direct Connect(DX) provides you with a dedicated, private network connection of either 1 or 10 Gbps.

Reduces data transfer costs

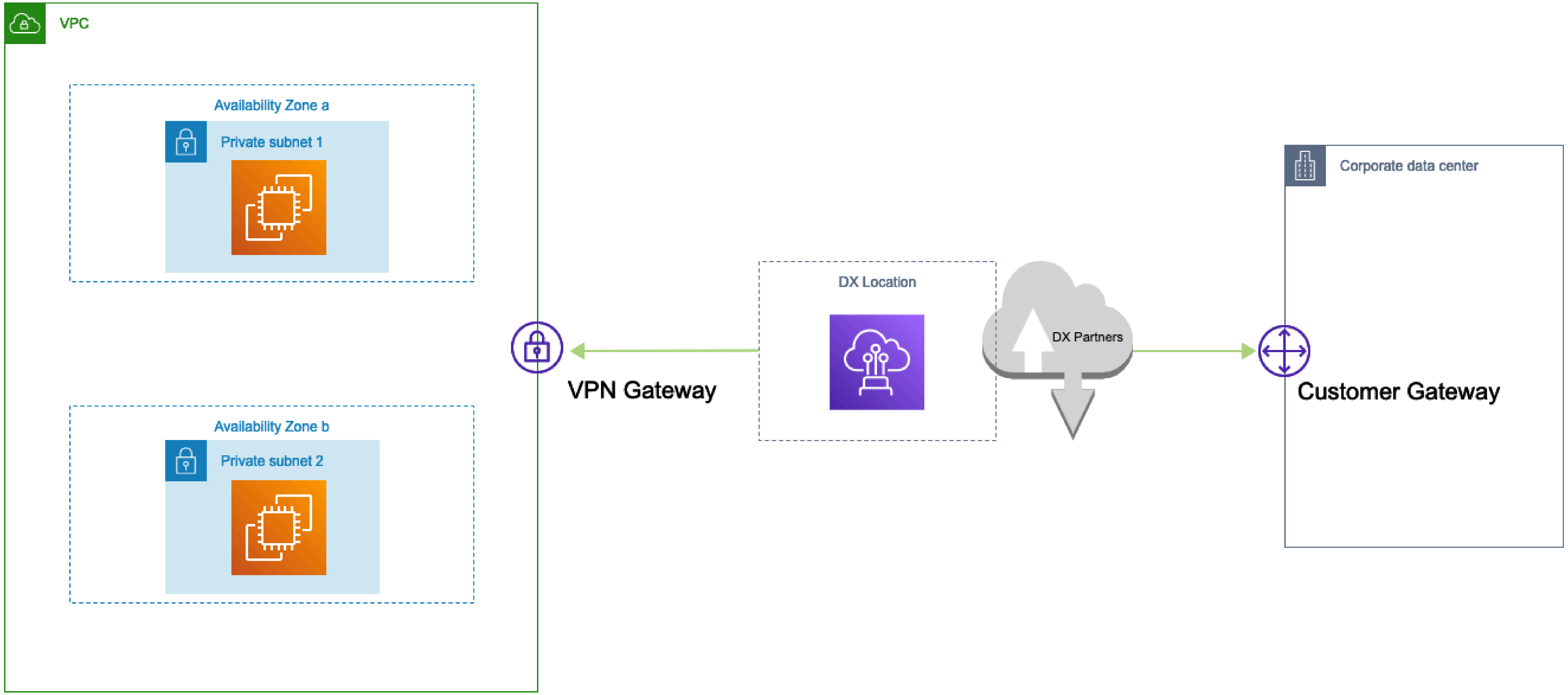
Improve application performance with predictable metrics

