MODULE 8 · STUDY NOTES

Connecting Networks

Connecting VPCs and on-premises networks with VPC peering, Transit Gateway, Site-to-Site VPN, and Direct Connect — and the Well-Architected principles that decide the choice

A single VPC rarely stays alone for long. As a business grows it separates workloads into many VPCs — for security, architecture, or management — and enterprises can run hundreds that still need to reach one another and reach on-premises data centers. This module is the connectivity toolkit for those situations. It starts inside the AWS Cloud: how to scale from a handful of VPCs (full mesh) to thousands (hub-and-spoke) with AWS Transit Gateway, and how VPC peering makes a simple, free, one-to-one link between two VPCs. It then reaches across the boundary to on-premises networks two ways — AWS Site-to-Site VPN, which builds encrypted IPsec tunnels over the public internet, and AWS Direct Connect, a dedicated private line for consistent performance. Throughout, the real work is route tables: which destinations point at a transit gateway, a peering connection, a virtual private gateway, or an internet gateway. The module closes by mapping every option to the AWS Well-Architected Framework, because a cloud architect always works backward from the business need — failover and bandwidth, cost and performance, and protecting data in transit — to choose the right components.

LEARNING OBJECTIVES

- Describe how to connect an on-premises network to the AWS Cloud
- Describe how to connect multiple VPCs in the AWS Cloud
- Connect VPCs in the AWS Cloud by using VPC peering
- Describe how to scale VPCs in the AWS Cloud
- Use the AWS Well-Architected Framework principles when connecting networks

8.1 Scaling your VPC network with AWS Transit Gateway

Once you can build a VPC and its components, real architectures push you to multiple VPCs — separating logical elements for security, architecture, or management simplicity — and most of those VPCs still need to talk to each other. There are two ways to wire many networks together. In a **full mesh**, every VPC connects directly to every other VPC; a network of N nodes requires `N × (N − 1) / 2` connections, so four VPCs need six connections and 100 VPCs need `100 × 99 / 2 = 4,950`. Mesh delivers fast, low-latency paths but becomes an enormous operational and maintenance burden as it grows. In a **hub-and-spoke** design, a central hub manages connectivity and each node needs only one connection to the hub, so 100 VPCs need only 100 connections. Hub-and-spoke trades a little hub-processing latency for a design that scales.

FULL MESH ARCHITECTURE	HUB-AND-SPOKE ARCHITECTURE
every node connected to every node	*a central hub manages connectivity*
▪ Requires `N × (N − 1) / 2` connections (4 VPCs → 6, 100 VPCs → 4,950)	▪ Each node needs only one connection to the hub (100 VPCs → 100)
▪ Fast, low network latency — direct point-to-point paths	▪ Significantly simpler management, much less operational effort
▪ Best for a small number of VPCs	▪ Best for a large number of VPCs
▪ Heavy operational and maintenance effort for network engineers	▪ Adds some latency for hub processing — Transit Gateway is the AWS hub

AWS Transit Gateway is that hub: a centralized, Regional router built on the hub-and-spoke model. You create and manage only a single connection from the central gateway into each VPC, data center, or remote office, and any newly attached VPC is automatically reachable by every other attached network. It is a managed service that **auto-scales based on traffic throughput** and can connect **thousands** of VPCs and on-premises networks. It supports both static and dynamic routing and both IPv4 and IPv6. Transit gateways can be **peered** with each other within a Region or across Regions and accounts, and Transit Gateway traffic always stays on the **global AWS backbone** — it never traverses the public internet, which reduces exposure to common exploits and DDoS attacks. Billing is per hour for the number of connections plus the amount of traffic that flows through the gateway. **Transit Gateway Flow Logs** capture IP traffic to and from the gateway and can publish to Amazon CloudWatch Logs, Amazon S3, or Amazon Kinesis Data Firehose for traffic visibility.

