AWS Global Networking in Terraform

Chad Smith

Principal Cloud Architect

Pearson

Introduction to Terraform

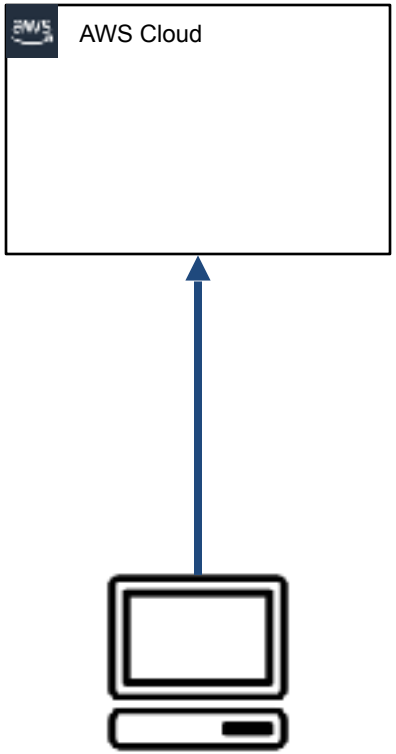
What is Hashicorp Terraform?

Terraform is an infrastructure as code (IaC) tool that allows you to build, change, and version infrastructure safely and efficiently.

This includes low-level components such as compute instances, storage, and networking, as well as high-level components such as DNS entries, SaaS features, etc.

Terraform can manage both existing service providers and custom in-house solutions.

Terraform in AWS (SPoF)



Client provisions or modifies AWS resources via Terraform

Relevant Terraform Constructs

Resources

All object types that are created in a terraform template.

Variables

Parameters which can make templates reusable while consolidating hardcoded values

Modules

Obfuscation of the dirty work of creating multiple related resources of different types

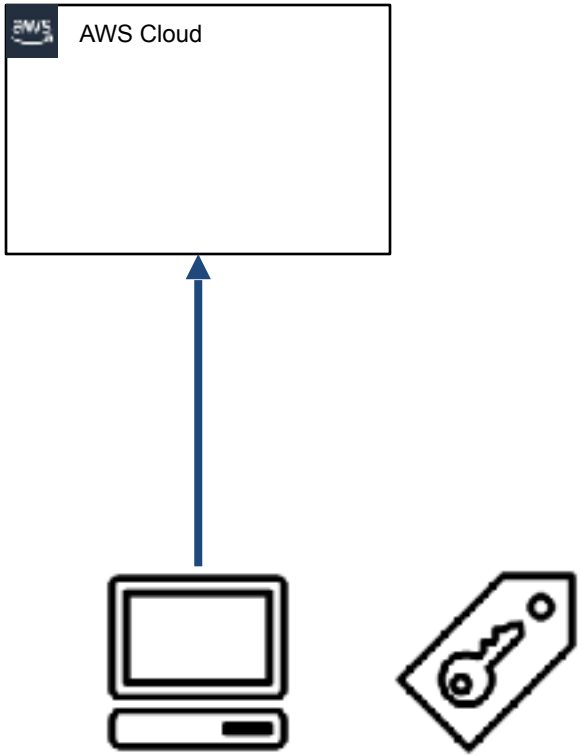
Outputs

Hardcoded values or properties of created objects which can be queried using the backend state objects

Backend

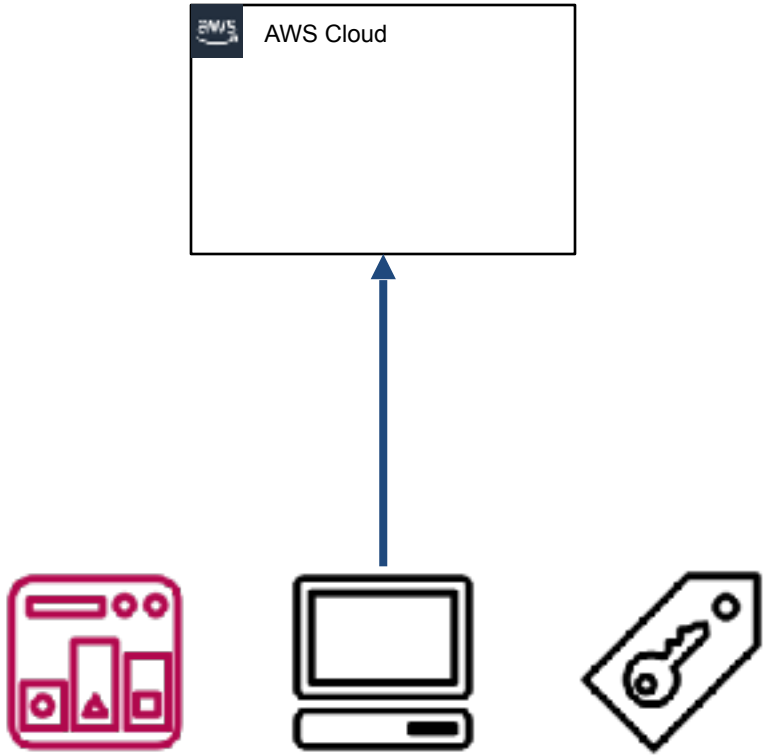
Location of state files and lockfile storage

Terraform in AWS (SPoF)



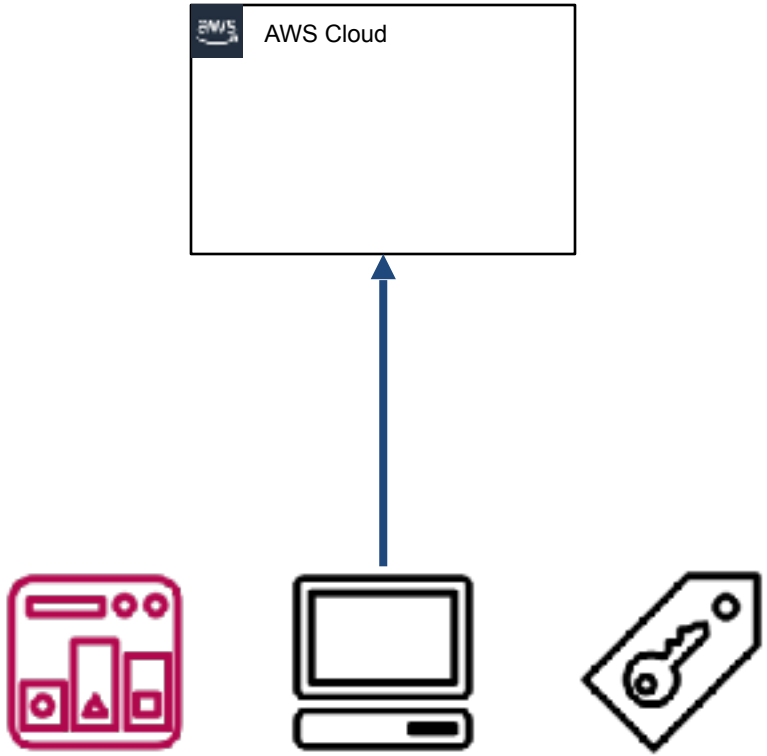
Lockfile prevents multiple actions from same client

Terraform in AWS (SPoF)



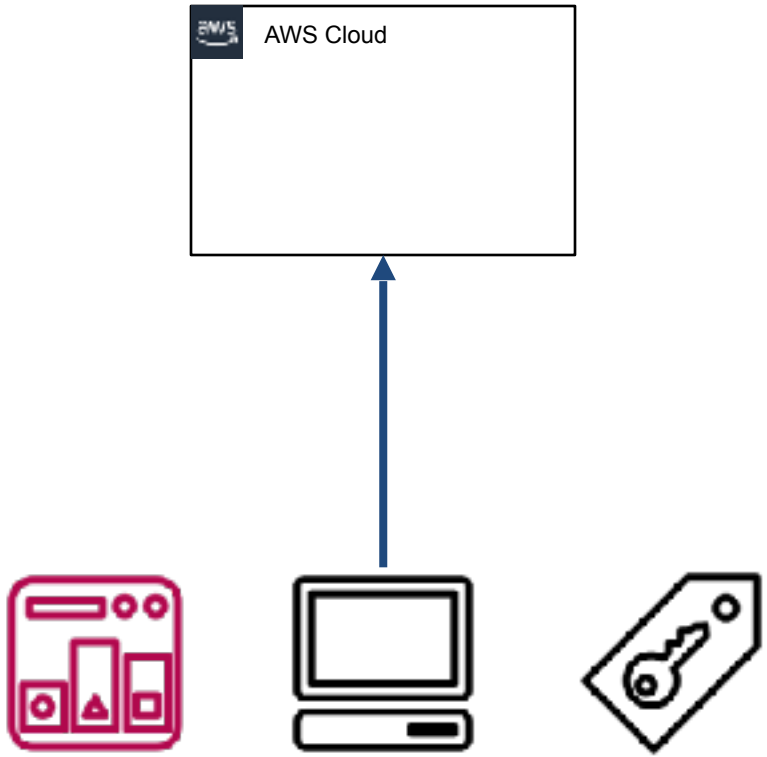
Resource state and outputs are stored on the client

Terraform in AWS (SPoF)



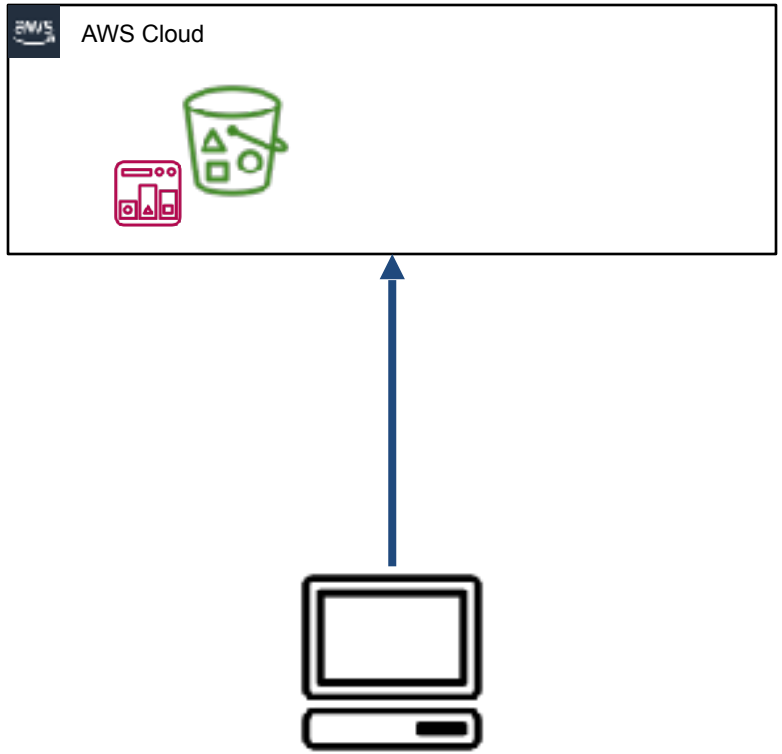
This paradigm does not account for multiple users or shared access to state and outputs

Terraform in AWS (SPoF)



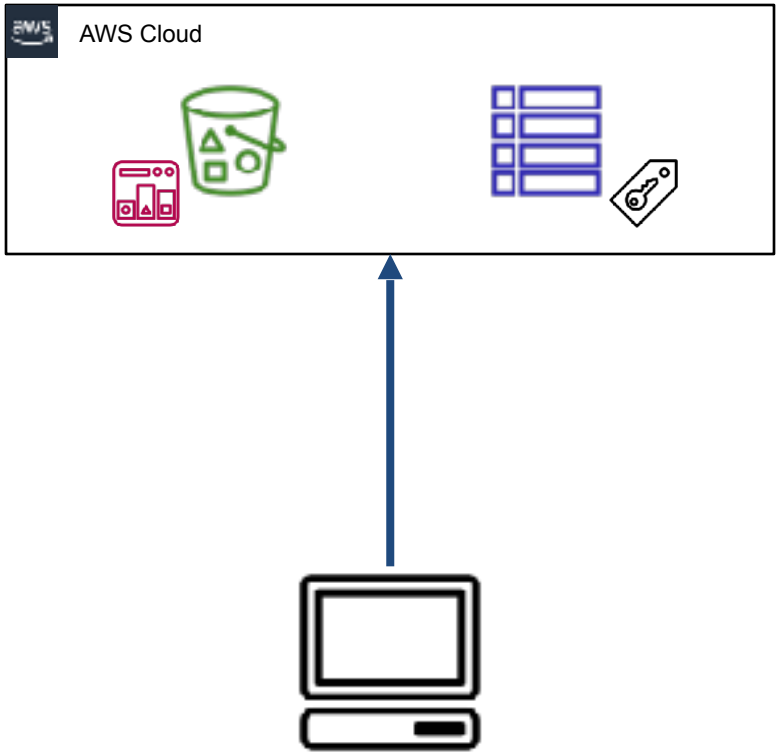
Nor is this a resilient design, by any measure

Terraform in AWS (Resilient)



Store the state files as S3 objects for remote access from other stacks

Terraform in AWS (Resilient)



Store the lockfiles as DynamoDB table items to allow shared access without conflict

Demonstration

Evaluate the S3 and DynamoDB configuration for use by Terraform

AWS Global Network Requirements and Architecture

Global Network Scenario

A company has tasked an AWS network architect with the creation of a global network for placement of various application infrastructure

The architecture must support multiple environments, regions and be extensible for future expansion

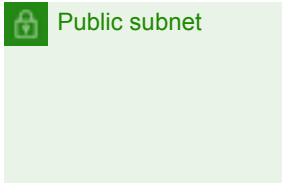
How can the network architect use Terraform to create a reliable and scalable global network for the applications to use?

Global Network Support Requirements

- Public-facing apps
- Internal only apps
- Multiple regions
- Highly available
- Use private networking where possible

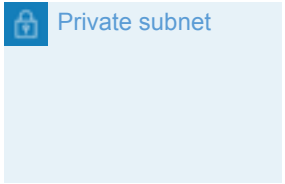
Requirement Translation to Features

Public Facing Applications



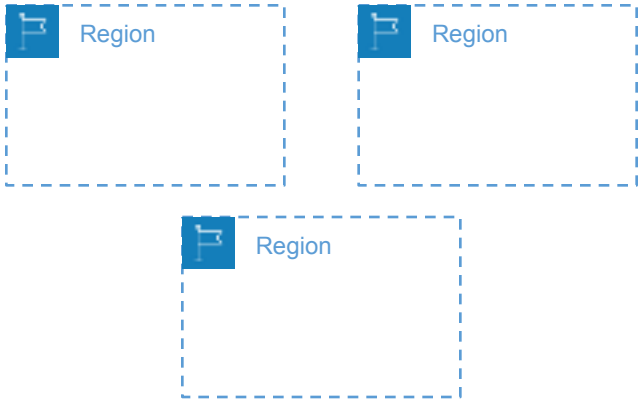
Requirement Translation to Features

Internal-only Applications



Requirement Translation to Features

Geographic Separation (For Business Continuity)

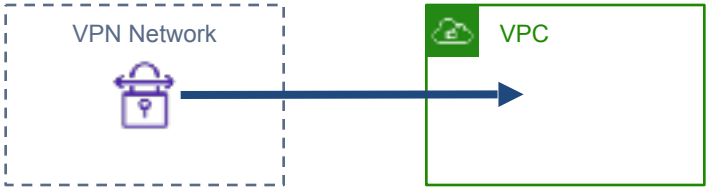


Requirement Translation to Features

Highly Available

Requirement Translation to Features

VPN Access (either external or private)