AWS Direct Connect Use Cases

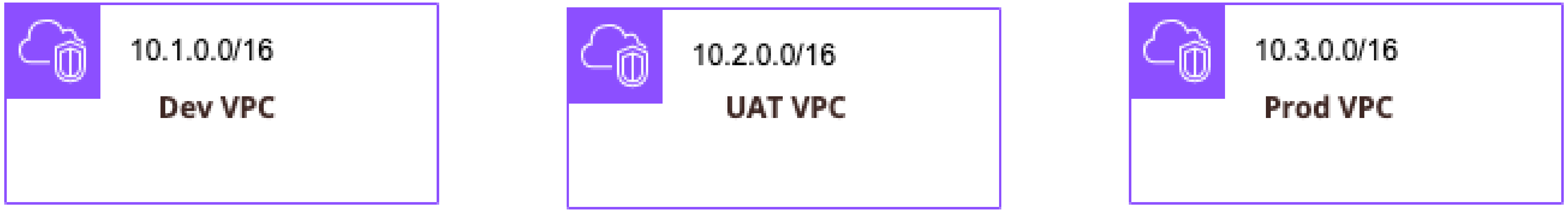
- Hybrid cloud architectures
    - Applications that require access to existing data centre
- Transferring large data sets
    - The high bandwidth link reduces the potential for network congestion and degraded application performance
- Performance predictability
    - Applications that operate on real-time data feeds, such as audio or video streams.
- Security and compliance
    - Enterprise security or regulatory policies sometimes require applications hosted on the AWS Cloud to be accessed through private network circuits only.

Direct Connect Example

DevKTOps

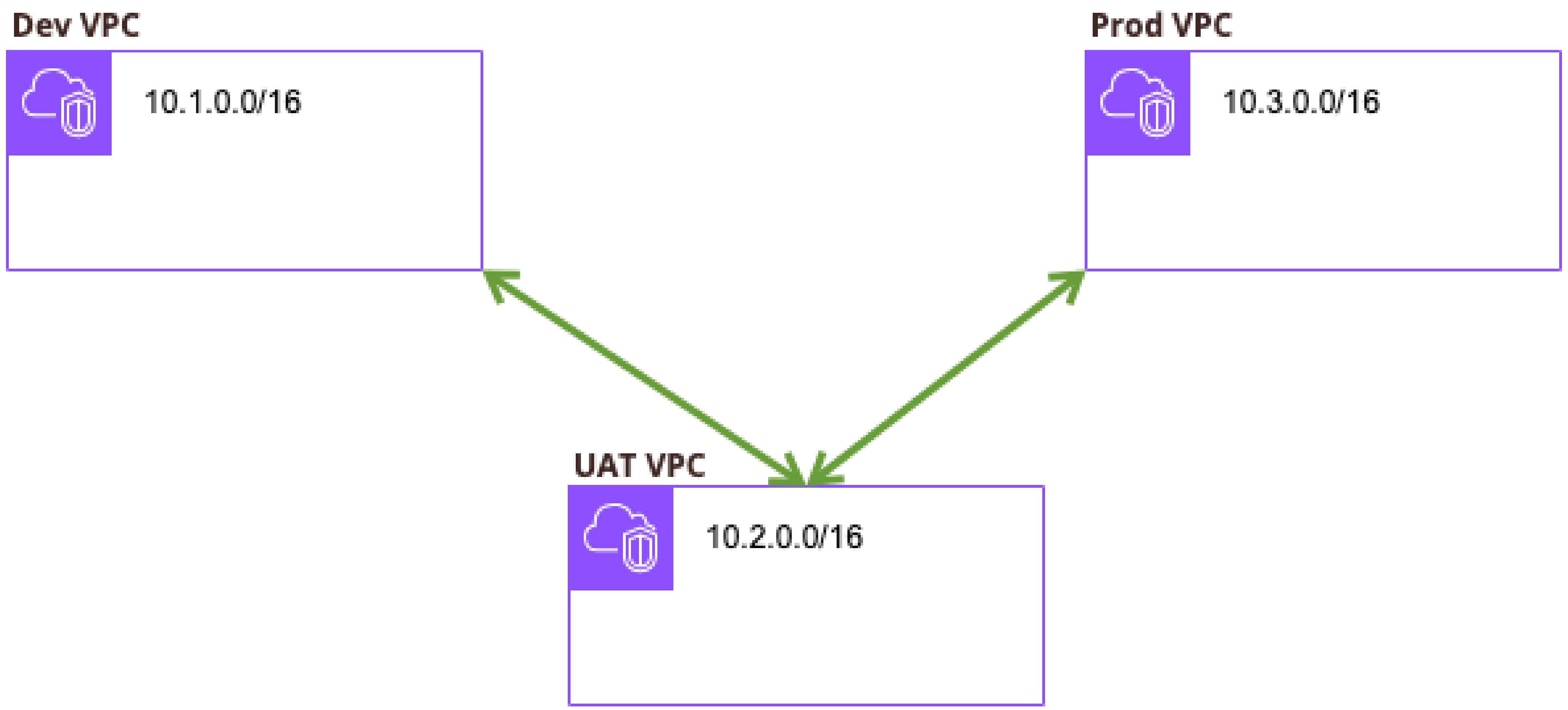


Connecting VPCs



- Isolating some of your workloads is generally a good practice.
- But you may need to transfer data between two or more VPCs.

