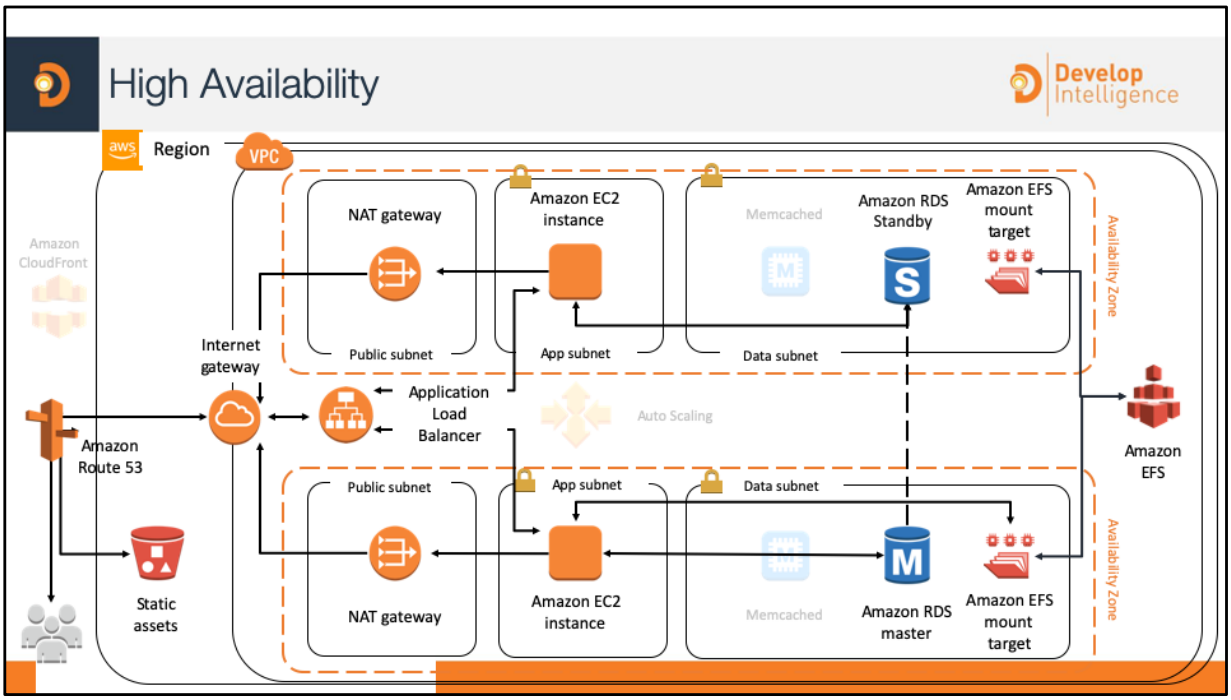
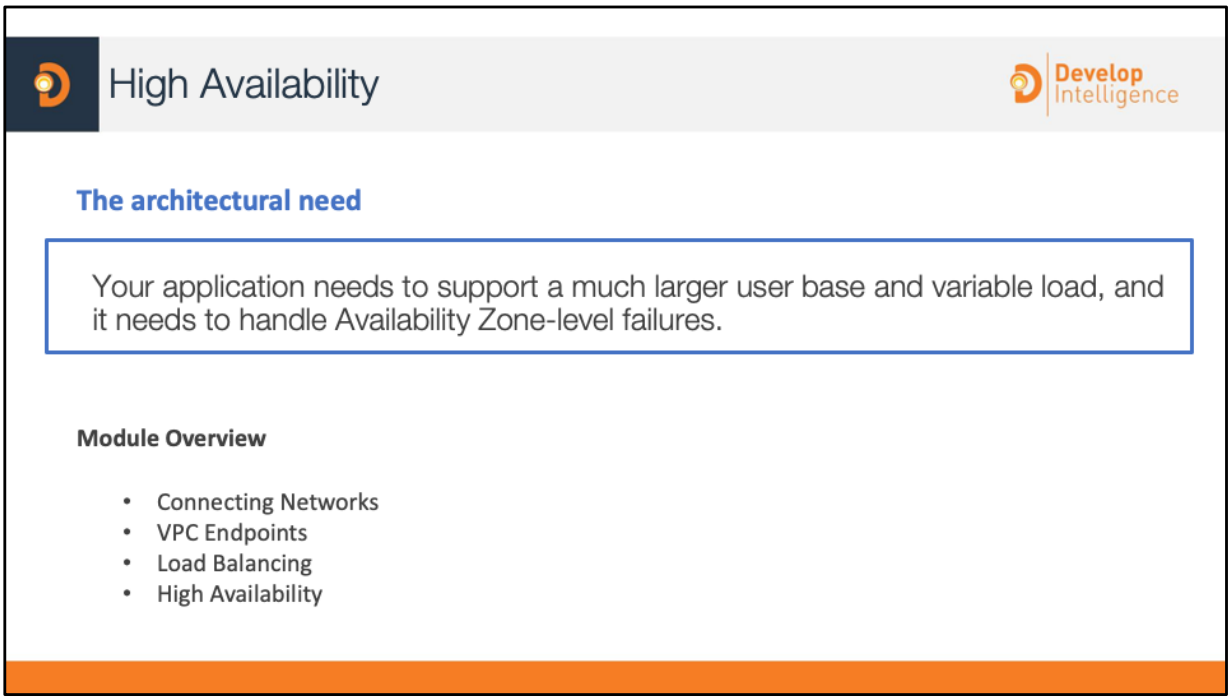
Networking In AWS Part 2

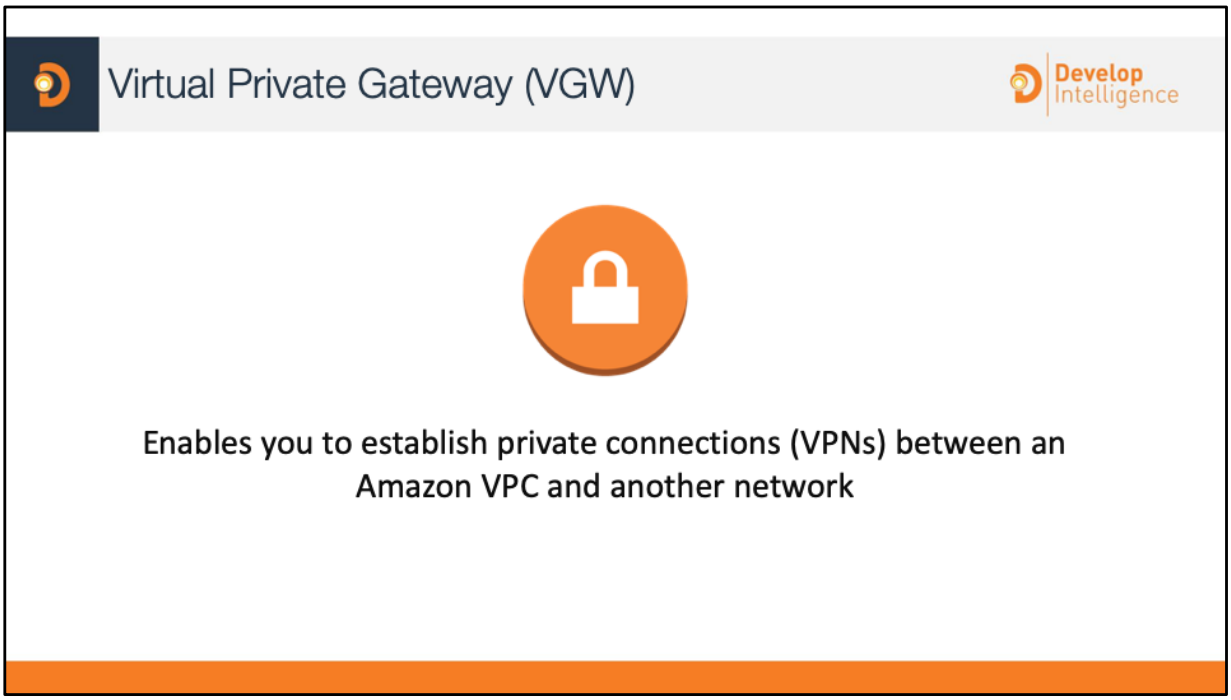
Develop Intelligence

- Module 6



By the end of class, you will be able to understand all of the components of this architectural diagram. You will also be able to construct your own architectural solutions that are just as large and robust.


High Availability
Develop Intelligence
The architectural need
Your application needs to support a much larger user base and variable load, and it needs to handle Availability Zone-level failures.
Module Overview
• Connecting Networks
• VPC Endpoints
• Load Balancing
• High Availability

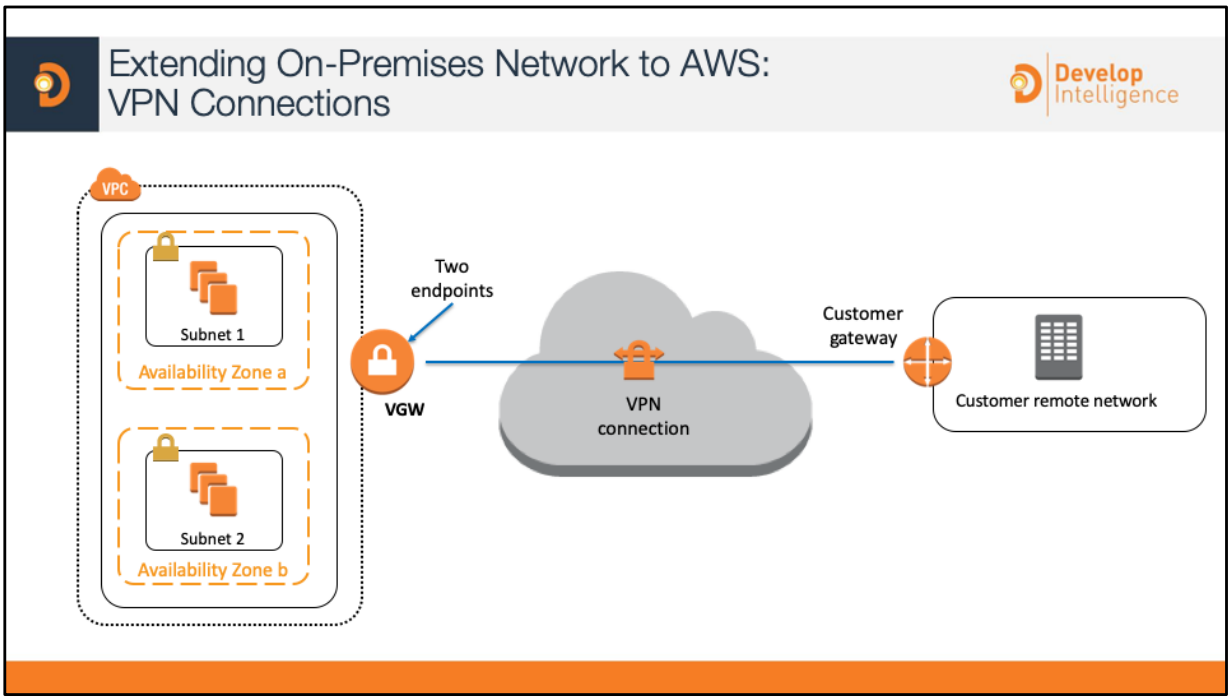


By default, instances that you launch into an Amazon VPC can't communicate with your own (remote) network. You can enable access to your remote network from your VPC by attaching a virtual private gateway (VGW) to the VPC, creating a custom route table, updating your security group rules, and creating an AWS-managed VPN connection.

Although the term *VPN connection* is a general term, in the Amazon VPC documentation, it refers to the connection between your VPC and your own network. AWS supports internet protocol security (IPsec) VPN connections.

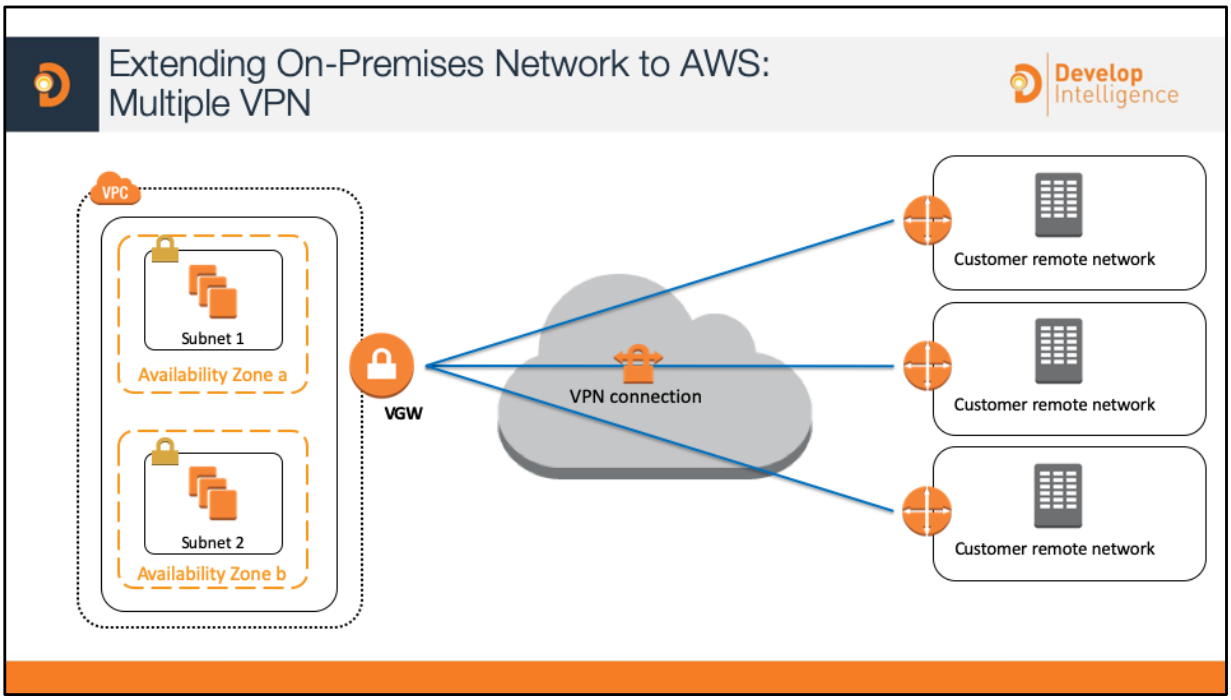
A VGW is the VPN concentrator on the Amazon side of the VPN connection. You create a VGW and attach it to the VPC from which you want to create the VPN connection.

When you create a VGW, you can specify the private Autonomous System Number (ASN) for the Amazon side of the gateway. If you don't specify an ASN, the VGW is created with the default ASN (64512). You cannot change the ASN after you've created the VGW.

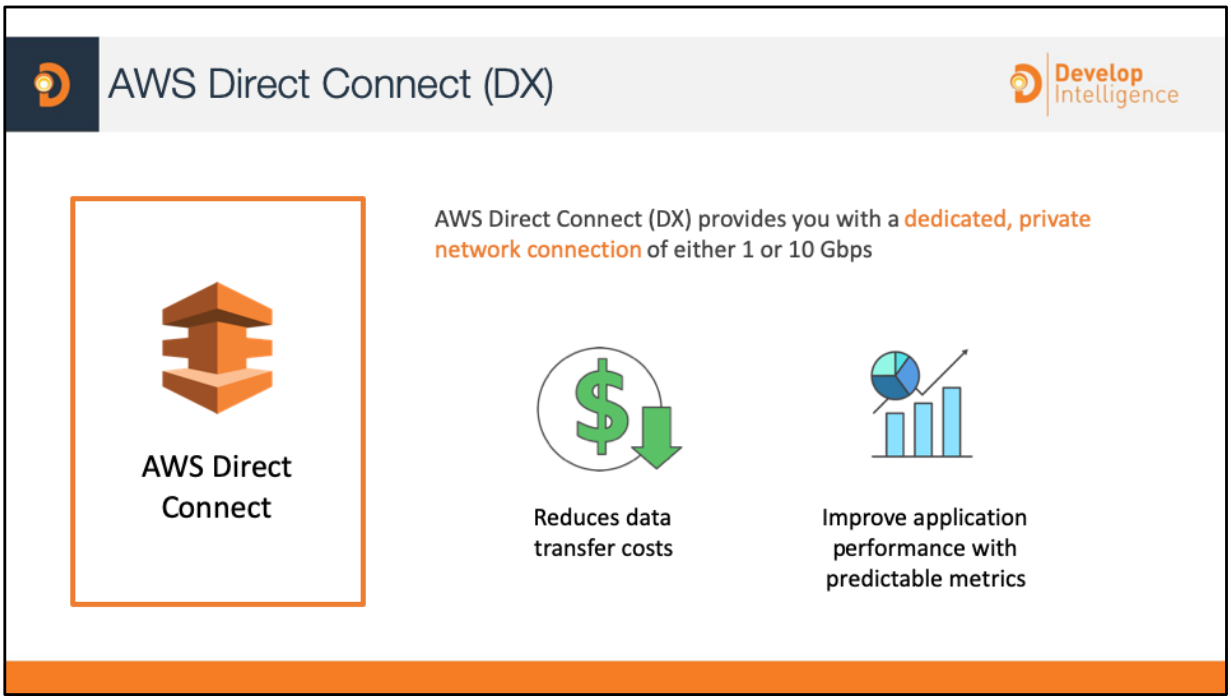


One solution is to use a VPN connection between your VPC's virtual gateway and your data center. With an AWS hardware VPN, you given two VPN endpoints to provide basic, automatic failover. For more information about creating an AWS hardware VPN, see http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_VPN.html.