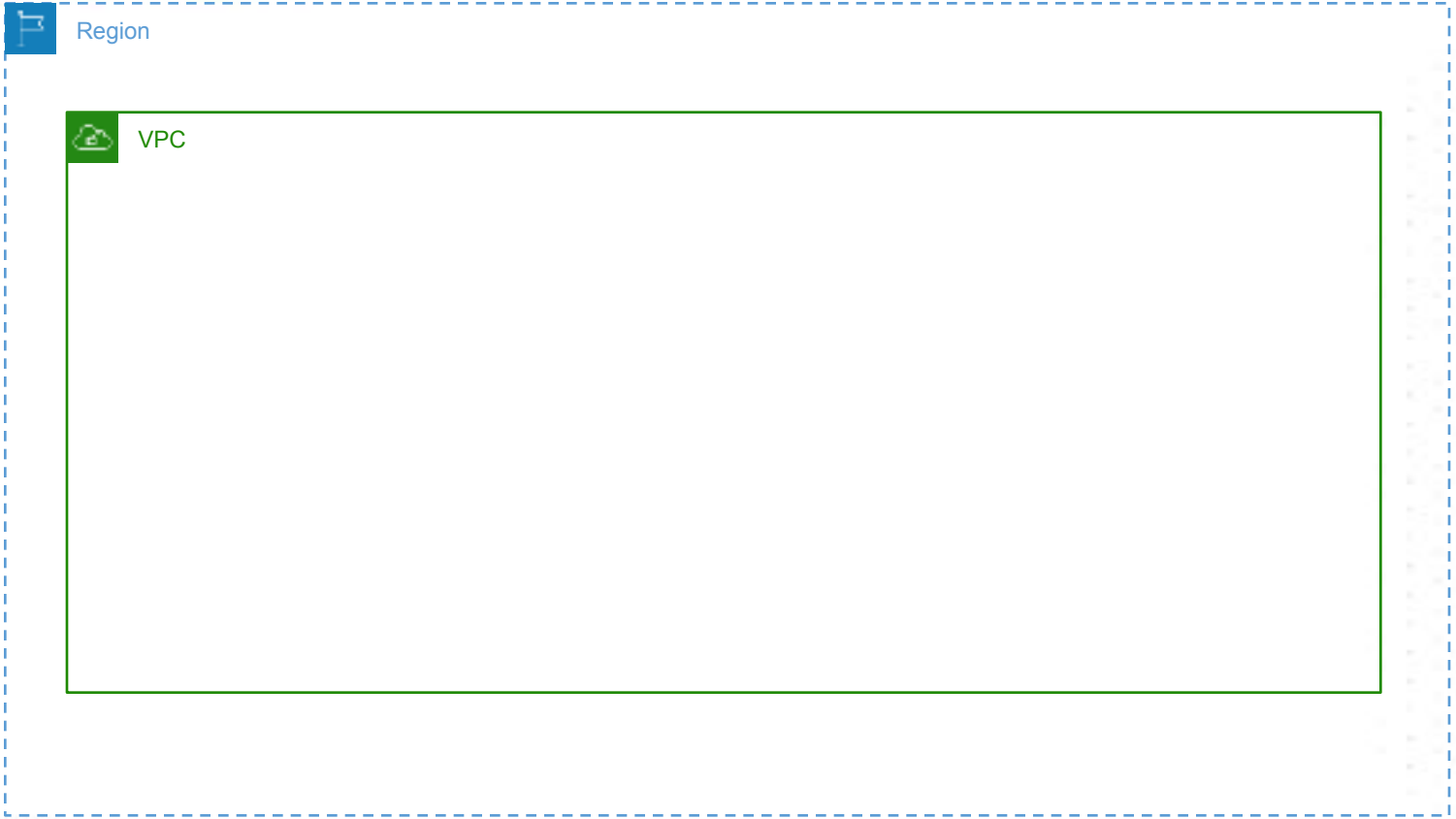
Requirement Translation to Features

Use private networking where possible

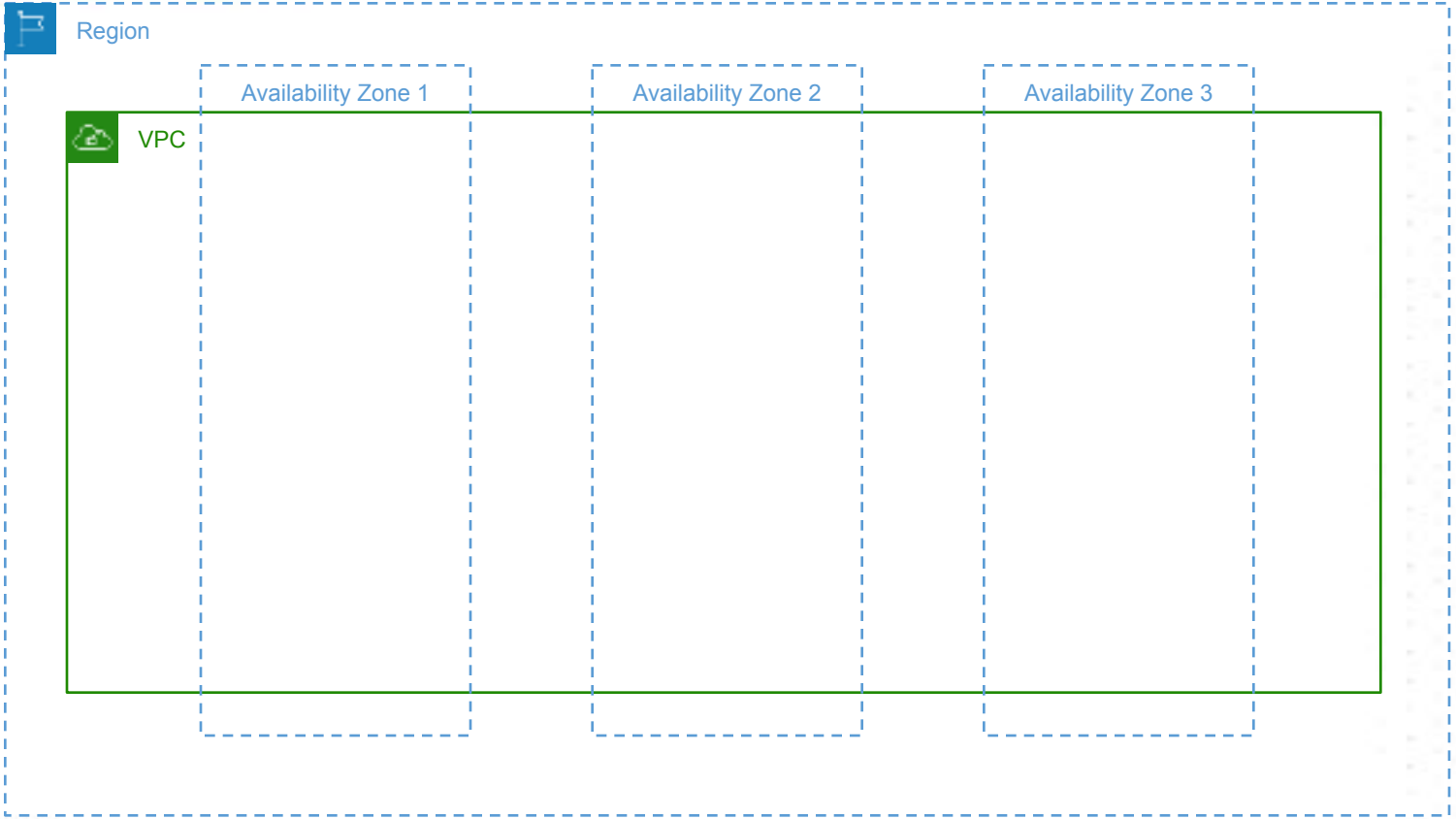
Gateway Endpoint

Interface Endpoint

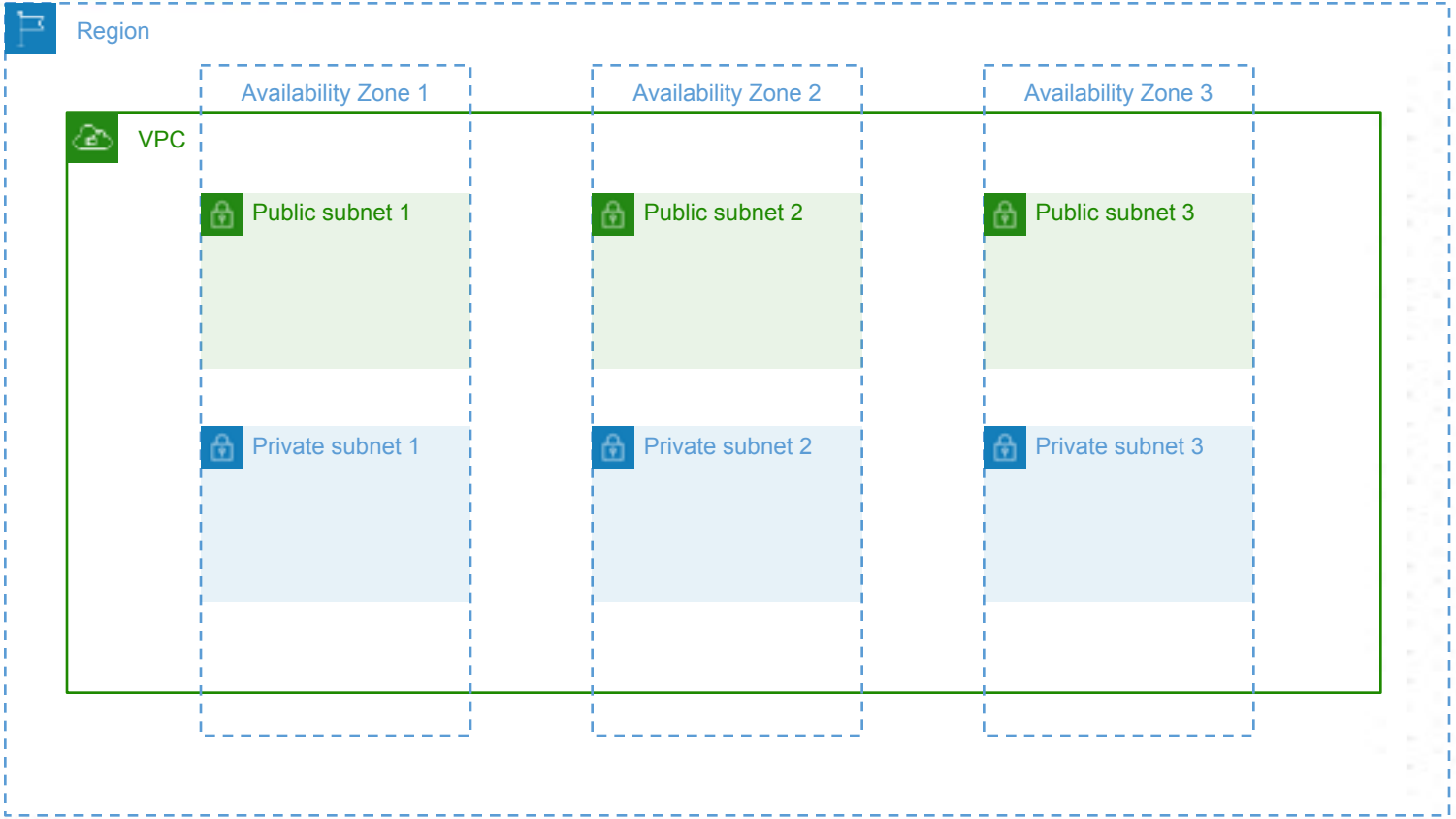
Single Region Network Diagram



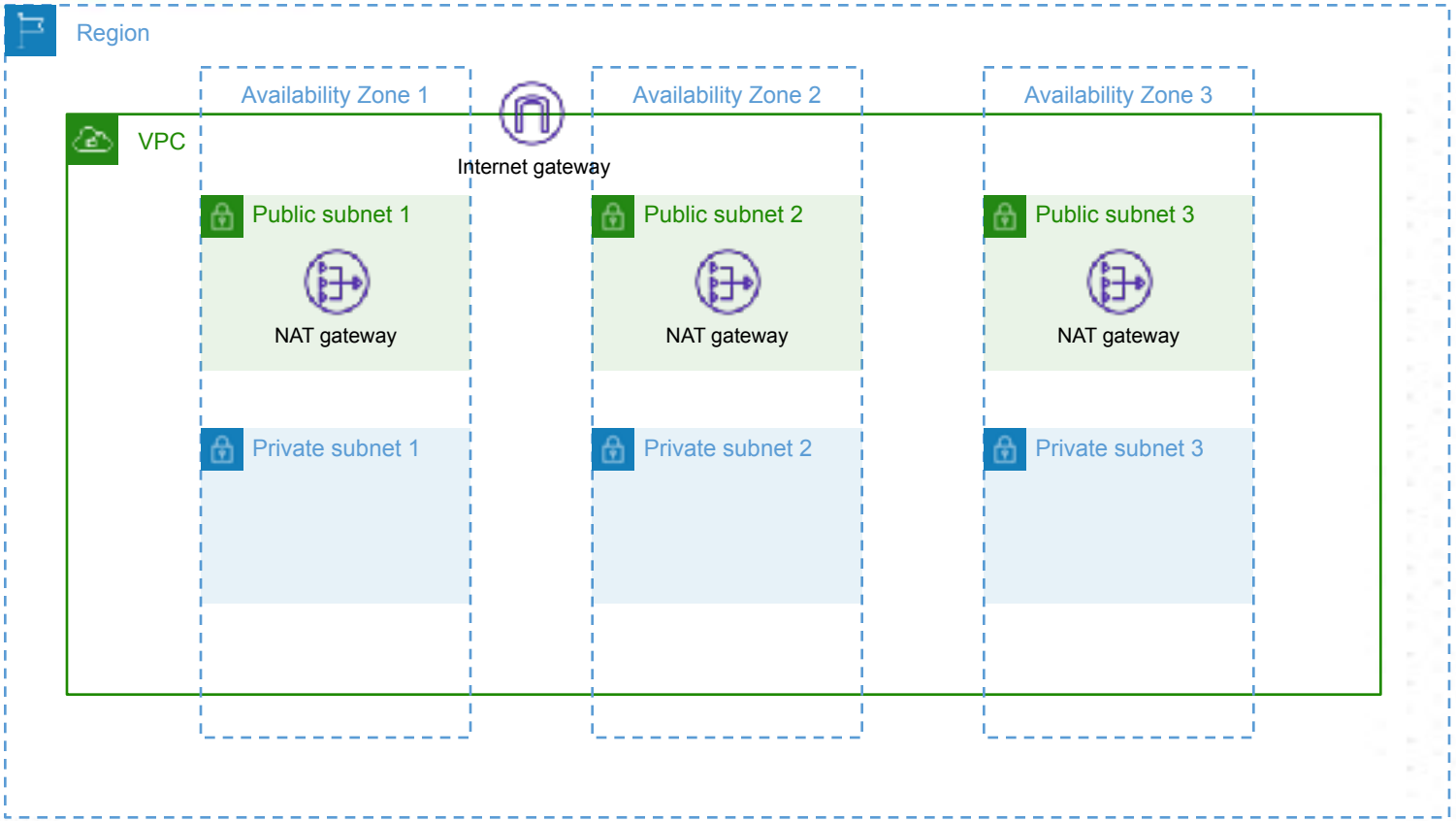
Single Region Network Diagram



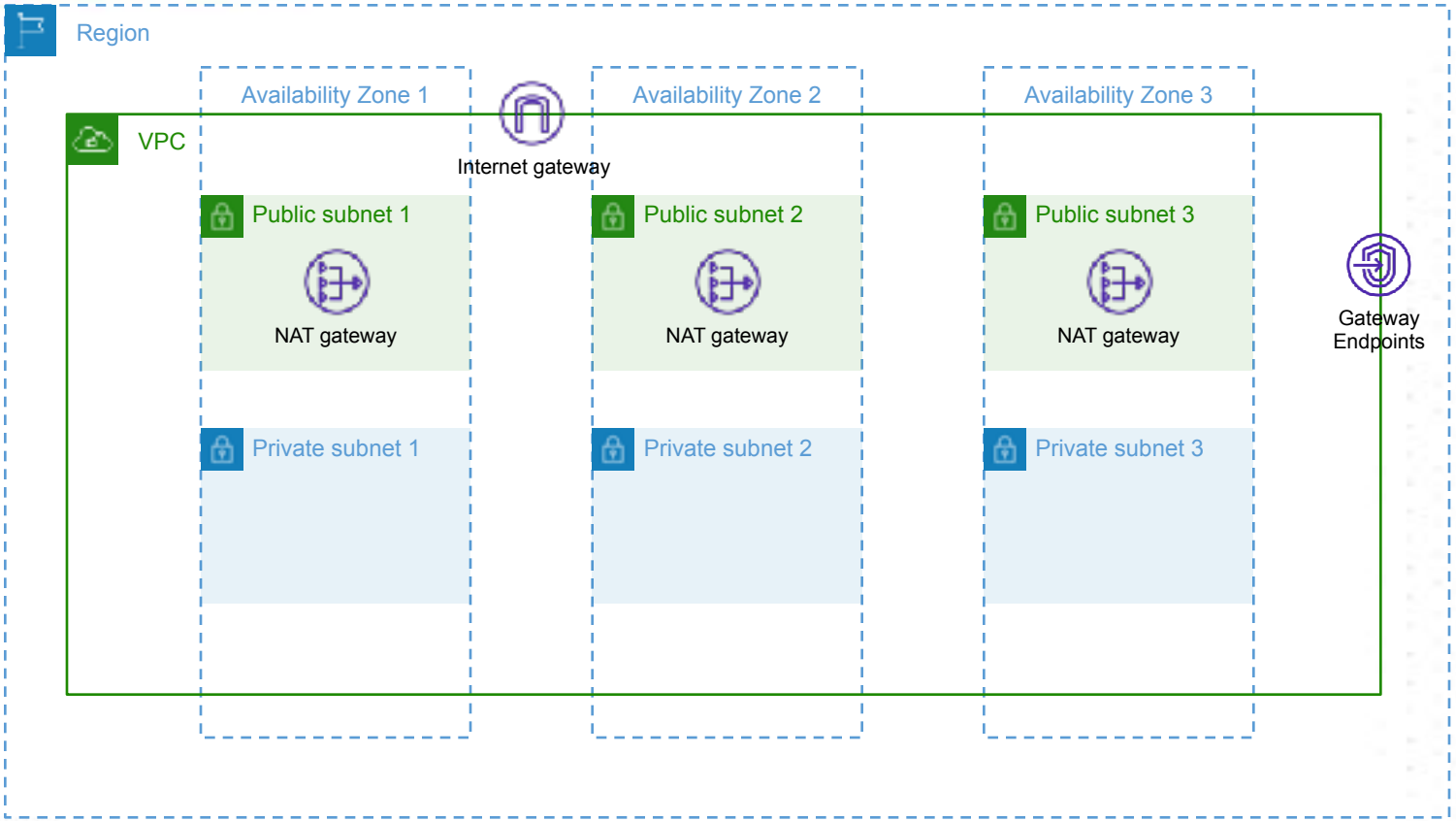
Single Region Network Diagram



Single Region Network Diagram



Single Region Network Diagram



Single-Region Network Deployment

Terraform VPC Resources (Atomic)

- VPC object
- Subnet(s)
- IGW
- IGW attachment
- NAT GW(s)
- NAT GW EIP(s)
- Route table(s)
- RT attachment(s)
- Tag(s)

All of these require several parameters, many of which are redundant from one resource to the next

Terraform VPC Module

- CIDR
- AZs
- Public subnets
- Private subnets
- NAT GW #s
- Tags

The module provisions all individual resources with a single set of parameters

Demonstration

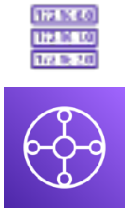
Deploy a single-region VPC

Transit Gateway Deployment

Transit Gateway Basics

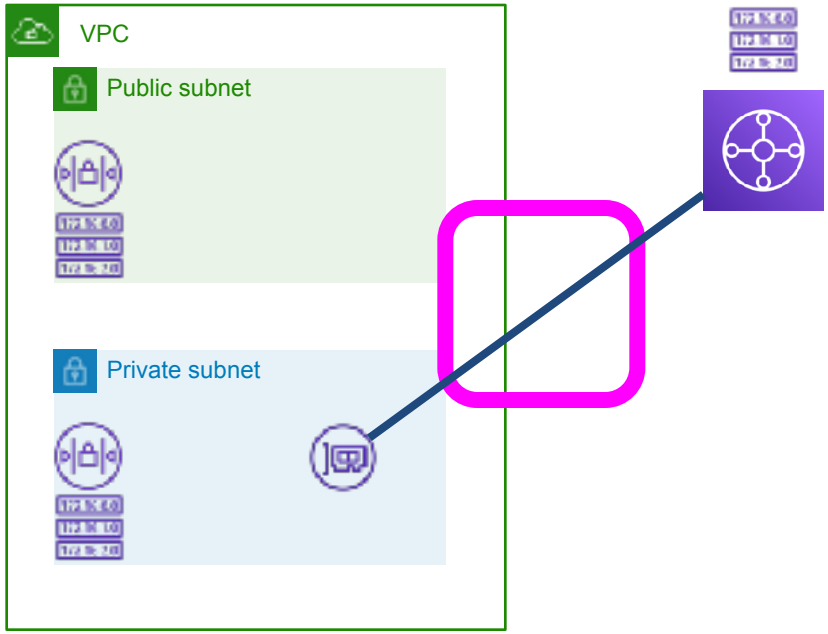
- Region scoped
- Attach to subnet
- Standalone resource
- Connect to VPCs
- Connect to on-prem networks

Transit Gateway Provisioning



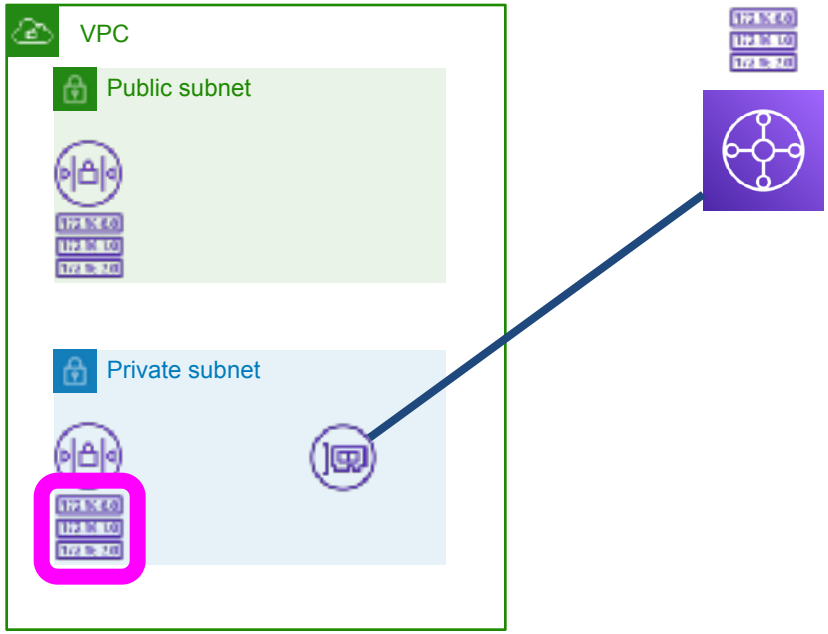
Transit Gateway can be deployed as a standalone resource

Transit Gateway Attachments



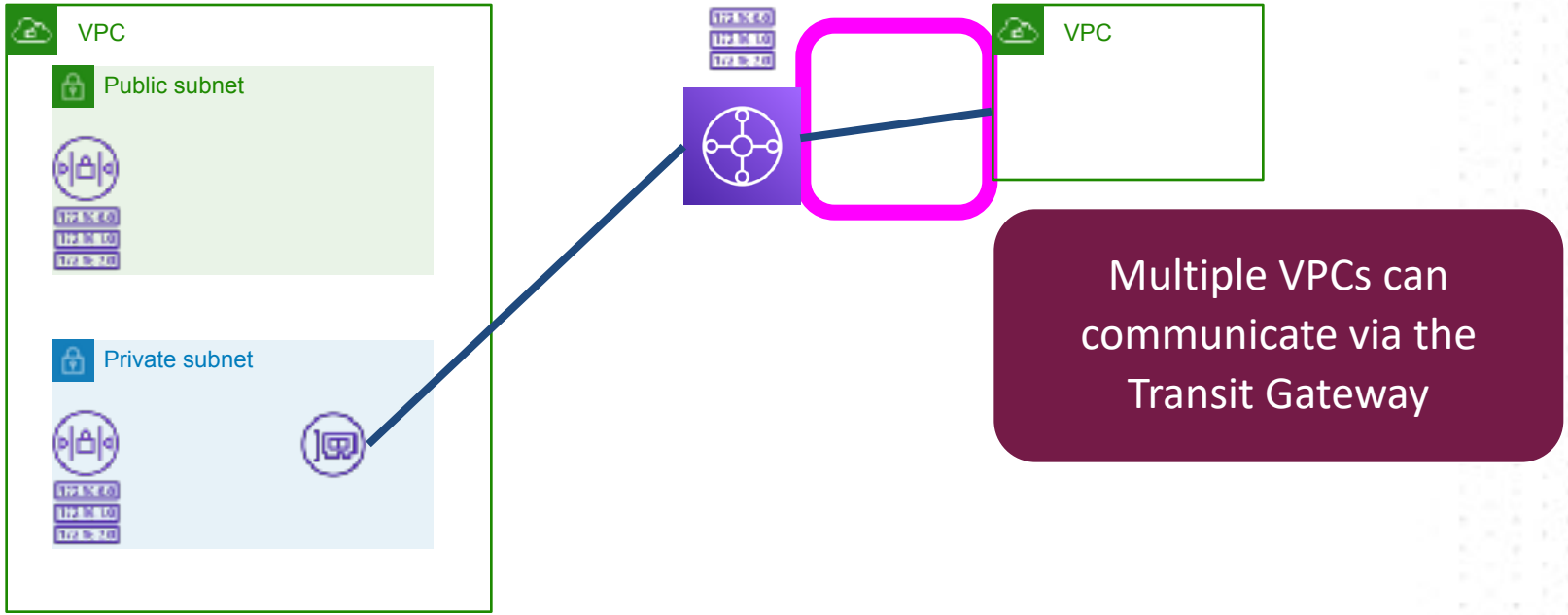
Attachments can be created into subnets in a VPC

