Securely Access Services Over AWS PrivateLink

January 2019

Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Contents

Abstract

Amazon Virtual Private Cloud (Amazon VPC) gives AWS customers the ability to define a virtual private network within the AWS cloud. Customers can build services securely within an Amazon VPC and provide access to these services internally and externally using traditional methods such as an internet gateway , VPC peering, network address translation (NAT), a virtual private network (VPN), and AWS Direct Connect. This whitepaper presents how AWS PrivateLink keeps network traffic private and allows connectivity from Amazon VPCs to services and data hosted on AWS in a secure and scalable manner.

This paper is intended for IT professionals who are familiar with the basic concepts of networking and AWS. Each section has links to relevant AWS documentation.

Introduction

The introduction of Amazon Virtual Private Cloud (Amazon VPC) in 2009 made it possible for customers to provision a logically-isolated section of the AWS cloud and launch AWS resources in a virtual network that they define. Traditional methods to access third-party applications or public AWS services from an Amazon VPC include using an internet gateway , virtual private network (VPN), AWS Direct Connect with a virtual private gateway, and VPC peering.

Figure 1 illustrates an example Amazon VPC and its associated components:

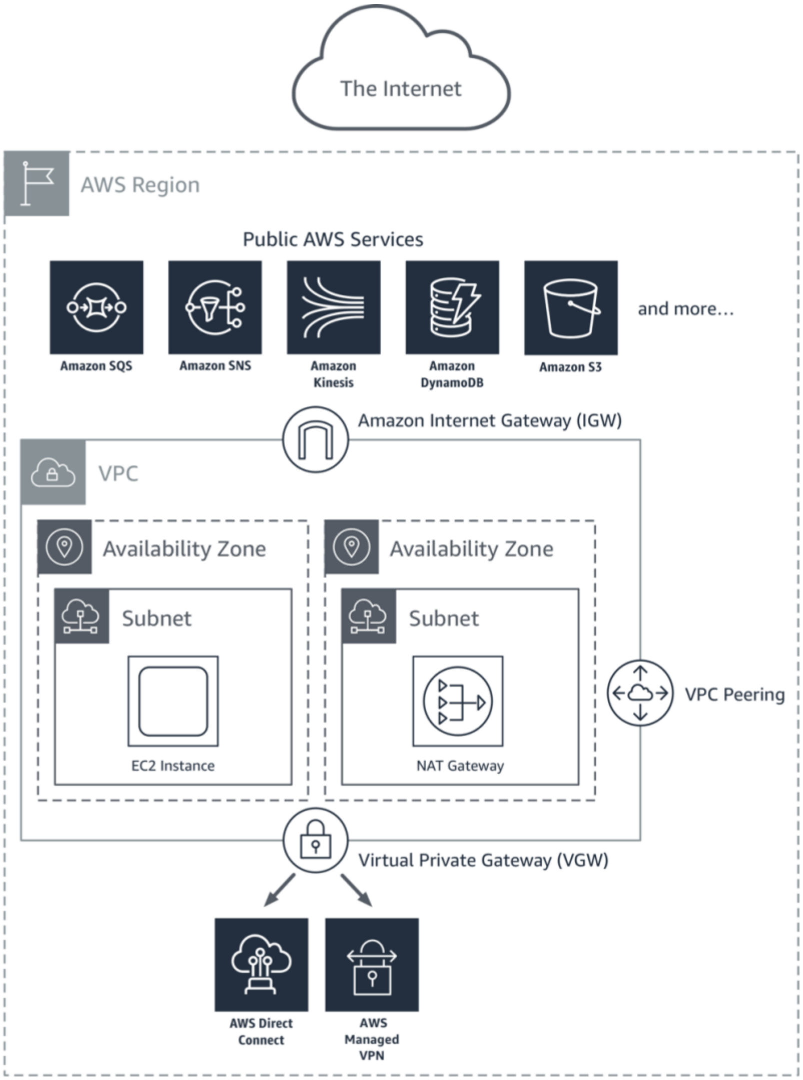

Figure 1: Traditional access from an Amazon VPC

What is AWS PrivateLink?

AWS PrivateLink provides secure, private connectivity between Amazon VPCs, AWS services, and on-premises applications on the AWS network. As a result, customers can simply and securely access services on AWS using Amazon's private network, powering connectivity to AWS services through interface Amazon VPC endpoints. Refer to Figure 2 for Amazon VPC-to-VPC connectivity using AWS PrivateLink.

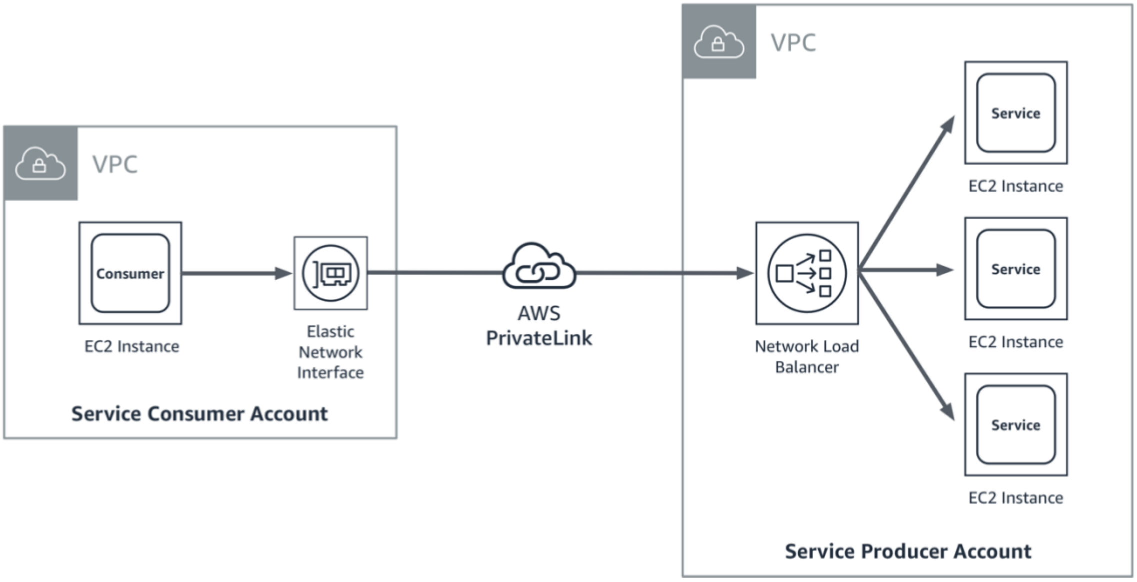

Figure 2: Amazon VPC-to-VPC connectivity with AWS PrivateLink

AWS PrivateLink also allows customers to create an application in their Amazon VPC, referred to as a service provider VPC, and offers that application as an AWS PrivateLink-enabled service or VPC endpoint service. A VPC endpoint service lets customers host a service and have it accessed by other consumers using AWS PrivateLink.

Why use AWS PrivateLink?

Prior to the availability of AWS PrivateLink, services residing in a single Amazon VPC were connected to multiple Amazon VPCs either (1) through public IP addresses using each VPC's internet gateway or (2) by private IP addresses using VPC peering. With AWS PrivateLink, service connectivity over Transmission Control Protocol (TCP) can be established from the service provider's VPC to the service consumers' VPCs in a secure and scalable manner.