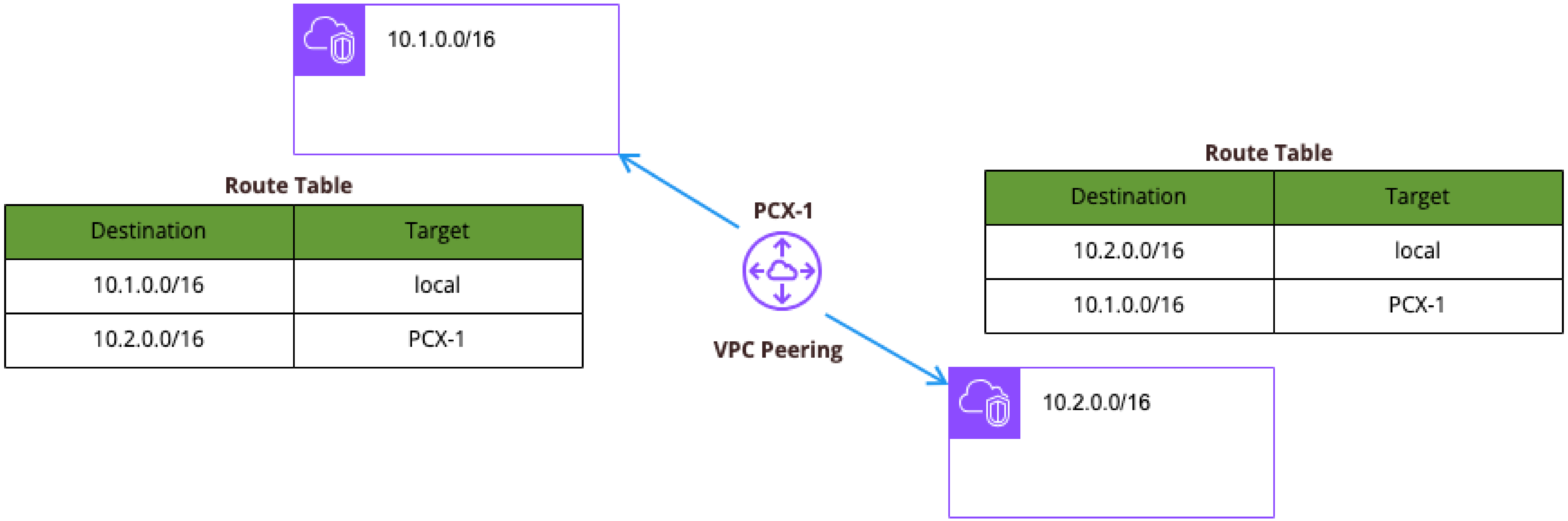
Connecting VPCs - VPC Peering



- Use private IP address
- Intra and inter-region support
- IP spaces cannot overlap
- Only one peering resource between any two VPCs
- Transitive peering relationships are not supported
- Can be established between different AWS Accounts

Instances can communicate across a peering connection as if they were in the same network.

VPC Peering



Route Table

Destination	Target
10.1.0.0/16	local
10.2.0.0/16	PCX-1

Route Table

Destination	Target
10.2.0.0/16	local
10.1.0.0/16	PCX-1

- No internet gateway or virtual gateway required
- Highly Available connections; not a single point of failure
- No bandwidth bottlenecks
- Traffic always stays on the AWS global backbone

Multiple VPCs Peering

Consider some universal network-design principles

- Ensure that your VPC network ranges (CIDR Blocks do not overlap.
- Make sure the solution you choose can scale according to your current and future VPC connectivity needs.
- Ensure that you implement a highly available (HA) design with no single point of failure.
- Consider your data-transfer needs.
- Connect only those VPCs that really need to communicate with each other.