Transit Gateway Attachments

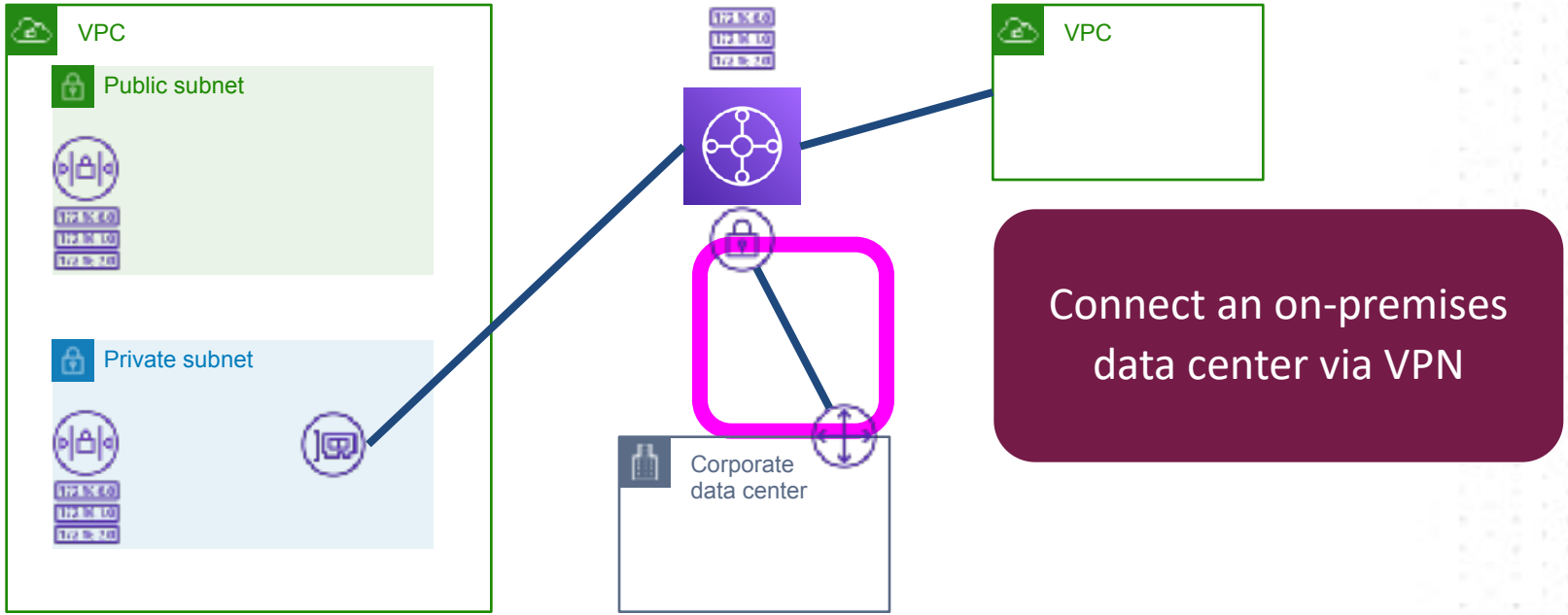


Route table entries are also required for each subnet

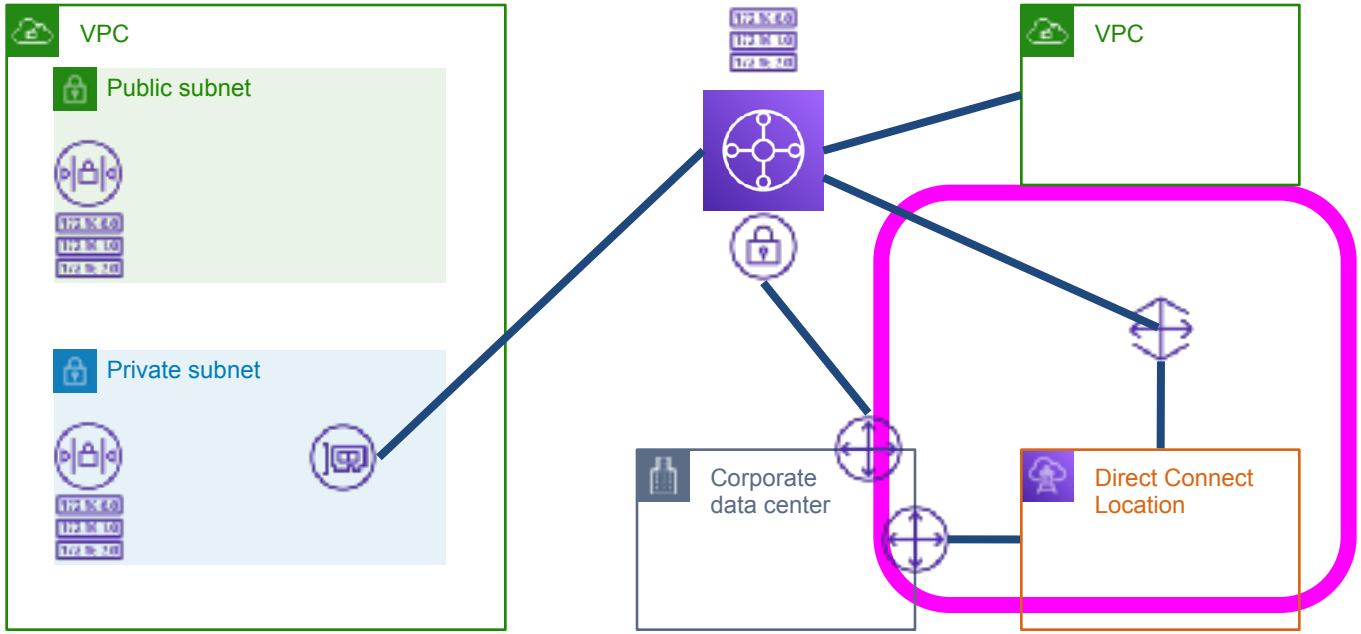
Transit Gateway Attachments



Transit Gateway Attachments

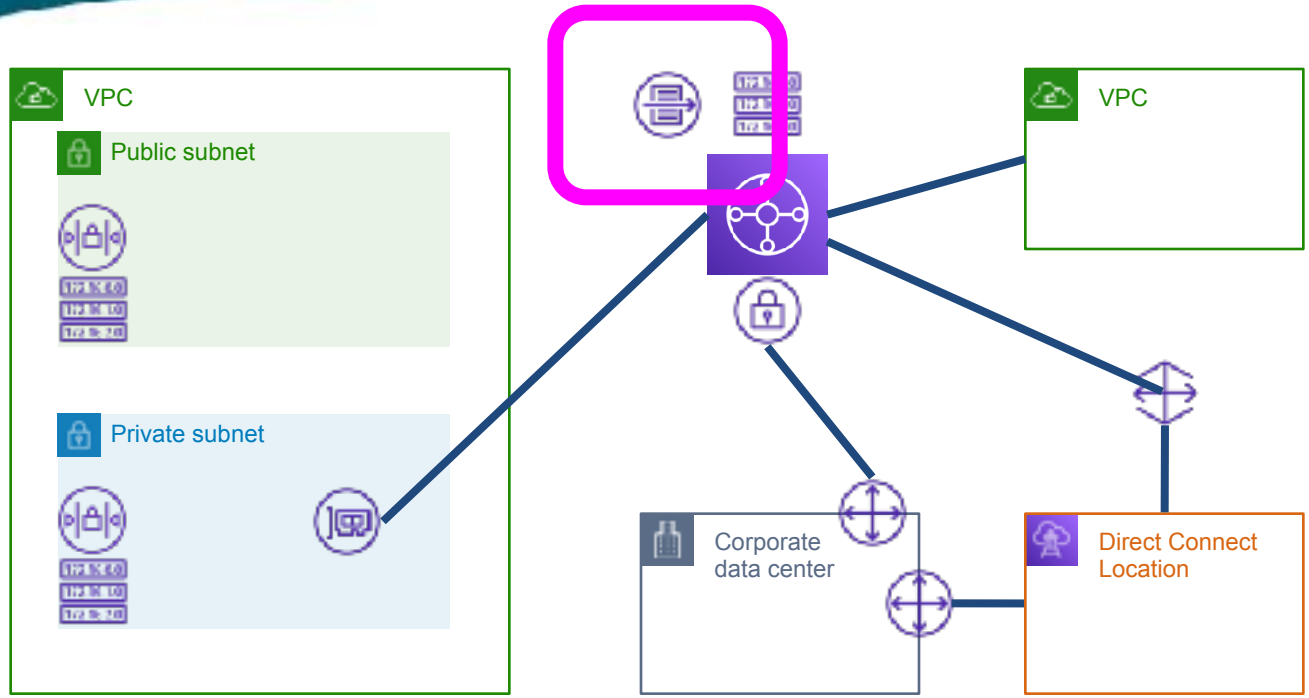


Transit Gateway Attachments



Connect an on-premises data center via Direct Connect Gateway

Transit Gateway Attachments



Enable VPC Flow logs for monitoring (NEW!!)

Terraform Transit Gateway Module

- TGW
- TGW VPC attachments
- TGW routes
- Tags

What is missing from the module?

- Cross-region peering attachments
- Cross-region TGW route table entries
- VPC route table entries

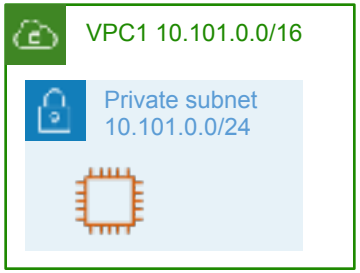
Because of all these missing elements, it makes sense to make OUR OWN module!

Demonstration

Create a custom module in Terraform for peering and routes which recursively uses the built-in TGW module

Multi-Region Network Connectivity using Transit Gateway

Multi-Region TGW Infrastructure

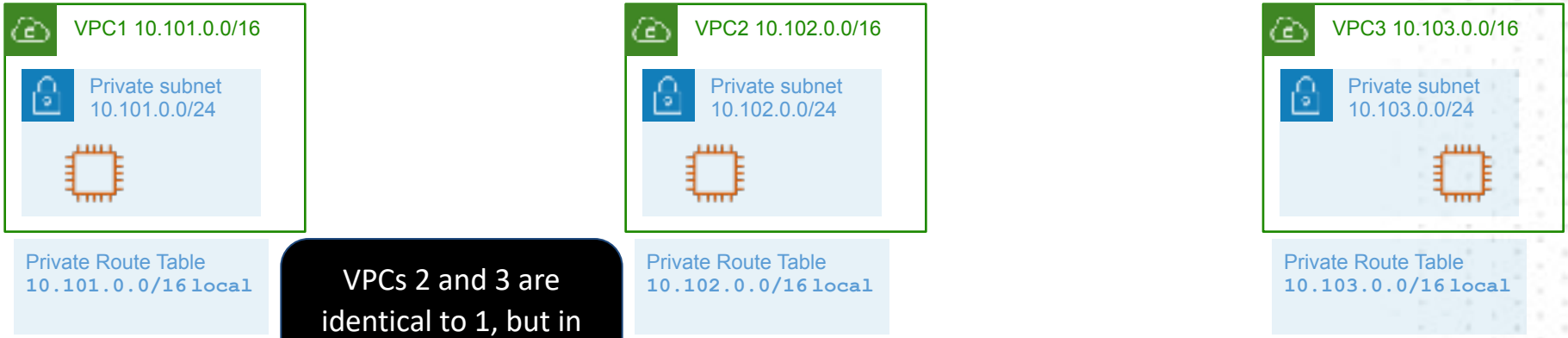


Private Route Table

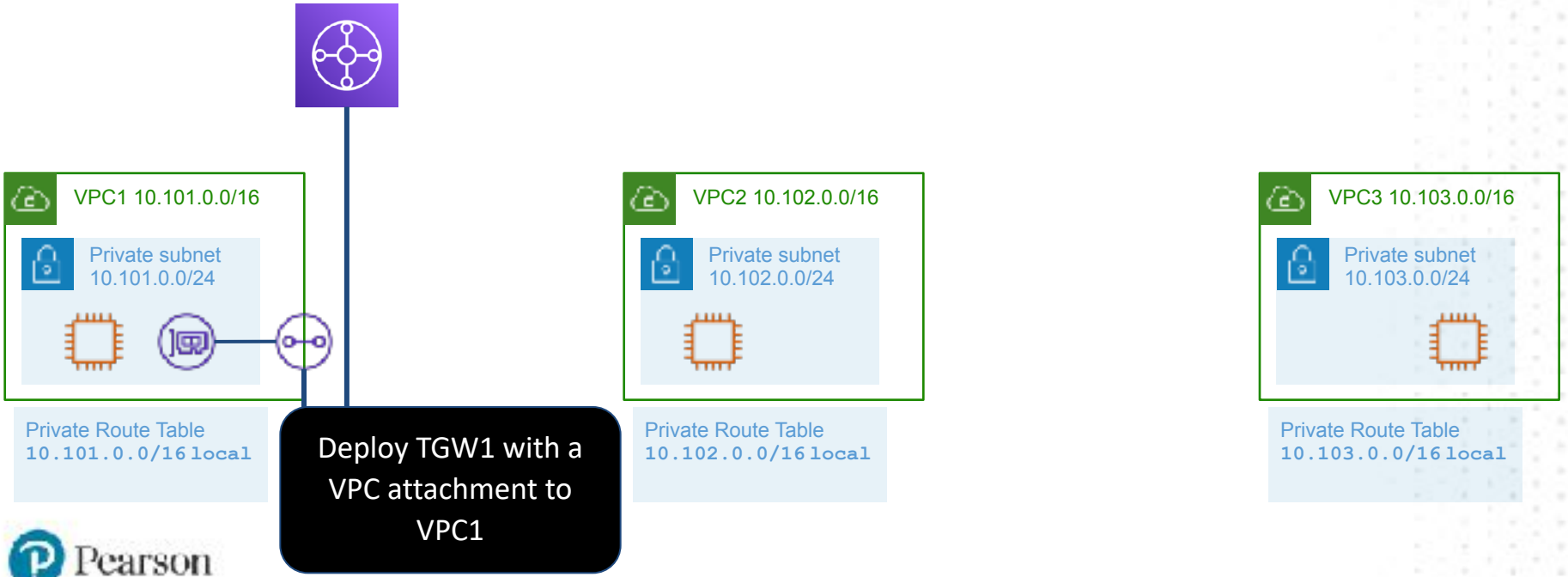
```
10.101.0.0/16 local
```

VPC1 (only showing private subnet)