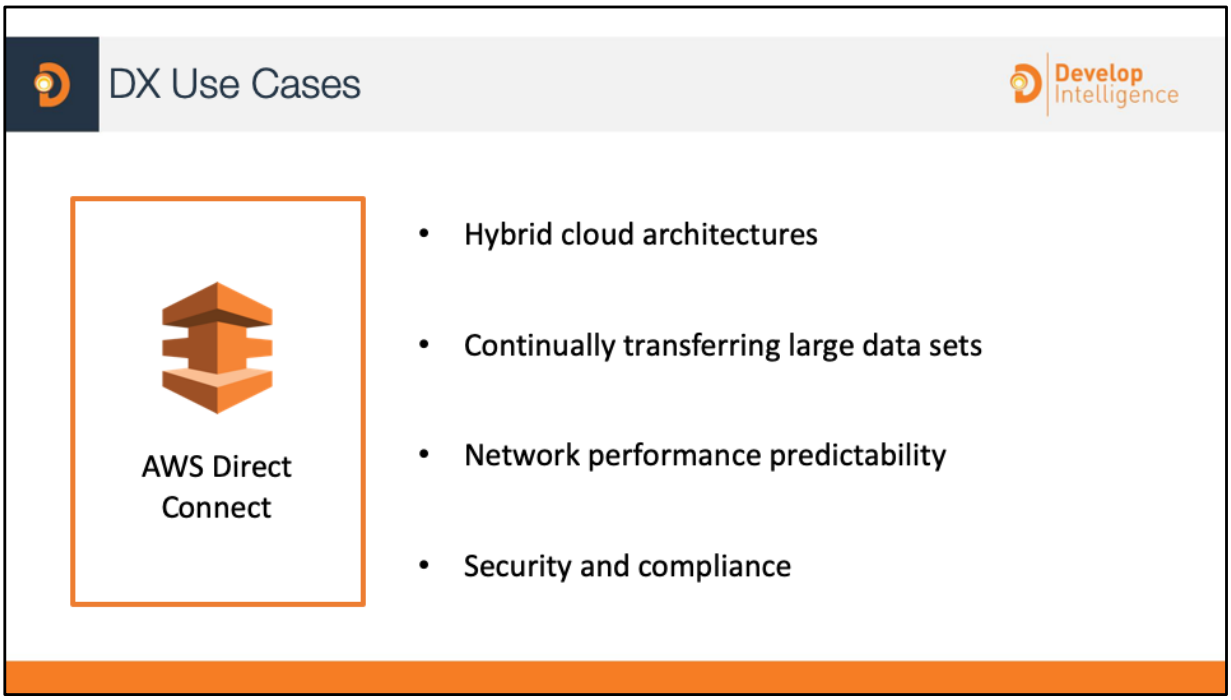
You can also create a VPN connection to your remote network by using an Amazon EC2 instance in your VPC that's running a software VPN appliance. AWS does not provide or maintain software VPN appliances; however, you can choose from a range of products provided by partners and open source communities on the AWS Marketplace.



In AWS, VGW also supports and encourages multiple customer gateway connections so that customers can implement redundancy and failover on their side of the VPN connection, as shown on this slide. Both dynamic and static routing options are provided, which gives customers flexibility in their routing configuration. Dynamic routing uses BGP peering to exchange routing information between AWS and these remote endpoints. Dynamic routing also allows customers to specify routing priorities, policies, and weights (metrics) in their BGP advertisements and to influence the network path between their networks and AWS.



AWS Direct Connect (DX) is a unique solution to go beyond simple connectivity over the internet and, instead, get access with scale, speed, and consistency to the AWS network for these important applications. DX does not involve the internet; instead, it uses dedicated, private network connections between your on-premises solutions and AWS.



Service Benefits
DX is useful for several scenarios, some of which are described below.

Transferring Large Data Sets
Consider an HPC application that operates on large data sets that must be transferred between your data center and the AWS Cloud. For such applications, connecting to the AWS Cloud using DX is a good solution:
network transfers will not compete for internet bandwidth at your data center or office location.
The high bandwidth link reduces the potential for network congestion and degraded application performance.

Reduced Network Transfer Costs
By using DX to transfer large data sets, you can limit the internet bandwidth used by your application. By doing so, you can reduce network fees that you pay to your internet service provider (ISP) and avoid having to pay for increased internet bandwidth commitments or new contracts.
In addition, all data transferred over DX is charged at the reduced DX data transfer rate rather than internet data transfer rates, which can greatly reduce your network costs.

Improved Application Performance

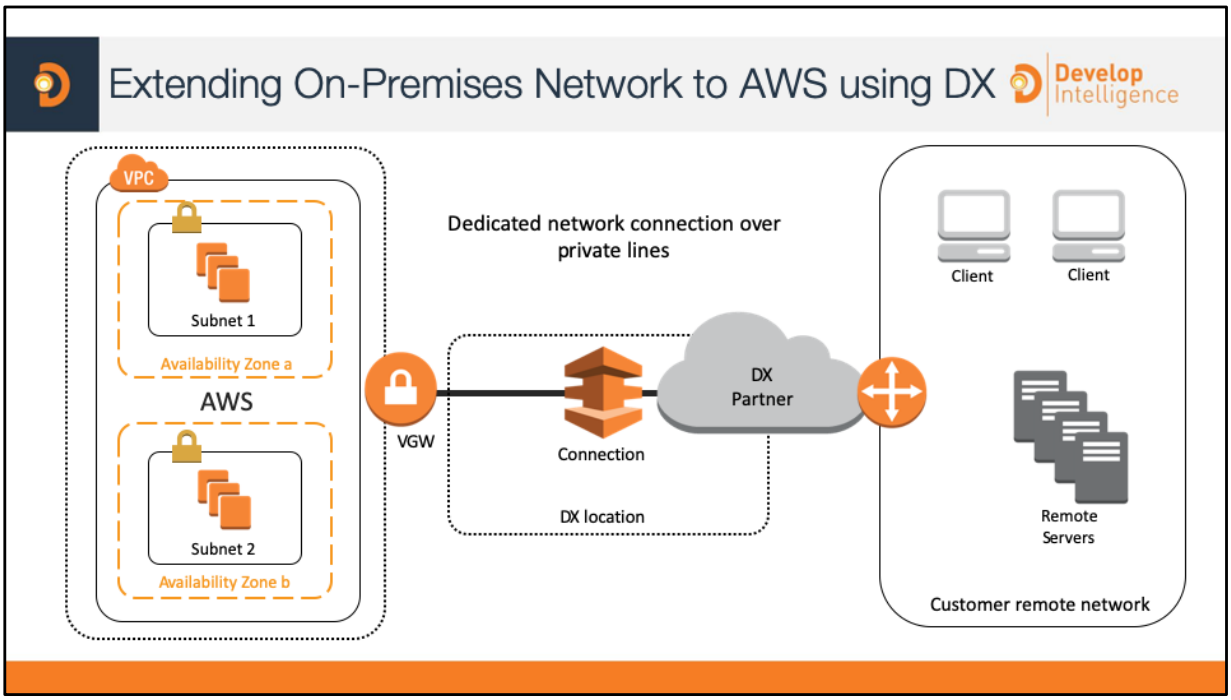
Applications that require predictable network performance can also benefit from DX. Examples include applications that operate on real-time data feeds, such as audio or video streams. In such cases, a dedicated network connection can provide more consistent network performance than standard internet connectivity.

Security and Compliance

Enterprise security or regulatory policies sometimes require applications hosted on the AWS Cloud to be accessed through private network circuits only. DX is a natural solution to this requirement because traffic between your data center and your application flows through the dedicated private network connection.

Hybrid Cloud Architectures

Applications that require access to existing data center equipment that you own can also benefit from DX. The next section discusses this use case and illustrates different scenarios that can be supported by DX.



Benefits:

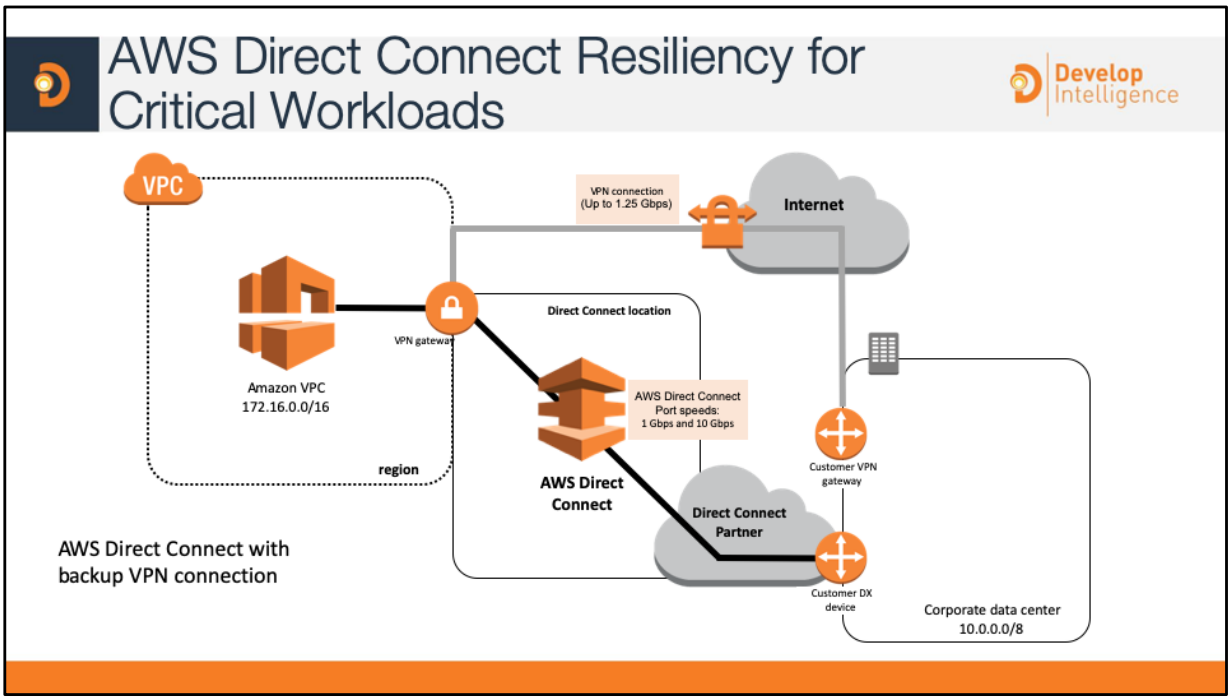
- More predictable network performance
- Reduced bandwidth costs
- 1- or 10-Gbps provisioned connections
- Supports BGP peering and routing policies

We have partnered closely with Equinix, Coresite, Eircom, TelecityGroup, and Terramark to create global DX access to every AWS Region in the world. In a few of our locations, particularly in LA, NYC, and London, we have extended the capability of the service to provide access to additional IT hot spots.

Limitation:
May require additional telecom and hosting provider relationships or new network circuits to be provisioned.
DX makes it easy to establish a dedicated network connection from on-premises to Amazon VPC. Using DX, customers can establish private connectivity between AWS and their data center, office, or colocation environment. This private connection can reduce network costs, increase bandwidth throughput, and provide a more consistent network experience than internet-based connections can.

DX lets customers establish 1-Gbps or 10-Gbps dedicated network connections (or multiple connections) between AWS networks and one of the DX locations, and uses industry standard VLANs to access Amazon EC2 instances running within a VPC using private IP addresses. Customers may choose from an ecosystem of WAN service providers for integrating their DX endpoint in an DX location with their remote networks.
For a list of current AWS DX locations, see
http://aws.amazon.com/directconnect/details/

For information on using AWS Direct Connect for high resiliency for critical workloads, see https://aws.amazon.com/directconnect/resiliency-recommendation/



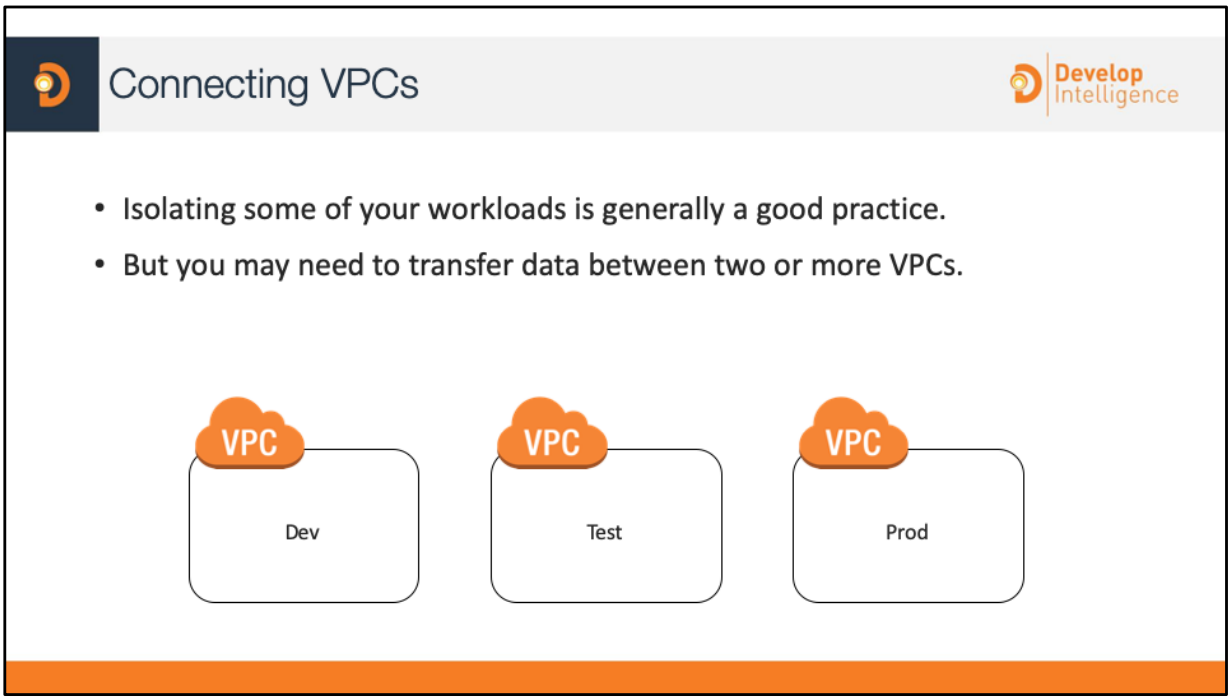
AWS customers can benefit of one or more AWS Direct Connect connections for their primary connectivity to AWS, coupled with a lower-cost backup connection. To achieve this objective, you can establish AWS Direct Connect connections with a VPN backup, as depicted in the diagram above.

The configuration in this example consists of two dynamically routed connections, one using AWS Direct Connect and the other using a VPN connection from two different customer devices. AWS provides example router configurations to assist in establishing both AWS Direct Connect and dynamically routed VPN connections. By default, AWS will always prefer to send traffic over your AWS Direct Connect connection, so no additional AWS-specific configuration is required to define primary and backup connections. However, customers should configure AWS Direct Connect and VPN-specific internal-route propagation to ensure internal systems select the appropriate paths. The Multiple Data Center HA Network Connectivity Solution Brief has more details on route manipulation options for this scenario; however, the default configuration is typically sufficient for most customers connecting to AWS from a single data center.

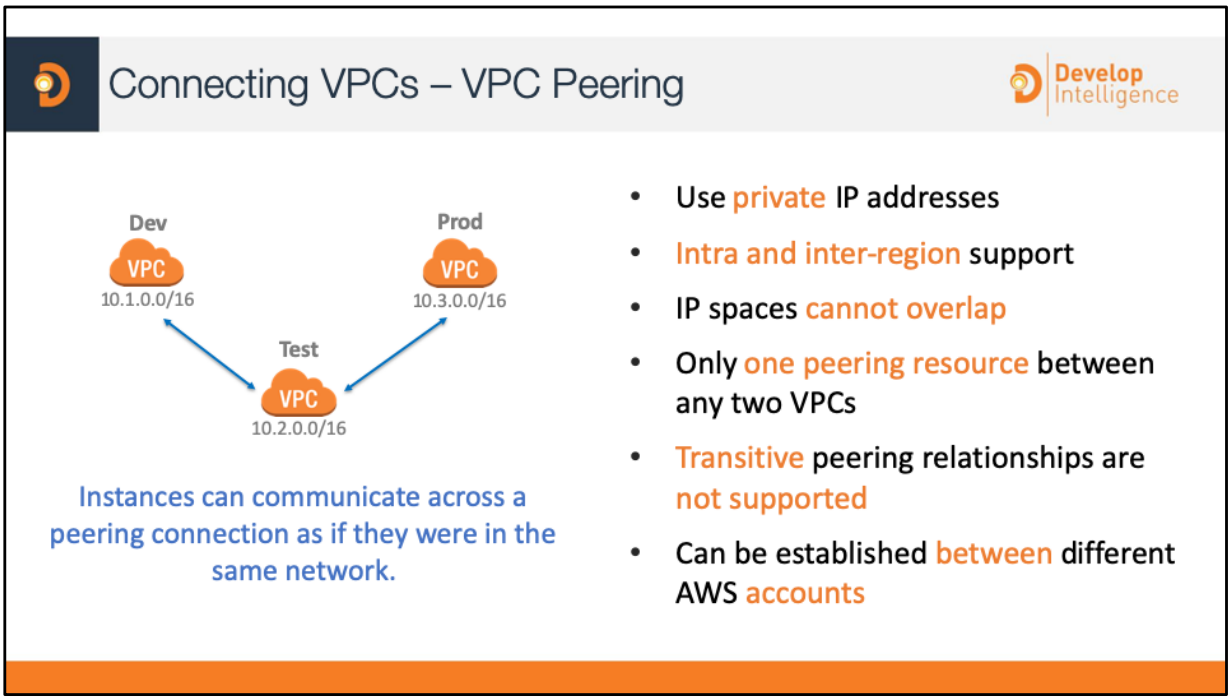
AWS Direct Connect supports these port speeds over single-mode fiber: 1 Gbps: 1000BASE-LX (1310nm) and 10 Gbps: 10GBASE-LR (1310nm).