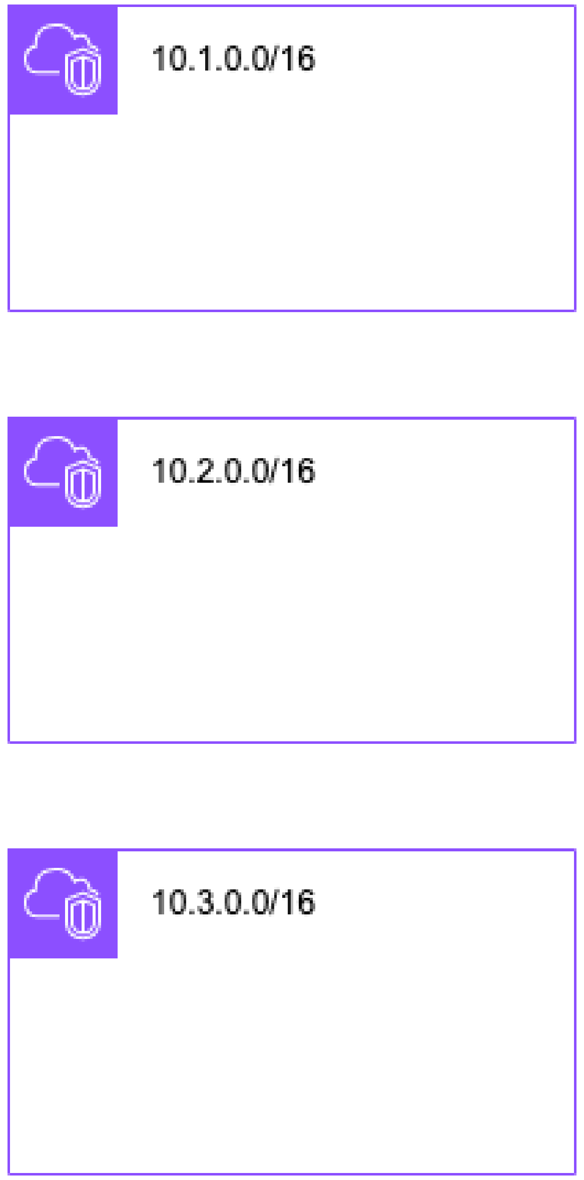
Connecting VPCs - Transit Gateway

AWS Transit Gateway

- Connects up to 5,000 VPCs and on-premises environments with a single gateway
- Serve as a hub for all traffic to flow through between your networks
- Fully managed, highly available, flexible routing service
- Allows for multicast and inter-regional peering

Transit Gateway Sample

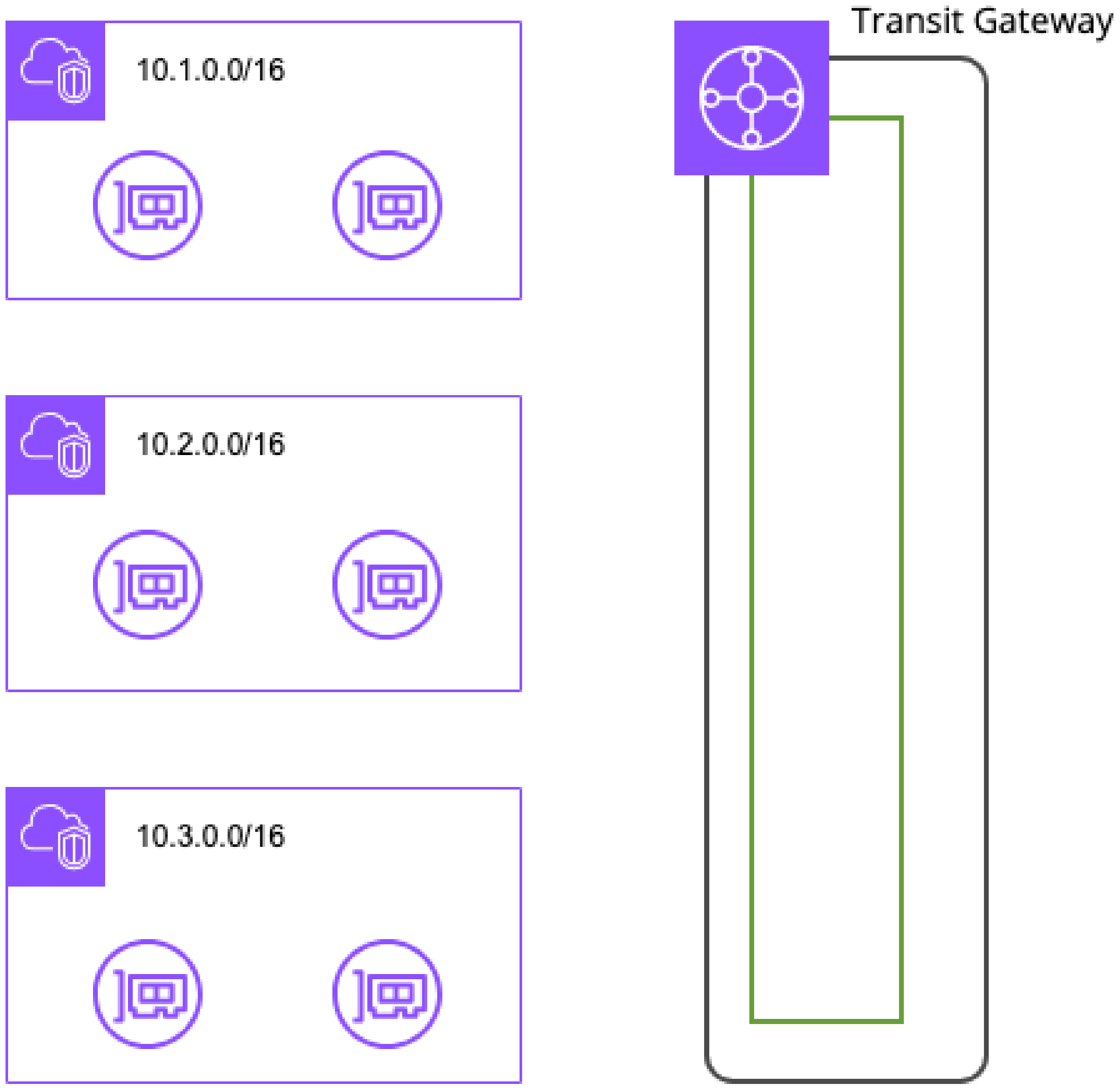
We want all three VPCs to be able to be fully connected.