Multi-Region TGW Infrastructure

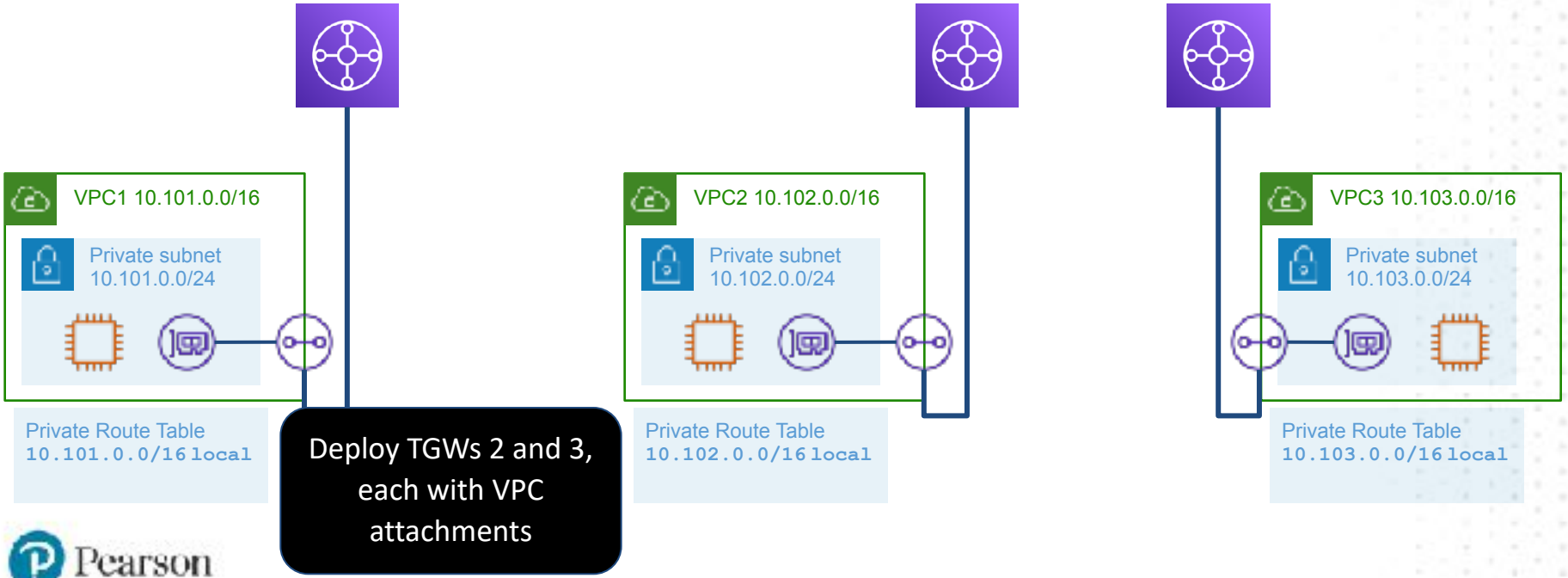


Multi-Region TGW Infrastructure

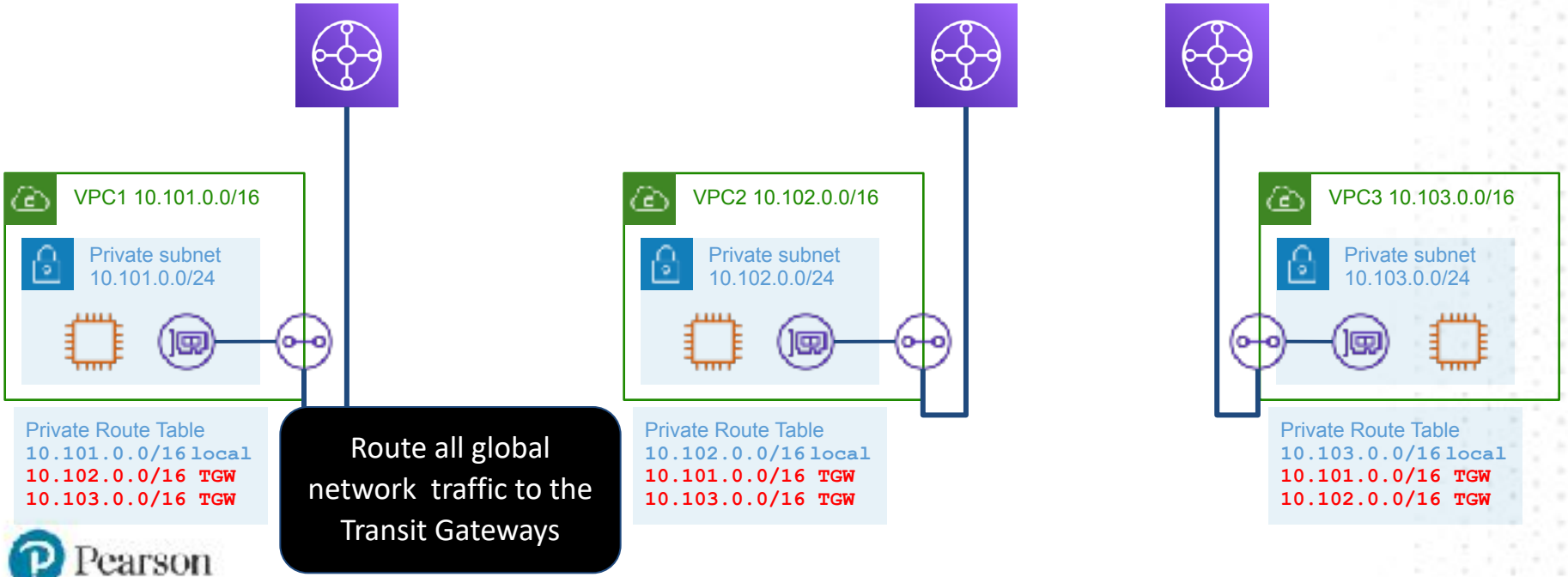


Pearson

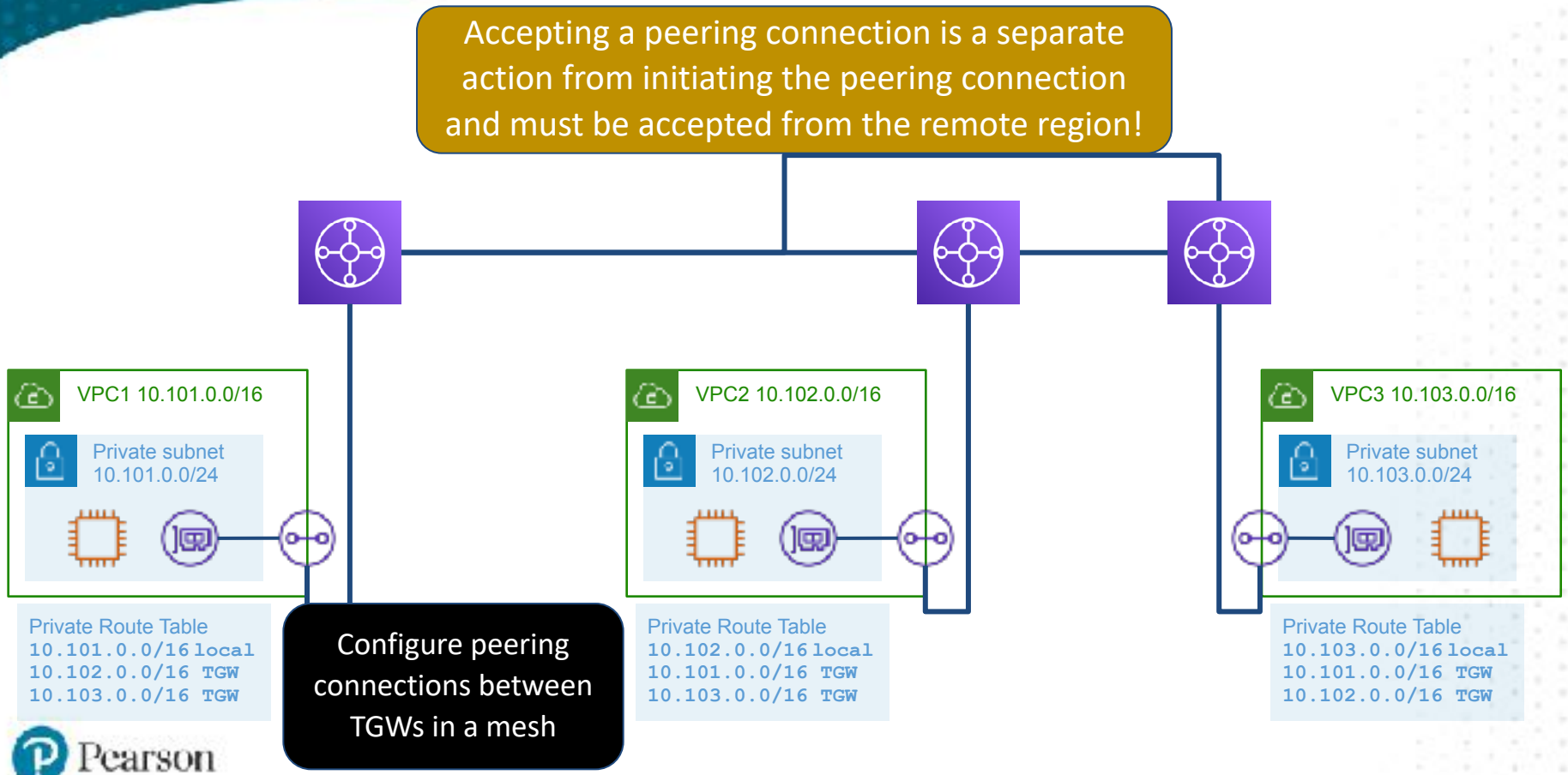
Multi-Region TGW Infrastructure



Multi-Region TGW Infrastructure

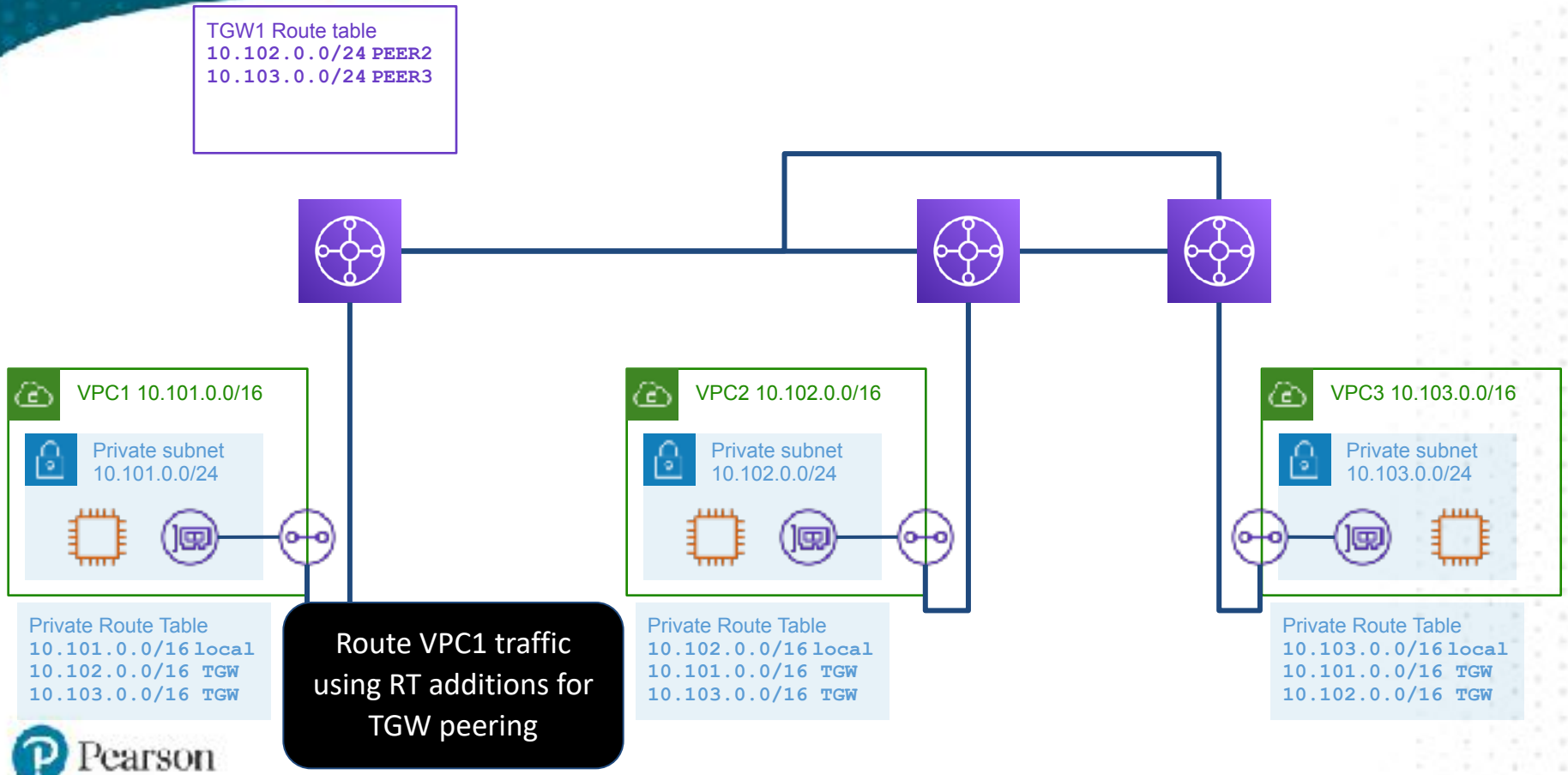


Multi-Region TGW Infrastructure

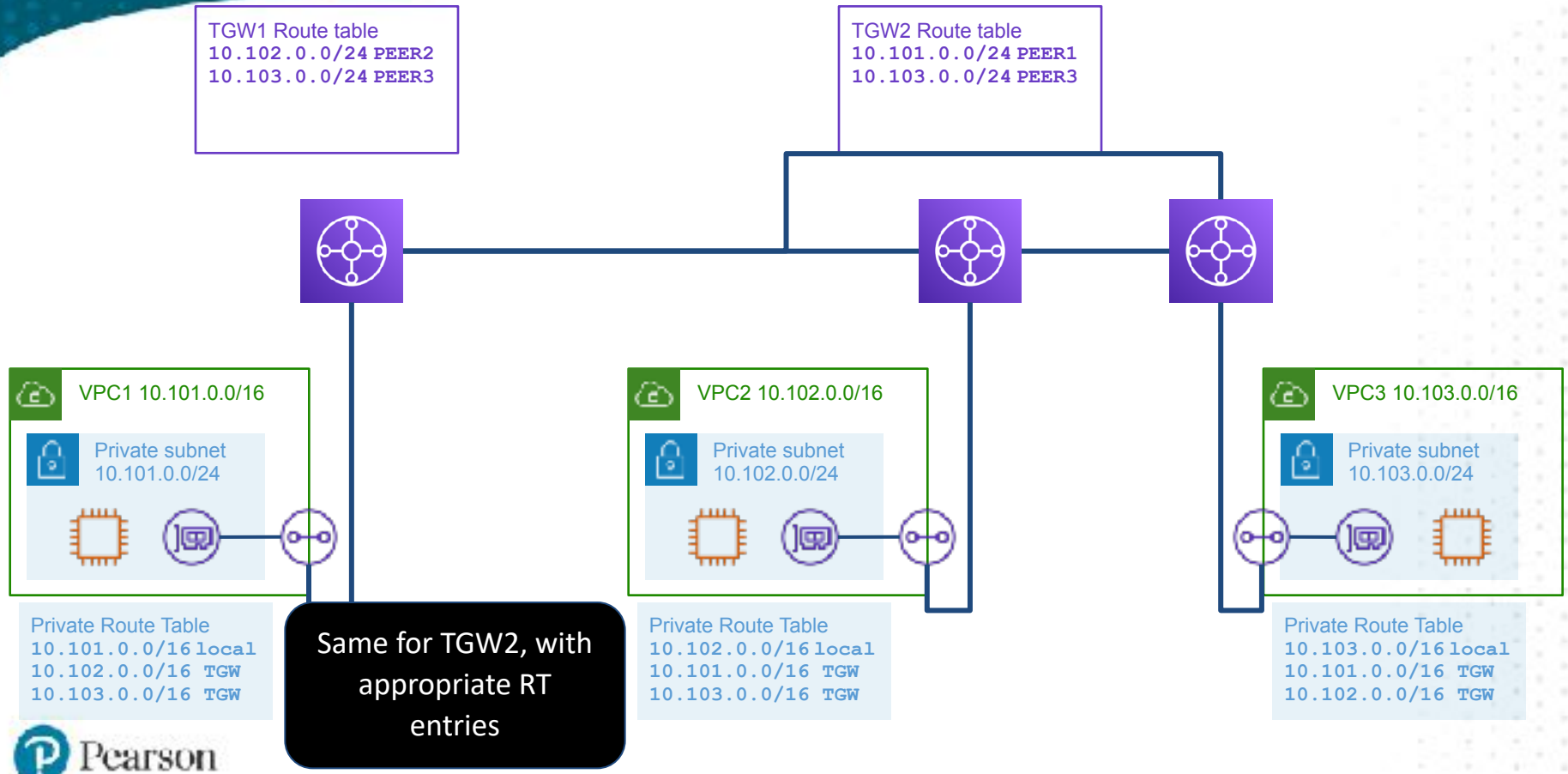


Pearson

Multi-Region TGW Infrastructure

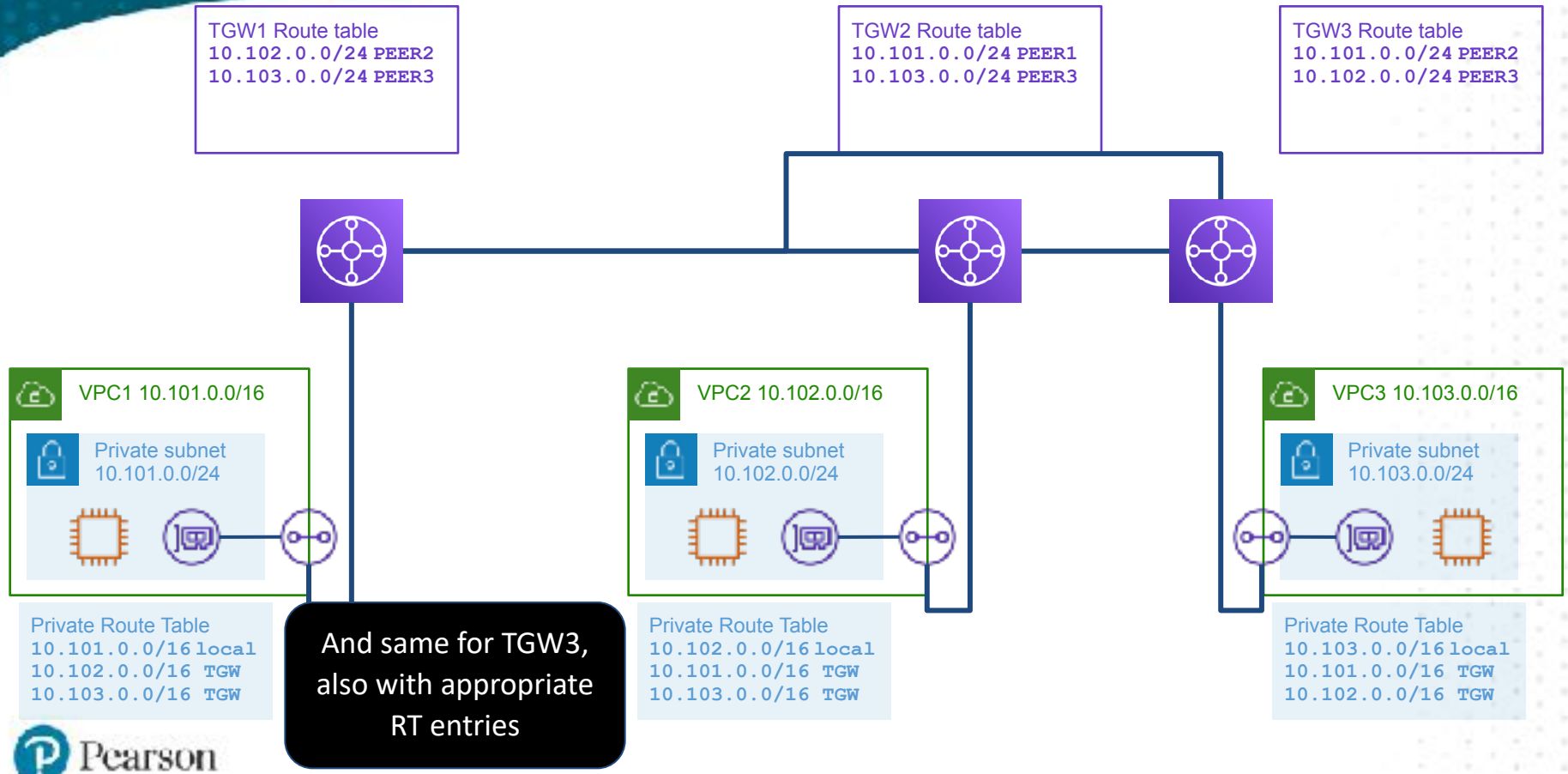


Multi-Region TGW Infrastructure



Pearson

Multi-Region TGW Infrastructure



Pearson

Cross-region Peering Considerations

- TGW route table entries
- Requestor and acceptor are separate resources in separate regions
- What about DNS resolution?
- Can we deploy all at once?

Demonstration

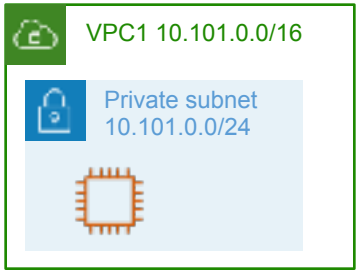
Deploy Transit Gateway Infrastructure with cross-region peering

Multi-Region Network Connectivity using VPC Peering

VPC Peering Basics

- Attach 2 VPC networks
- Same region
- Cross region
- Same account
- Cross account
- No overlapping CIDR

Multi-Region VPC Peering Infrastructure

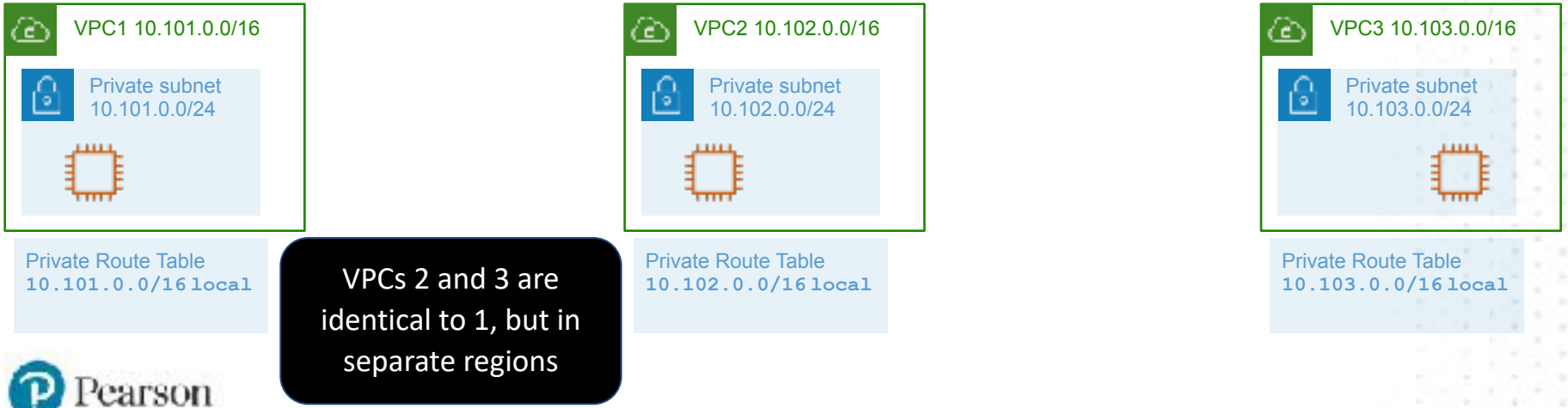


Private Route Table

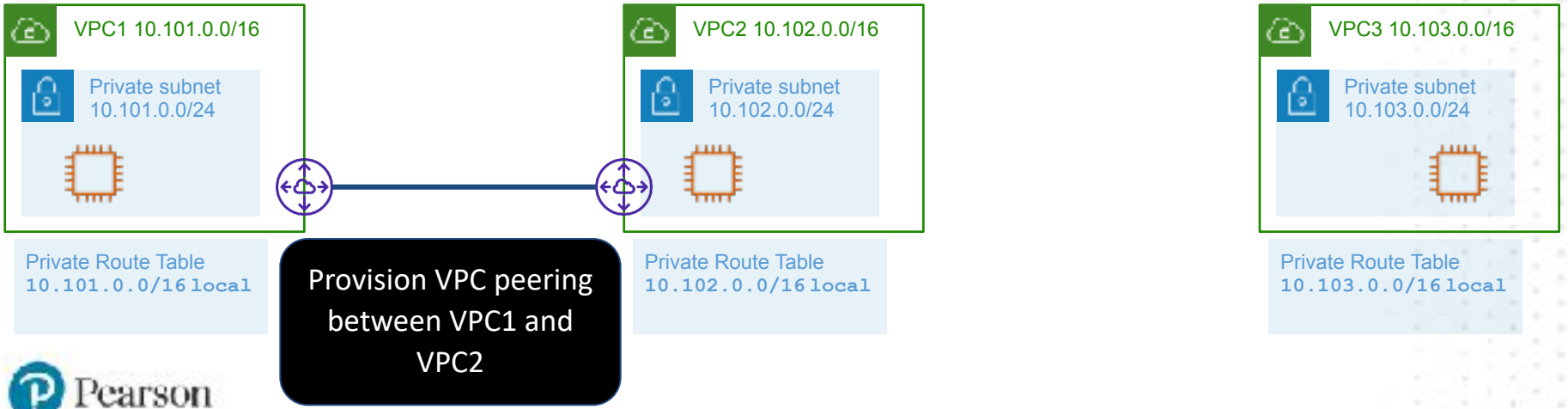
```
10.101.0.0/16 local
```

VPC1 (only showing private subnet)