It is important to understand that AWS Managed VPN supports up to 1.25 Gbps throughput per VPN tunnel and does not support Equal Cost Multi Path (ECMP) for egress data path in the case of multiple AWS Managed VPN tunnels terminating on the same VGW.

This approach allows you to choose the primary network path and network provider for your AWS traffic, with the option of using a different provider for a backup VPN connection. Choose network providers and AWS Direct Connect locations that align with your organization's risk tolerance, financial expectations, and data-center connectivity policies. For example, if you are concerned about the risk associated with an individual network-provider outage, consider different network providers for AWS Direct Connect and Internet connectivity. However, because this design relies on a single customer location to provide connectivity to AWS, any location-specific disruption (e.g., loss of power or cable cut outside the facility) can still affect network connectivity to AWS, despite leveraging redundant network providers. Additionally, make sure to monitor AWS Direct Connect utilization to ensure that a VPN connection will be a sufficient backup to support your application's latency and bandwidth requirements.



When your business or architecture becomes large enough, you will find the need to separate logical elements either for security or architectural needs, or just for simplicity's sake.

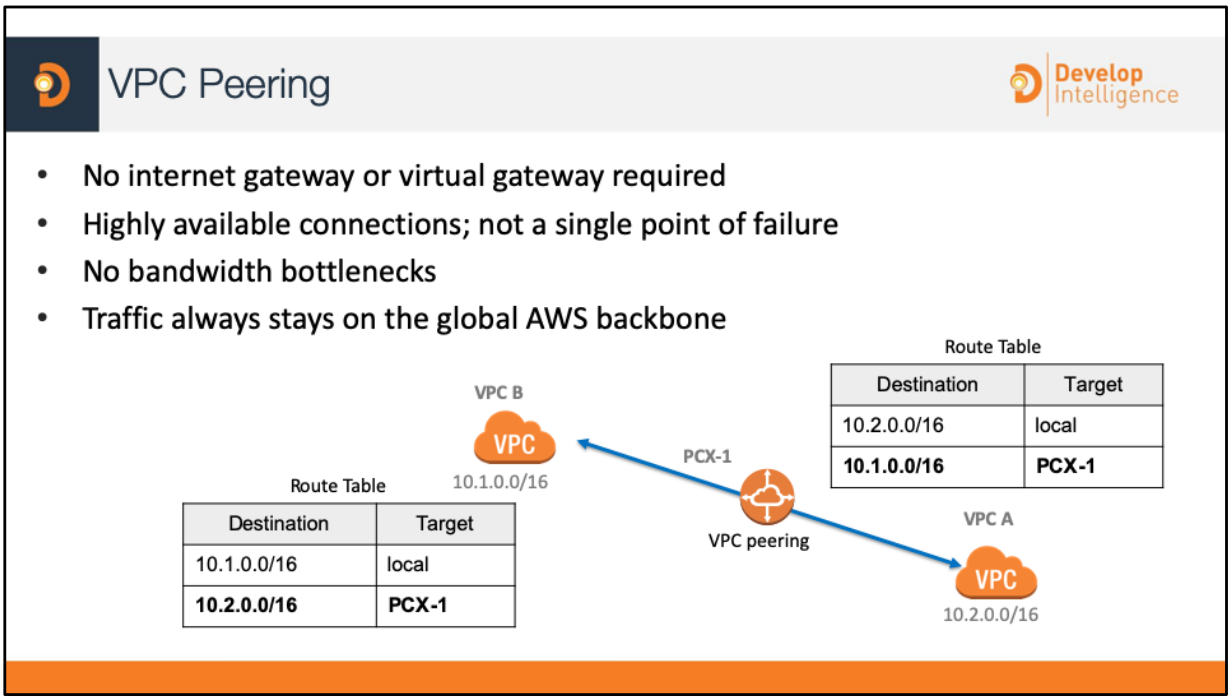


In the diagram, VPCs Dev and Test are peered, which does not mean that *Prod* can talk to Dev. By default, VPC peering does not allow Prod to connect to Dev unless they are *explicitly established as peers.* Therefore, you control which VPCs can talk to each other.

To establish a VPC peering connection, the owner of the requester VPC (or local VPC) sends a request to the owner of the peer VPC to create the VPC peering connection. The peer VPC can be owned by you or another AWS account, and cannot have a CIDR block that overlaps with the requester VPC's CIDR block. The owner of the peer VPC has to accept the VPC peering connection request to activate the VPC peering connection. To enable the flow of the traffic between the peer VPCs using private IP addresses, add a route to one or more of your VPC's route tables that points to the IP address range of the peer VPC. The owner of the peer VPC adds a route to one of their VPC's route tables that points to the IP address range of your VPC. You may also need to update the security group rules that are associated with your instance to ensure that traffic to and from the peer VPC is not restricted.

A VPC peering connection is a one-to-one relationship between two VPCs. You can create multiple VPC peering connections for each VPC that you own, but transitive peering relationships are not supported: you will not have any peering relationship with VPCs that your VPC is not directly peered with. You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account within a single region.

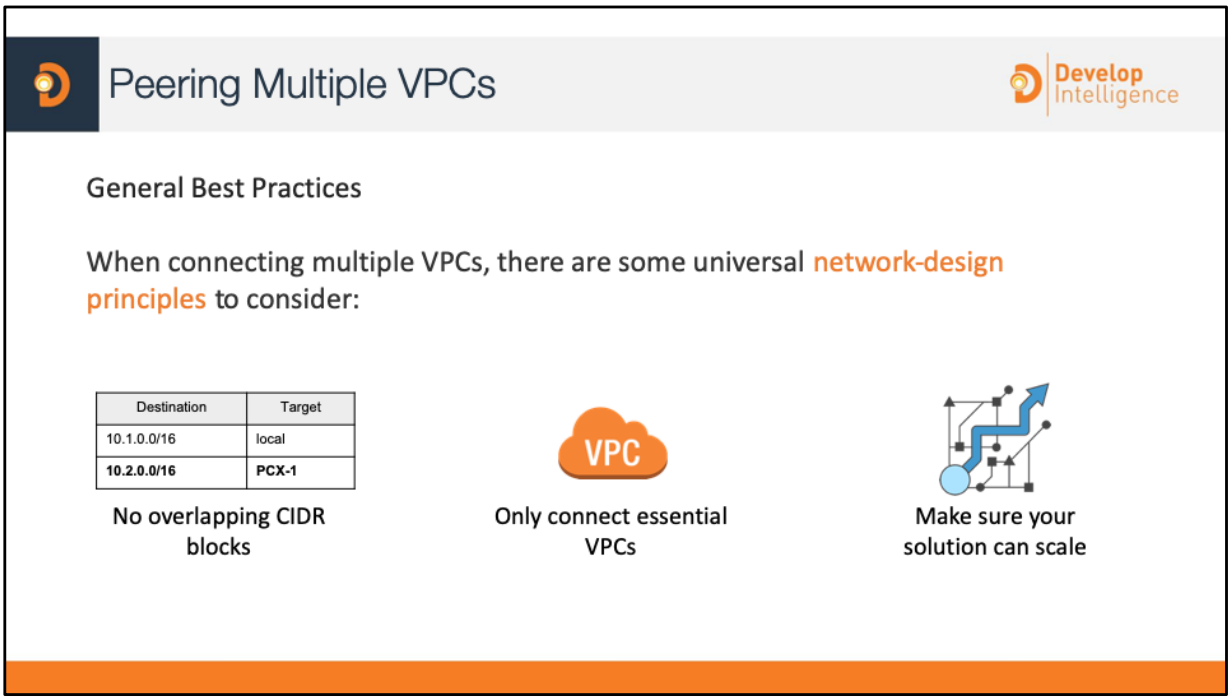
Amazon EC2 now allows peering relationships to be established between VPCs across different regions. Inter-region VPC peering allows VPC resources like Amazon EC2 instances, Amazon RDS databases, and Lambda functions running in different regions to communicate with each other using private IP addresses, without requiring gateways, VPN connections, or separate network appliances. Data transferred across inter-region VPC peering connections is charged at the standard inter-region data transfer rates.



To create a VPC peering connection with another VPC, you need to be aware of the following limitations and rules:

- There is a limit on the number of active and pending VPC peering connections that you can have per VPC.
- VPC peering does not support transitive peering relationships; in a VPC peering connection, your VPC will not have access to any other VPCs that the peer VPC may be peered with. This includes VPC peering connections that are established entirely within your own AWS account.
- You cannot have more than one VPC peering connection between the same two VPCs at the same time.
- The maximum transmission unit (MTU) across a VPC peering connection is 1500 bytes.
- A placement group can span peered VPCs; however, you will not get full-bisection bandwidth between instances in peered VPCs.
- Unicast reverse path forwarding in VPC peering connections is not supported.
- Private DNS values cannot be resolved between instances in peered VPCs.
- Traffic using inter-region VPC peering always stays on the global AWS backbone and never traverses the public internet, thereby reducing threat vectors, such as common exploits and DDoS attacks.

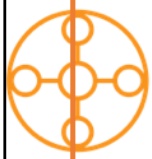
You can now reference security groups in a peered VPC in both inbound and outbound rules. This functionality is supported cross-account, so the two VPCs can be in different accounts. Support for security group references in a peered VPC simplifies configuration by controlling peering traffic via security group membership instead of CIDR ranges. You can reference security group from a peered VPC using the console, AWS CLI, and SDKs.



When connecting multiple VPCs in a single AWS Region, there are some universal network-design principles to consider:

- 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, as this will affect the solution you choose. Some solutions may prove to be more expensive than others based on the amount of data transferred.
- Connect only those VPCs that really need to communicate with each other.

Connecting VPCs - Transit Gateway

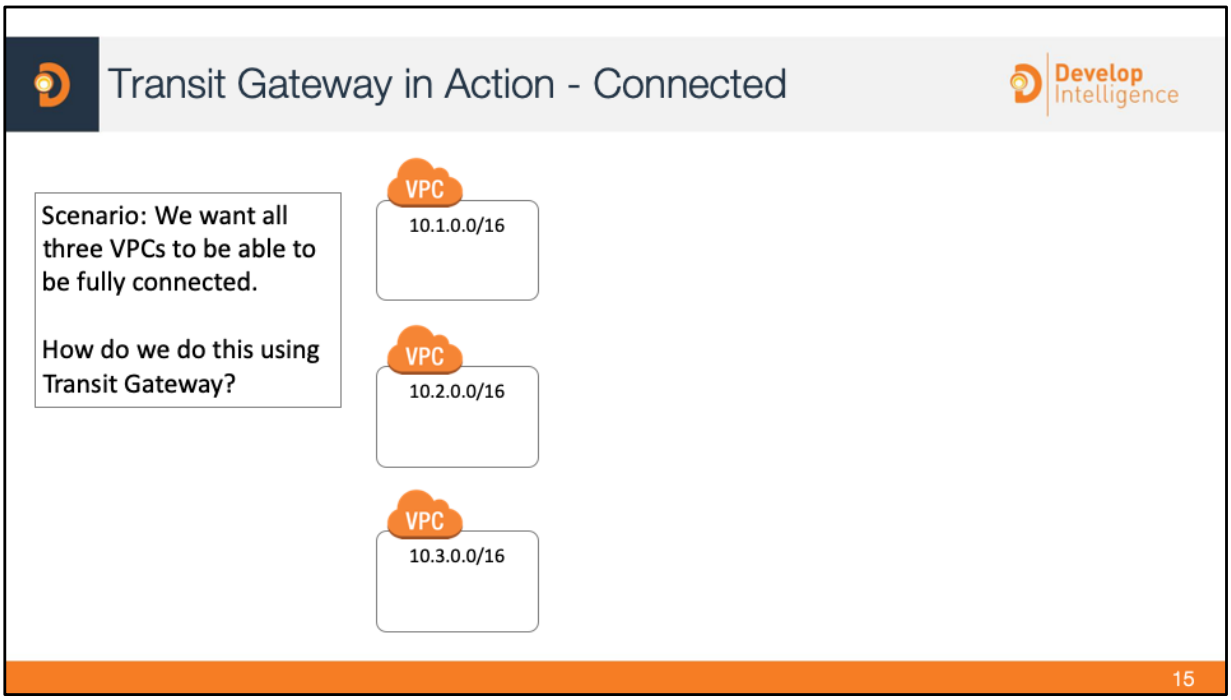


AWS Transit Gateway

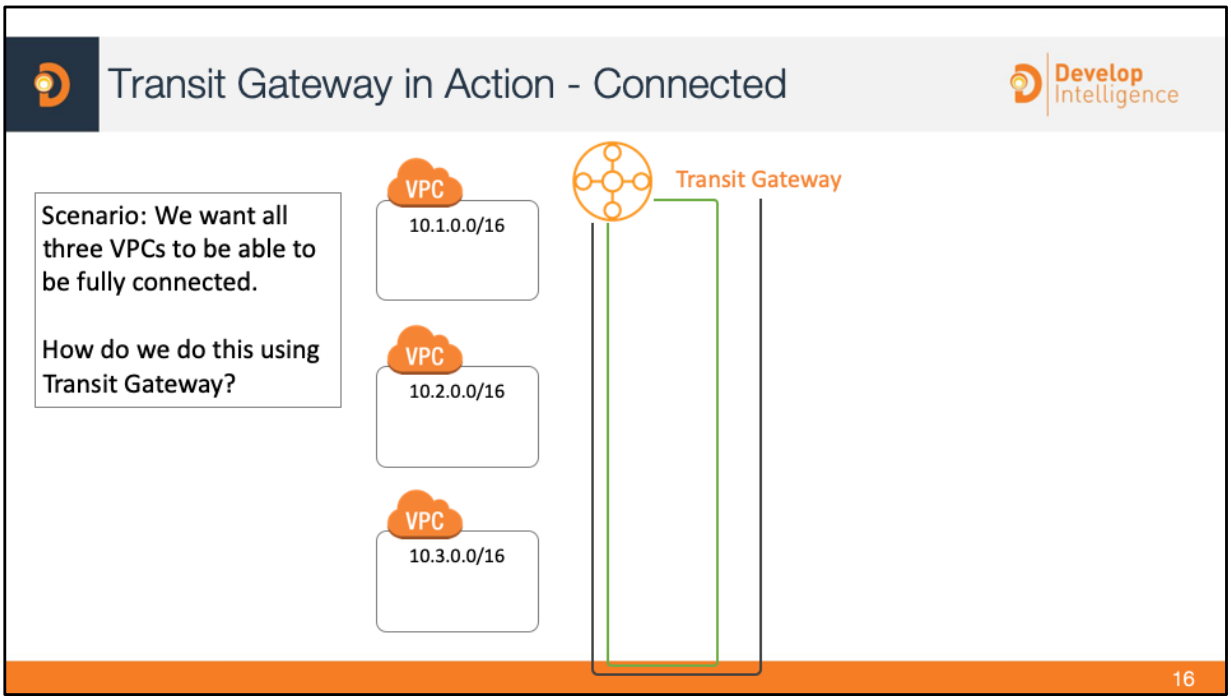
Connects up to **5,000 VPCs** and **on-premises environments** with a single gateway