AWS PrivateLink provides three main benefits:

Use Private IP Addresses for Traffic

AWS PrivateLink provides Amazon VPCs with a secure and scalable way to privately connect to AWS-hosted services. AWS PrivateLink traffic does not use public internet protocols (IP) addresses nor traverse the internet. AWS PrivateLink uses private IP addresses and security groups within an Amazon VPC so that services function as though they were hosted directly within an Amazon VPC.

Simplify Network Management

AWS PrivateLink helps avoid both (1) security policies that limit benefits of internet gateways and (2) complex networking across a large number of Amazon VPCs. AWS PrivateLink is easy to use and manage because it removes the need to whitelist public IPs and manage internet connectivity with internet gateways, NAT gateways, or firewall proxies.

AWS PrivateLink allows for connectivity to services across different accounts and Amazon VPCs with no need for route table modifications. There is no longer a need to configure an internet gateway, VPC peering connection, or Transit VPC to enable connectivity.

A Transit VPC connects multiple Amazon Virtual Private Clouds that might be geographically disparate or running in separate AWS accounts, to a common Amazon VPC that serves as a global network transit center. This network topology simplifies network management and minimizes the number of connections that you need to set up and manage. It is implemented virtually and does not require any physical network gear or a physical presence in a colocation transit hub.

Facilitate Your Cloud Migration

AWS PrivateLink gives on-premises networks private access to AWS services via AWS Direct Connect. Customers can more easily migrate traditional on-premises applications to services hosted in the cloud and use cloud services with the confidence that traffic remains private.

What are VPC Endpoints?

A VPC endpoint enables customers to privately connect to supported AWS services and VPC endpoint services powered by AWS PrivateLink. Amazon VPC instances do not require public IP addresses to communicate with resources of the service. Traffic between an Amazon VPC and a service does not leave the Amazon network.

VPC endpoints are virtual devices. They are horizontally scaled, redundant, and highly available Amazon VPC components that allow communication between instances in an Amazon VPC and services without imposing availability risks or bandwidth constraints on network traffic. There are two types of VPC endpoints: (1) *interface endpoints* and (2) *gateway endpoints*.

Interface endpoints

Interface endpoints enable connectivity to services over AWS PrivateLink. These services include some AWS managed services, services hosted by other AWS customers and partners in their own Amazon VPCs (referred to as endpoint services), and supported AWS Marketplace partner services. The owner of a service is a service provider. The principal creating the interface endpoint and using that service is a service consumer.

An interface endpoint is a collection of one or more elastic network interfaces with a private IP address that serves as an entry point for traffic destined to a supported service. Interface endpoints currently support over 17 AWS managed services. Check the AWS documentation for VPC endpoints for a list of AWS services that are available over AWS PrivateLink.

Gateway endpoints

A gateway endpoint targets specific IP routes in an Amazon VPC route table, in the form of a prefix-list, used for traffic destined to Amazon DynamoDB or Amazon Simple Storage Service (Amazon S3). Gateway endpoints do not enable AWS PrivateLink. More information about gateway endpoints is in the Amazon VPC User Guide.

Instances in an Amazon VPC do not require public IP addresses to communicate with VPC endpoints, as interface endpoints use local IP addresses within the consumer Amazon VPC. Gateway endpoints are destinations that are reachable from within an

Amazon VPC through prefix-lists within the Amazon VPC's route table. Refer to Figure 3 showing connectivity to AWS services using VPC endpoints.

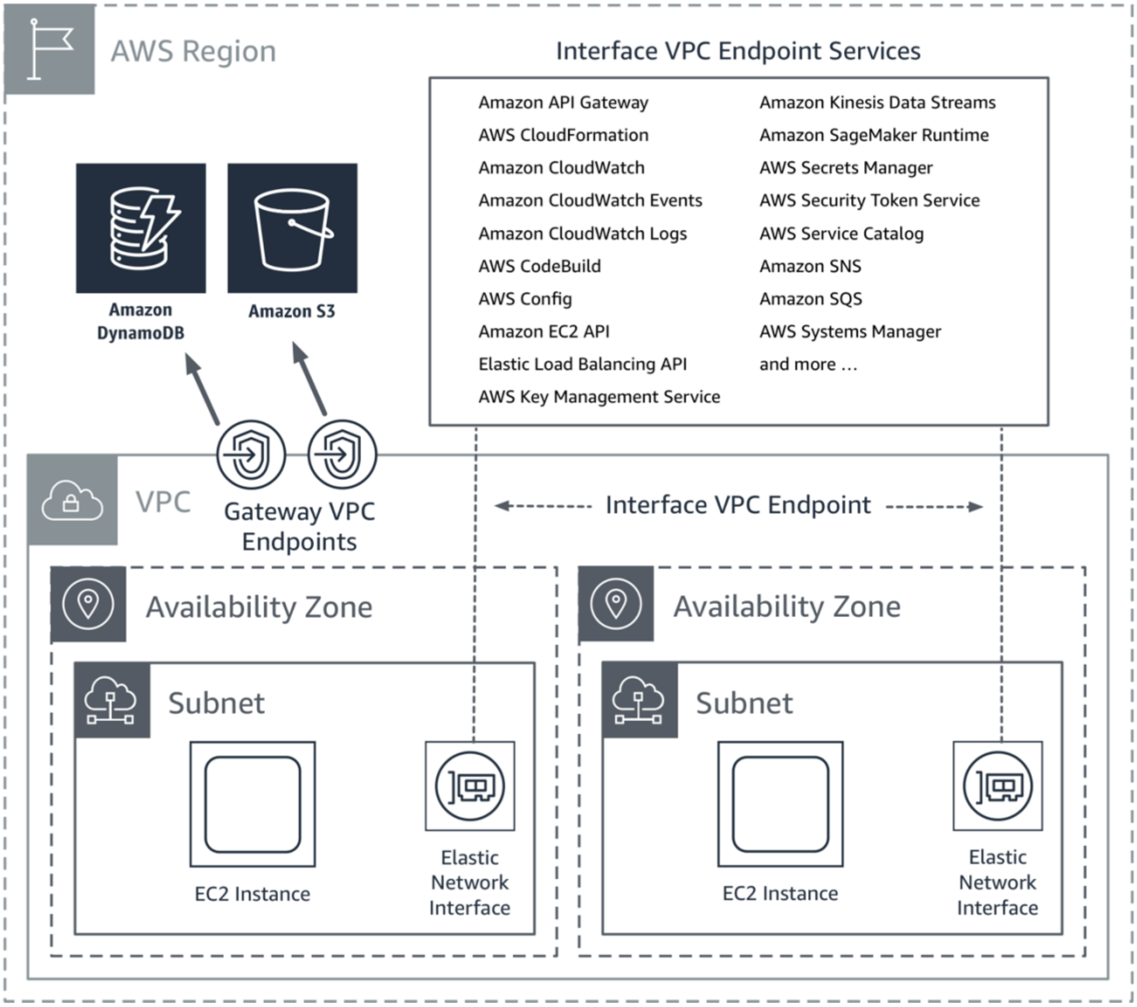

Figure 3 – Connectivity to AWS services using VPC endpoints

How does AWS PrivateLink work?

AWS PrivateLink uses Network Load Balancers to connect interface endpoints to services. A Network Load Balancer functions at the network transport layer (layer 4) and can handle millions of requests per second. In the case of AWS PrivateLink, it is represented inside the consumer Amazon VPC as an endpoint network interface. Customers can specify multiple subnets in different Availability Zones to ensure that their service is resilient to an Availability Zone service disruption. To achieve this, they can create endpoint network interfaces in multiple subnets mapping to multiple Availability Zones.