Multi-Region VPC Peering Infrastructure

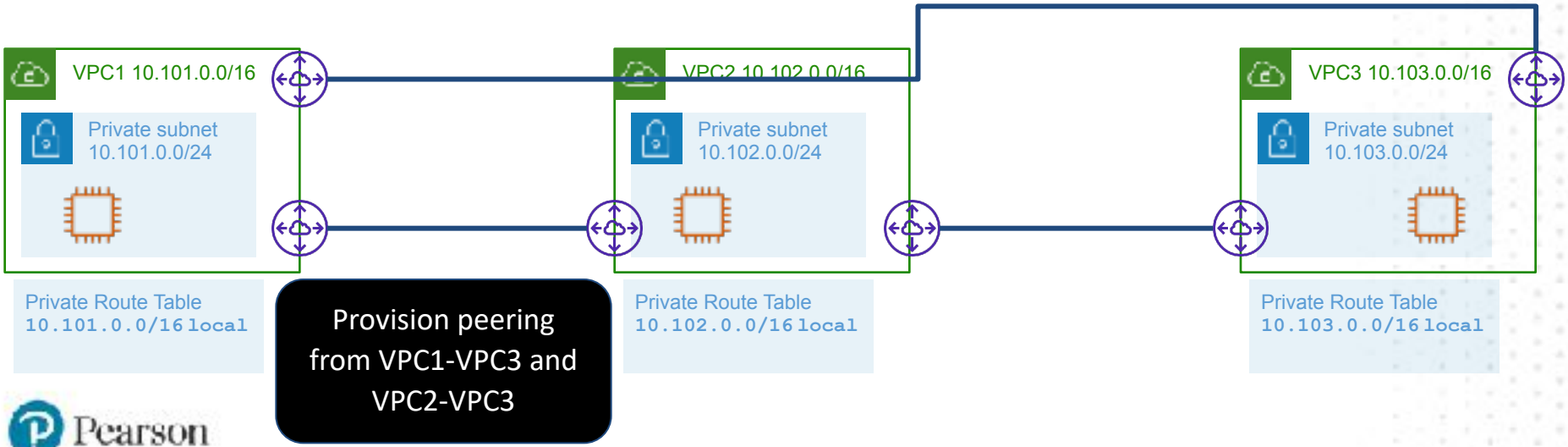


Multi-Region VPC Peering Infrastructure



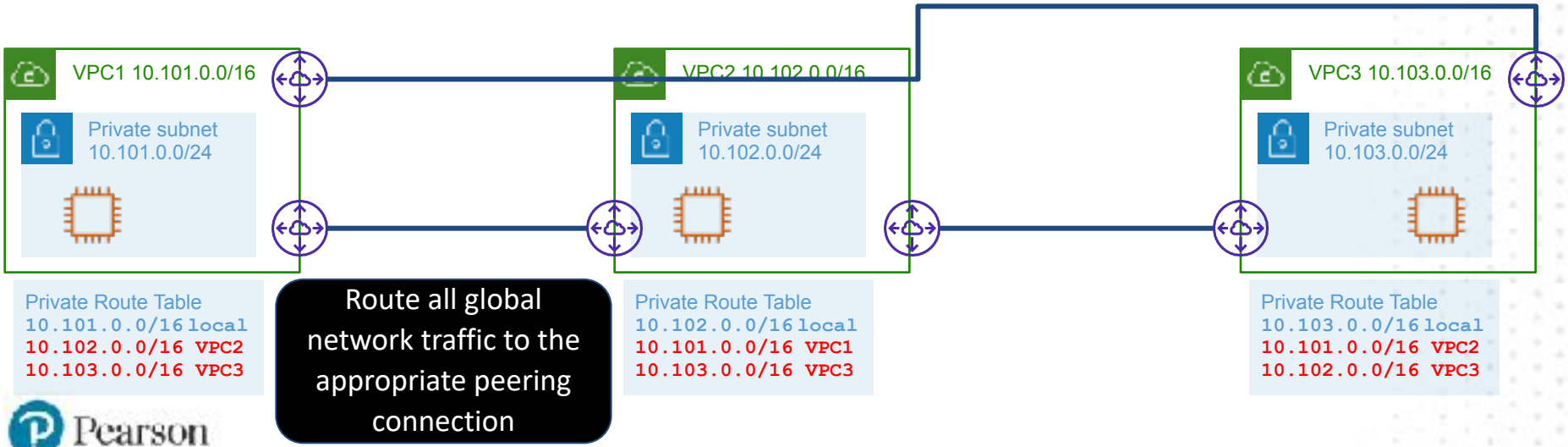
Multi-Region VPC Peering Infrastructure

Accepting a peering connection is a separate action from initiating the peering connection and must be accepted from the remote region!



Pearson

Multi-Region VPC Peering Infrastructure



VPC Peering vs TGW Comparison

	VPCp	TGWp
Auto SR accept	✓	✓

VPC Peering vs TGW Comparison

	VPCp	TGWp
Auto SR accept	✓	✓
Auto CR accept		

VPC Peering vs TGW Comparison

	VPCp	TGWp
Auto SR accept	✓	✓
Auto CR accept		
Route table entry	✓	✓

VPC Peering vs TGW Comparison

	VPCp	TGWp
Auto SR accept	✓	✓
Auto CR accept		
Route table entry	✓	✓
Attach to subnet		✓

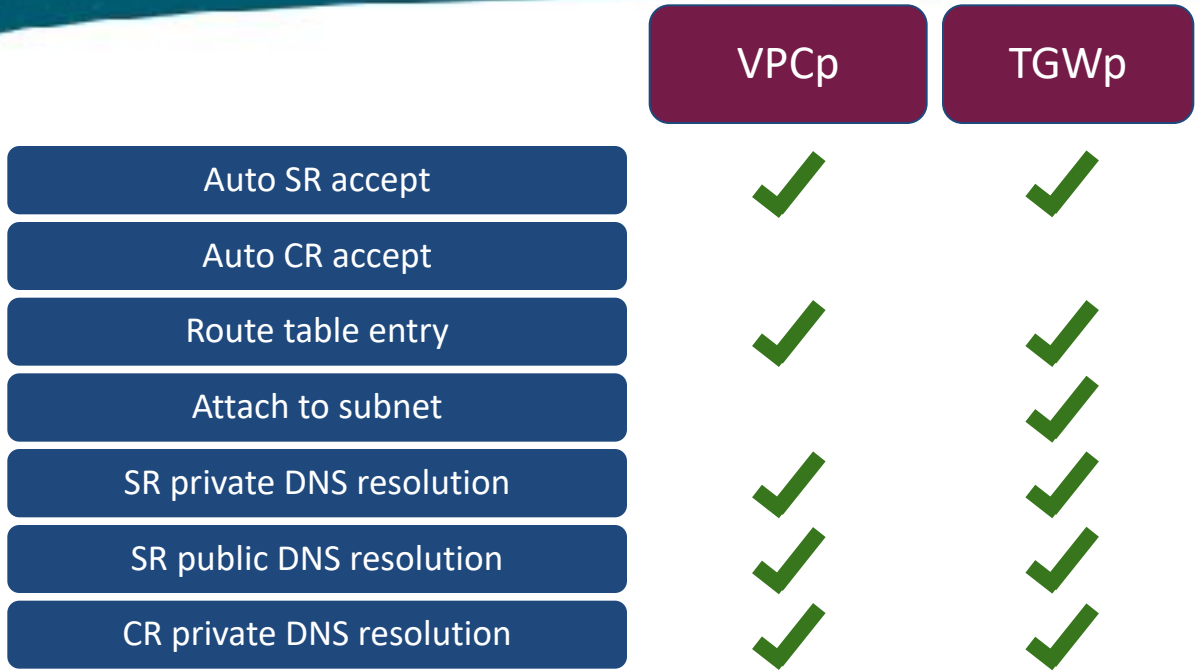
VPC Peering vs TGW Comparison

	VPCp	TGWp
Auto SR accept	✓	✓
Auto CR accept		
Route table entry	✓	✓
Attach to subnet		✓
SR private DNS resolution	✓	✓

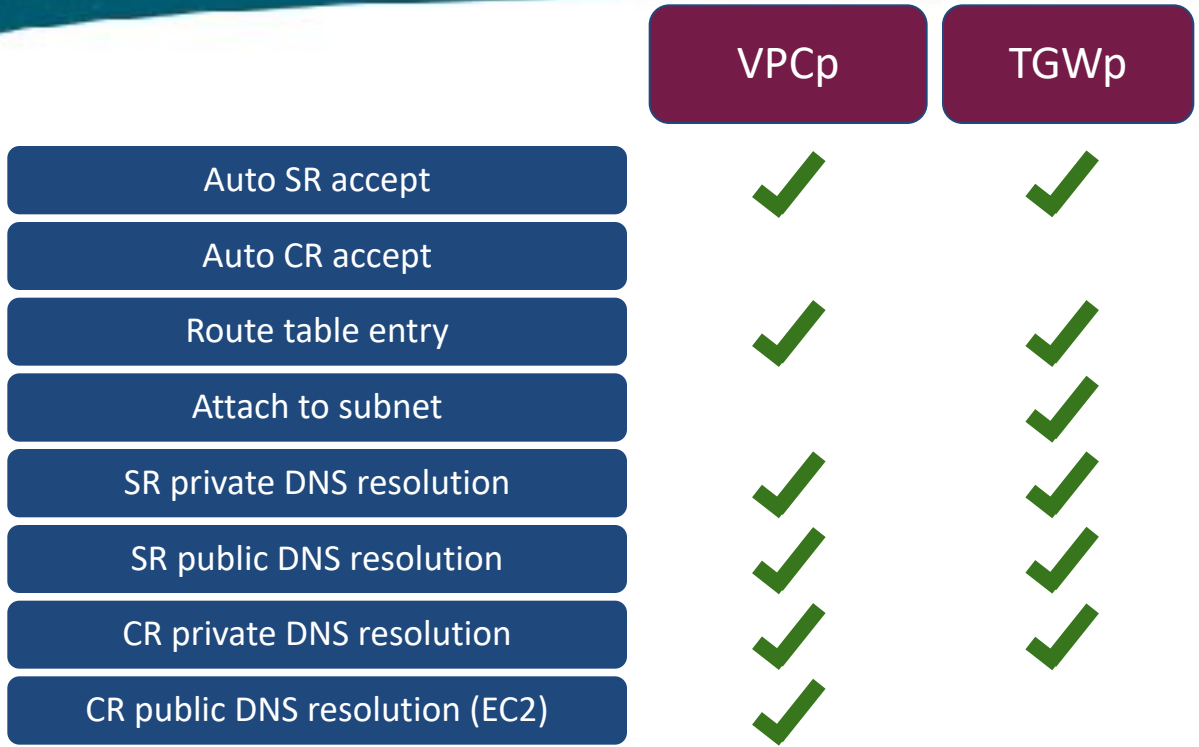
VPC Peering vs TGW Comparison

	VPCp	TGWp
Auto SR accept	✓	✓
Auto CR accept		
Route table entry	✓	✓
Attach to subnet		✓
SR private DNS resolution	✓	✓
SR public DNS resolution	✓	✓

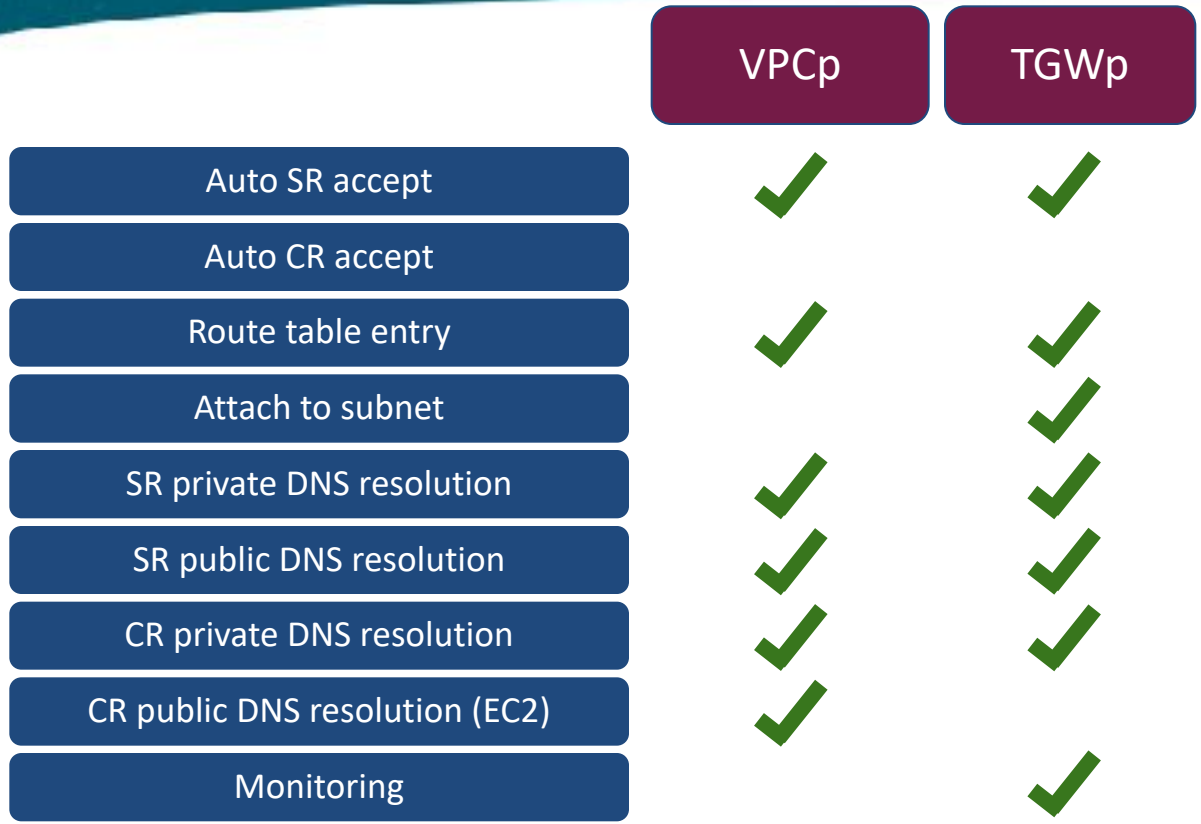
VPC Peering vs TGW Comparison



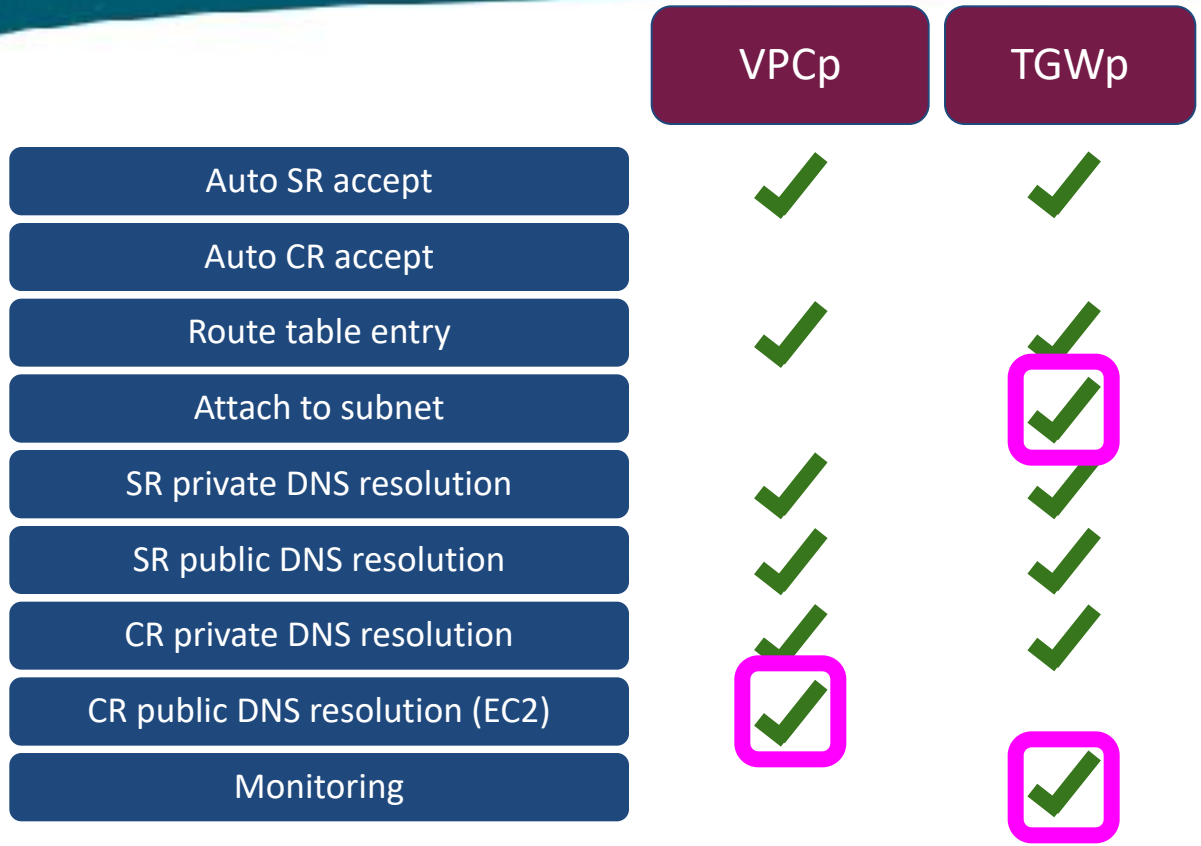
VPC Peering vs TGW Comparison



VPC Peering vs TGW Comparison



VPC Peering vs TGW Comparison



What about VPC Peering in Terraform?

- Same-region peers
- Cross-region peers
- Does NOT handle route table entries!

Let's create a small module for handling route table entries for accepter and requestor

Demonstration

De-provision Transit Gateway and replace with VPC Peering

Deprovision resources in the correct order

Bonus Discussion

Bonus Discussion Scenario

Your company's engineering team created two VPC networks and now want to connect them. They are ok with using any native AWS networking feature to accomplish this.

However, the two VPC network ranges are identical, which eliminates VPC peering as a possibility.

How can you implement the network connectivity, considering the overlapping CIDR ranges?