Transit Gateway Sample

DevKTOps

We can use Transit Gateway To Implement it.

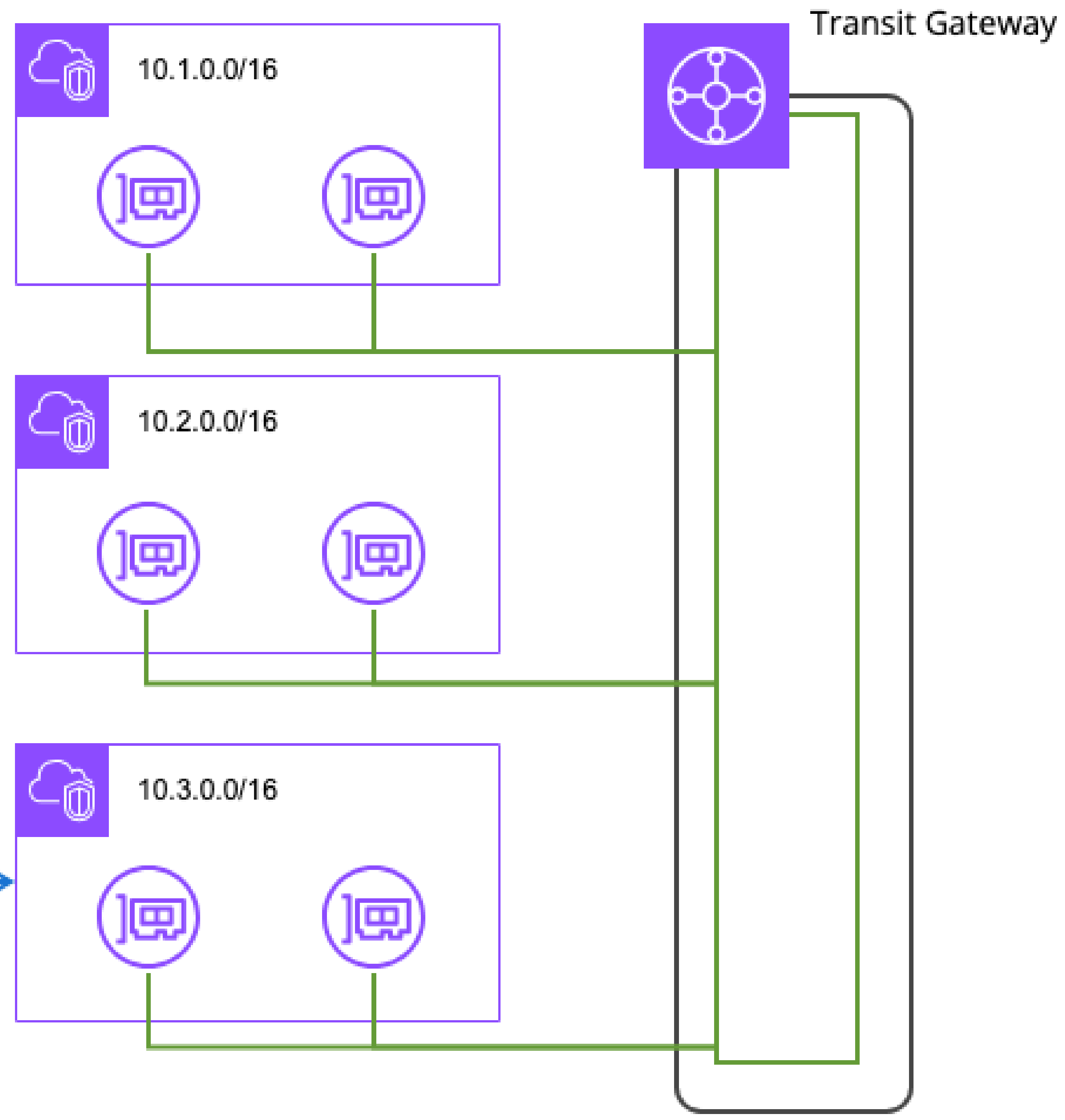


Transit Gateway Sample

DevKTOps

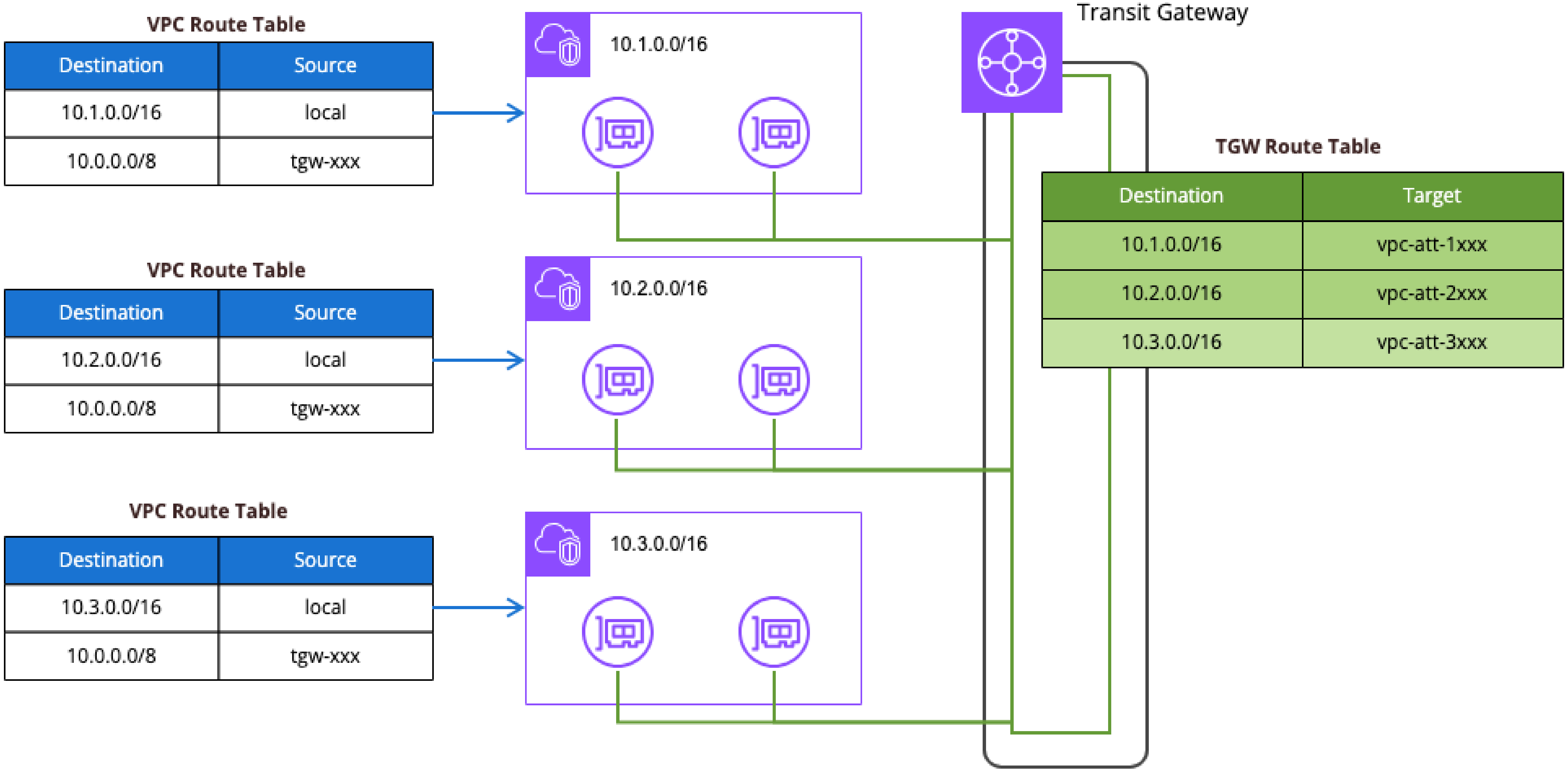