An endpoint network interface can be viewed in the account, but customers cannot manage it themselves. For more information, see Elastic Network Interfaces.

Creating Highly-Available Endpoint Services

The creation of VPC endpoint services goes through four stages, which we develop here. The generation of a DNS hostname, the use of private IP address, the deployment of the endpoint, and its configuration.

In Figure 4, the account owner of VPC B is a service provider and has a service running on instances in subnet B. The owner of VPC B has a service endpoint (vpce-svc-1234) with an associated Network Load Balancer that points to the instances in subnet B as targets. Instances in subnet A of VPC A use an interface endpoint to access the services in subnet B.

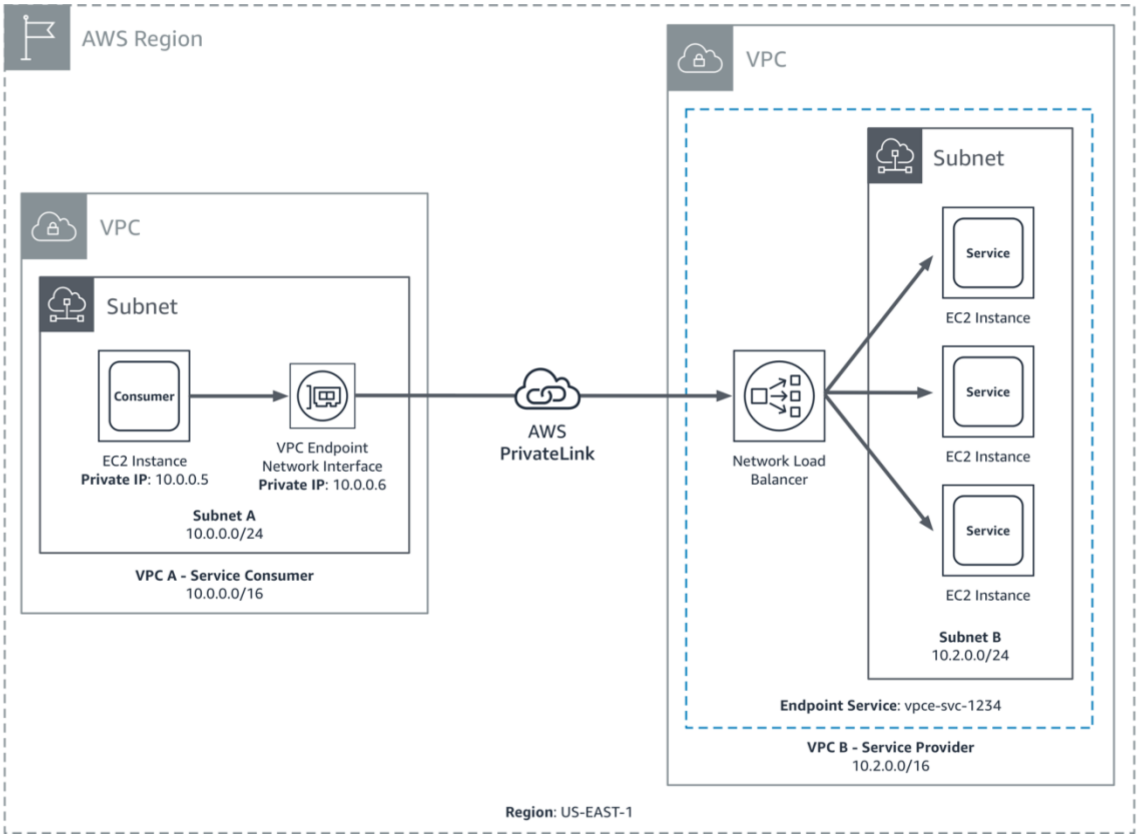

Figure 4: Detailed Amazon VPC-to-VPC connectivity with AWS PrivateLink

When an interface endpoint is created, endpoint-specific Domain Name System (DNS) hostnames are generated that can be used to communicate with the service. After creating the endpoint , requests can be submitted to the provider's service through one of the following methods:

The Endpoint-Specific Regional DNS Hostname

Customers generate an endpoint-specific DNS hostname which includes all zonal DNS hostnames generated for the interface endpoint. The hostname includes a unique endpoint identifier, service identifier, the region, and *vpce.amazonaws.com* in its name; for example:

```
vpce-0fe5b17a0707d6abc-29p5708s.ec2.us-east-
1.vpce.amazonaws.com
```

The Zonal-specific DNS Hostname

Customers generate a zonal specific DNS hostname for each Availability Zone in which the endpoint is available. The hostname includes the Availability Zone in its name; for example:

```
vpce-0fe5b17a0707d6abc-29p5708s-us-east-1a.ec2.us-east-
1.vpce.amazonaws.com
```

A Private DNS Hostname

If enabled, customers can use a private DNS hostname to alias the automatically-created zonal-specific or regional-specific DNS hostnames into a friendly hostname such as:

```
myservice.example.com
```

The Private IP Address of the Endpoint Network Interface

The private IP address of the endpoint network interface in the VPC is directly reachable to access the service in and across Availability Zones, in the same way the zonal-specific DNS hostname is.

Service providers that use zonal DNS hostnames to access the service can help achieve high availability by enabling cross-zone load balancing. Cross-zone load balancing enables the load balancer to distribute traffic across the registered targets in all enabled Availability Zones. Regional data transfer charges may apply to a service provider's account when they enable cross-zone load balancing, as data could potentially transfer between Availability Zones.

In Figure 5, the owner of VPC B is the service provider, and has configured a Network Load Balancer with targets in two different Availability Zones. The service consumer (VPC A) has created interface endpoints in the same two Availability Zones in their Amazon VPC. Requests to the service from instances in VPC A can use either interface endpoint. The DNS name resolution of the Endpoint Specific Regional DNS Hostname will alternate between the two IP addresses.