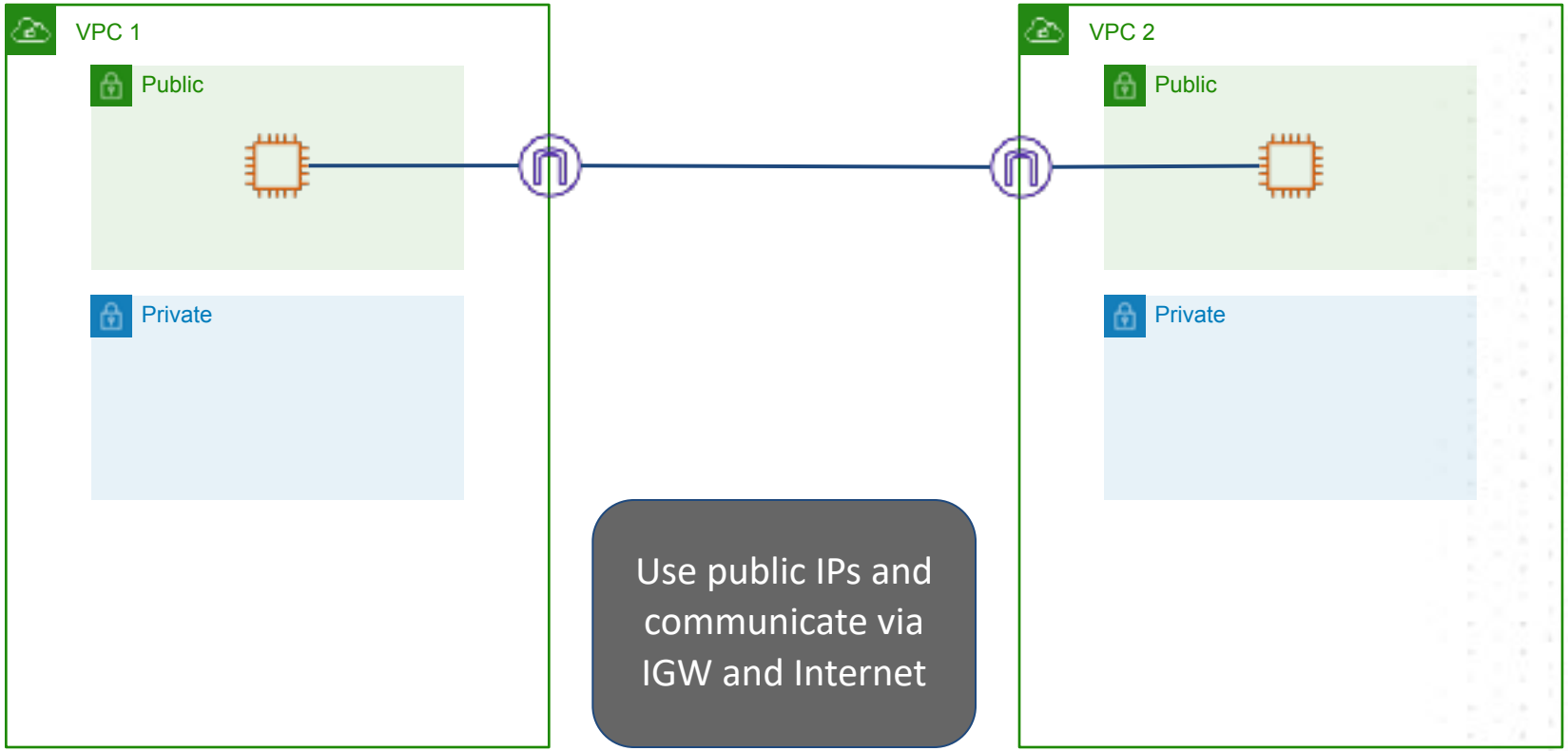
Scenario Questions to Ask

- Why are overlapping CIDR ranges so problematic?
- What are the AWS connectivity options between two VPC networks?
- Do any of those connectivity options support overlapping networks?

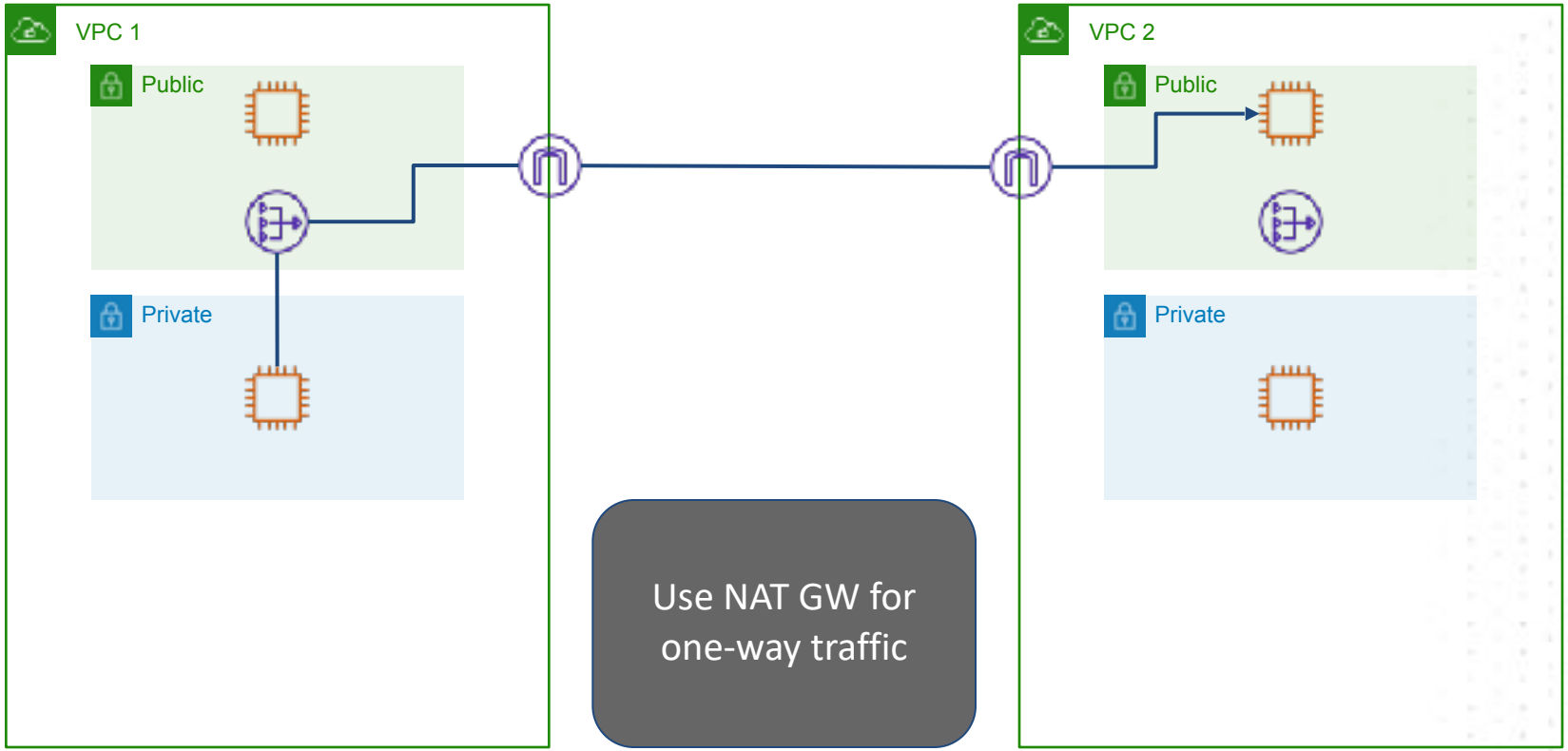
Overlapping CIDR Range Challenges

- Route tables cannot have identical cidr ranges and different destinations
- Double NAT may be required
- VPC Peering is eliminated as a choice