Connectivity is realized in **route tables**. First, create a VPC attachment, which links the transit gateway to the VPC through an elastic network interface deployed in the subnets — select at least one subnet per Availability Zone so every AZ has an interface. Then add a route in each VPC's route table sending traffic bound for the other VPCs to the transit gateway. A wildcard CIDR like `10.0.0.0/8` is handy here: it covers the individual `10.x.0.0/16` VPC blocks, so one route reaches them all (this is a routing wildcard, not a VPC CIDR — a VPC CIDR block maxes out at `/16`). Finally, the transit gateway's own route table sends each VPC's CIDR to the corresponding attachment. A transit gateway can have multiple route tables, and each attachment associates with one of them.

TABLE 8.1 Centralized routing with Transit Gateway (single Region)

ROUTE TABLE	DESTINATION	TARGET
VPC 1	`10.1.0.0/16`	local
VPC 1	`10.0.0.0/8`	Transit gateway ID (wildcard reaches all VPCs)
Transit gateway	`10.1.0.0/16`	Transit gateway attachment VPC 1
Transit gateway	`10.2.0.0/16`	Transit gateway attachment VPC 2

The same hub can centralize more than east-west VPC traffic. In a **centralized outbound (egress) routing** pattern, a dedicated egress VPC holds a NAT gateway; application VPCs send their `0.0.0.0/0` internet traffic to the transit gateway, which forwards it to the egress VPC, whose route table points to the NAT gateway and then an internet gateway. This gives one controlled gate for outbound monitoring and security, and it is cheaper to run NAT gateways in one VPC than in every VPC (run one per AZ for redundancy). The same idea centralizes shared services such as traffic inspection or interface VPC endpoints in a single designated service VPC.

To span **Regions or accounts**, create a **transit gateway peering** attachment on your transit gateway that targets another transit gateway; the accepter (owner of the target) must accept the request. VPC CIDRs propagate into each transit gateway's default route table, and you add a **static route** on each side pointing to the peering attachment. The peered gateway can live in a different Region or account, giving seamless VPC-to-VPC communication across accounts and Regions with efficient, reliable, and encrypted inter-Region transfers that never touch the public internet.

> **WORKED EXAMPLE – GROUP OF DEPARTMENTS**
>
> A company has multiple IT departments, each with its own VPC — some in the same AWS account and others in a different account — and all should be peered so departments have full access to each other's resources, with room to add other business groups later. The lowest-maintenance solution is to connect each department's VPC to a **transit gateway**; it provides adequate capacity for future growth and one attachment per VPC. (If instead one VPC needed access to only one other VPC, you could simply update route tables to limit traffic to that target VPC's IP range.) The module activity reinforces this: for five VPCs `10.1.0.0/16` through `10.5.0.0/16` attached to `tgw-1`, each VPC route table gets `10.0.0.0/8` → `tgw-1`, and the transit gateway route table gets one route per VPC CIDR pointing to that VPC's attachment.

> **COMMON PITFALL**
>
> Do not reach for full-mesh VPC peering when the VPC count is large. Mesh needs `N × (N − 1) / 2` connections and grows unmanageable fast; Transit Gateway's hub-and-spoke model needs one attachment per VPC and is the Well-Architected recommendation for connecting more than two VPCs.

8.2 Connecting VPCs in AWS with VPC peering

When you have only a **small number of VPCs**, or your budget is constrained by paying for a transit gateway, **VPC peering** establishes a one-to-one network connection between two VPCs. Peering is a feature of Amazon VPC, and **creating a peering connection incurs no cost**. Because it is a point-to-point peer, network overhead is very small and latency is low. Peering lets EC2 instances in two VPCs communicate by **private IP** as if they were on the same network, and you can peer your own VPCs, a VPC in another account, or a VPC in another Region. Inter-Region peering is a cost-effective way to share resources or replicate data for geographic redundancy; data across inter-Region or inter-AZ peering is charged at standard data-transfer rates. Peering traffic stays on the **AWS backbone** (never the public internet), is not shared with other peering connections, and inter-Region traffic is encrypted with no single point of failure or bandwidth bottleneck.