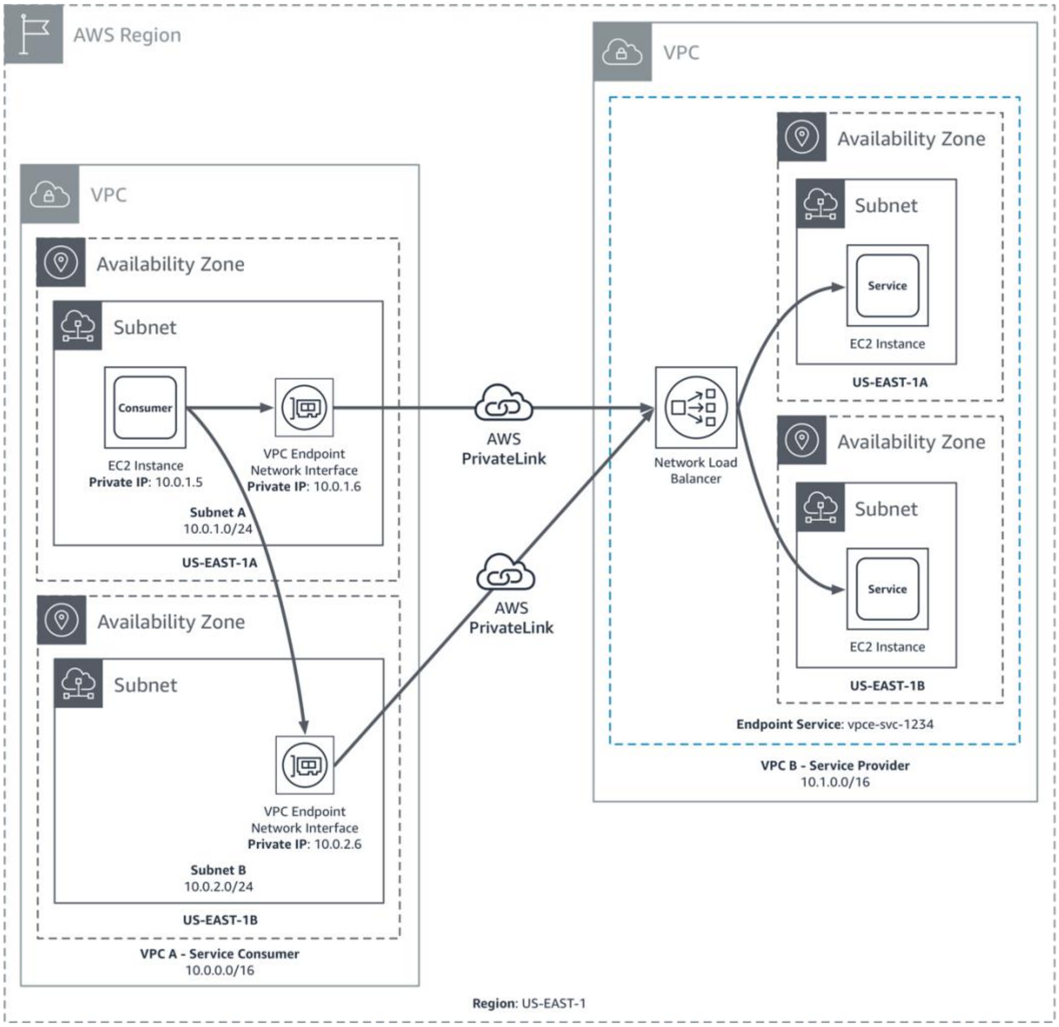

Figure 5: Round-robin DNS load balancing

Deploying AWS PrivateLink

AWS PrivateLink Considerations

When deploying an endpoint, customers should consider the following:

- Traffic will be sourced from the Network Load Balancer inside the service provider Amazon VPC. When service consumers send traffic to a service through an interface endpoint, the source IP addresses provided to the application are the private IP addresses of the Network Load Balancer nodes, and not the IP addresses of the service consumers.

- Proxy Protocol v2 can be enabled to gain insight into the network traffic. Network Load Balancers use Proxy Protocol v2 to send additional connection information such as the source and destination. This may require changes to the application.

- Proxy Protocol v2 can be enabled on the load balancer and the client IP addresses can be obtained from the Proxy Protocol header when IP addresses of the service consumers and their corresponding interface endpoint IDs are needed.

- Customers can create an Amazon Simple Notification Service (SNS) to receive alerts for specific events that occur on the endpoints that are attached or when they attempt to attach to their endpoint service. For example, one can receive an email when an endpoint request is accepted or rejected for the endpoint service.