Attach TGW at Route Table

Destination	Source
10.3.0.0/16	local
10.0.0.0/8	tgw-xxx



Transit Gateway Sample

DevKTOps



VPC Route Table

Destination	Source
10.1.0.0/16	local
10.0.0.0/8	tgw-xxx

VPC Route Table

Destination	Source
10.2.0.0/16	local
10.0.0.0/8	tgw-xxx

VPC Route Table

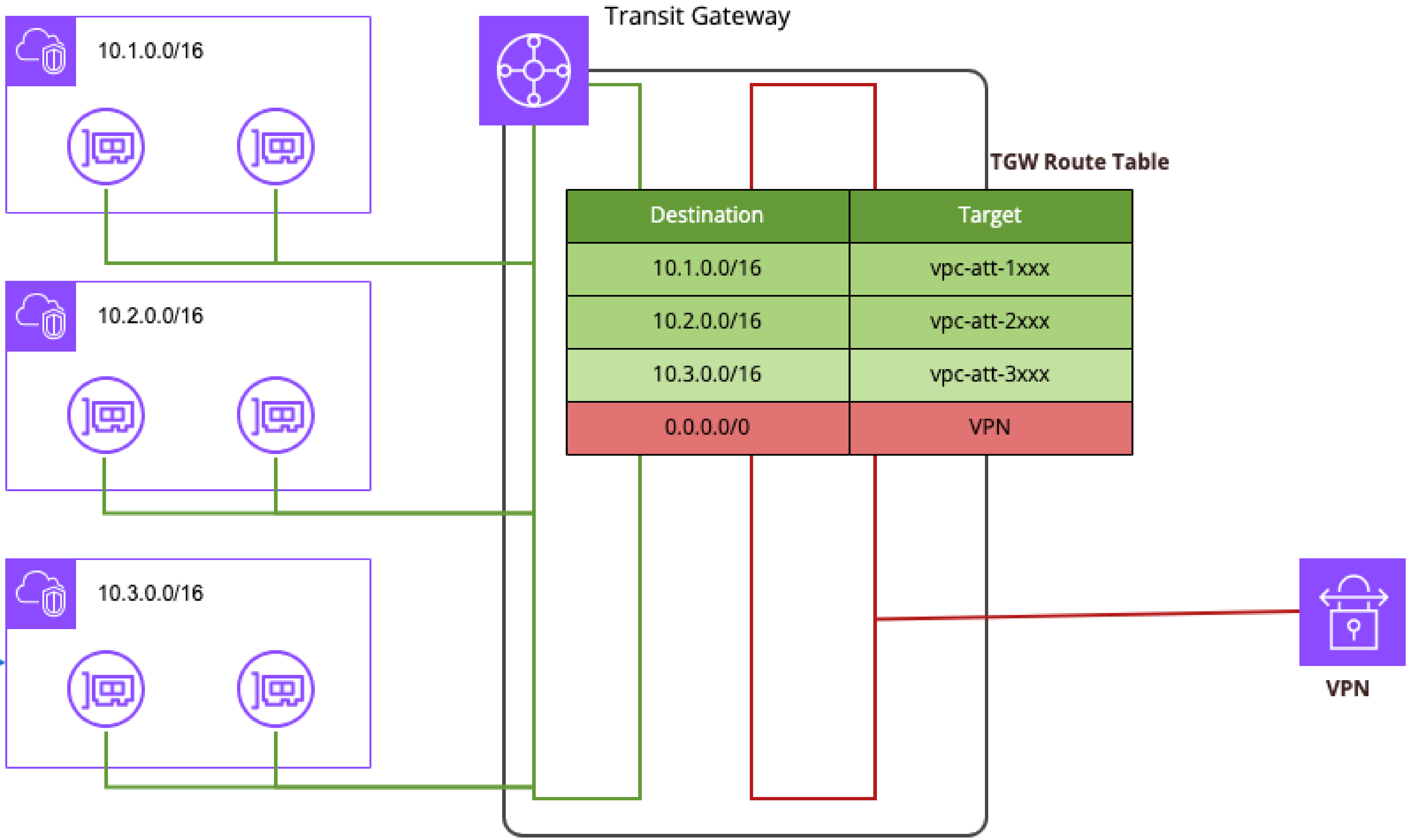
Destination	Source
10.3.0.0/16	local
10.0.0.0/8	tgw-xxx