Acts as a hub for all traffic to flow through between your networks

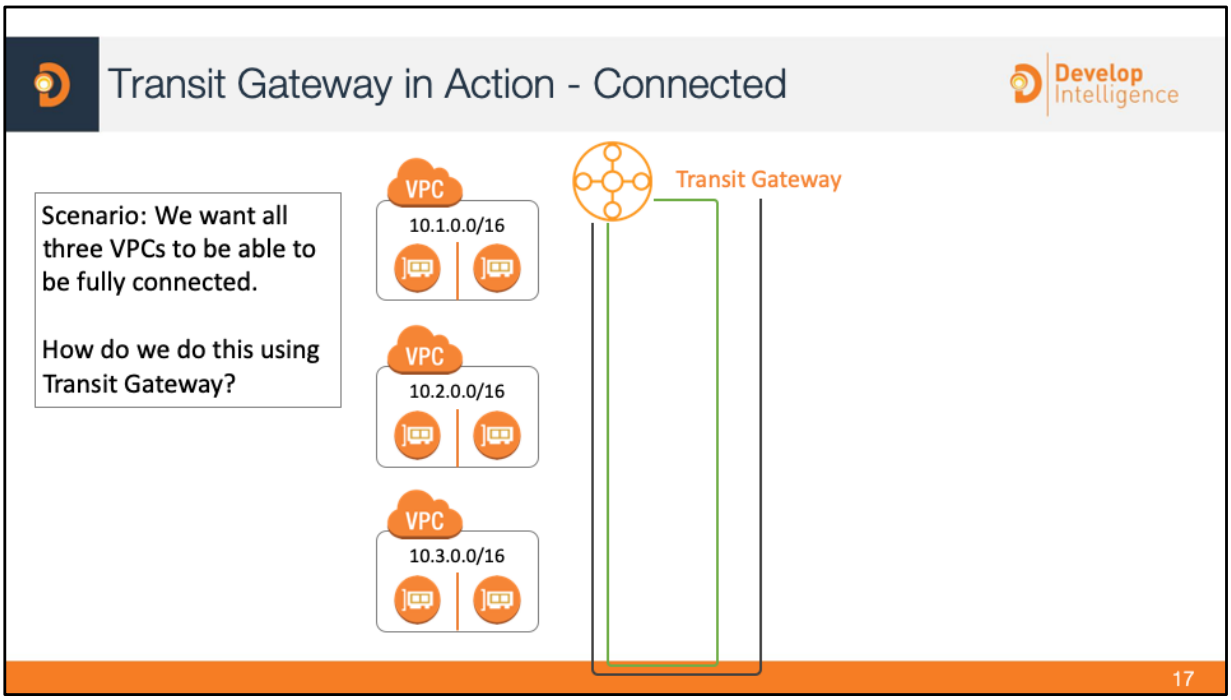
Fully managed, highly available, flexible routing service



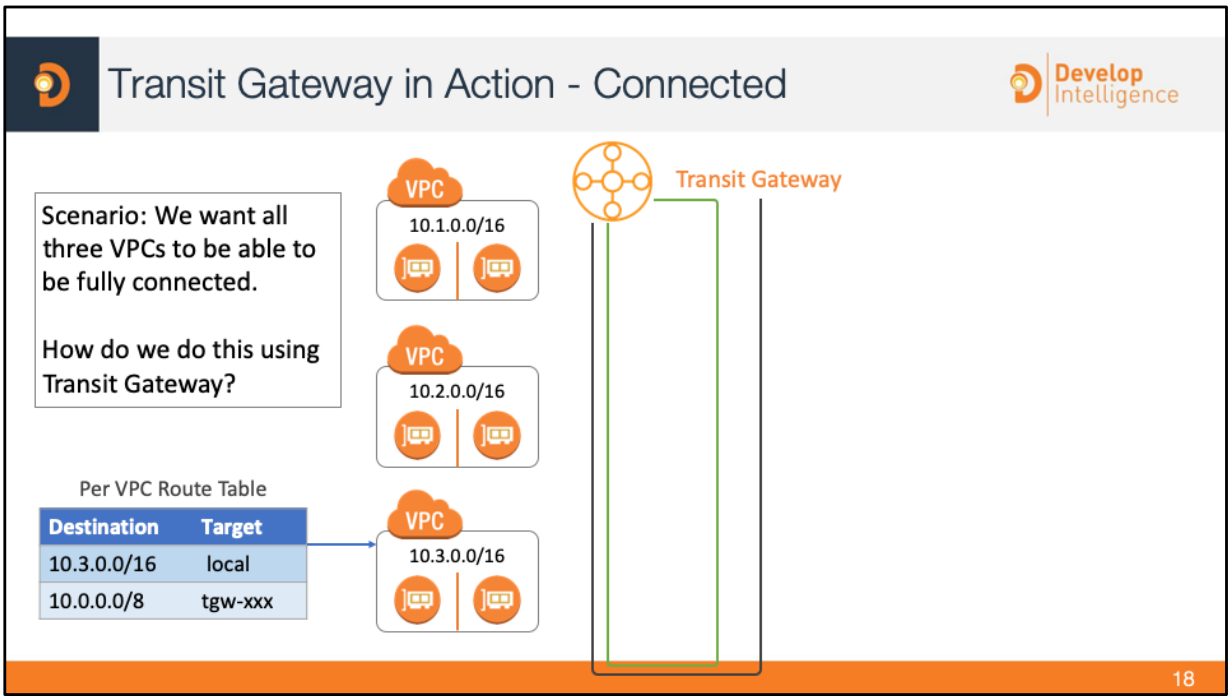
There are many situations where having connectivity between multiple VPCs is highly desirable. Having to manage VPC peering connections over a large group can be annoying and difficult. It's vital to keep in mind how large your environment might become over time, how well it will scale, and how you will organize these VPCs.



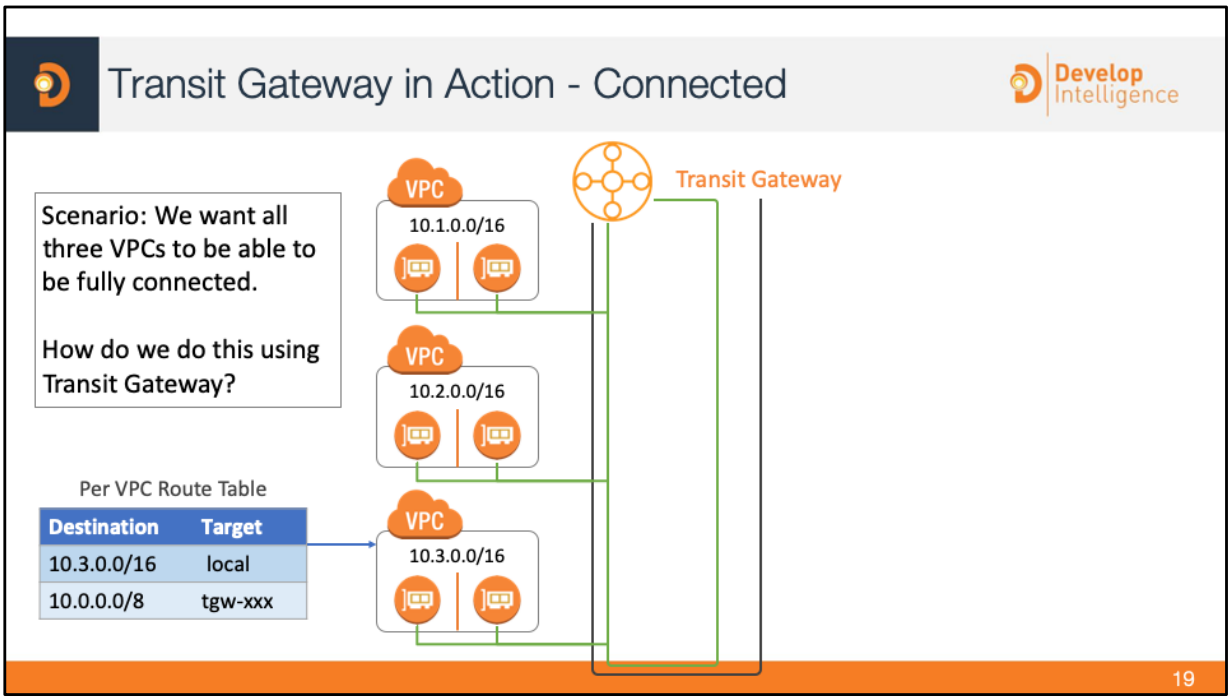
The first step to creating this connectivity is to set up a transit gateway. This can be done through the Amazon EC2 dashboard. Various charges apply for using transit gateway – make sure your architecture and budget can support this.



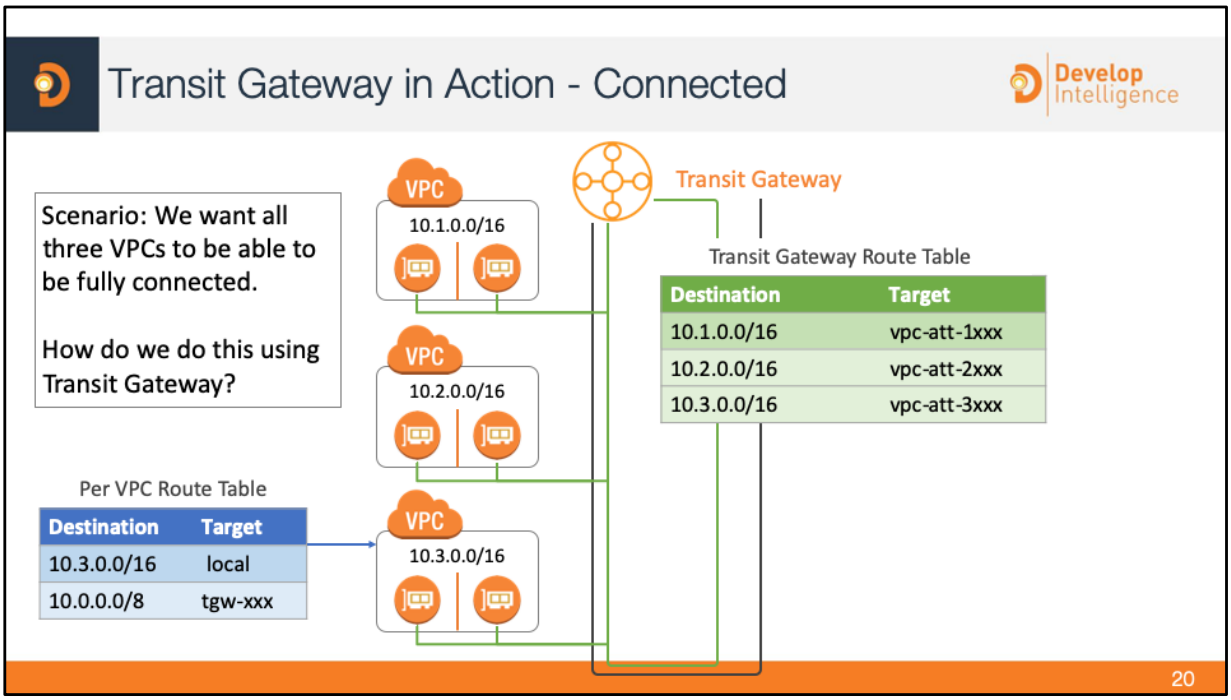
Transit Gateway operates though network interfaces which are deployed into subnets. To effectively use Transit Gateway, you need to deploy one attachment into each Availability Zone your target VPC occupies.



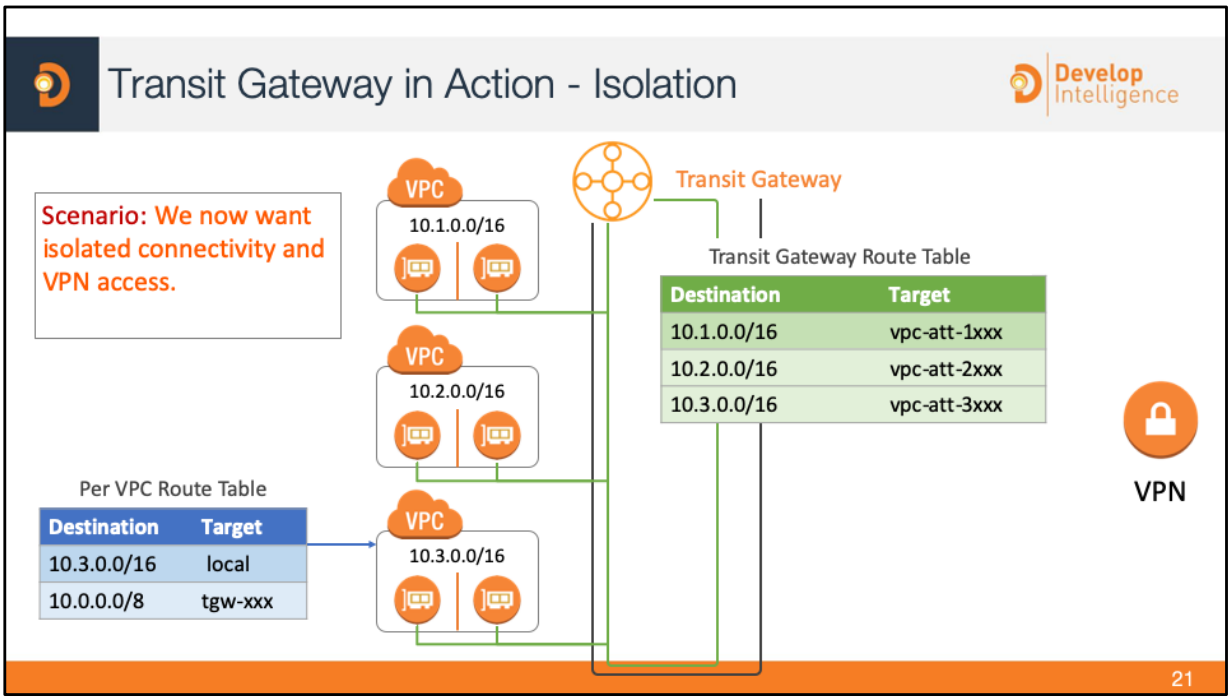
In each route table of your VPC, make sure traffic is routing outwards towards the Transit Gateway attachment.



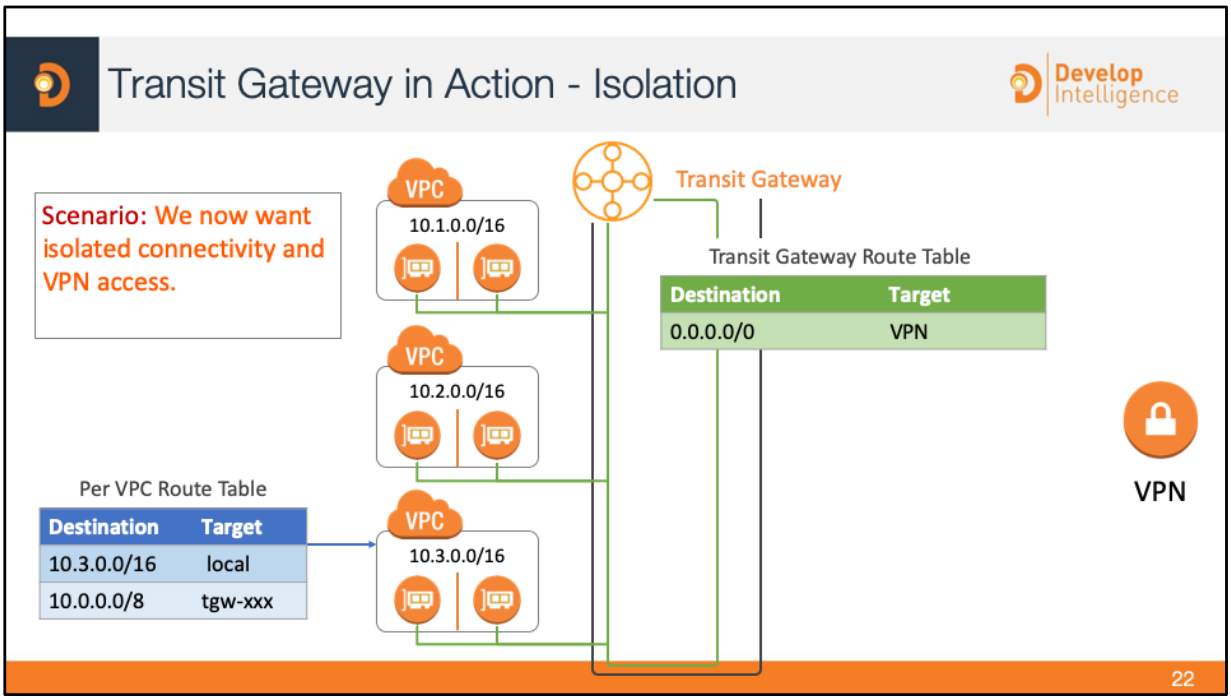
These attachments are connected to the Transit Gateway



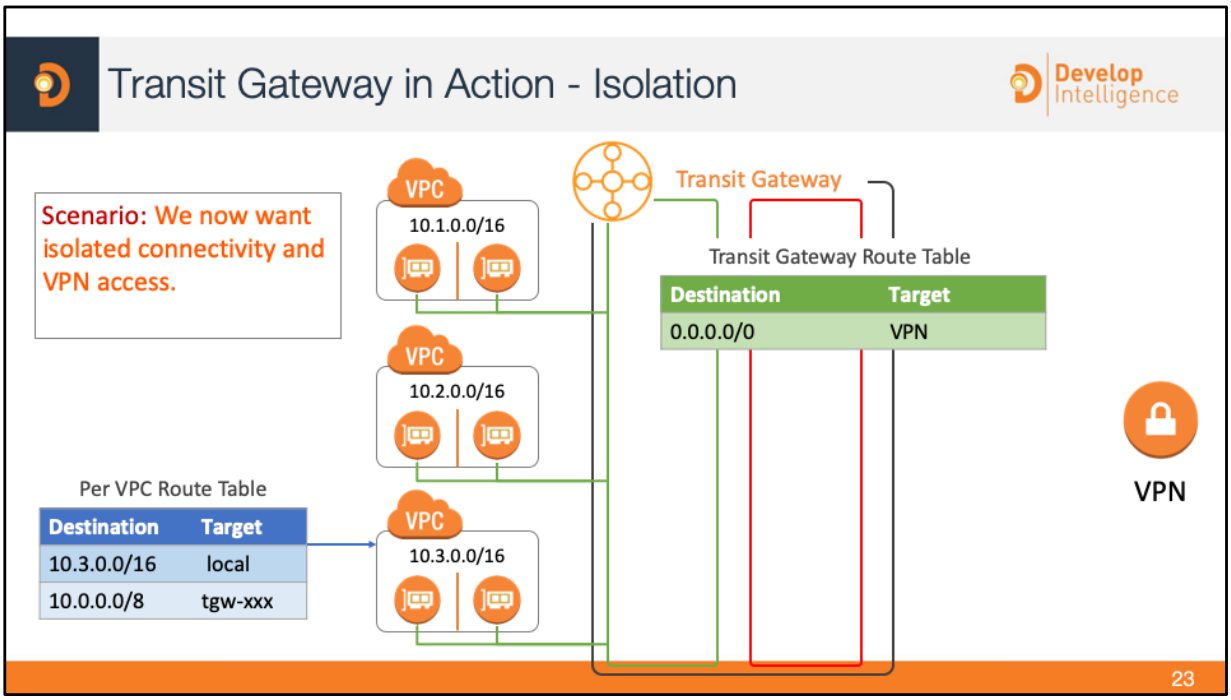
Inside the Transit Gateway, you can create route tables to direct traffic how you see fit. You can have multiple route tables for very specific interactions. In our case, we just have the one to allow full connectivity.