- The Amazon SNS topic that a customer can use for notifications must have a topic policy that allows the VPC endpoint service to publish notifications on your behalf. Include the following statement in the topic policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "vpce.amazonaws.com"
      },
      "Action": "SNS:Publish",
      "Resource": "arn:aws:sns:region:account:topic-
name"
    }
  ]
}
```

  For more information see the documentation on Authentication and Access Control for Amazon SNS.

- Endpoint services cannot be tagged.

- The private DNS of the endpoint does not resolve outside of the Amazon VPC. For more information, read accessing a service through an interface endpoint. Note that private DNS hostnames can be configured to point to endpoint network interface IP addresses directly. Endpoint services are available in the AWS Region in which they are created and can be accessed in remote AWS Regions using Inter-Region VPC Peering.

- If an endpoint service is associated with multiple Network Load Balancers, then for a specific Availability Zone, an interface endpoint will establish a connection with one load balancer only.

- Availability Zone names in a customer account might not map to the same locations as Availability Zone names in another account. For example, the Availability Zone US-EAST-1A might not be the same Availability Zone as US-EAST-1A for another account. An endpoint service gets configured in Availability Zones according to their mapping in a customer's account.

- For low latency and fault tolerance, we recommend creating a Network Load Balancer with targets in each available Availability Zone of the AWS Region.

AWS PrivateLink Configuration

Full details on how to configure AWS PrivateLink can be found from the documentation on interface VPC endpoints.

Use-Case Examples

This section showcases some of the most common use cases for consuming and providing AWS PrivateLink endpoint services.

Private Access to SaaS Applications

AWS PrivateLink enables Software-as-a-Service (SaaS) providers to build highly scalable and secure services on AWS. Service providers can privately expose their service to thousands of customers on AWS with ease.

A SaaS (or service) provider can use a Network Load Balancer to target instances in their Amazon VPC which will represent their endpoint service. Customers in AWS can then be granted access to the endpoint service and create an interface VPC endpoint in their own Amazon VPC that is associated with the endpoint service. This allows customers to access the SaaS provider’s service privately from within their own Amazon VPC.

Follow the best practice of creating an AWS PrivateLink endpoint in each Availability Zone within the region that the service is deployed into. This provides a highly available and low-latency experience for service consumers.

Service consumers who are not already on AWS and want to access a SaaS service hosted on AWS can utilize AWS Direct Connect for private connectivity to the service provider. Customers can use an AWS Direct Connect connection to access service provider services hosted in AWS.

For example, a customer is interested in understanding their log data and selects a logging analytics SaaS offering hosted on AWS to ingest their logs in order to create visual dashboards. One way of transferring the logs into the SaaS provider’s service is to send them to the public-facing AWS endpoints of the SaaS service for ingestion.

With AWS PrivateLink, the service provider can create an endpoint service by placing their service instances behind a Network Load Balancer enabling customers to create an interface VPC endpoint in their Amazon VPC that is associated with their endpoint service. As a result, customers can privately and securely transfer log data to an interface VPC endpoint in their Amazon VPC and not over public facing AWS endpoints. See Figure 6 for an illustration.

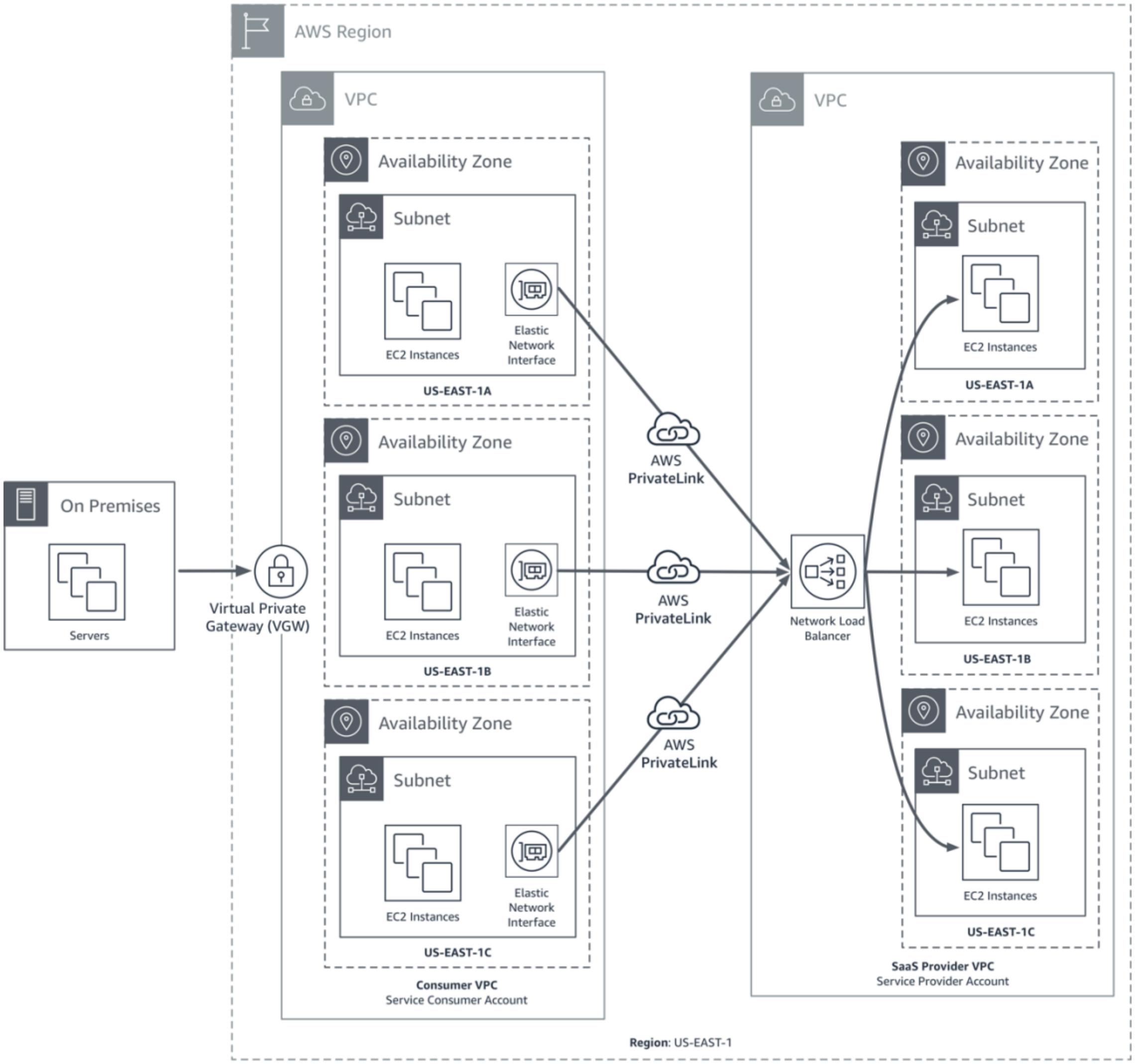

Figure 6: Private connectivity to cloud-based SaaS services

Shared Services