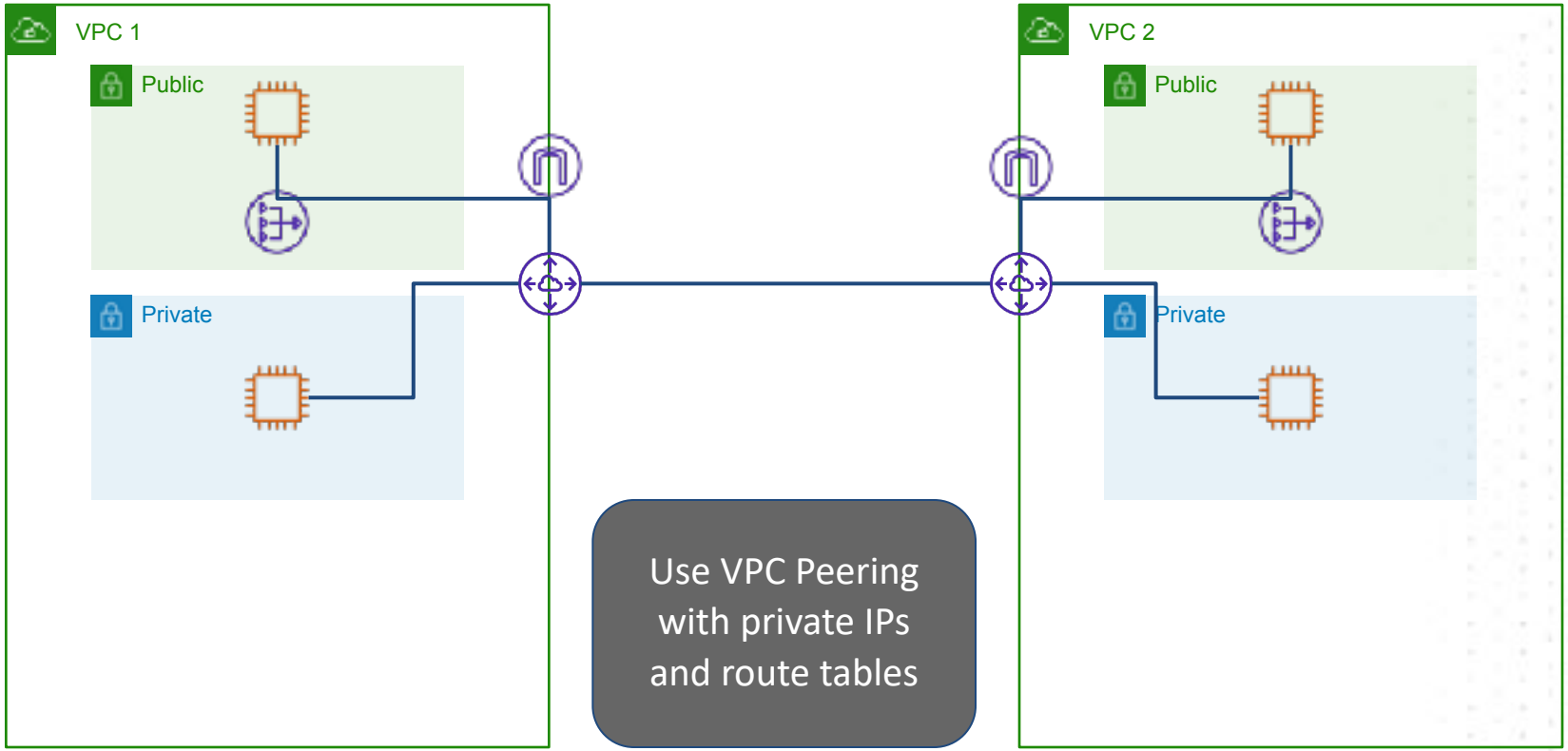
VPC to VPC Connectivity Options



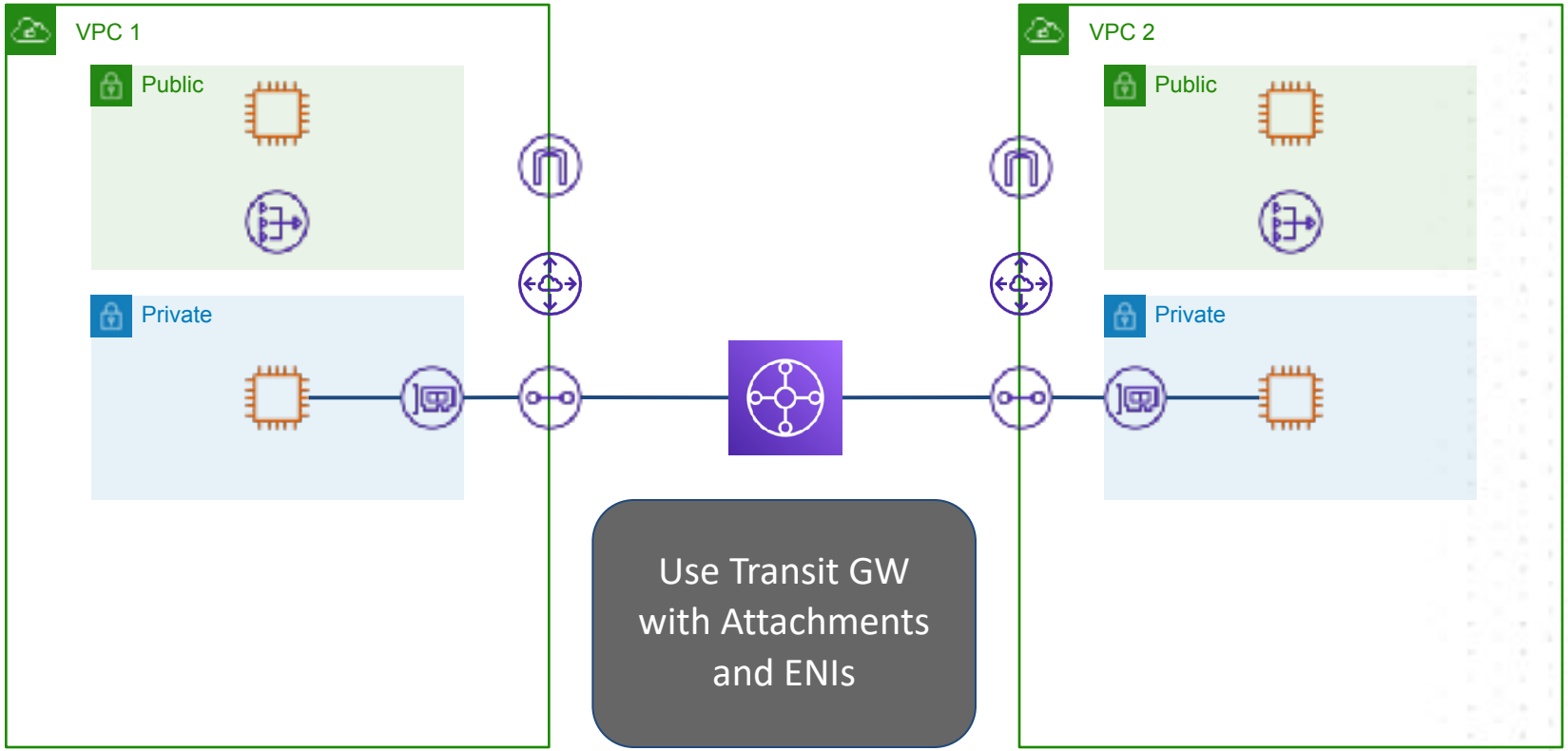
VPC to VPC Connectivity Options



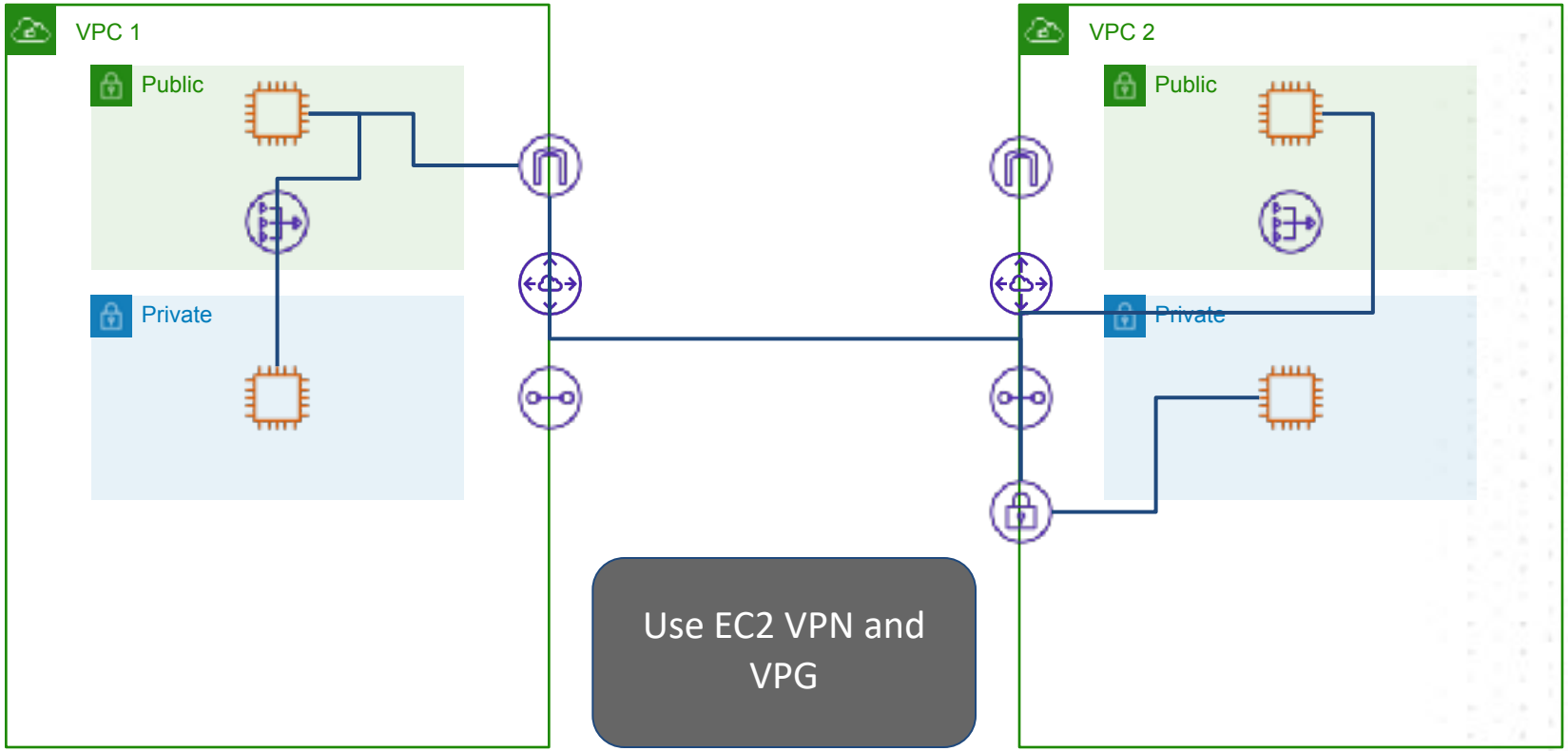
VPC to VPC Connectivity Options



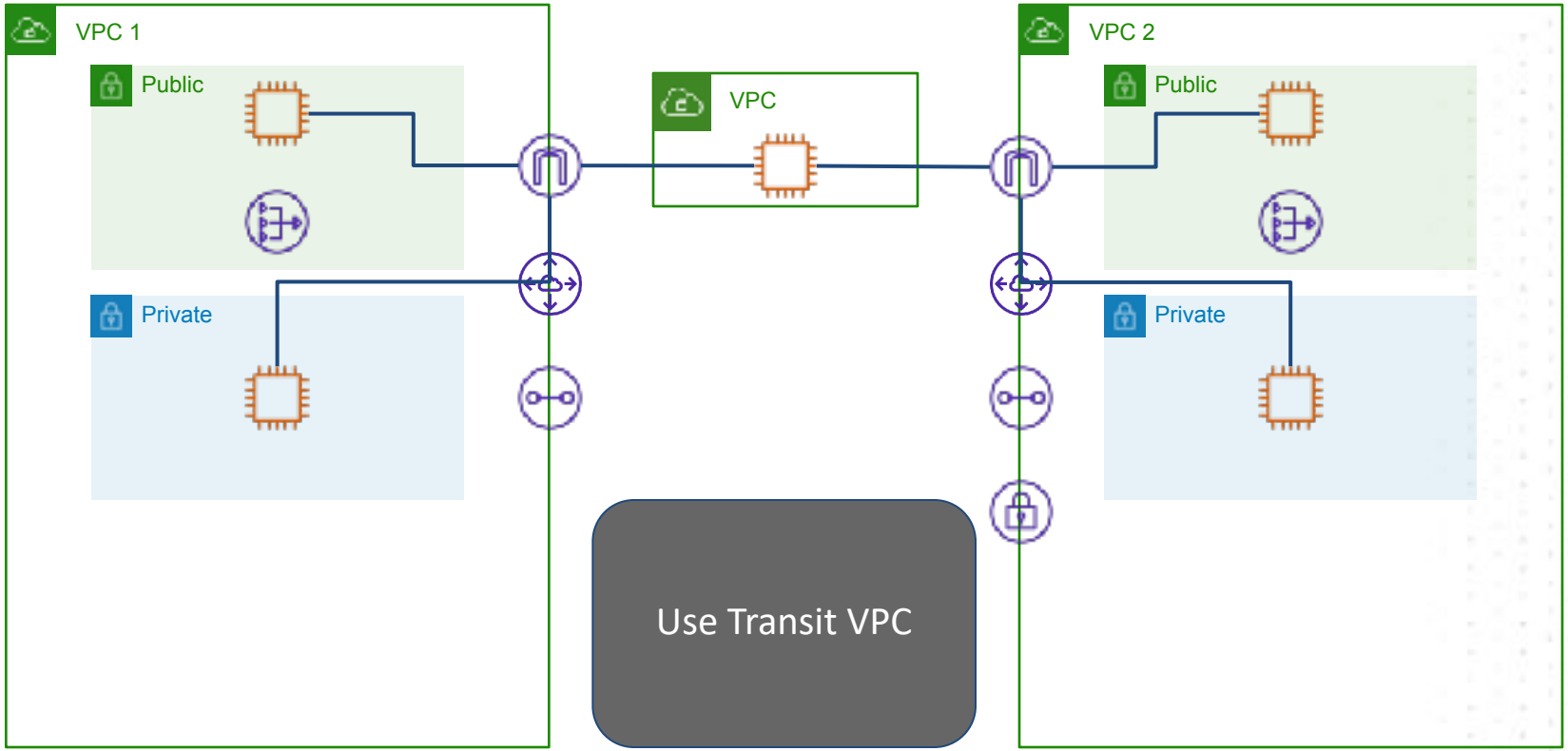
VPC to VPC Connectivity Options



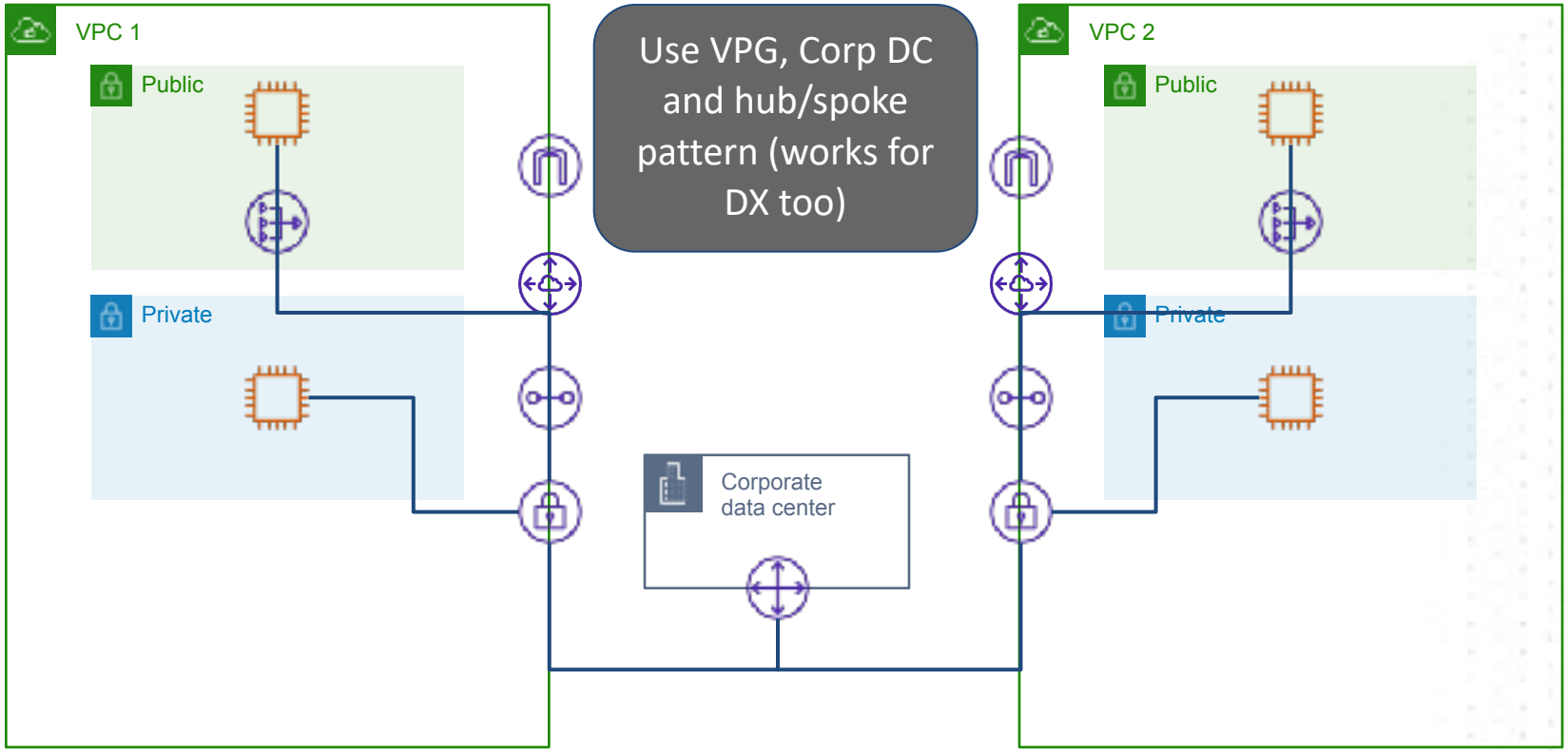
VPC to VPC Connectivity Options



VPC to VPC Connectivity Options



VPC to VPC Connectivity Options



Overlapping CIDR Range Support

Public IP/IGW

Overlapping CIDR Range Support

Public IP/IGW	✅
Private IP/NAT GW 1-way	✅

Overlapping CIDR Range Support

Public IP/IGW	✅
Private IP/NAT GW 1-way	✅
Private IP/VPC Peering	⛔

Overlapping CIDR Range Support

Public IP/IGW	
Private IP/NAT GW 1-way	
Private IP/VPC Peering	
Private IP/TGW	

Overlapping CIDR Range Support

Public IP/IGW	✅
Private IP/NAT GW 1-way	✅
Private IP/VPC Peering	⛔
Private IP/TGW	❓
EC2 VPN/VPG	⛔

Overlapping CIDR Range Support

Public IP/IGW	✅
Private IP/NAT GW 1-way	✅
Private IP/VPC Peering	⛔
Private IP/TGW	❓
EC2 VPN/VPG	⛔
Transit VPC	❓

Overlapping CIDR Range Support

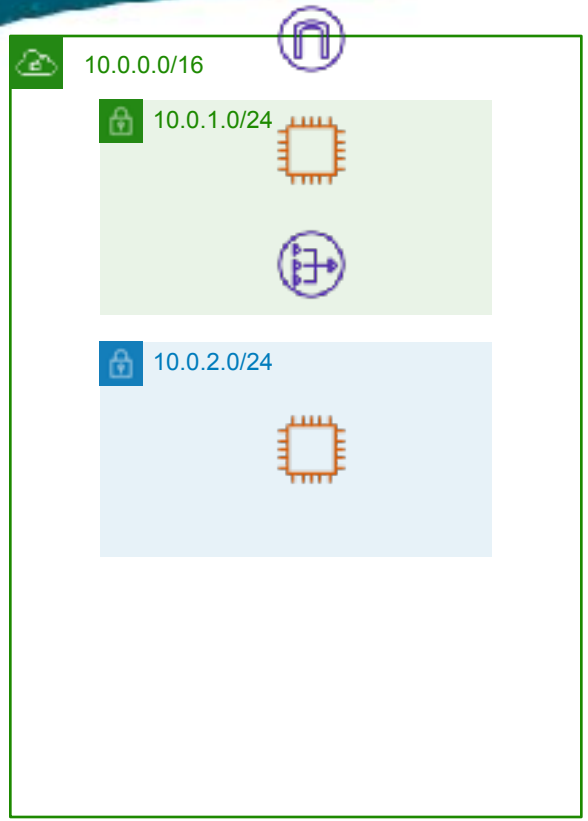
Public IP/IGW	✅
Private IP/NAT GW 1-way	✅
Private IP/VPC Peering	⛔
Private IP/TGW	❓
EC2 VPN/VPG	⛔
Transit VPC	❓
VPG(and/or DX) and Corp DC	❓

Overlapping CIDR Range Support

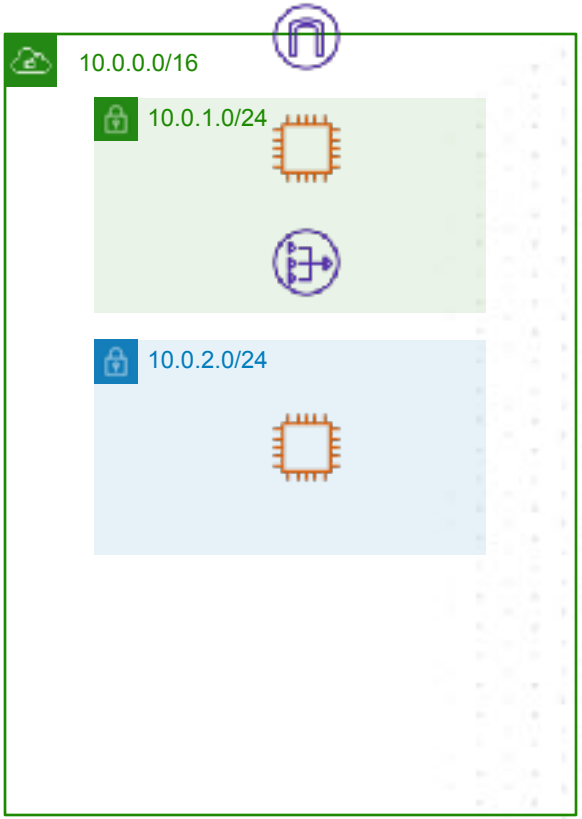
Public IP/IGW	✅
Private IP/NAT GW 1-way	✅
Private IP/VPC Peering	⛔
Private IP/TGW	❓
EC2 VPN/VPG	⛔
Transit VPC	❓
VPG(and/or DX) and Corp DC	❓

Let's explore this!

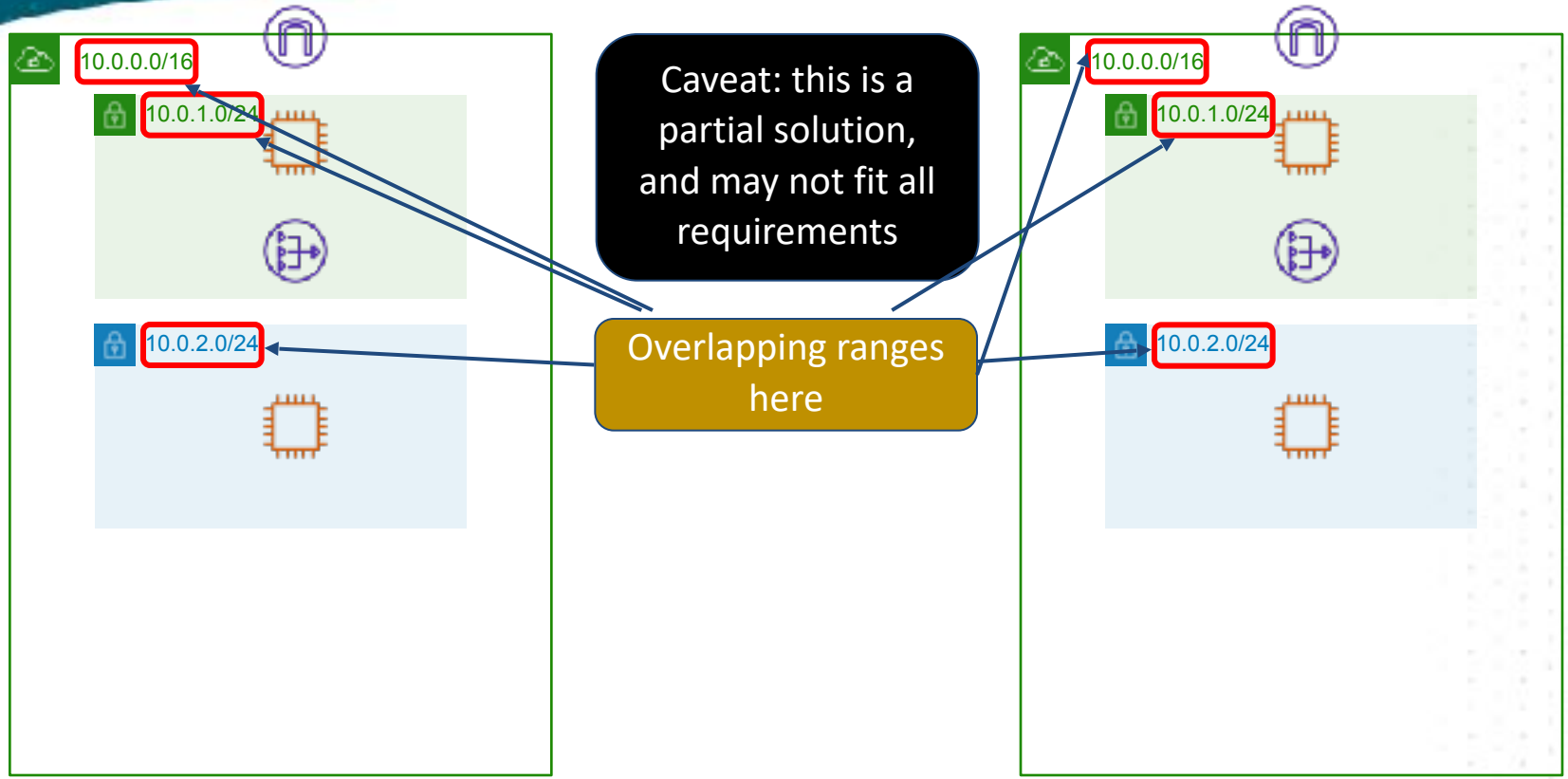
Transit Gateway and Overlapping CIDR Range



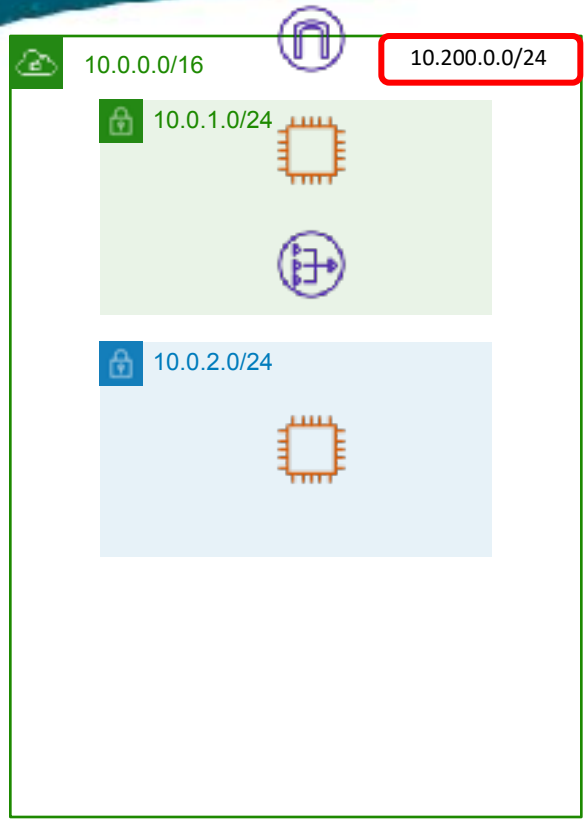
Caveat: this is a partial solution, and complicated to implement!



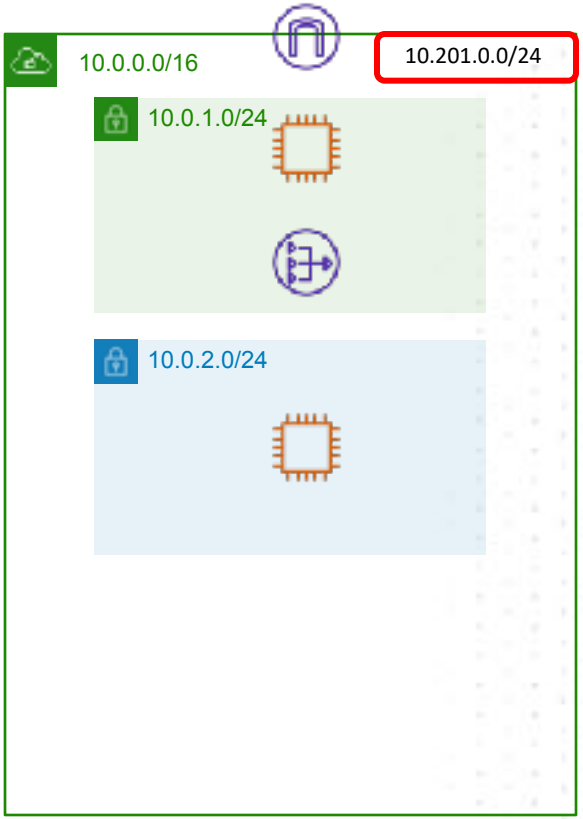
Transit Gateway and Overlapping CIDR Range



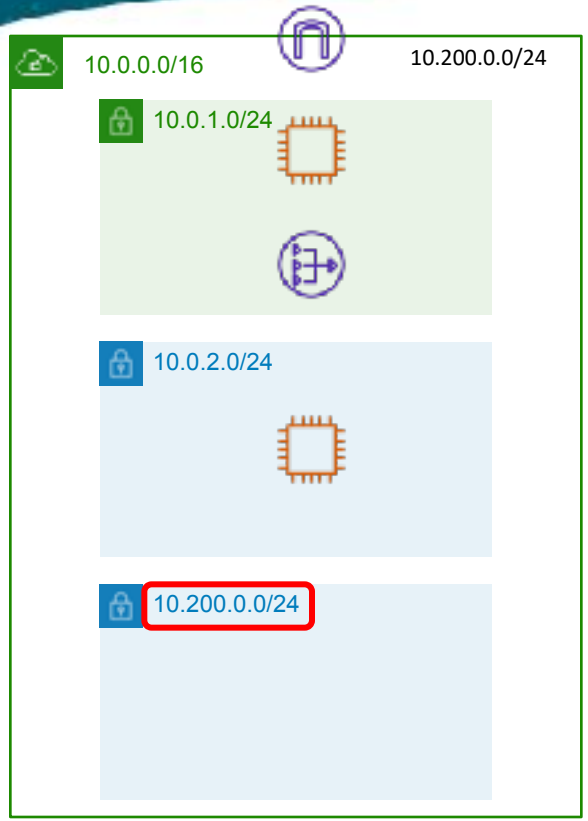
Transit Gateway and Overlapping CIDR Range



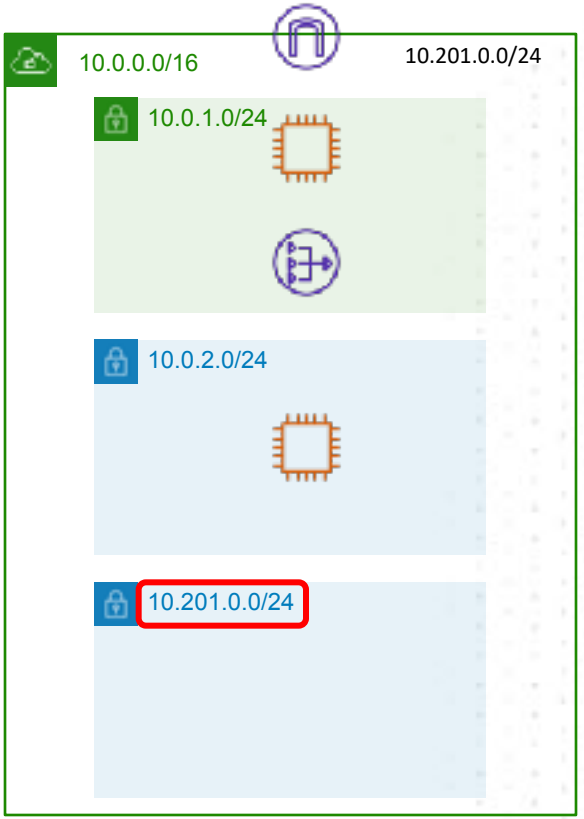
Add a new CIDR range to each VPC within the same base range but not overlapping



