Architecture Diagrams

Amazon VPC Lattice Reference Architectures

Amazon VPC Lattice Reference Architectures: Architecture Diagrams

Table of Contents

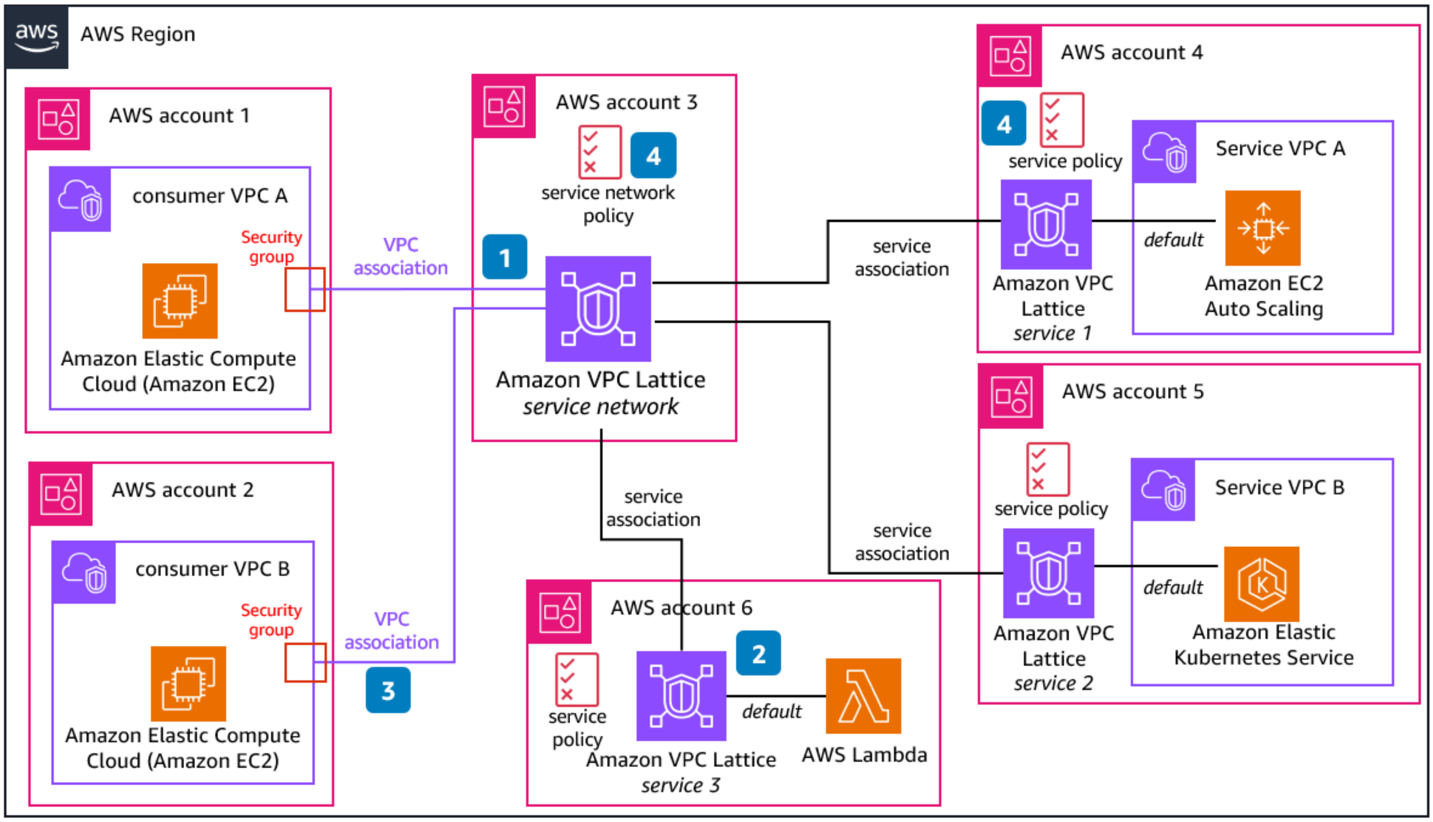
Amazon VPC Lattice Reference Architectures

Publication date: **October 16, 2023 (Diagram history)**

Amazon VPC Lattice is an application layer service that consistently connects, monitors, and secures communications between your services. This series shows different connectivity patterns for multi-AWS account environments.

Components Diagram

VPC Lattice gives you a consistent way to connect, secure, and monitor communication between your services, across AWS compute services (instances, containers, and serverless functions). This diagram will show you the different components in VPC Lattice and how they interact within each other.