As customers deploy their workloads on AWS, common service dependencies will often begin to emerge among the workloads. These shared services include security services, logging, monitoring, Dev Ops tools, and authentication to name a few. These common services can be abstracted into their own Amazon VPC and shared among the workloads that exist in their own separate Amazon VPCs. The Amazon VPC that contains and shares the common services is often referred to as a Shared Services VPC.

Traditionally, workloads inside Amazon VPCs use VPC peering to access the common services in the Shared Services VPC. Customers can implement VPC peering effectively, however, there are caveats. VPC peering allows instances from one Amazon VPC to talk to any instance in the peered VPC. Customers are responsible for implementing fine grained network access controls to ensure that only the specific resources intended to be consumed from within the Shared Services VPC are accessible from the peered VPCs. In some cases, a customer running at scale can have hundreds of Amazon VPCs, and VPC peering has a limit of 125 peering connections to a single Amazon VPC.

AWS PrivateLink provides a secure and scalable mechanism that allows common services in the Shared Services VPC to be exposed as an endpoint service, and consumed by workloads in separate Amazon VPCs. The actor exposing an endpoint service is called a service provider. AWS PrivateLink endpoint services are scalable and can be consumed by thousands of Amazon VPCs.

The service provider creates an AWS PrivateLink endpoint service using a Network Load Balancer that then only targets specific ports on specific instances in the Shared Services VPC. For high availability and low latency, we recommend using a Network Load Balancer with targets in at least two Availability Zones within a region.

A service consumer is the actor consuming the AWS PrivateLink endpoint service from the service provider. When a service consumer has been granted permission to consume the endpoint service, they create an interface endpoint in their VPC that connects to the endpoint service from the Shared Services VPC. As an architectural best practice to achieve low latency and high availability, we recommend creating an Interface VPC endpoint in each available Availability Zones supported by the endpoint service. Service consumer VPC instances can use a VPC’s available endpoints to access the endpoint service via one of the following ways: (1) the private endpoint-specific DNS hostnames that are generated for the interface VPC endpoints or (2) the Interface VPC endpoint’s IP addresses.

On-premises resources can also access AWS PrivateLink endpoint services over AWS Direct Connect. Create an Amazon VPC with up to 20 interface VPC endpoints and associate with the endpoint services from the Shared Services VPC. Terminate the AWS Direct Connect connection’s private virtual interface to a virtual private gateway. Next, attach the virtual private gateway to the newly created Amazon VPC. Resources on-premises are then able to access and consume AWS PrivateLink endpoint services over the AWS Direct connection. Figure 7 illustrates a shared services Amazon VPC using AWS PrivateLink.

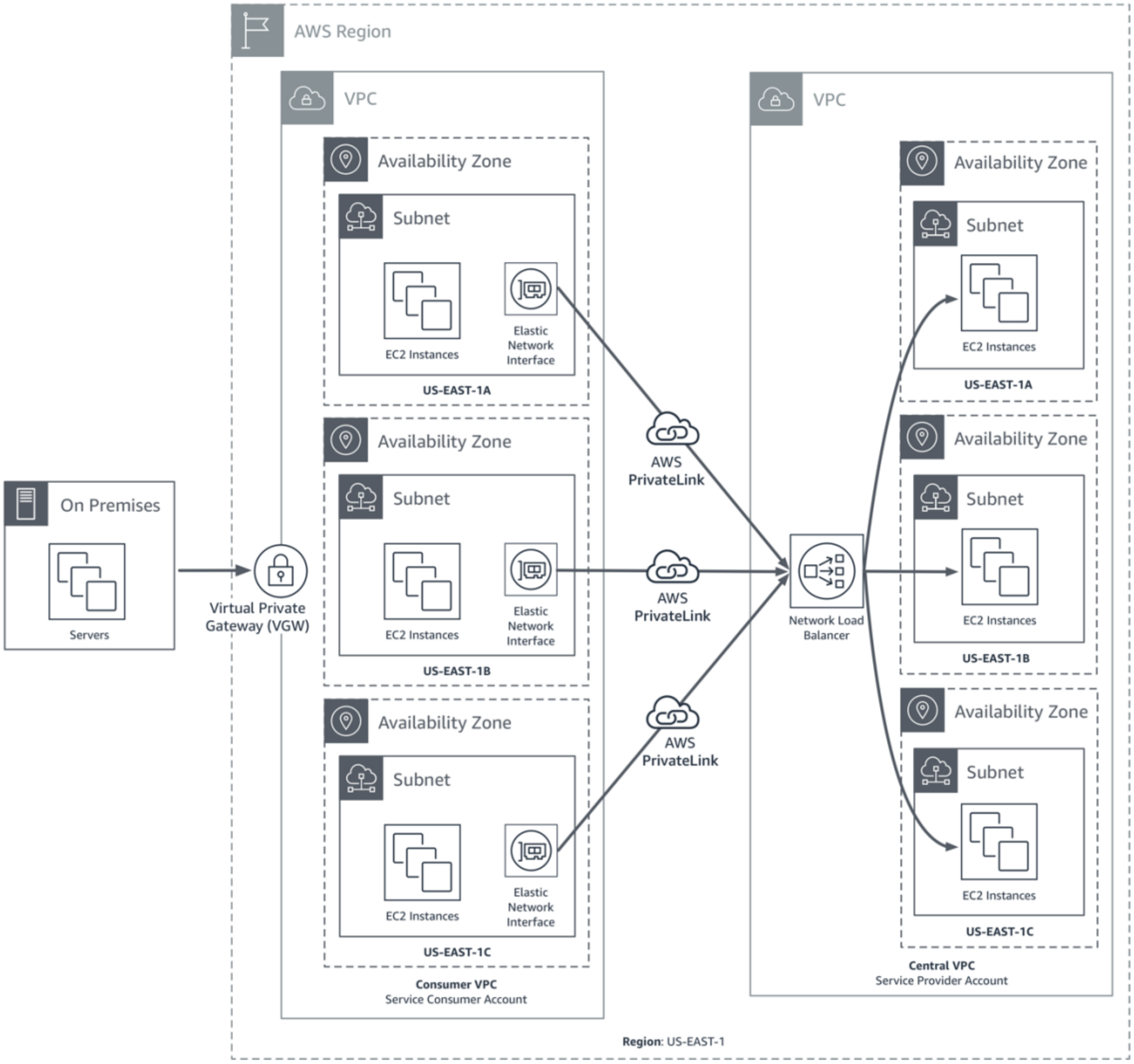

Figure 7: Shared Services VPC using AWS PrivateLink

Hybrid Services

As customers start their migration to the cloud, a common architecture pattern used is a hybrid cloud environment. This means that customers will begin to migrate their workloads into AWS over time, but they will also start to use native AWS services to serve their clients.

In a Shared Services VPC, the instances behind the endpoint service exist on the AWS cloud. AWS PrivateLink allows you to extend resource targets for the AWS PrivateLink endpoint service to resources in an on-premises data center.

The Network Load Balancer for the AWS PrivateLink endpoint service can use resources in an on-premises data center as well as instances in AWS. Service consumers on AWS still access the AWS PrivateLink endpoint service by creating an interface VPC endpoint that is associated with the endpoint service in their VPC, but the requests they make over the interface VPC endpoint will be forwarded to resources in the on-premises data center.