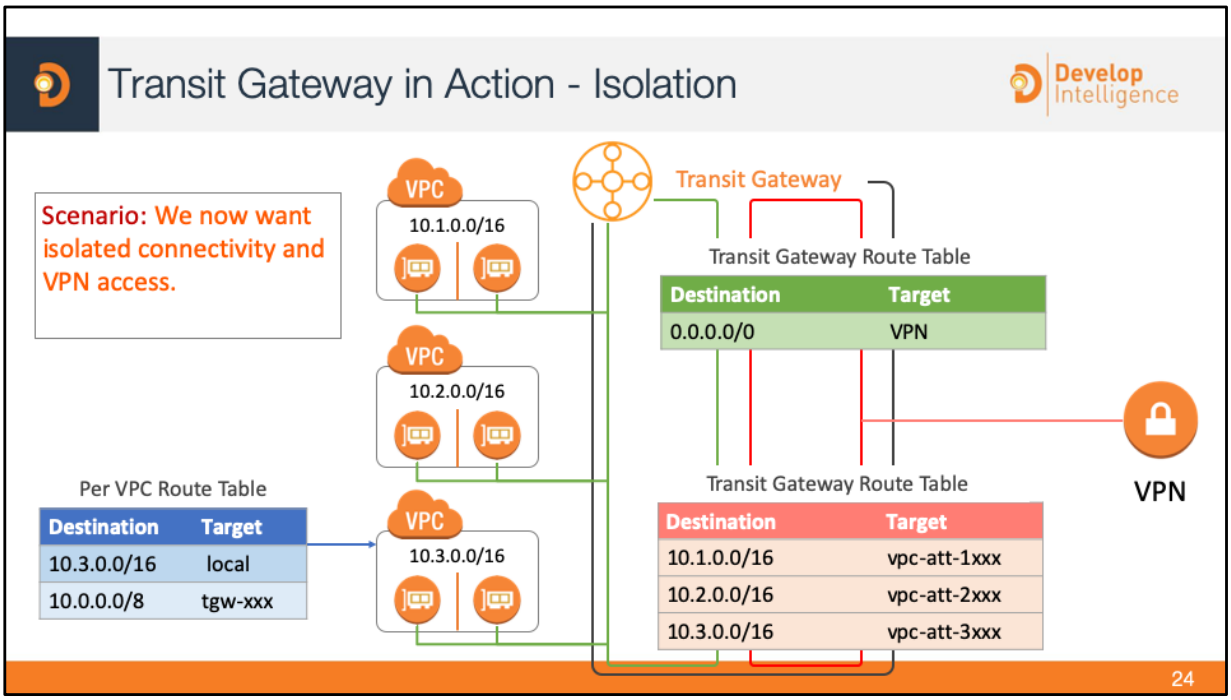
A common architecture is having full access to your environment from a VPN source. In this scenario, we also do not want our VPCs to be able to talk with each other.



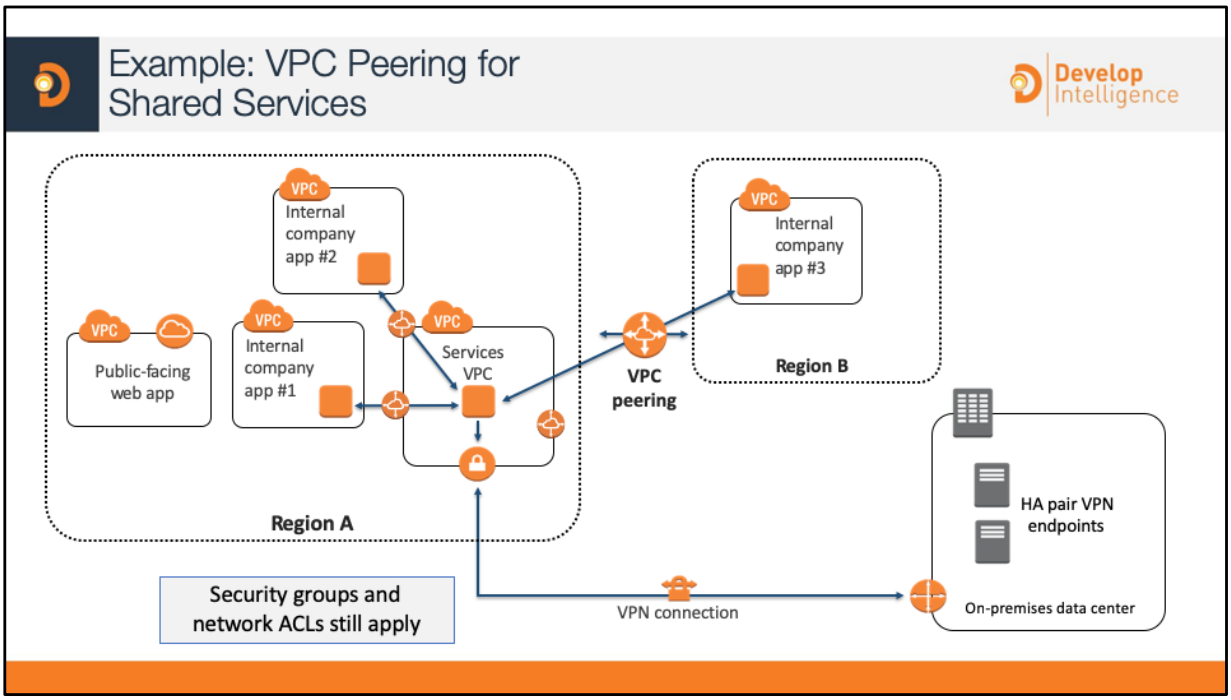
First, we change the routes for the initial table to point towards the VPN connection. This will stop the VPCs from communicating with each other and provide outbound access.



Now that we want to create an isolated environment, that is connected to VPN only. Add another Route table inside the Transit Gateway.



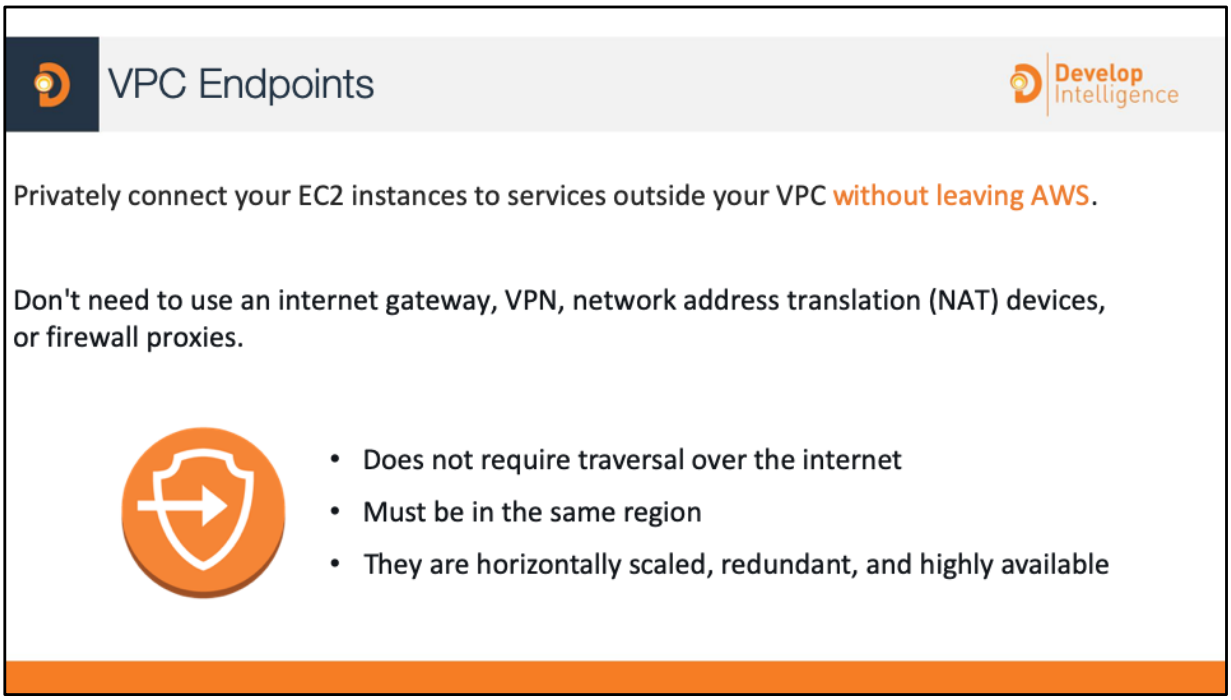
Set these destinations to point from the VPN to the target VPCs. There you have it: isolated and secure VPN access with no cross communication.



In this example, in order to deliver its responsibilities, the Corporate IT and Corporate Information Security groups provide a “Services VPC” that each department may peer with. This VPC contains connections to Active Directory, security scanning tools, monitoring/logging tools, and a variety of other capabilities. It also provides a proxy through which the department VPCs can access some on-premises resources.
VPC Peering:

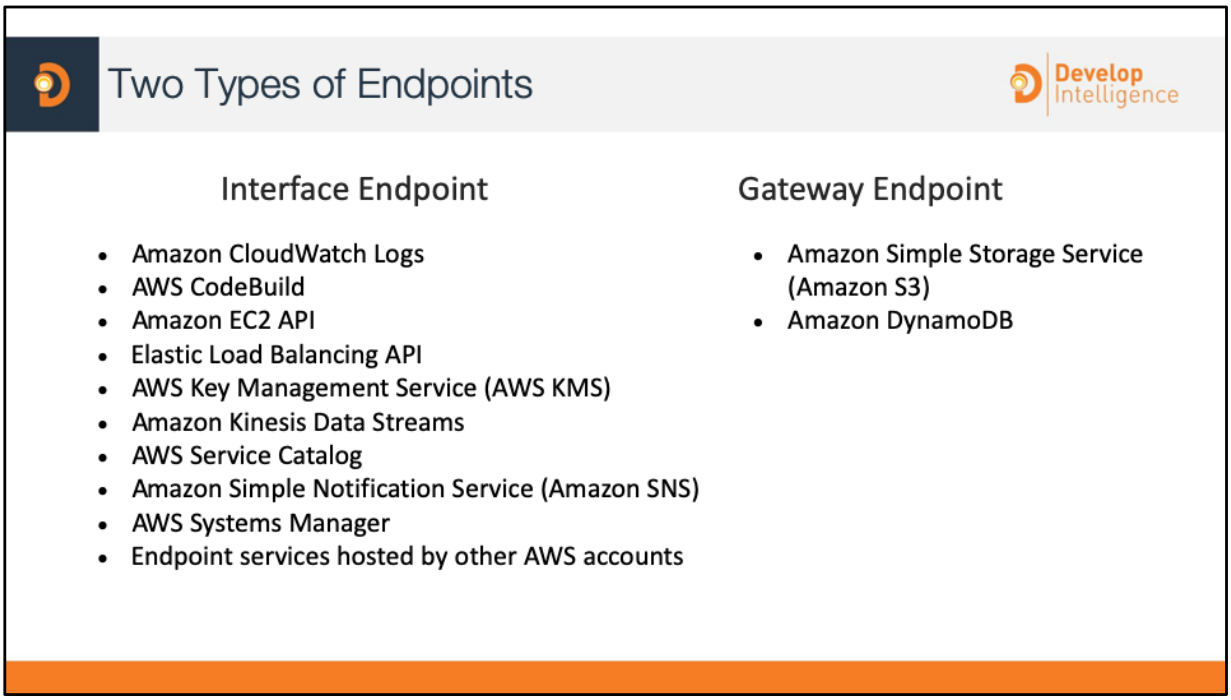
- 1 to 1 Peer = company app isolation from other apps in other VPCs, but this could always facilitate TEMP connect b/t dev/qa and prod, TRANSFER data, tear down.
- Security groups and network ACLs still apply.

Note that a VPC peering connection with a VPC in a different region is present. Amazon EC2 now allows peering relationships to be established between VPCs across different AWS Regions. Inter-region VPC peering allows VPC resources like EC2 instances, Amazon RDS, and Lambda functions that run in different AWS Regions to communicate with each other using private IP addresses, without requiring gateways, VPN connections or separate physical hardware.



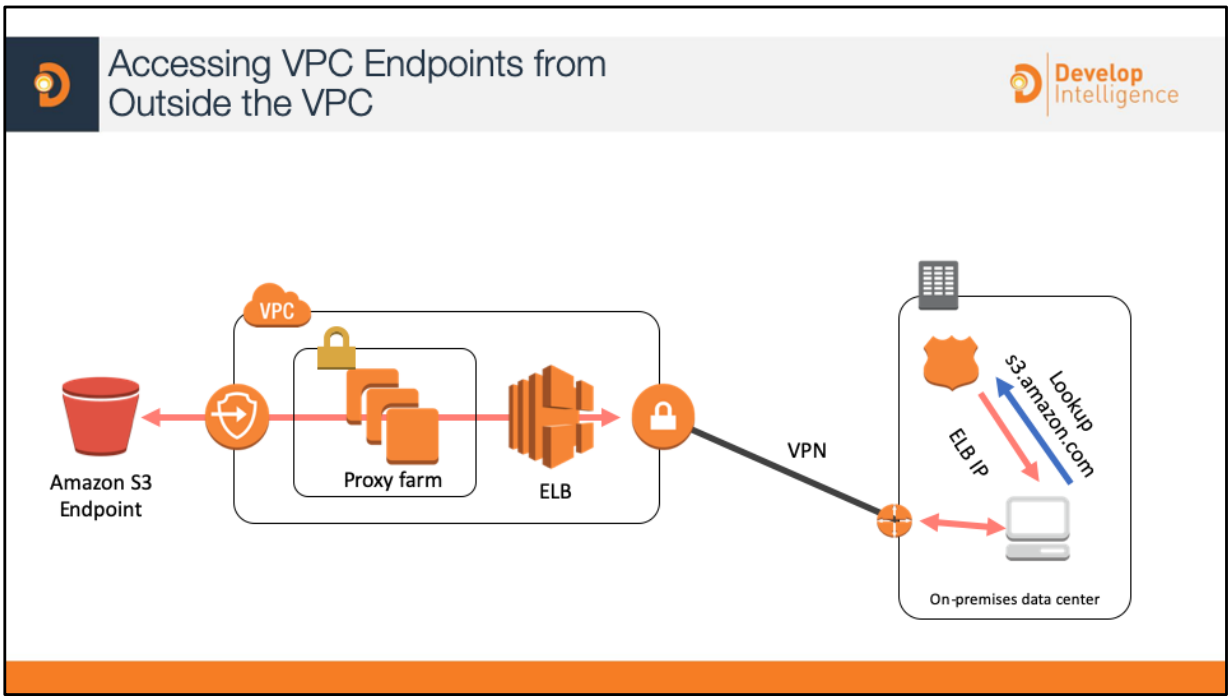
An Amazon VPC *endpoint* enables a private connection between a VPC and another AWS service without leaving the AWS network. An endpoint enables Amazon EC2 instances to communicate with an AWS service in the same region from their private IP addresses. It does not require traversal over the internet or through a NAT instance, a VPN connection, or DX. VPC endpoints also provide additional security features such as the ability to add policies to control which Amazon S3 buckets in a VPC can access or to lock down S3 buckets to specific VPCs. Currently, AWS supports VPC endpoints for connections with Amazon S3 and Amazon DynamoDB only.