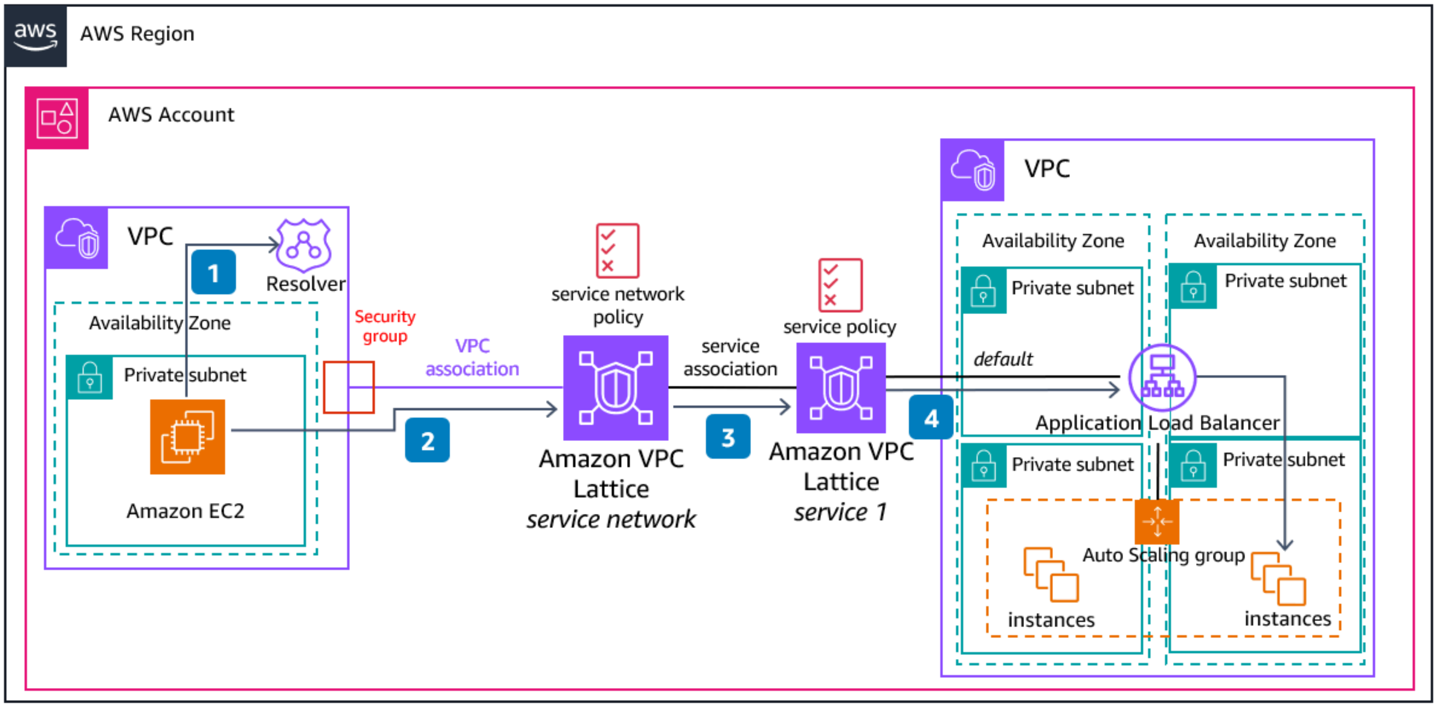


1. A *service network* is logical boundary for a collection of services. Services associated with the network can be authorized for discovery, connectivity, accessibility, and observability. To make requests to services in the network, the client must be in a VPC that is associated with the service network.

2. A *service* represents an independently deployable unit of software that delivers a specific task or function. Each service has a listener that uses rules to target to one or several target groups. Targets can be **Amazon Elastic Compute Cloud** (Amazon EC2) instances, IP addresses, **AWS Lambda** functions, Application Load Balancers, or Kubernetes Pods.
3. When a service is associated to a service network, it enables clients to make requests to this service, but only if the VPC where the client is located is associated as well to the service network, and the policies allow it.
4. When you do a *VPC association* with the service network, it enables all the targets within that VPC to be clients and communicate with other services in the service network. A security group can be attached to this association to control the network access from the VPC, while service network or service policies can be used to apply fine grained access controls.
5. You can use *auth policies* both in the service network or services to configure access control.