The Network Load Balancer enables the extension of a service architecture to load balance workloads across resources in AWS and on-premises resources, and makes it easy to migrate-to-cloud, burst-to-cloud, or failover-to-cloud. As customers complete the migration to the cloud, on-premises targets would be replaced by target instances in AWS and the hybrid scenario would convert to a Shared Services VPC solution. Refer to Figure 8 for a diagram on hybrid connectivity to services over AWS Direct Connect.

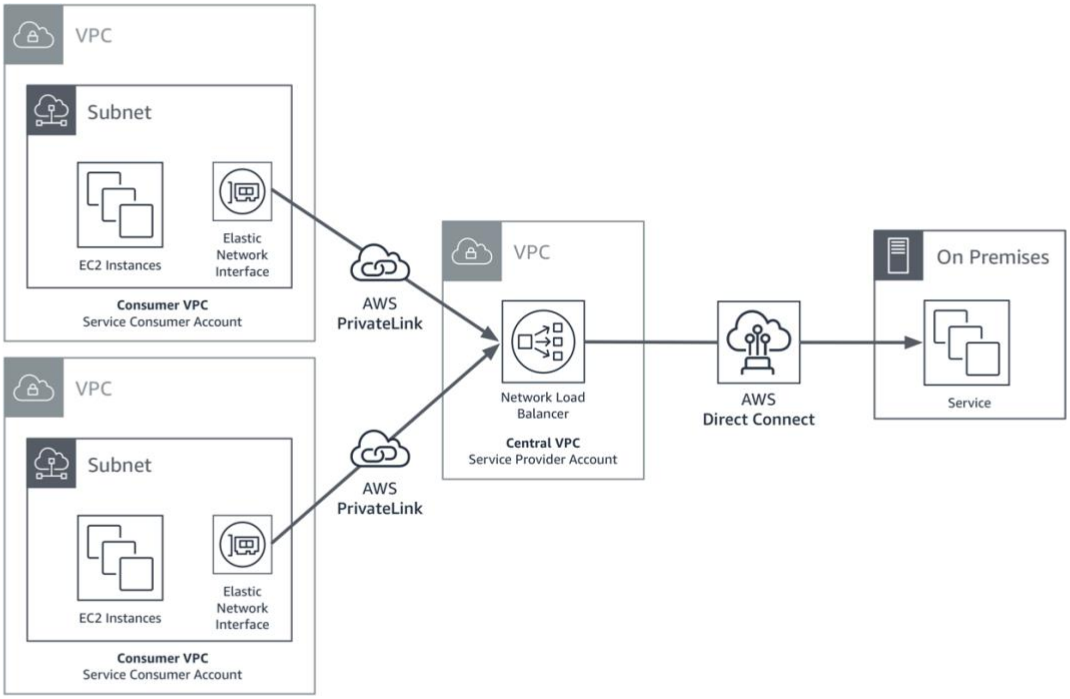

Figure 8: Hybrid connectivity to services over AWS Direct Connect

Presenting Microservices

Customers are continuing to adopt modern, scalable architecture patterns for their workloads. A microservice is a variant of the service-oriented architecture (SOA) that structures an application as a collection of loosely-coupled services that do one specialized job and do it well.

AWS PrivateLink is well suited for a microservices environment. Customers can give teams who own a particular service an Amazon VPC to develop and deploy their service in. Once they are ready to deploy the service for consumption by other services, they can create an endpoint service. For example, endpoint service may consist of a Network Load Balancer that can target Amazon Elastic Compute Cloud (Amazon EC2) instances or containers on Amazon Elastic Container Service (Amazon ECS). Service teams can then deploy their microservices on either one of these platforms and the Network Load Balancer would provide access to the service.

A service consumer would then request access to the endpoint service and create an interface VPC endpoint associated with an endpoint service in their Amazon VPC. The service consumer can then begin to consume the microservice over the interface VPC endpoint.

The architecture in Figure 9 shows microservices which are segmented into different Amazon VPCs, and potentially different service providers. Each of the consumers who have been granted access to the endpoint services would simply create interface VPC endpoints associated with the given endpoint service in their Amazon VPC for each of the microservices it wishes to consume. The service consumers will communicate with the AWS PrivateLink endpoints via endpoint-specific DNS hostnames that are generated when the endpoints are created in the Amazon VPCs of the service consumer.

The nature of a microservice is to have a call stack of various microservices throughout the lifecycle of a request. What is illustrated as a service consumer in Figure 9 can also become a service provider. The service consumer can aggregate what it needs from the services it consumed and present itself as a higher-level microservice.

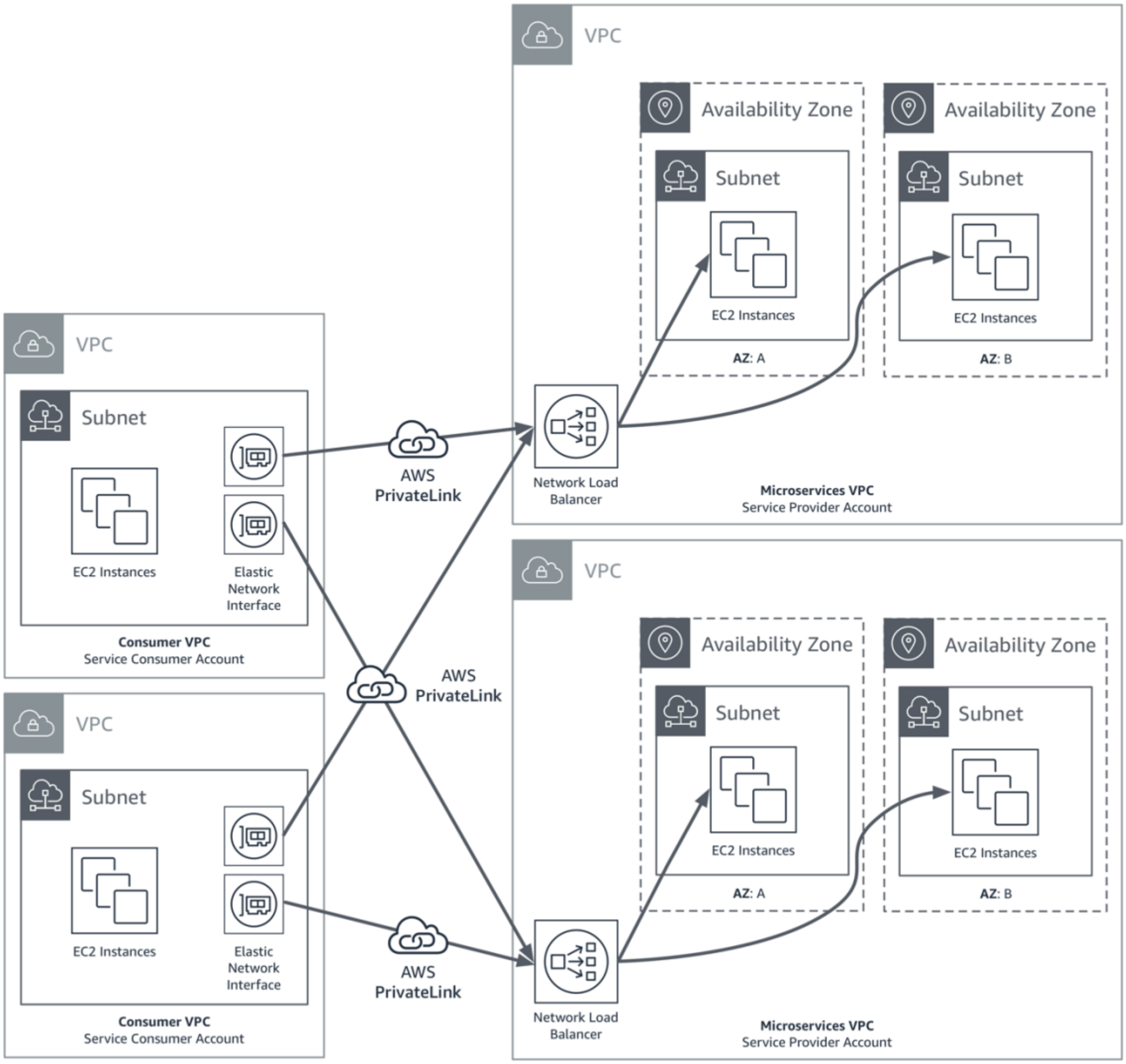

Figure 9: Presenting Microservices via AWS PrivateLink

Inter-Region Endpoint Services

Customers and SaaS providers who host their service in a single region can extend their service to additional regions through Inter-Region VPC Peering. Service providers can leverage a Network Load Balancer in a remote region and create an IP target group that uses the IPs of their instance fleet in the remote region hosting the service.

Inter-Region VPC Peering traffic leverages Amazon's private fiber network to ensure that services communicate privately with the AWS PrivateLink endpoint service in the remote region. This allows the service consumer to use local interface VPC endpoints to connect to an endpoint service in an Amazon VPC in a remote region.

Figure 10 shows Inter-Region Endpoint services. A service provider is hosting an AWS PrivateLink endpoint service in the US-EAST-1 region. Service consumers of the endpoint service require the service provider to provide a local interface VPC endpoint that is associated with the endpoint service in the EU-WEST-2 region.