TGW Route Table

Destination	Target
10.1.0.0/16	vpc-att-1xxx
10.2.0.0/16	vpc-att-2xxx
10.3.0.0/16	vpc-att-3xxx

Transit Gateway Sample

We want to connect with VPN tunnel



TGW Route Table

Destination	Target
10.1.0.0/16	vpc-att-1xxx
10.2.0.0/16	vpc-att-2xxx
10.3.0.0/16	vpc-att-3xxx
0.0.0.0/0	VPN

VPC Route Table

Destination	Source
10.3.0.0/16	local
10.0.0.0/8	tgw-xxx

VPC Endpoints

Privately connect your EC2 instances to services outside your VPC without leaving AWS.

Don't need to use an internet gateway, VPN, network address translation (NAT) devices, or firewall proxies.

- Does not require traversal over the internet
- Must be in the same region
- They are horizontally scaled, redundant, and highly available

Two Types of Endpoint

DevKTOps

Interface endpoint is an elastic network interface with a private IP address that serves as an entry point for traffic destined to a supported service. Mainly used AWS PrivateLink and working on a subnet.

- Amazon CloudWatch Logs
- AWS CodeBuild
- Amazon EC2 API
- Elastic Load Balancing API
- AWS Key Management Service (AWS KMS)
- Amazon Kinesis Data Streams
- AWS Service Catalog
- Amazon Simple Notification Service (Amazon SNS)
- AWS Systems Manager
- Endpoint services hosted by other AWS accounts
- And MANY MORE!

Two Types of Endpoint

Gateway endpoint is a gateway that is a target for a specified route in your route table, used for traffic destined o a supported AWS service. Working on a VPC level.

- Amazon Simple Storage Service (Amazon S3)
- Amazon DynamoDB

Route 53

Route 53 is a highly available and scalable cloud Domain Name System (DNS) service.

- DNS translates domain names into IP addresses
- Able to purchase and manage domain names and automatically configure DNS settings
- Provides tools for flexible, high-performance, highly available architectures on AWS
- Multiple routing options

Route 53 Routing Options

Simple routing (round robin)

- Distributes the number of requests as evenly possible between all participating servers

Weighted round robin

- Allows you to assign weights to resource record sets in order to specify the frequency which different responses are served

Latency-based routing

- Helps you improve your application's performance for a global audience.

Geolocation routing

- You can choose the resources that serve your traffic based on the geographic location of your users. You can localize your content and present some or all of your website in the language of your users.

Geoproximity routing

- You can route traffic based on the physical distance between your users and your resources if you're using Route 53 traffic flow.

Route 53 Routing Options

DNS Failover

- Route 53 can help detect an outage of your website and redirect your end users to alternate locations where your application is operating properly.

Multi-value Answers

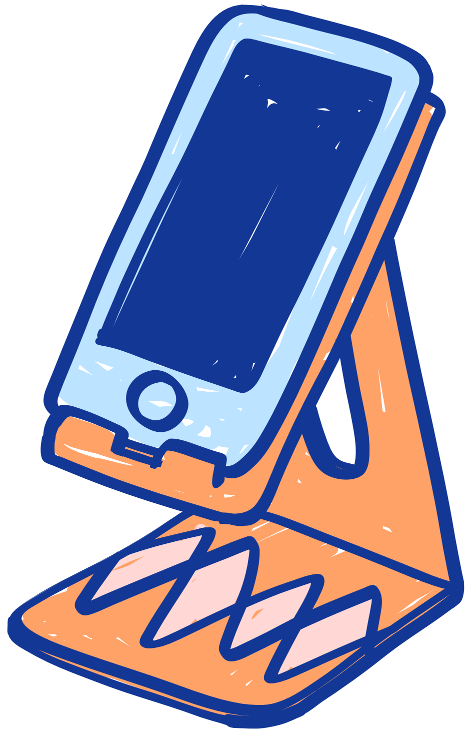
- If you want to route traffic approximately randomly to multiple resources, such as web servers.

Amazon Route 53 Health Checks

- Monitor the health and performance of your web applications, web servers and other resources.

DevKTOps

THANK YOU



See you in next lecture!