> **WHAT VPC PEERING GIVES YOU**
>
> - A **one-to-one** private connection between exactly two VPCs.
> - Communication by **private IP**, as if the VPCs shared a network.
> - Works across accounts and across Regions (inter-Region peering).
> - **No charge** for the connection; cross-AZ and cross-Region data transfer is charged.
> - Traffic stays on the AWS backbone; inter-Region traffic is encrypted.

Establishing a peering connection is a request-and-accept handshake: the owner of VPC A sends a request and the owner of VPC B must **accept** it (the target VPC may belong to a different account, or the requester may lack permission to accept on its behalf). The **CIDR blocks of the two VPCs must not overlap**. To make traffic flow, each side adds a route: VPC A's route table sends VPC B's range (for example `10.2.0.0/16`) to the A-B peering connection, and VPC B's route table sends VPC A's range (`10.1.0.0/16`) to the same peering connection. You may also need to update **security**

group rules so traffic to and from the peer VPC is allowed. You can scope routes tightly — to a single subnet CIDR, a specific CIDR when a VPC has several, or even one resource's IP (a `/32` host route to a single EC2 instance in the peer VPC).

The defining limitation is that **VPC peering is not transitive**. If VPC A is peered with VPC B, and VPC B is peered with VPC C, that does **not** connect A to C — they must be explicitly peered. This is a security feature: it isolates errors and limits the blast radius of network attacks, and it lets you control exactly which VPCs can communicate. This makes peering ideal for a **central VPC that provides a service** to many others without letting those others reach each other.

VPC PEERING LIMITATIONS

- **No transitive peering** — you must peer each pair of VPCs explicitly.
- VPCs with **matching or overlapping CIDR blocks cannot be peered**.
- If a VPC has an internet gateway or NAT gateway, the **peer VPC cannot use it** to reach the internet.
- **Only VPC owners** can work with their peering connections.

TABLE 8.2 Use cases – peering to one central VPC

CENTRAL VPC	HOW PEERING IS USED
File-sharing VPC	Peer many VPCs to the IT file-sharing VPC, but keep the peered VPCs from sending traffic to each other
Customer-shared VPC	Each customer peers with your VPC; customers cannot route to other customers' VPCs and are unaware of their routes
Active Directory VPC	Specific instances in peer VPCs get full access to central AD servers; the central VPC only routes responses back to those instances

Peering connects VPCs at the **network** level. To connect at the **application** level — or to connect VPCs whose IP ranges **overlap** — use **AWS PrivateLink**. PrivateLink privately exposes a service in a provider VPC to consumer VPCs in a Region, where **only the consumer VPCs initiate connections**. The provider runs a **Network Load Balancer** in front of its EC2 targets, and each consumer creates an interface (elastic network interface) endpoint to that load balancer. Because the connection is application-level, the consumer and provider VPCs can even share the same range (for example both `10.1.0.0/16`) — something peering forbids. No peering connection is required.

WORKED EXAMPLE – MODULE SAMPLE EXAM (PRIVATE S3 ACCESS)

An application on EC2 instances in a VPC reads sensitive data from Amazon S3 through S3's **public Regional endpoint over the internet**, and the security team wants that internet exposure gone using the **most efficient** network route. The answer is a **gateway VPC endpoint for Amazon S3**: it routes privately over the AWS network at **no additional cost**, with **no bandwidth or packet-size limits**. Why not the others? An **internet gateway** still traverses the internet; you **cannot reach S3 over a VPN** at all; and a **NAT gateway** adds extra network hops across the Region and is not the most efficient path. (Gateway endpoints serve S3 and DynamoDB via the route table; PrivateLink interface endpoints use an ENI for other services.)

COMMON PITFALL

Two peering myths sink exam questions. First, peering is **never transitive** — a shared middle VPC does not bridge the outer two. Second, peering **cannot** join overlapping CIDR blocks — when ranges overlap (or you need app-level access), the answer is **PrivateLink with a Network Load Balancer**, not another peering connection.