Service providers can use Inter-Region VPC Peering to provide local endpoint service access to their customers in remote regions. This approach can help the service providers gain the agility to provide the access their customers want while not having to immediately deploy their service resources in the remote regions, but instead deploying them when they are ready. If the service provider has chosen to expand their service resources into remote regions that are currently using Inter-Region VPC Peering, the service provider will have to remove the targets from the Network Load Balancer in the remote region and point them to the targets in the local region.

Since the remote endpoint service is communicating with resources in a remote region, additional latency will be incurred when the service consumer communicates with the endpoint service. The service provider will also have to cover the costs for the Inter-Region VPC Peering data transfer. Depending on the workload, this could be a long-term approach for some service providers so long as they evaluate the pros and cons of the service consumer experience and their own operating model.

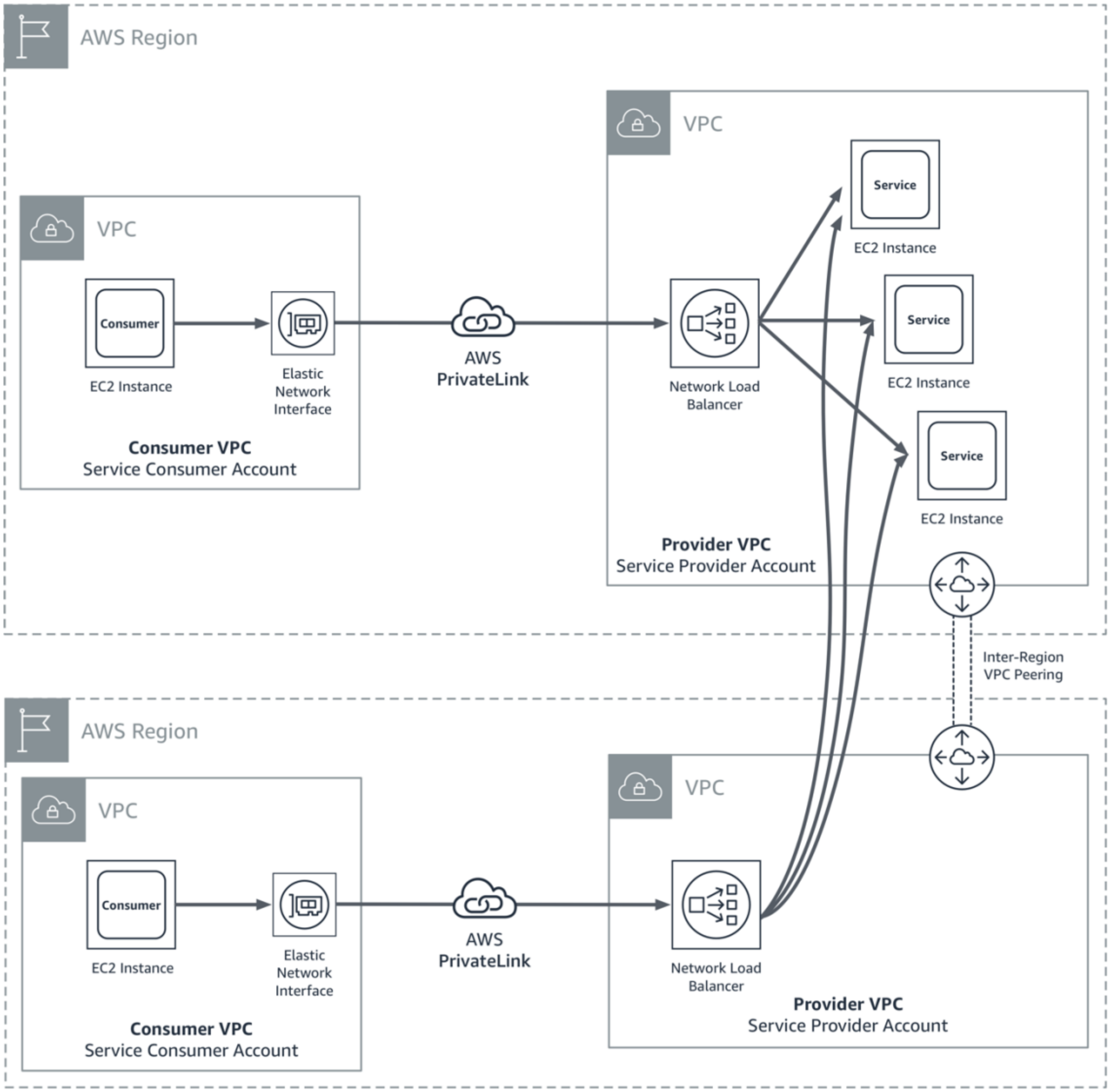

Figure 10: Inter-Region Endpoint Services

Inter-Region Access to Endpoint Services

As customers expand their global footprint by deploying workloads in multiple AWS regions across the globe, they will need to ensure that the services that depend on AWS PrivateLink endpoint services have connectivity from the region they are hosted in. Customers can leverage Inter-Region VPC Peering to enable services in another region to communicate with interface VPC endpoint terminating the endpoint service which directs traffic to the AWS PrivateLink endpoint service hosted in the remote region. Inter-Region VPC Peering traffic is transported over Amazon's network and ensures that your services communicate privately to the AWS PrivateLink endpoint service in the remote region.

Figure 11 visualizes the inter-region access to endpoint services. A customer has deployed a workload in the EU-WEST-1 region that needs to access an AWS PrivateLink endpoint service hosted in the US-EAST-1 region. The service consumer will first need to create an Amazon VPC in the region where the AWS PrivateLink endpoint service is currently being hosted in. They will then need to create an Inter-Region VPC Peering connection from the Amazon VPC in their region to the Amazon VPC in the remote region. The service consumer will then need to create an interface VPC endpoint in the Amazon VPC in the remote region that is associated with the endpoint service. The workload in the service consumers Amazon VPC can now communicate with the endpoint service in the remote region by leveraging Inter-Region VPC Peering. The service consumer will have to consider the additional latency when communicating with endpoint service hosted in the remote region, as well as the inter-region data transfer costs between the two regions.

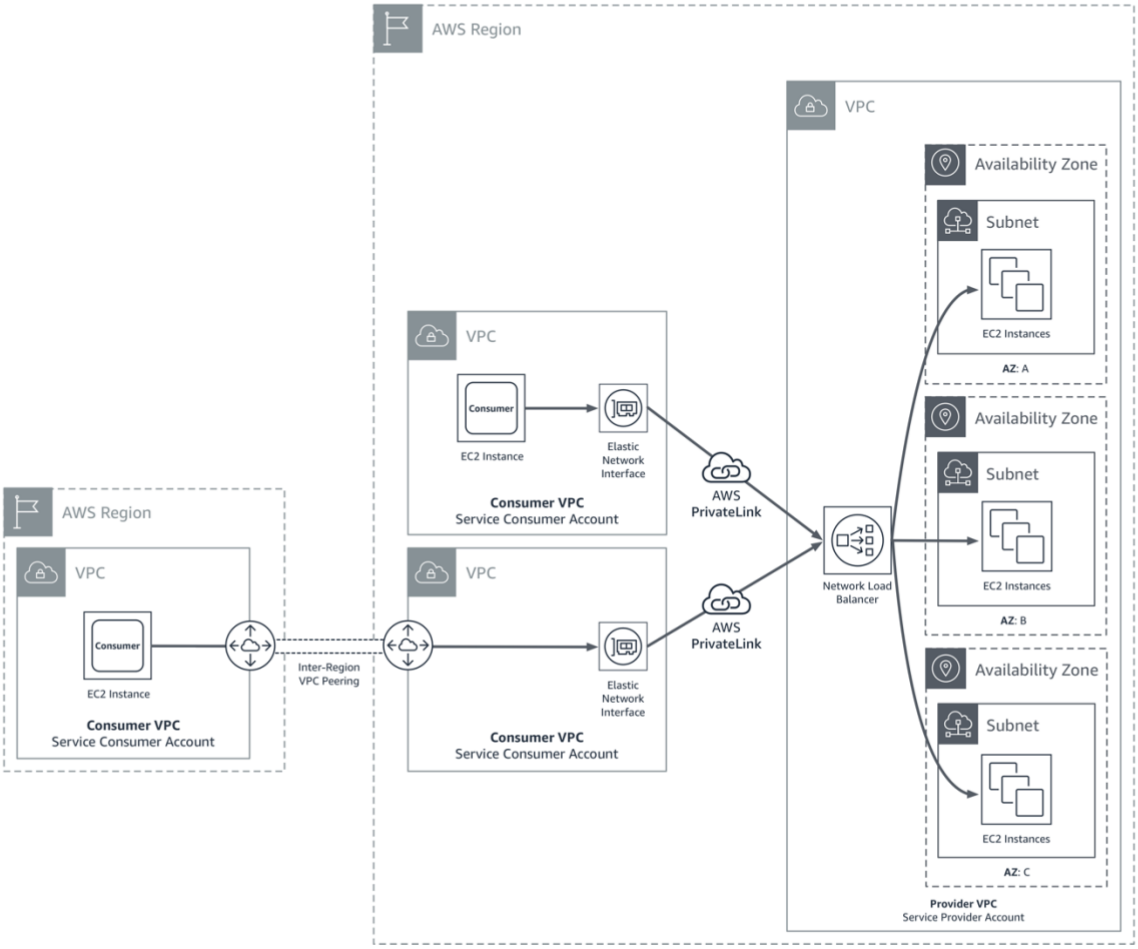

Figure 11: Inter-Region access to endpoint services

Conclusion

The AWS PrivateLink scenarios and best practices outlined in this paper can help you build secure, scalable, and highly available architectures for your services on AWS. Consider your application’s connectivity requirements before choosing an Amazon VPC connectivity architecture for your internal or external customers.

Contributors

The following individuals contributed to this document:

- Ahsan Ali, Global Accounts Solutions Architect, Amazon Web Services
- David Murray, Strategic Solutions Architect, Amazon Web Services
- James Devine, Senior Solutions Architect, Amazon Web Services
- Ikenna Izugbokwe, Senior Solutions Architect, Amazon Web Services
- Matt Lehwess, Principal Solutions Architect, Amazon Web Services
- Miguel Cervantes, Associate Solutions Architect, Amazon Web Services
- Tom Clavel, Senior Product Marketing Manager, Amazon Web Services
- Puneet Konghot, Senior Product Manager, Amazon Web Services

Further Reading

For additional information on building secure, highly-available connectivity architectures, see the following:

- Section on Network-to-Amazon VPC connectivity options in the AWS whitepaper.
- AWS Answers on single-region multi-VPC connectivity.

Document Revisions

Date	Description
January 2019	First publication