Endpoints are virtual devices. They are horizontally scaled, redundant, and highly available VPC components that allow communication between instances in your VPC and services without imposing availability risks or bandwidth constraints on your network traffic.



An *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.

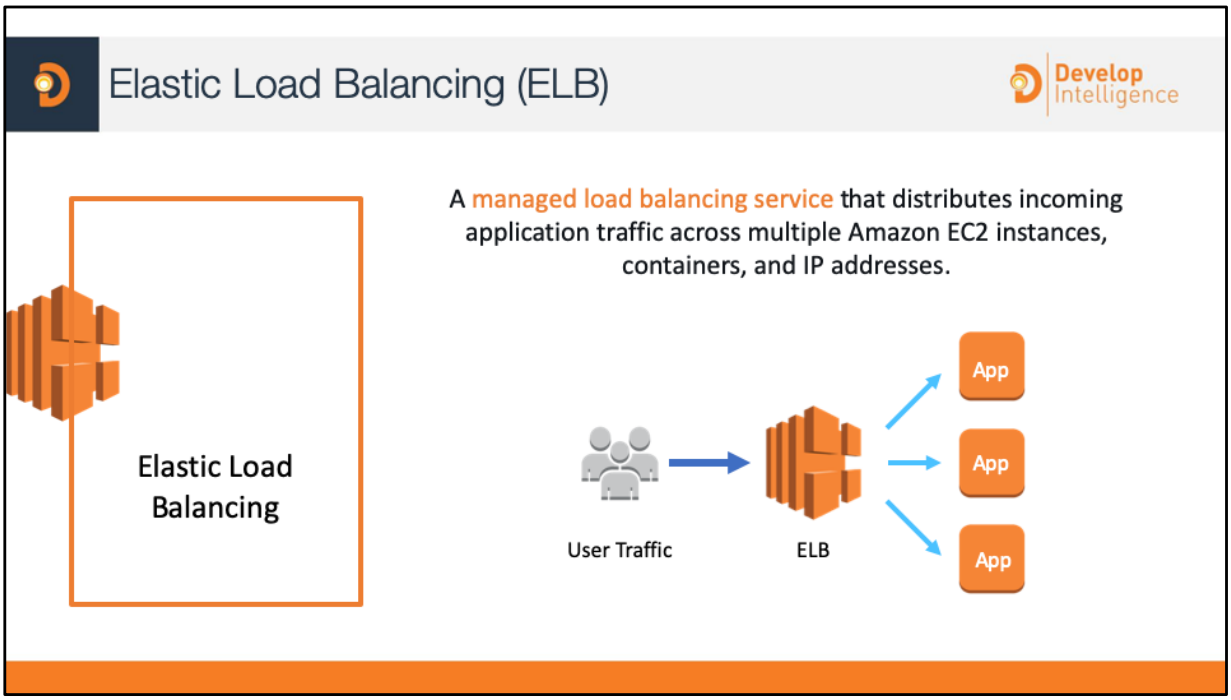
A *gateway endpoint* is a gateway that is a target for a specified route in your route table, used for traffic destined to a supported AWS service.



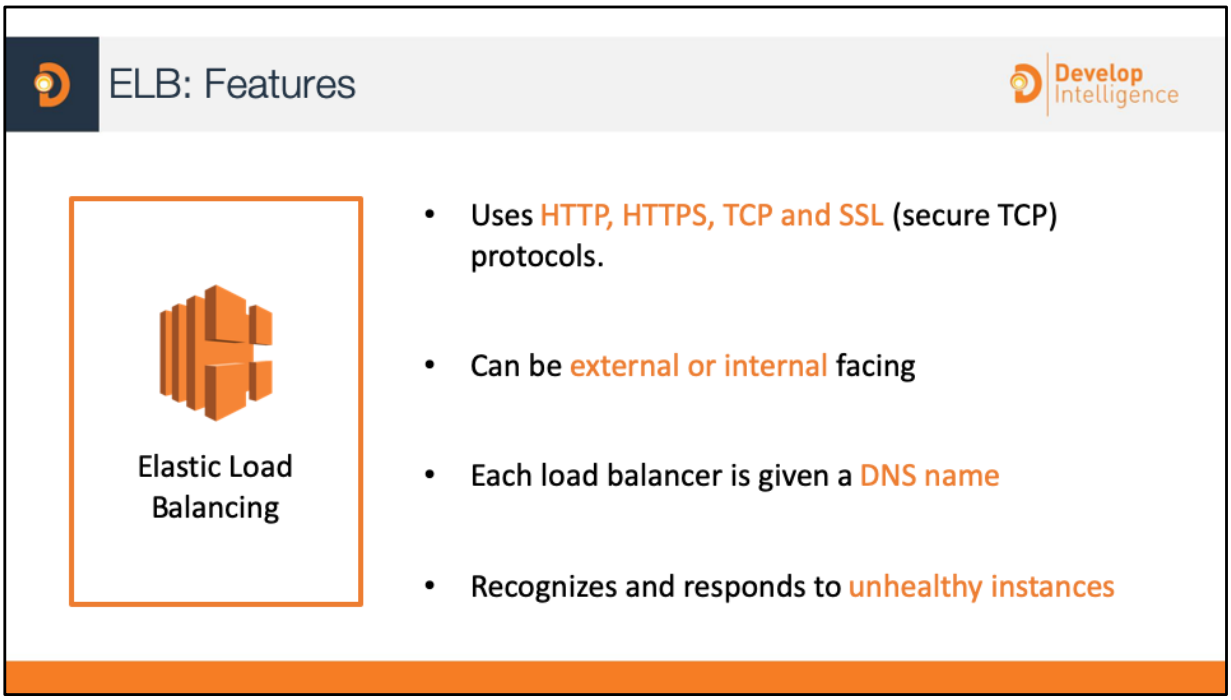
Corporate Domain Name Service (DNS)
The first step in using a VPC endpoint from a remote network is to identify the traffic to redirect through the endpoint. This solution uses corporate DNS servers to override DNS resolution for VPC-endpoint-specific traffic. In the example above, the DNS servers are configured to resolve s3.amazonaws.com to an internal ELB load balancer, which redirects traffic destined for US Standard S3 buckets to the VPC endpoint. This sends Amazon S3 requests from the corporate network to the S3 bucket over a private VPN or DX connection instead of over the internet.

Elastic Load Balancing (ELB)
ELB automatically distributes incoming Amazon S3 TCP connections across multiple Amazon EC2 proxy instances. It enables greater levels of fault tolerance for the proxy farm by seamlessly providing the required amount of load balancing capacity needed to distribute S3 traffic across multiple proxy servers. Additionally, configure the ELB load balancer to leverage multiple Availability Zones for maximum fault tolerance.

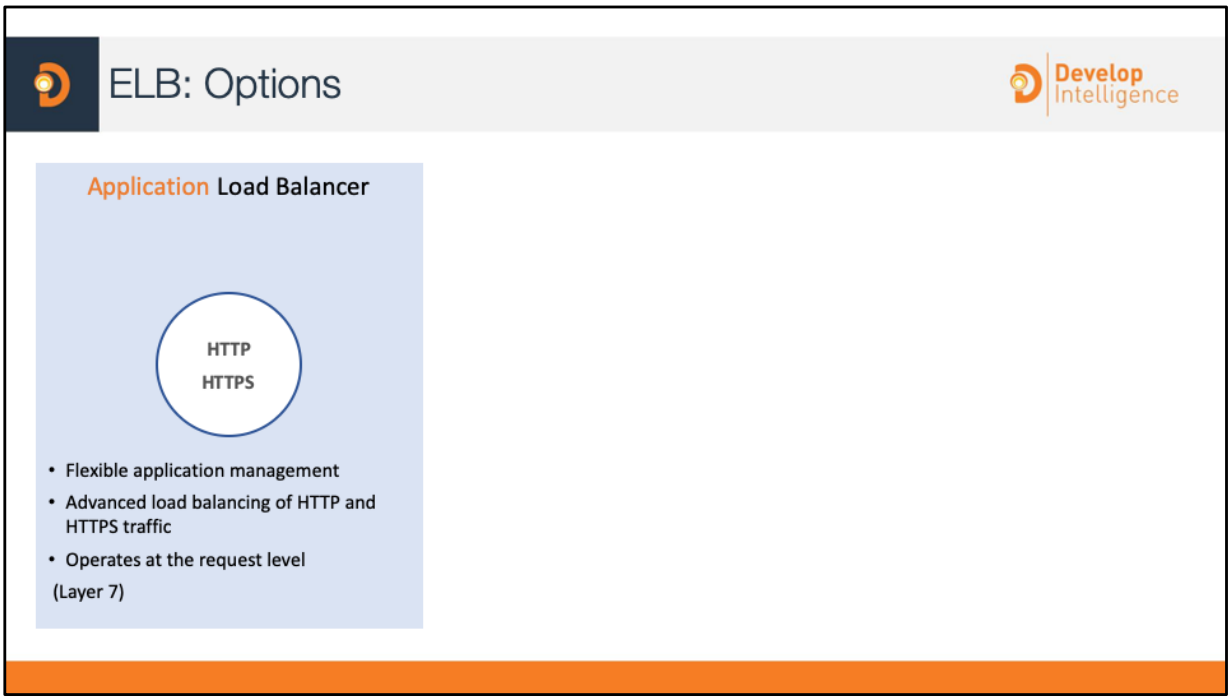
Proxy Farm
The proxy farm proxies Amazon S3 traffic to the VPC endpoint. The proxy farm can use access control lists (ACLs) to provide additional control over VPC endpoint traffic. An ACL can specify which remote users or networks are authorized to leverage the solution, and can further restrict the VPC endpoints or destination domains that clients can access. Configure an Auto Scaling group to manage the proxy servers and automatically grow or shrink the number of required instances based on proxy server load.



The foundation of the web tier includes the use of ELB in the architecture. These load balancers not only send traffic to EC2 instances, but can also send metrics to Amazon CloudWatch, a managed monitoring service provided. The metrics from Amazon EC2 and ELB can act as triggers—so that if you notice a particularly high latency or that our servers are becoming over-utilized, you can take advantage of Auto Scaling to add more capacity to your web server fleet.



ELB automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, and IP addresses. It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones. ELB offers three types of load balancers that all feature the high availability, automatic scaling, and robust security necessary to make your applications fault-tolerant.



ELB supports three types of load balancers: Application Load Balancers, Network Load Balancers, and Classic Load Balancers. You can select a load balancer based on your application needs.

An **Application Load Balancer** functions at the application layer, the seventh layer of the Open Systems Interconnection (OSI) model. Application Load Balancers support content-based routing and supports applications that run in containers. They support native Web Sockets over HTTP or HTTPS as well as HTTP/2 with HTTPS listeners. They also check the health of the targets, whether it's an EC2 instance or a container. Websites and mobile apps, running in containers or on EC2 instances, will benefit from the use of Application Load Balancers.

The **Network Load Balancer** is designed to handle tens of millions of requests per second while maintaining high throughput at ultra-low latency, with no effort on your part. It accepts incoming traffic from clients and distributes this traffic across the targets within the **same** Availability Zone. Network Load Balancers operate at the connection level (Layer 4), routing connections to targets—Amazon EC2 instances, containers, and IP addresses based on IP protocol data. The Network Load Balancer is API-compatible with the Application Load Balancer, including full programmatic control of target groups and targets.

The Network Load Balancer is ideal for balancing TCP traffic. Network Load Balancers are optimized to handle sudden and volatile traffic patterns while using a single static IP address per Availability Zone.

The **Classic Load Balancer** provides basic load balancing across EC2 instances in multiple Availability Zones and operates at both the request level and connection level of the OSI.

ELB: Options

Develop Intelligence

Application Load Balancer

- Flexible application management
- Advanced load balancing of HTTP and HTTPS traffic
- Operates at the request level (Layer 7)

Network Load Balancer

- Extreme performance and static IP for your application
- Load balancing of TCP traffic
- Operates at the connection level (Layer 4)

ELB: Options

Develop Intelligence

Application Load Balancer

- Flexible application management
- Advanced load balancing of HTTP and HTTPS traffic
- Operates at the request level (Layer 7)

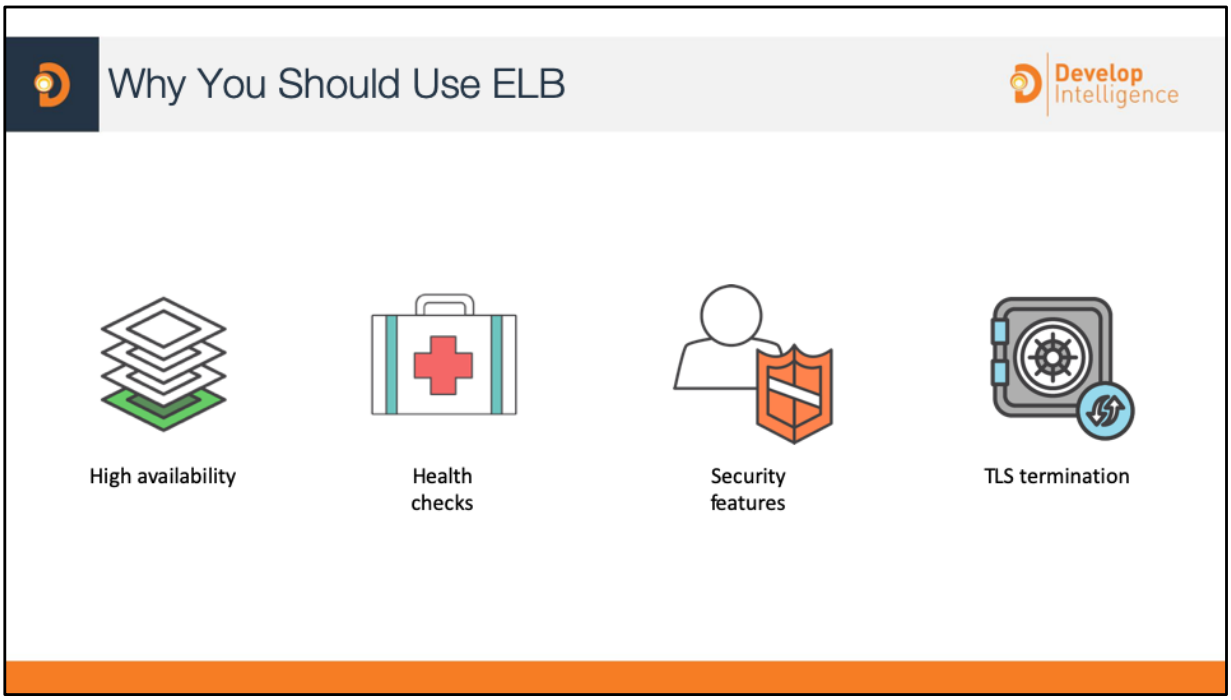
Network Load Balancer

- Extreme performance and static IP for your application
- Load balancing of TCP traffic
- Operates at the connection level (Layer 4)

Classic Load Balancer

PREVIOUS GENERATION
for HTTP, HTTPS, and TCP

- Existing application that was built within the EC2 Classic network
- Operates at both the request level and connection level



High availability
ELB automatically distributes traffic across multiple targets—Amazon EC2 instances, containers and IP addresses—in a single Availability Zone or multiple Availability Zones.

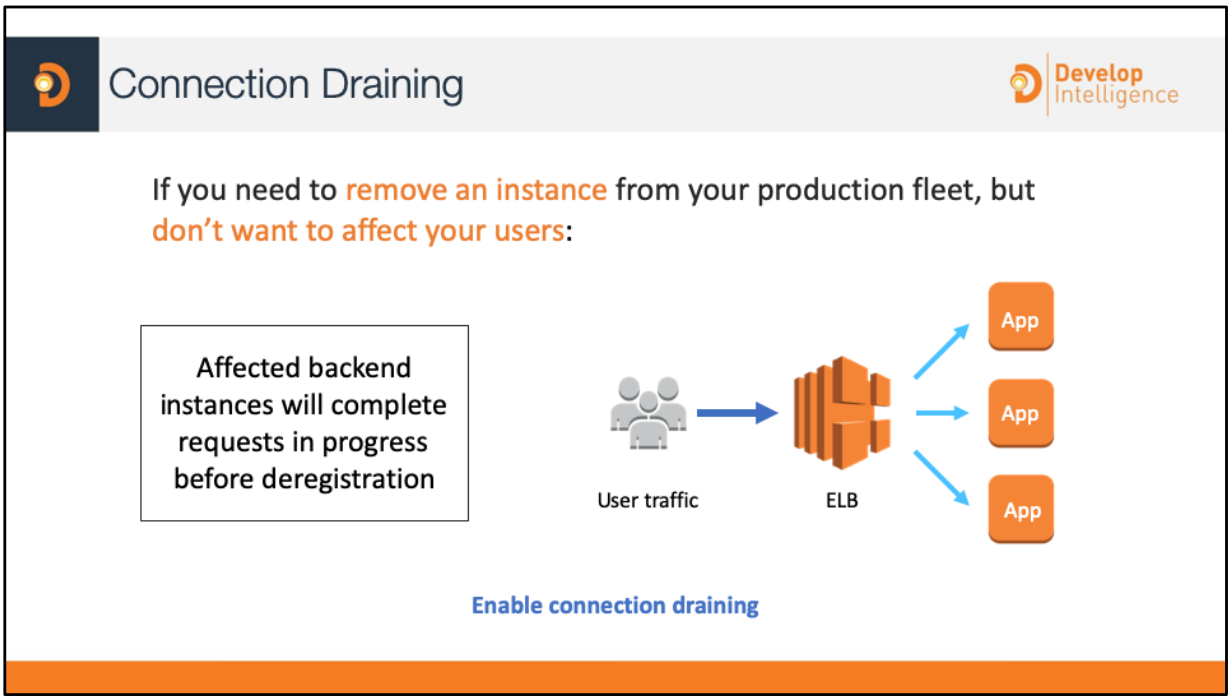
Health checks
To discover the availability of your Amazon EC2 instances, the load balancer periodically sends pings, attempts connections, or sends requests to test the Amazon EC2 instances. These tests are called *health checks*. Each registered Amazon EC2 instance must respond to the target of the health check with an HTTP status code 200 to be considered healthy by your load balancer.