8.3 Connecting to your remote network with AWS Site-to-Site VPN

By default, instances you launch into a VPC **cannot communicate with your on-premises network** — you have to build the bridge. **AWS Site-to-Site VPN** is one option: it uses **IPsec** to create encrypted VPN tunnels between two locations, letting data pass between your customer network and AWS over the **public internet**. The on-premises side of the connection is the **customer gateway**; the AWS side is the **virtual private gateway** (or, alternatively, a **transit gateway** attachment). Each connection provides **two tunnels across multiple Availability Zones** that you can use simultaneously for high availability — stream primary traffic through one and keep the second for redundancy so traffic still reaches your VPC if a tunnel drops. Site-to-Site VPN can be stood up relatively quickly (available in hours), and you are charged for each **VPN connection-hour** the connection is provisioned and available.

TABLE 8.3 Creating a Site-to-Site VPN connection

STEP	ACTION
1. Customer gateway	Create a customer gateway to represent the on-premises device (physical or software); assign a BGP ASN if the device supports BGP
2. Virtual private gateway	Create a virtual private gateway with a custom or Amazon-default BGP ASN — different from the customer gateway's; a transit gateway can be used instead
3. Configure routing	Add routes pointing to the virtual private gateway; you can enable route propagation to add Site-to-Site VPN routes automatically
4. Update security groups	Allow SSH, RDP, ICMP, or other required protocols from the on-premises network
5. Create the VPN connection	Create the connection with two tunnels terminating in separate Availability Zones for high availability
6. Configure the device	Download the configuration file and use it to configure the customer gateway device

A Site-to-Site VPN supports **dynamic or static routing** depending on the customer gateway device. If the device supports **BGP**, choose dynamic routing — it advertises routes to the virtual private gateway automatically; otherwise choose static routing and specify the IP prefixes for your network. **AWS recommends BGP-capable devices** because BGP offers robust health checks that assist failover to the other VPN tunnel. Note that the VPN relies on internet connectivity for its tunnels and is not guaranteed to be available.

Large organizations often have **multiple on-premises sites** that need to reach each other. **AWS VPN CloudHub** connects them with a simple hub-and-spoke model, usable with or without Amazon VPC. It uses one **virtual private gateway** with **multiple customer gateways**, each with a **unique BGP ASN**; the customer gateways advertise their routes, which are received and re-advertised to every other BGP peer so each site can exchange data with the others. The remote sites **must not have overlapping IP ranges**.

Because the public internet can be unpredictable, you can use **AWS Global Accelerator** to **accelerate** a Site-to-Site VPN. An accelerated VPN routes traffic from your on-premises network to the nearest AWS **edge location**, then rides the AWS backbone to the transit gateway for better, more consistent response times. The key limitation: **acceleration is supported only for VPNs attached to a transit gateway, not a virtual private gateway**.

You can also give on-premises **full access to your VPCs while keeping the VPCs isolated from one another** using a single transit gateway with **multiple route tables**. Each attachment associates with one route table; attachments on an isolated routing table can route to each other but not to attachments on a different isolated router. In a typical setup, a **transit gateway VPN route table** (associated with the VPN attachment) has routes to each VPC, while the **transit gateway VPC route table** (associated with the VPC attachments) has a route only back to the VPN attachment — so

VPC-1-to-on-prem traffic flows, but VPC-1-to-VPC-2 traffic reaches the transit gateway and is dropped because no route exists.

> **COMMON PITFALL**
>
> Watch the acceleration boundary: **Global Accelerator accelerates a Site-to-Site VPN only when it is attached to a transit gateway**, never a virtual private gateway. And remember every VPN connection ships with **two tunnels** across AZs — a single-tunnel design is a distractor.

8.4 Connecting to your remote network with AWS Direct Connect

Site-to-Site VPN sends data through encrypted tunnels over the public internet. **AWS Direct Connect** (DX) goes beyond the internet: it uses **virtual local area networks (VLANs)** to establish a **dedicated, private network connection** from your on-premises network to AWS, extending your private network to include AWS resources. Its benefits are **increased bandwidth throughput** and a **more consistent, predictable network experience** than internet-based connections.