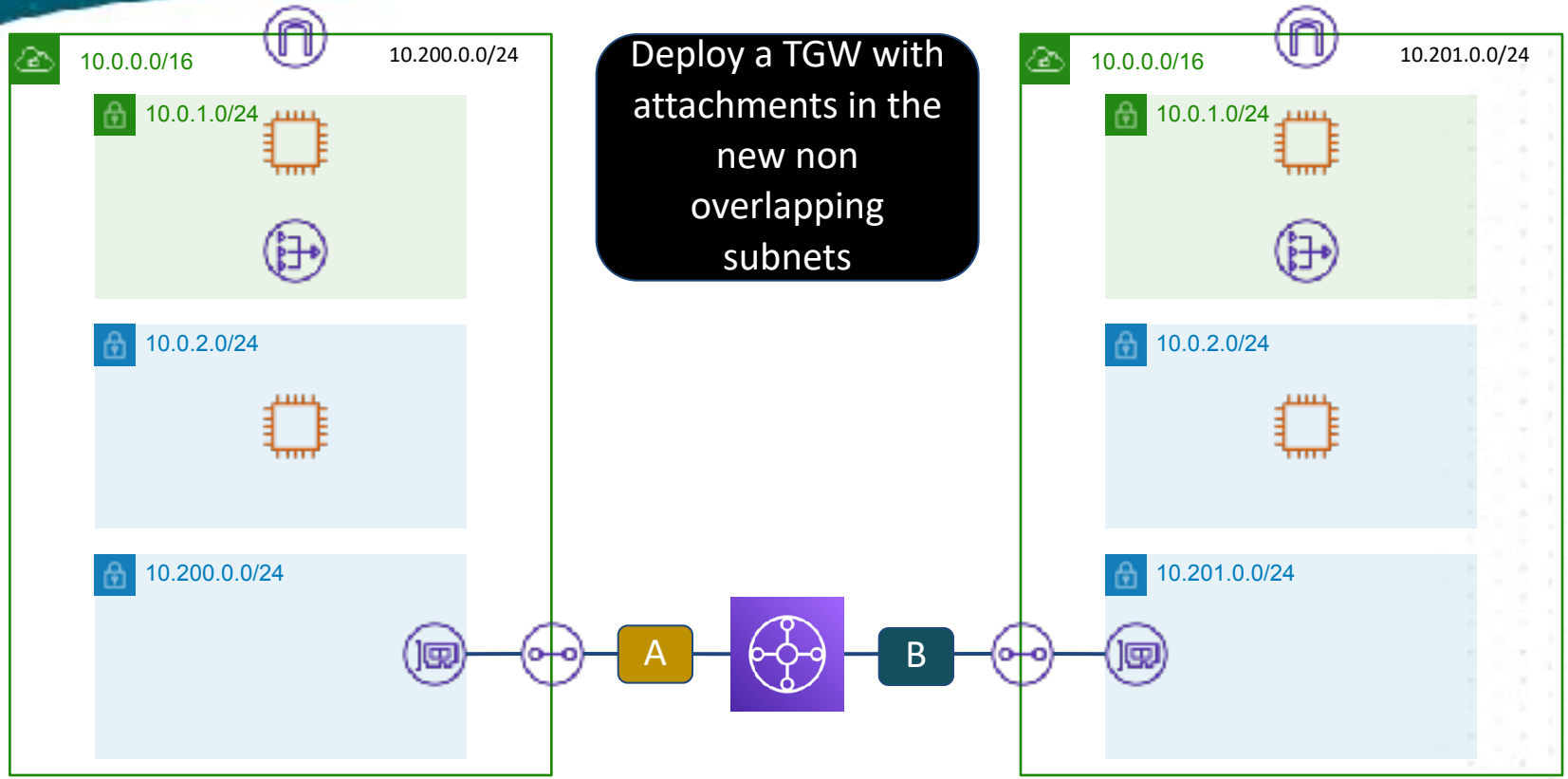
Transit Gateway and Overlapping CIDR Range



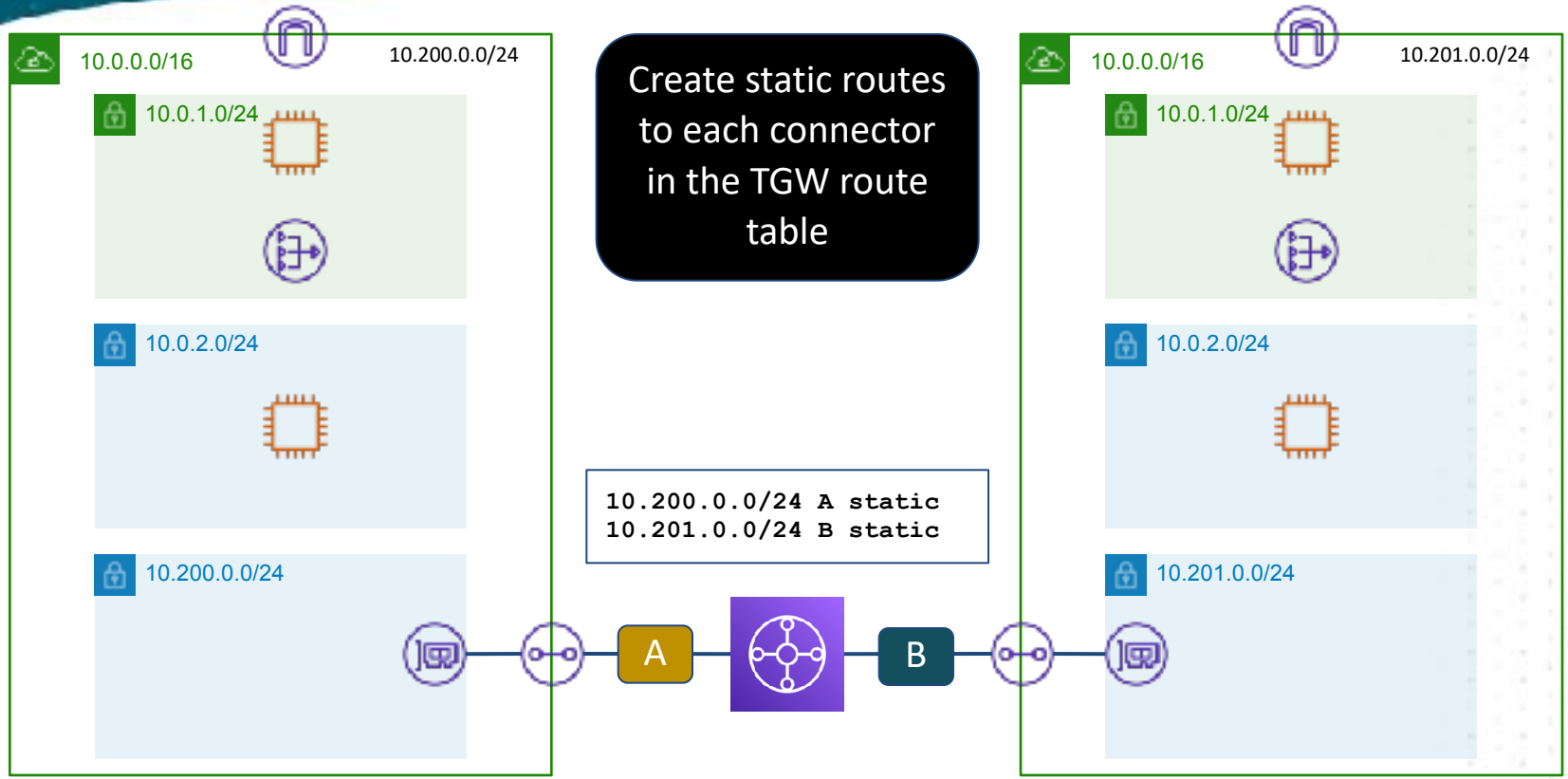
Create a new subnet in each VPC using the new secondary ranges



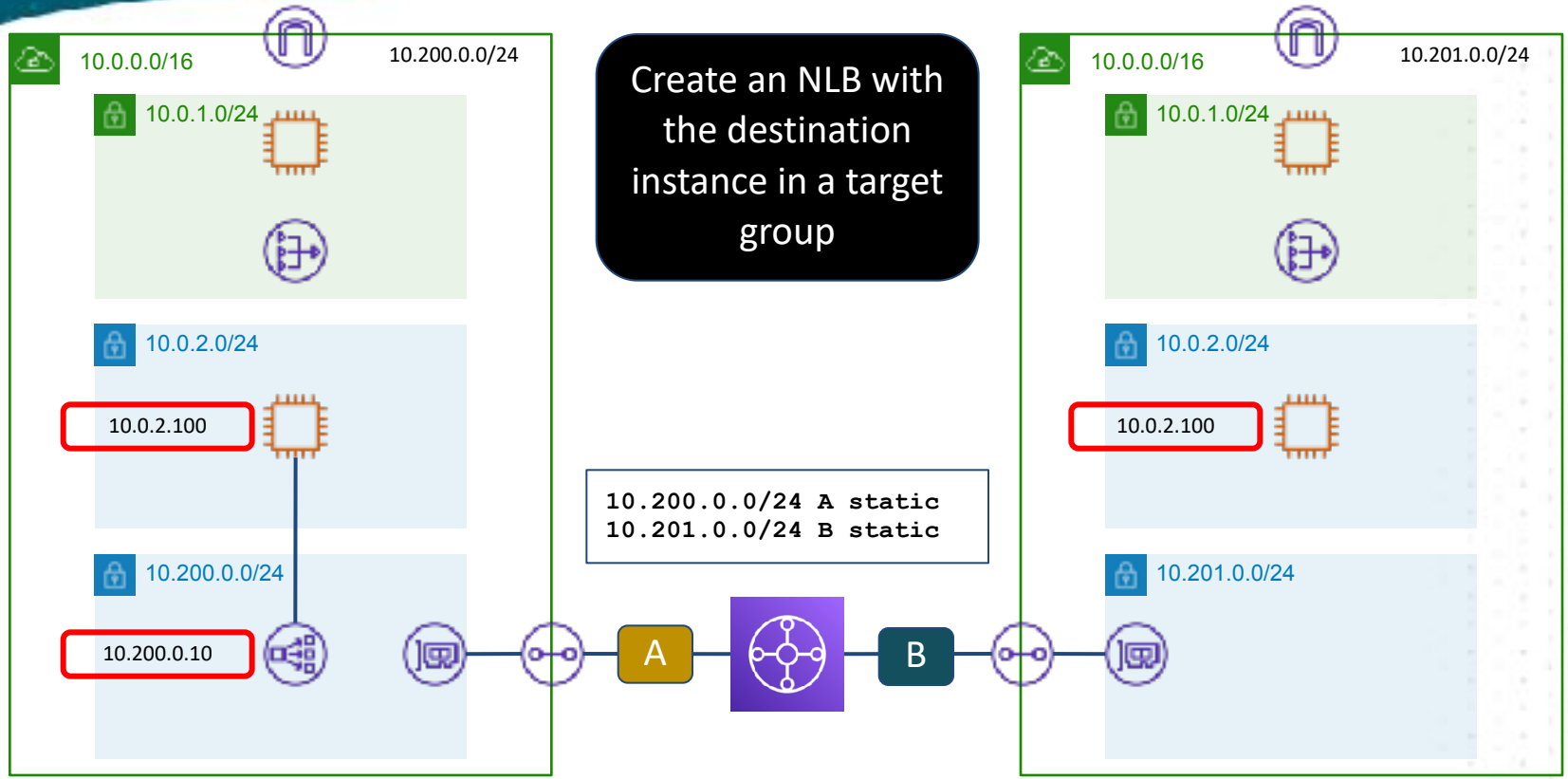
Transit Gateway and Overlapping CIDR Range



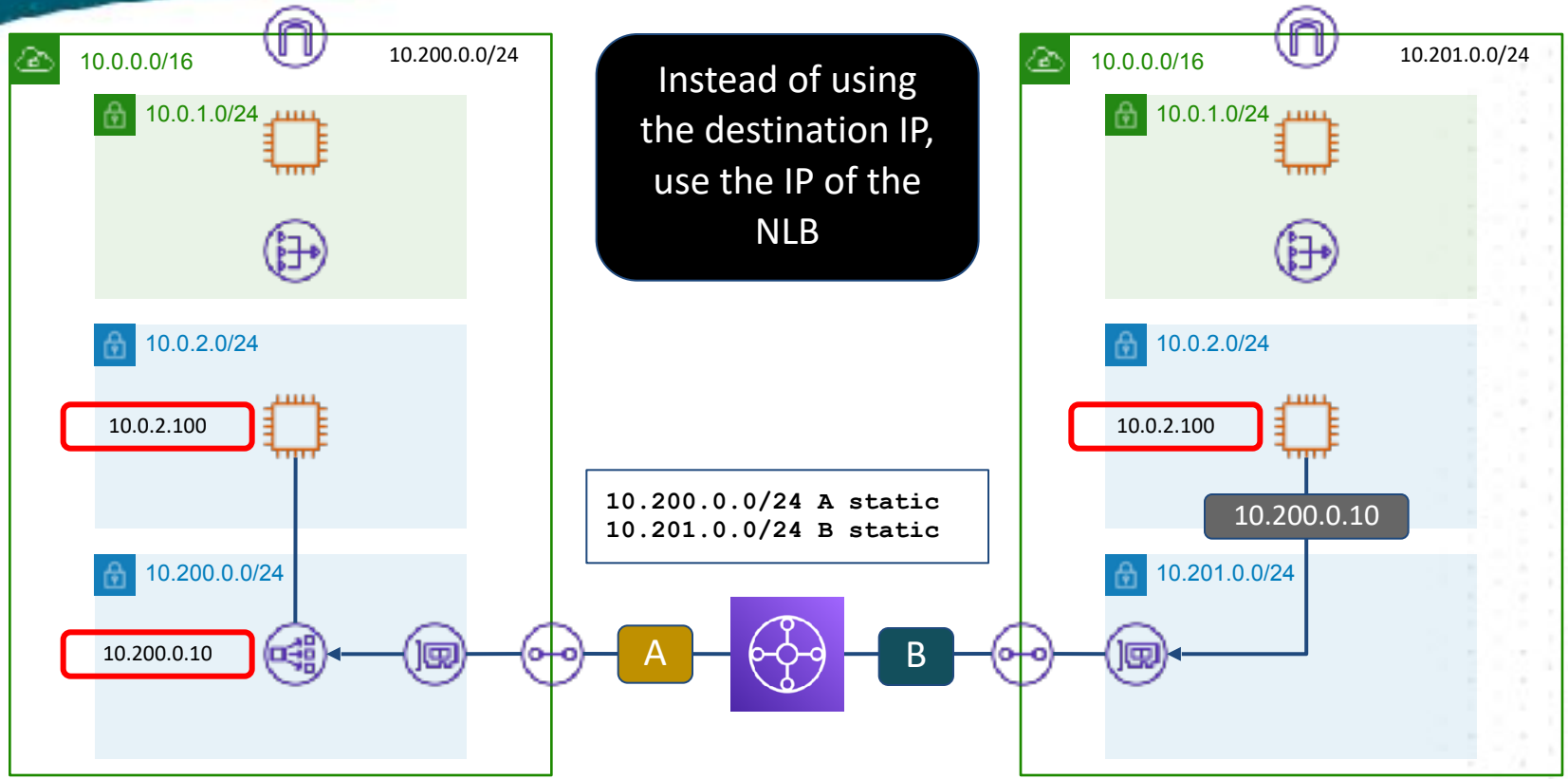
Transit Gateway and Overlapping CIDR Range



Transit Gateway and Overlapping CIDR Range

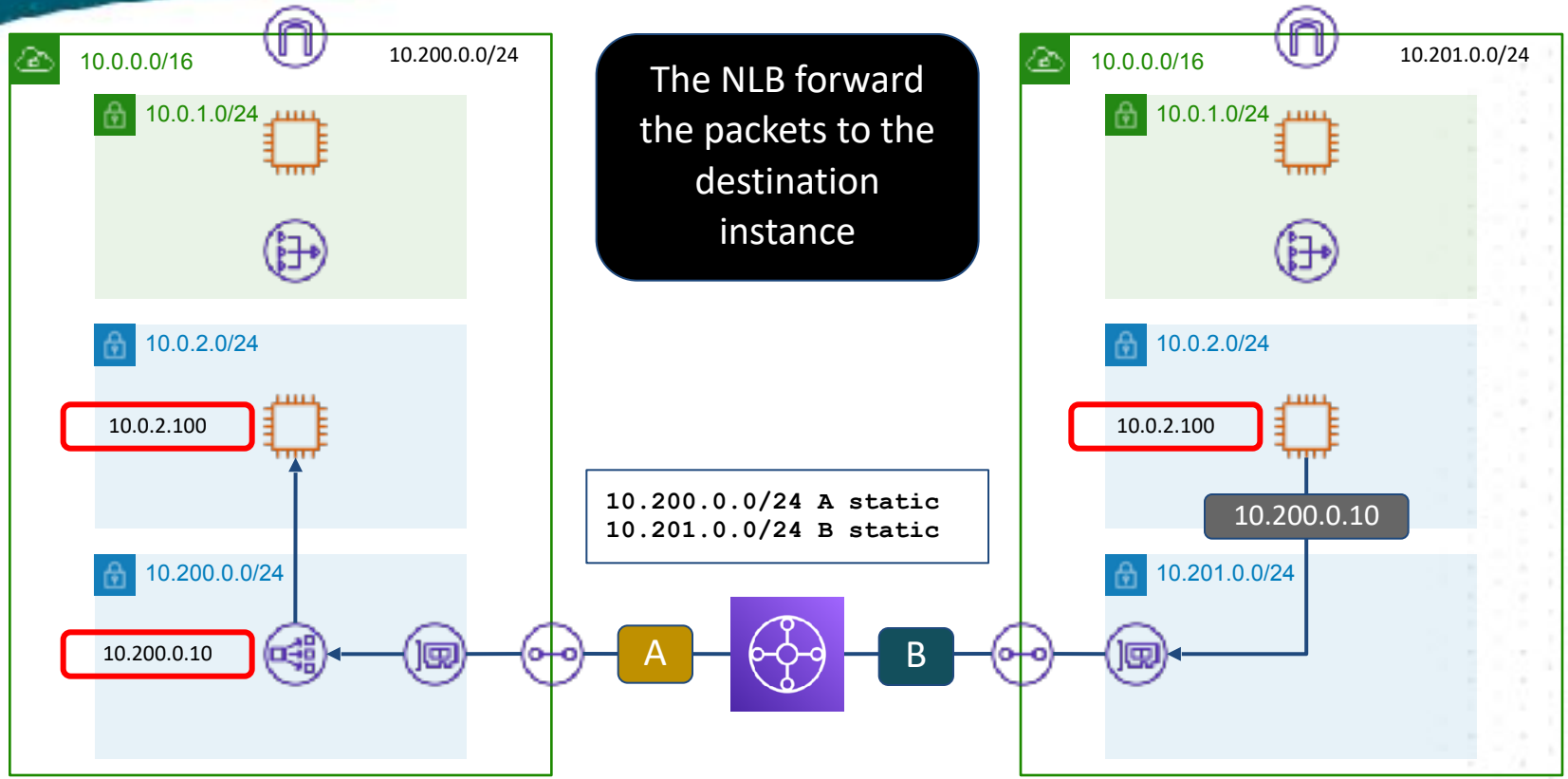


Transit Gateway and Overlapping CIDR Range

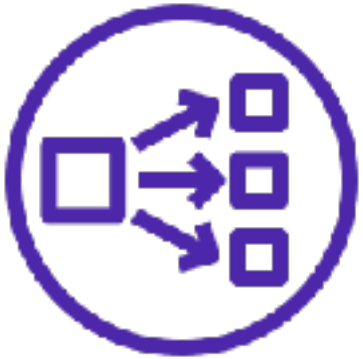


Pearson

Transit Gateway and Overlapping CIDR Range



TGW->NLB Considerations



- Requires 1 NLB per EC2 instance (~16USD/month)
- Only allows 1-way traffic (need NLBs in both networks for 2-way)
- Scalable solution (somewhat)
- Better solution: create 1 new VPC with non-overlapping range and move resources into it

Q&A