Security features
ELB load balancers provisioned within an Amazon VPC can leverage its network security features, such as security groups.

Transport Layer Security Termination
ELB provides integrated certificate management and SSL decryption, allowing you the flexibility to centrally manage the SSL settings of the load balancer and offload CPU-intensive work from your application.

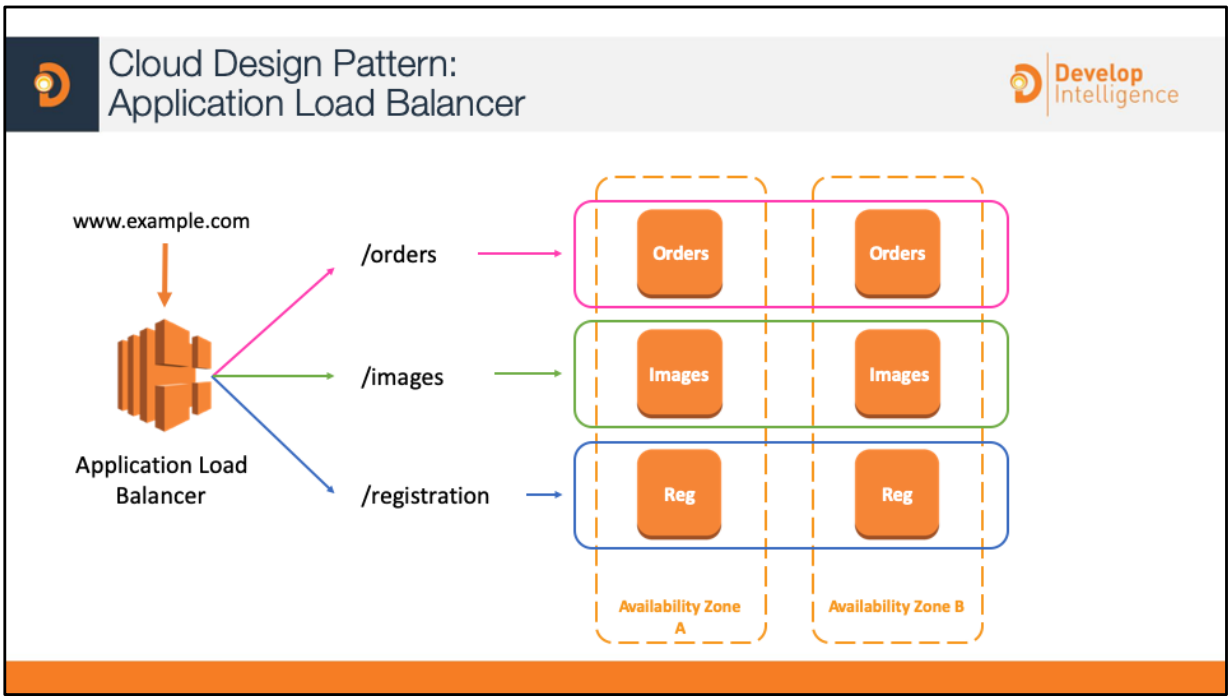
Layer 4 or Layer 7 load balancing
You can balance HTTP/HTTPS applications for Layer 7-specific features, or use strict Layer 4 load balancing for applications that rely purely on the TCP protocol.

For a side-by-side feature comparison, see https://aws.amazon.com/elasticloadbalancing/details/#compare.



When you enable connection draining on a load balancer, any backend instances that you deregister will complete requests that are in progress before deregistration. Likewise, if a backend instance fails health checks, the load balancer will not send any new requests to the unhealthy instance. It will allow existing requests to be completed, while ensuring that in-flight requests continue to be served. That means you can perform maintenance like deploying software upgrades or replacing back-end instances without affecting your customers' experience.

Connection draining is also integrated with Auto Scaling, which makes it even easier to manage the capacity behind your load balancer. When connection draining is enabled, Auto Scaling will wait for outstanding requests to be completed before terminating instances.



For more information, see https://aws.amazon.com/blogs/devops/introducing-application-load-balancer-unlocking-and-optimizing-architectures/

What is High Availability?

Develop Intelligence

Your application can recover from a failure or roll over to a secondary source within an acceptable amount of degraded performance time.

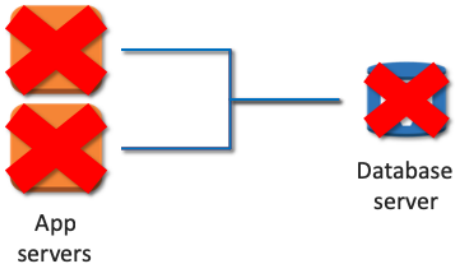
Percent of Uptime	Max Downtime per Year	Equivalent Downtime per Day
90%	36.5 days	2.4 hrs
99%	3.65 days	14 min
99.9%	8.76 hrs	86 sec
99.99%	52.6 min	8.6 sec
99.999%	5.25 min	.86 sec

High Availability Example

Develop Intelligence

Assume everything fails, and design backward

Implement redundancy where possible in order to prevent single failures from bringing down an entire system.

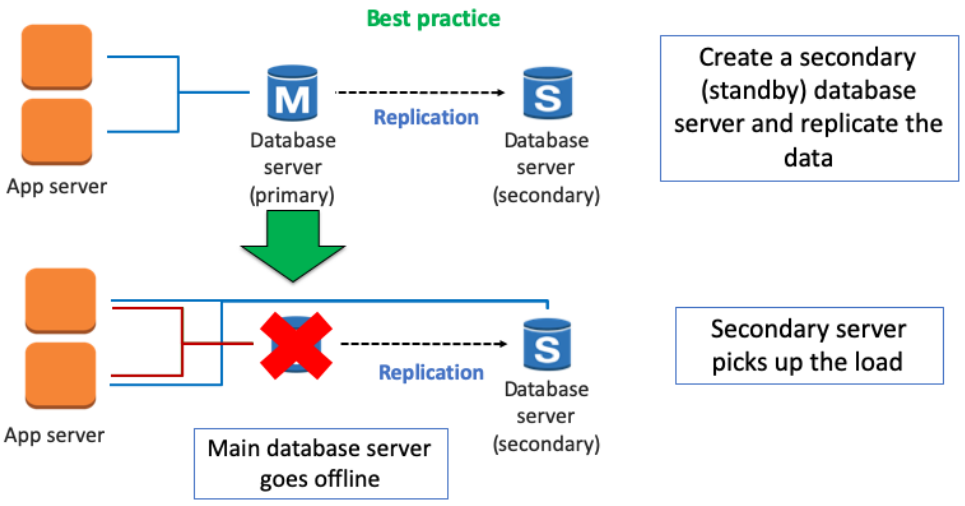


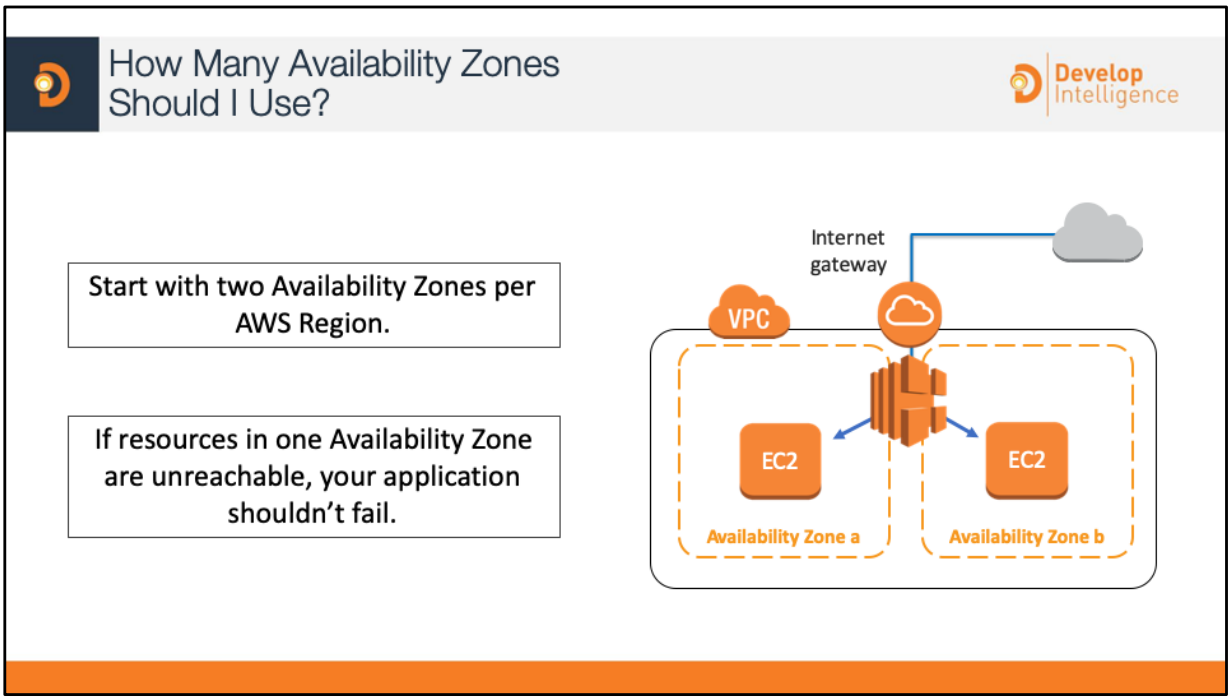
App servers

Database server

Anti-pattern

High Availability Example





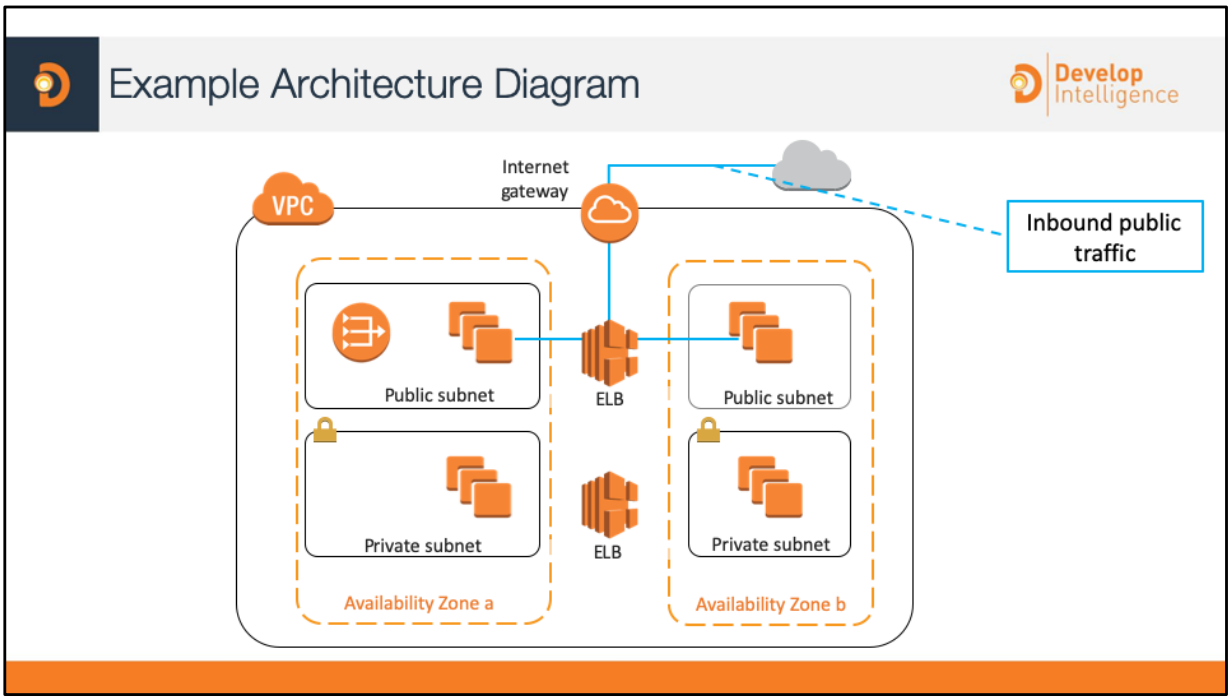
Most applications can be designed to support two Availability Zones. They may not benefit from the use of more Availability Zones than that, if you use data sources that only support primary/secondary failover. Availability Zones are spread out physically, so you won't receive much benefit from duplicating your resources in three or more Availability Zones in one AWS Region.

For heavy Amazon EC2 Spot Instance usage or data sources that go beyond active/passive, such as Amazon DynamoDB, there may be a benefit to using more than two Availability Zones.

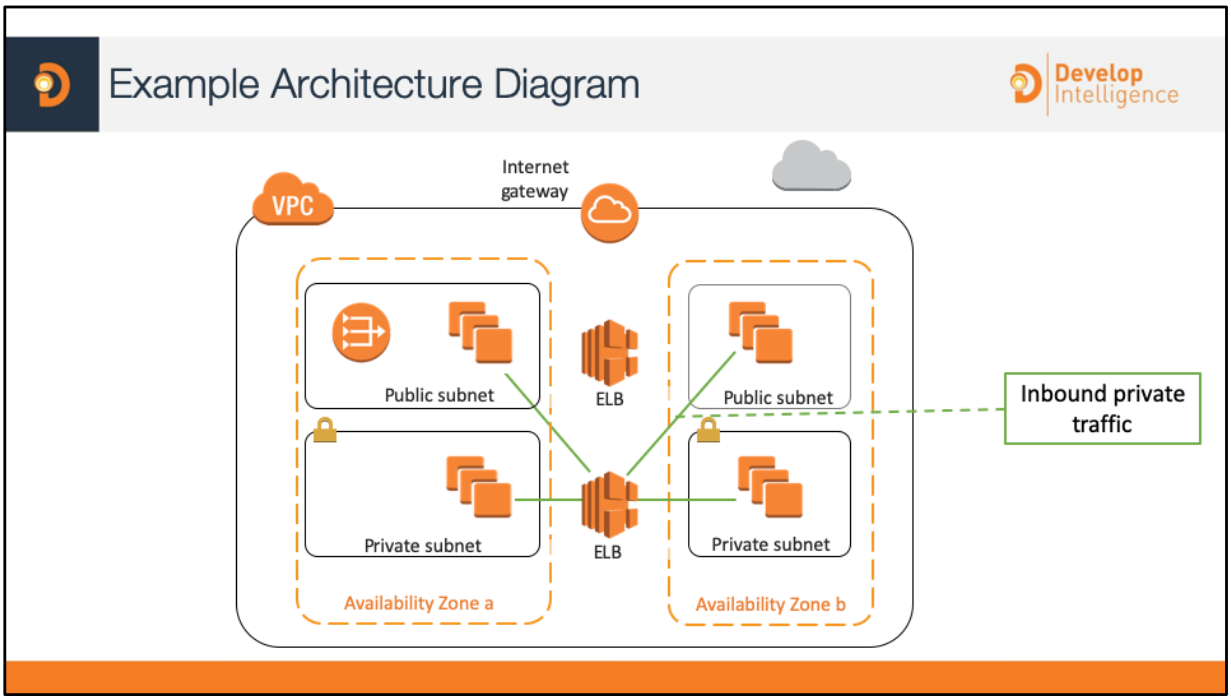
In this basic pattern, the two web servers (Amazon EC2) are positioned behind an ELB load balancer, which distributes traffic between them. If one of the servers becomes unavailable, the load balancer recognizes this. It stops distributing traffic to the unhealthy instance. That way, if there's a problem in one of the Availability Zones where a component resides, your application is still available.

You can further increase the availability of your infrastructure using other methods, which will be covered in a later module.