TABLE 8.4 Direct Connect use cases

USE CASE	WHY DIRECT CONNECT FITS
Hybrid environments	Reach on-premises equipment such as a local database while gaining AWS elasticity and economics
Large datasets	HPC and big transfers avoid competing for internet bandwidth; the high-bandwidth link reduces congestion, and the reduced DX data-transfer rate lowers cost
Predictable performance	Real-time feeds such as audio or video streams get the consistent performance a dedicated line provides
Security and compliance	Policies requiring private circuits are satisfied — traffic flows over the dedicated private connection, not the internet

Direct Connect uses **Direct Connect locations** and industry-standard **802.1Q VLANs** to extend your network to AWS; from a supported location you can reach any VPC or public AWS service in any Region (except China). You choose between two connection types: a **dedicated connection**, a physical fiber-optic cable provisioned for your exclusive use, or a **hosted connection**, provisioned by a Direct Connect Partner and shared — offering more bandwidth options in smaller increments. Either way, provisioning takes planning and physical resources, so the implementation **typically spans multiple weeks**. Over a DX connection you create **virtual interfaces (VIFs)** to segment access.

TABLE 8.5 Direct Connect virtual interfaces

VIRTUAL INTERFACE	PROVIDES ACCESS TO	THROUGH
Public VIF	Public AWS services such as Amazon S3	The Direct Connect endpoint to public services
Private VIF	Your VPC	A virtual private gateway
Transit VIF	Your VPCs	A transit gateway, via a Direct Connect gateway

To simplify routing between on-premises and **many VPCs**, connect Direct Connect to a **transit gateway** instead of a virtual private gateway. The transit gateway has a **Transit Gateway Direct Connect attachment** pointing to a **Direct Connect gateway**, and that Direct Connect gateway can connect to **multiple transit gateways**; the path from the

Direct Connect location uses a **transit virtual interface**. This lets one dedicated connection fan out to many VPCs through the hub.

Direct Connect is fast and predictable but a single connection is still a single point of failure, so design for availability. For **high availability**, pair one or more Direct Connect connections (primary) with a lower-cost **backup VPN**: by default AWS always prefers the Direct Connect path, so no extra AWS configuration is needed to make DX primary — though you should configure internal route propagation so on-premises systems pick the right paths. For **high resiliency** on critical production workloads, use **multiple Direct Connect locations** — connect on-premises network 1 through location 1 and network 2 through location 2, so a hardware or whole-location failure doesn't sever connectivity. If you use multiple Regions, you need Direct Connect locations in at least two Regions.

HIGHLY AVAILABLE, RESILIENT CONNECTIVITY

- Use **Direct Connect as primary with a VPN backup**; AWS prefers DX by default.
- Use **multiple Direct Connect connections from different locations** for physical redundancy.
- Connect from **multiple on-premises data centers** to guard against a site or location failure.
- Prefer **dynamically routed, active/active** connections for automatic load balancing and failover.
- Provision **enough capacity** so one connection's failure doesn't overwhelm the others.

COMMON PITFALL

Don't confuse the two remote-connection options on timing and privacy. **Direct Connect is a private, dedicated line but takes weeks to provision** — a VPN is the option available in hours. And DX carries traffic privately off the internet, but you should still **enforce encryption in transit** (TLS or an IPsec VPN over the top) for sensitive data.

8.5 Applying the AWS Well-Architected Framework to network connectivity

The AWS Well-Architected Framework has **six pillars**, each with best practices and questions to weigh when architecting. When you connect multiple networks, ask how a workload stays **resilient, secure, high performing, and cost effective** across on-premises and cloud. Resiliency is a **shared responsibility**: AWS is responsible for the resiliency of the cloud network backbone, and you are responsible for implementing resiliency on premises and in the AWS Cloud. The table below maps this module's services to the most relevant best practices.

TABLE 8.6 Well-Architected best practices for network connectivity

PILLAR – FOCUS AREA	BEST PRACTICE	HOW THIS MODULE APPLIES IT
Reliability — Foundations	Provision redundant connectivity; prefer hub-and-spoke over mesh	Redundant Direct Connect from separate locations, VPN backup for DX; use Transit Gateway instead of many-to-many VPC peering
Security — Infrastructure protection	Control traffic at all layers (zero trust)	Establish private traffic with Direct Connect or Site-to-Site VPN
Security — Data protection	Authenticate communications; enforce encryption in transit	Use protocols with authentication (TLS, IPsec); TLS 1.3 recommended; VPN uses IPsec, DX for private paths
Performance Efficiency — Selection	Choose workload location; size dedicated connectivity or VPN	Estimate bandwidth/latency to size DX vs. VPN; DX for predictable performance; accelerate VPN with Global Accelerator
Cost Optimization — Cost-effective resources	Plan for data transfer; select components to optimize and reduce cost	Direct Connect's reduced transfer rate; CDNs and WAN/app optimization; AWS prefers DX over VPN for predictable connectivity

Foundations (plan your network topology). A resilient network anticipates failures and future traffic growth. Provision **redundant connectivity** between private networks in the cloud and on premises — a second VPN appliance if yours isn't resilient, redundant Direct Connect connections from **different locations**, or a VPN as a backup for Direct Connect. And **prefer hub-and-spoke topologies over many-to-many mesh**: use Transit Gateway rather than connecting more than two VPCs with peering, and avoid establishing multiple BGP sessions per VPC across Regions.

Infrastructure protection (protecting networks). Apply a **zero-trust** approach and security at all layers. **Control traffic at all layers** by establishing private network traffic: implement **Site-to-Site VPN** for private traffic over the internet, and **Direct Connect** for a private, dedicated line to AWS.

Data protection (protecting data in transit). Data in transit is any data sent between systems, and you protect its confidentiality and integrity. **Authenticate network communications** using protocols that support authentication — **TLS** (used across many AWS services) and **IPsec** (used in AWS VPNs) — which establishes trust between parties and reduces tampering or interception. **Enforce encryption in transit**: encrypt all data in transit with TLS (AWS recommends **TLS 1.3**), and protect network-to-network traffic with an IPsec VPN or Direct Connect to keep it private.

Selection (network architecture selection). The optimal solution varies and often combines approaches. **Choose your workload's location based on network requirements** — placing resources to reduce latency and improve throughput, and using a dedicated Direct Connect connection where on-premises users or applications benefit from avoiding internet congestion. **Choose appropriately sized dedicated connectivity or VPN**: estimate bandwidth and latency for the hybrid workload, then size the connection and its encryption accordingly.

Cost optimization (plan for data transfer). Include network costs in the workload cost benchmark and pick the pricing model that fits. **Select components to optimize data transfer cost** — content delivery networks to move data closer to users, a dedicated Direct Connect connection, and WAN or application optimization to move less data. **Implement services to reduce data transfer cost**: model your transfer to find the largest, highest-volume flows, and prefer **Direct Connect over VPN** where predictable connectivity and the reduced Direct Connect data-transfer rate lower cost.

COMMON PITFALL

Well-Architected questions reward the same instincts throughout this module: **redundancy** (two connections, two locations, DX + VPN backup), **hub-and-spoke over mesh** (Transit Gateway), **private + encrypted** paths (DX or VPN, plus TLS/IPsec), **right-sizing** the connection to real bandwidth and latency, and **planning data transfer** to control cost. Always work backward from the business need to the components — never the reverse.