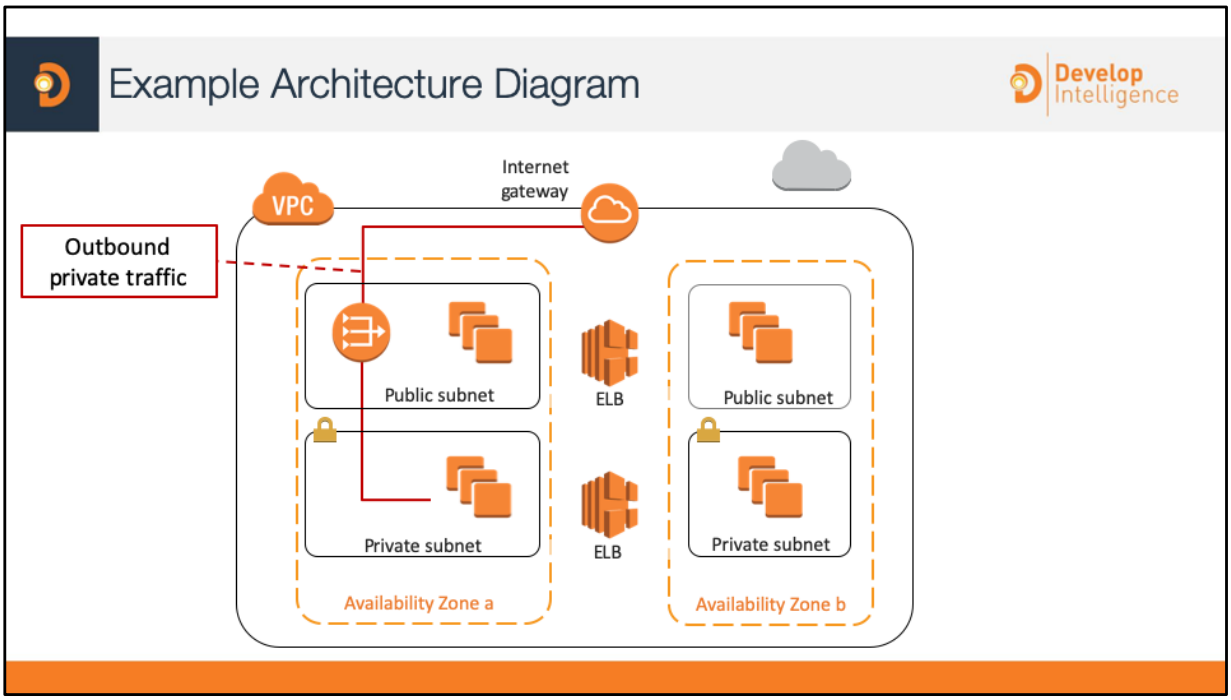
Example Architecture Diagram

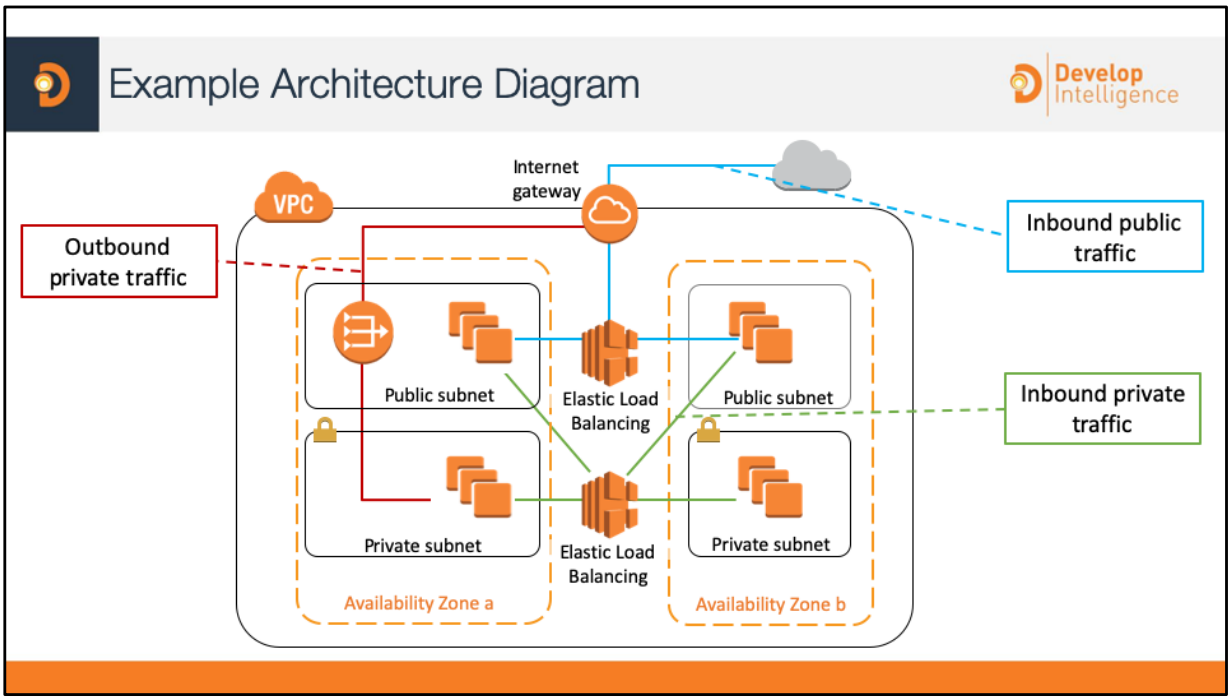


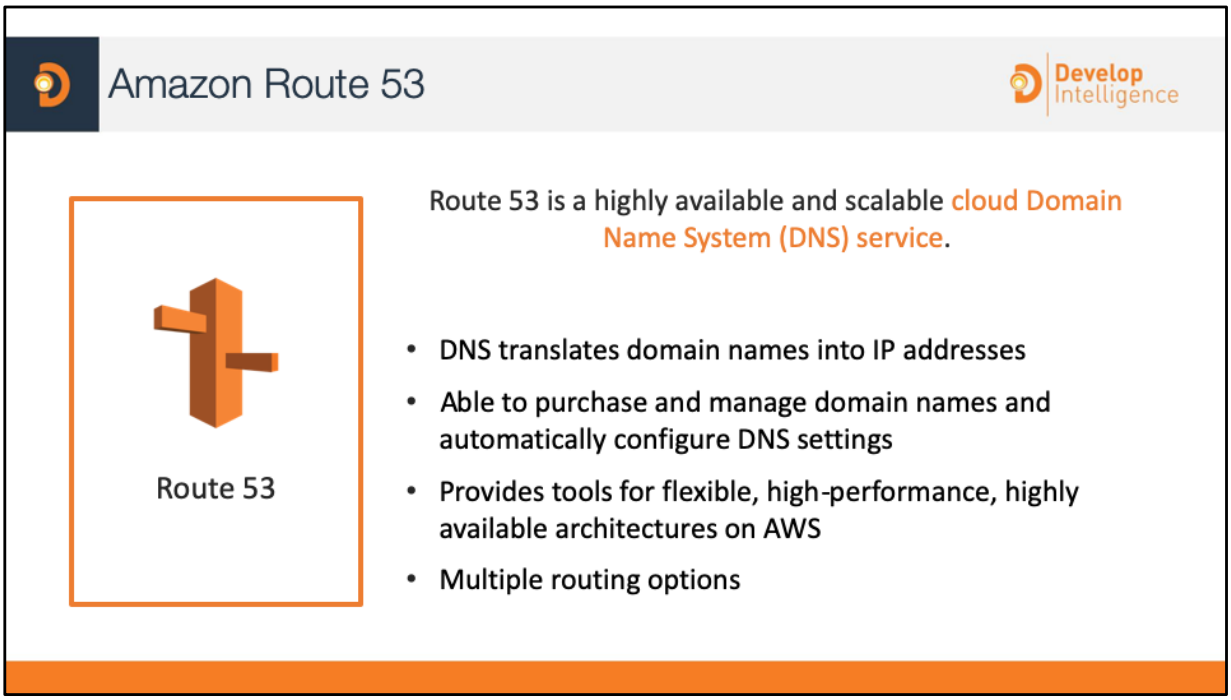
Example Architecture Diagram



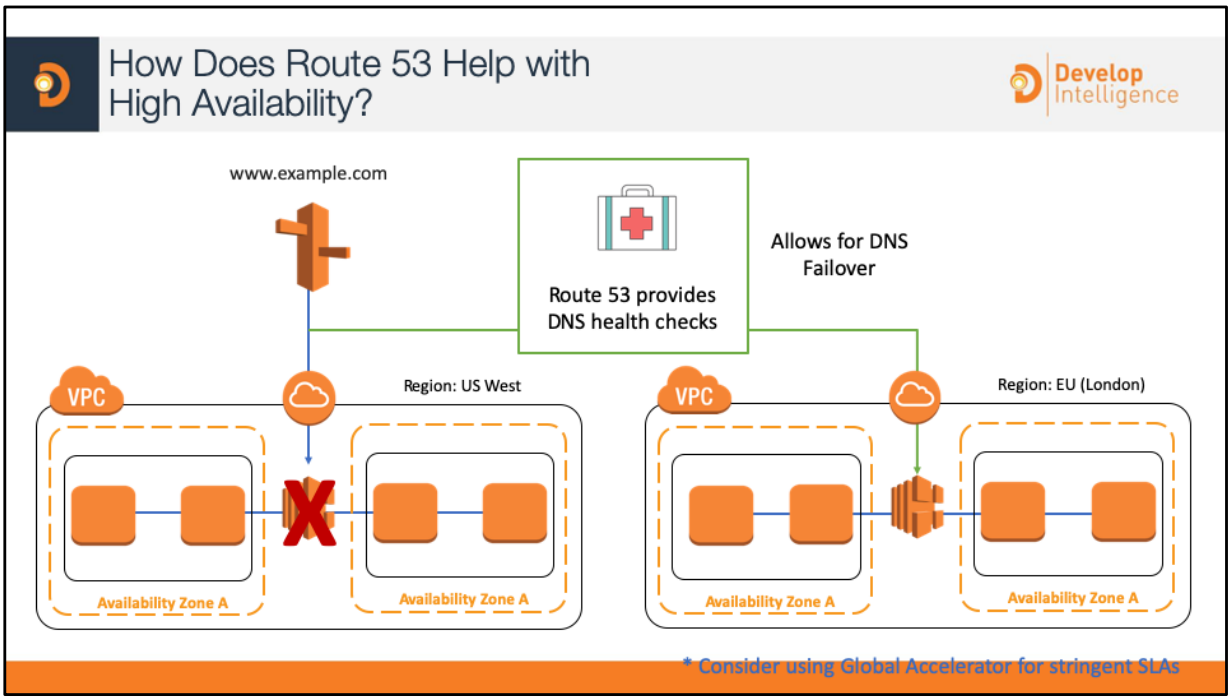
Example Architecture Diagram




Example Architecture Diagram
Develop Intelligence
Internet gateway
VPC
Outbound private traffic
Inbound public traffic
Inbound private traffic
Public subnet
Elastic Load Balancing
Public subnet
Private subnet
Elastic Load Balancing
Private subnet
Availability Zone a
Availability Zone b



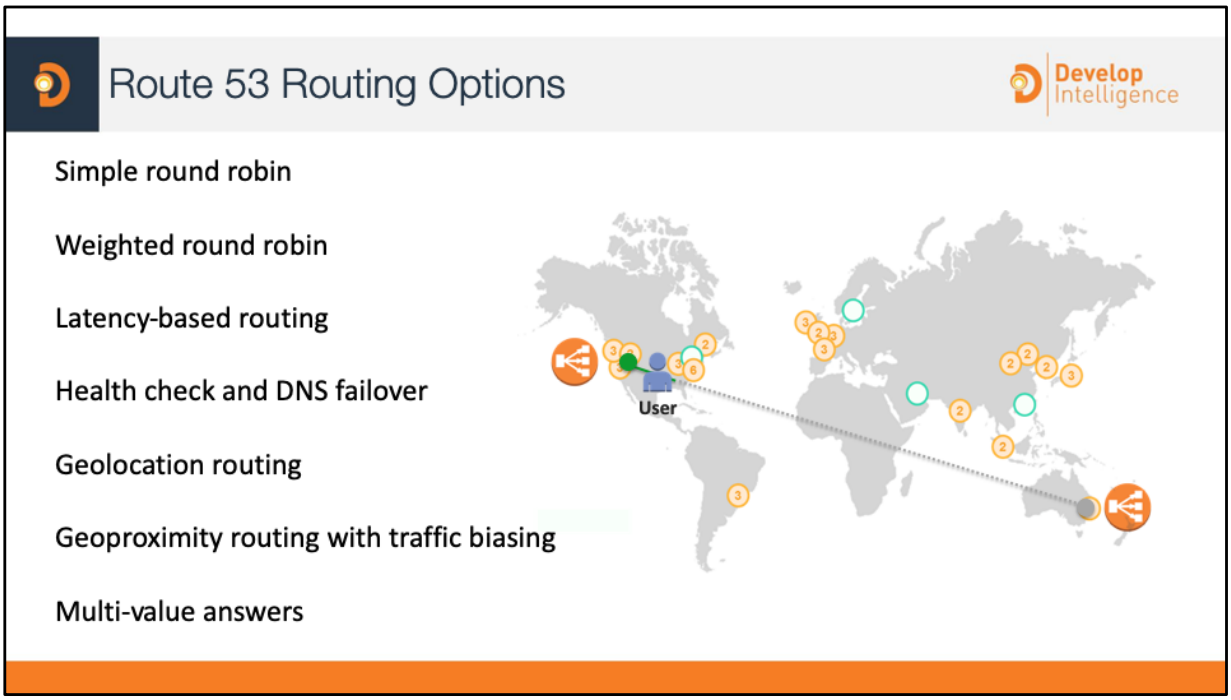
Amazon Route 53 provides a Domain Name System (DNS), domain name registration, and health-checking web services. The service was designed to give developers and businesses a reliable and cost-effective way to route end users to internet applications by translating names like *example.com* into the numeric IP addresses, such as *192.0.2.1*, that computers use to connect to each other. You can combine your DNS with health-checking services to route traffic to healthy endpoints or to independently monitor and/or alarm on endpoints. You can also purchase and manage domain names such as *example.com* and automatically configure DNS settings for your domains. Route 53 effectively connects user requests to infrastructure running in AWS—such as Amazon EC2 instances, ELB load balancers, or Amazon S3 buckets—and can also be used to route users to infrastructure outside of AWS.



If you have stringent SLAs or your application demands the fastest possible failover; please consider adding global accelerator into your architecture.

Global Accelerator provides your network with greater resiliency by removing the role of DNS in failover. It can protect your users and applications from caching issues and allows nearly instantaneous redirection of traffic to healthy endpoints. Additionally, when you add new endpoints into your architecture, they can receive traffic immediately without waiting for DNS propagation.

Global accelerator uses Static IP addresses that are Anycast from multiple AWS edge locations giving us a fixed entry point address that enables ingress of user traffic at an edge location closest to them



Simple routing (round robin) distributes the number of of requests as evenly as possible between all participating servers

Weighted round robin allows you to assign weights to resource record sets in order to specify the frequency with which different responses are served. You may want to use this capability to do A/B testing, sending a small portion of traffic to a server on which you've made a software change. For instance, suppose you have two record sets associated with one DNS name: one with weight 3 and one with weight 1. In this case, 75 percent of the time, Amazon Route 53 will return the record set with weight 3. Twenty-five percent of the time, Amazon Route 53 will return the record set with weight 1. Weights can be any number between 0 and 255.

Latency-based routing (LBR) helps you improve your application's performance for a global audience. LBR works by routing your customers to the AWS endpoint (e.g., Amazon EC2 instances, Elastic IP addresses, or load balancers) that provides the fastest experience based on actual performance measurements of the different AWS regions where your application is running.

Amazon Route 53 health checks monitor the health and performance of your web applications, web servers, and other resources. Each health check that you create can monitor one of the following:

- The health of a specified resource, such as a web server
- The status of other health checks
- The status of an Amazon CloudWatch alarm

After you create a health check, you can get the status of the health check, get notifications when the status changes, and configure DNS failover.

Geolocation routing lets you choose the resources that serve your traffic based on the geographic location of your users (the origin of DNS queries). When you use geolocation routing, you can localize your content and present some or all of your website in the language of your users. You can also use geolocation routing to restrict distribution of content to only the locations in which you have distribution rights. You can also balance load across endpoints in a predictable, easy-to-manage way, so each end-user location is consistently routed to the same endpoint.

With **DNS failover,** Amazon Route 53 can help detect an outage of your website and redirect your end users to alternate locations where your application is operating properly. When you enable this feature, Amazon Route 53 health-checking agents will monitor each location/endpoint of your application to determine its availability. You can take advantage of this feature to increase the availability of your customer-facing application.

Geoproximity routing lets you route traffic based on the physical distance between your users and your resources if you're using Route 53 traffic flow. You can also route more or less traffic to each resource by specifying a positive or negative bias. When you create a traffic flow policy, you can specify either an AWS Region (if you're using AWS resources) or the latitude and longitude for each endpoint.

With **multi-value answers***,* if you want to route traffic approximately randomly to multiple resources, such as web servers, you can create one multi-value answer record for each resource and, optionally, associate an Amazon Route 53 health check with each record. For example, suppose you manage an HTTP web service with 12 web servers that each have their own IP address. No one web server could handle all of the traffic, but if you create a dozen multi-value answer records, Amazon Route 53 responds to DNS queries with up to eight healthy records in response to each DNS query. Amazon Route 53 gives different answers to different DNS resolvers. If a web server becomes unavailable after a resolver caches a response, client software can try another IP address in the response.