Module summary

- Multiple-VPC designs scale two ways: full mesh directly links every pair (`N × (N − 1) / 2` connections) and suits a few latency-sensitive VPCs; hub-and-spoke centralizes on a hub and scales to thousands — AWS Transit Gateway is the AWS hub-and-spoke router.
- Transit Gateway is a centralized, Regional, auto-scaling router that connects thousands of VPCs and on-premises networks, supports static/dynamic and IPv4/IPv6 routing, can peer across Regions and accounts, keeps traffic on the AWS backbone, and charges per connection-hour plus throughput.
- VPC peering is a free, one-to-one, low-latency private link between two VPCs (any account or Region); it is not transitive, requires non-overlapping CIDRs, and needs both route tables (and often security groups) updated. Overlapping ranges or app-level access call for AWS PrivateLink with a Network Load Balancer.
- Site-to-Site VPN builds two IPsec tunnels across AZs over the public internet between an on-premises customer gateway and an AWS virtual private gateway (or transit gateway); it is quick to stand up, charged per connection-hour, supports BGP or static routing, and can be accelerated with Global Accelerator only when attached to a transit gateway.
- VPN CloudHub connects multiple on-premises sites hub-and-spoke through one virtual private gateway (unique BGP ASNs, no overlapping ranges); multiple Transit Gateway route tables can grant on-premises full VPN access while keeping VPCs isolated from each other.
- Direct Connect is a dedicated, private 802.1Q VLAN line delivering consistent performance, more bandwidth, and reduced data-transfer rates; it offers public, private, and transit virtual interfaces, connects to many VPCs via a Direct Connect gateway and transit gateway, and takes weeks to provision.
- For availability, use Direct Connect as primary with a VPN backup (AWS prefers DX by default); for resiliency, connect from multiple on-premises networks through multiple Direct Connect locations with dynamically routed active/active links and spare capacity.
- The Well-Architected Framework guides the choice: provision redundant connectivity and prefer hub-and-spoke (Reliability), control traffic at all layers (Security — infrastructure protection), authenticate and encrypt in transit with TLS 1.3/IPsec (Security — data protection), size connectivity and place workloads by network need (Performance Efficiency), and plan data transfer to optimize cost (Cost Optimization), often favoring Direct Connect over VPN.

Review questions

1. You must connect 50 VPCs with the least ongoing maintenance. Full mesh or Transit Gateway — and how many connections does each require?

Answer: Transit Gateway (hub-and-spoke) — one attachment per VPC, so 50 connections. Full mesh would need 50 × 49 / 2 = 1,225 connections and is far heavier to operate; Well-Architected prefers hub-and-spoke.

2. VPC A is peered with VPC B, and VPC B is peered with VPC C. Can A reach C? Why or why not?

Answer: No. VPC peering is not transitive — A can reach C only if A and C are explicitly peered. This isolation limits the blast radius and keeps peering manageable.

3. Two VPCs you must connect have overlapping CIDR blocks. What do you use, and why can't you peer them?

Answer: Use AWS PrivateLink with a Network Load Balancer (an interface endpoint), which works at the application level and tolerates overlapping ranges. Peering is impossible because it requires non-overlapping CIDR blocks.

4. Describe the two endpoints of a Site-to-Site VPN and how many tunnels it provides and why.

Answer: The on-premises side is the customer gateway; the AWS side is a virtual private gateway (or transit gateway). Each connection provides two IPsec tunnels terminating in separate Availability Zones, so traffic keeps flowing if one tunnel fails.

5. When can you accelerate a Site-to-Site VPN with Global Accelerator, and what does acceleration do?

Answer: Only when the VPN is attached to a transit gateway (not a virtual private gateway). Global Accelerator routes traffic from the on-premises network to the nearest AWS edge location, then over the AWS backbone, for better and more consistent response times.

6. Name the three Direct Connect virtual interfaces and what each accesses.

Answer: Public VIF → public AWS services such as Amazon S3; private VIF → your VPC via a virtual private gateway; transit VIF → your VPCs via a transit gateway through a Direct Connect gateway.

7. How do you make hybrid connectivity both highly available and highly resilient with Direct Connect?

Answer: High availability: pair a primary Direct Connect connection with a lower-cost backup VPN (AWS prefers DX by default). Resiliency: connect from multiple on-premises data centers through multiple Direct Connect locations, using dynamically routed active/active links with enough capacity to absorb a failover.

8. EC2 instances read sensitive data from Amazon S3 over its public endpoint, and security wants to remove internet exposure by the most efficient route. What is the solution, and why not a VPN or NAT gateway?

Answer: Use a gateway VPC endpoint for Amazon S3 — private access over the AWS network at no additional cost, with no bandwidth or packet-size limits. You cannot reach S3 over a VPN; a NAT gateway adds extra hops and isn't the most efficient route; an internet gateway still traverses the internet.