Traffic Flow Diagram

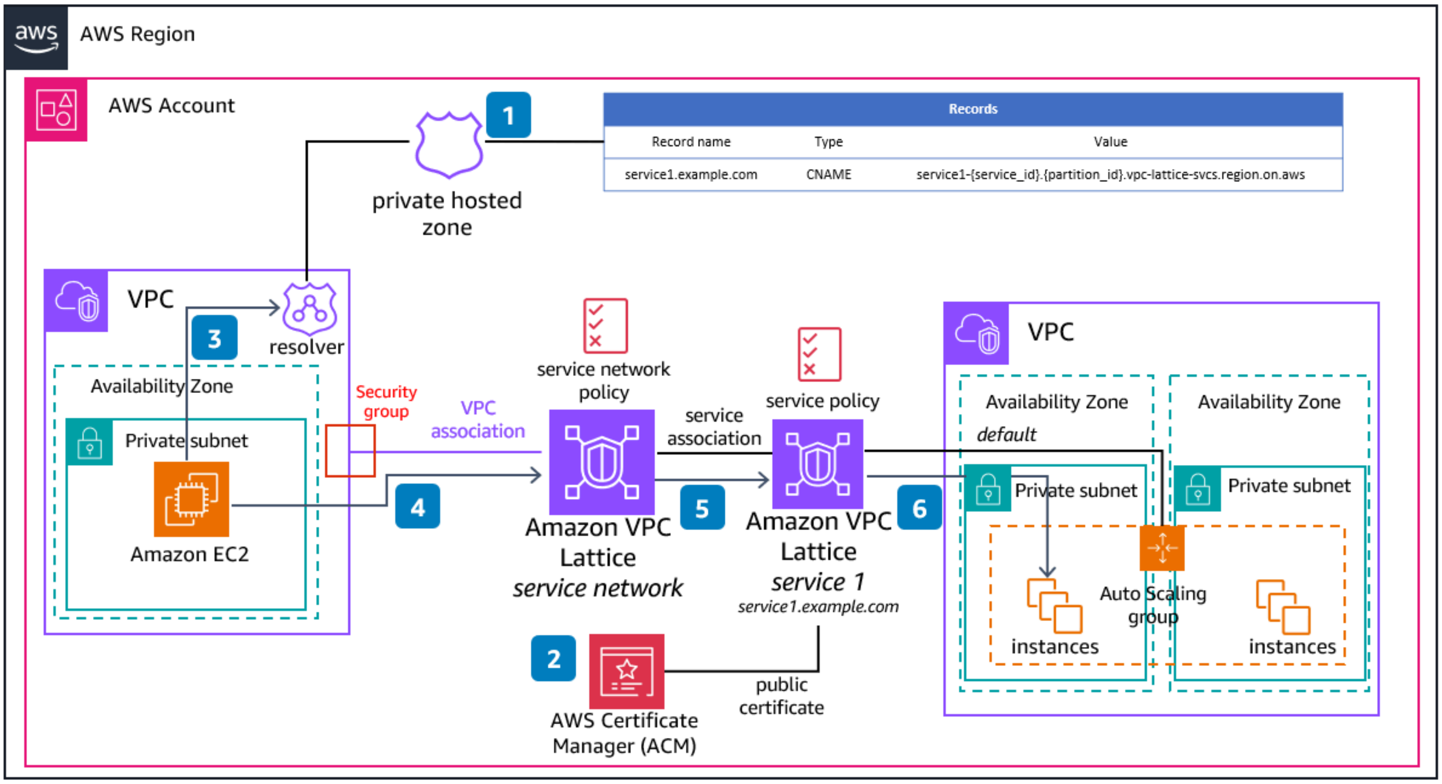
VPC Lattice is designed to help you easily and effectively discover, secure, connect, and monitor all of the services within it. The Amazon Route 53 VPC Resolver is used within the consumer VPC to resolve the VPC Lattice service's domain names, resolving to link-local addresses to send traffic locally to the service network.



1. The consumer application placed in the **Amazon Elastic Compute Cloud** (Amazon EC2) instance queries the **Amazon Route 53** VPC Resolver to get domain name resolution of *service1*.
2. DNS resolution determines that the traffic should be sent to the service network (link-local addresses). The first traffic control measure is a security group attached to the VPC association (you can attach more than one security group).
3. If the requested **VPC Lattice** service is associated with the service network that is associated with the consumer VPC and the service network policy allows communication to it, traffic will be forwarded to the specific target.
4. In this example, *service1* only has one target group (routed by default) which is an Application Load Balancer (ALB). This ALB will forward the traffic to the corresponding **Amazon EC2** instance.

Custom Domain Name for a VPC Lattice Service Diagram

When creating VPC Lattice services, you can configure a custom domain name to provide a more intuitive URL for your users. When a client makes a request using your custom domain name, the DNS server resolves it to the VPC Lattice-generated domain name (service-name-service_id.partition_id.vpc-lattice-svcs.region.on.aws). However, this happens only if you map your custom domain name to the VPC Lattice-generated domain name (CNAME record).

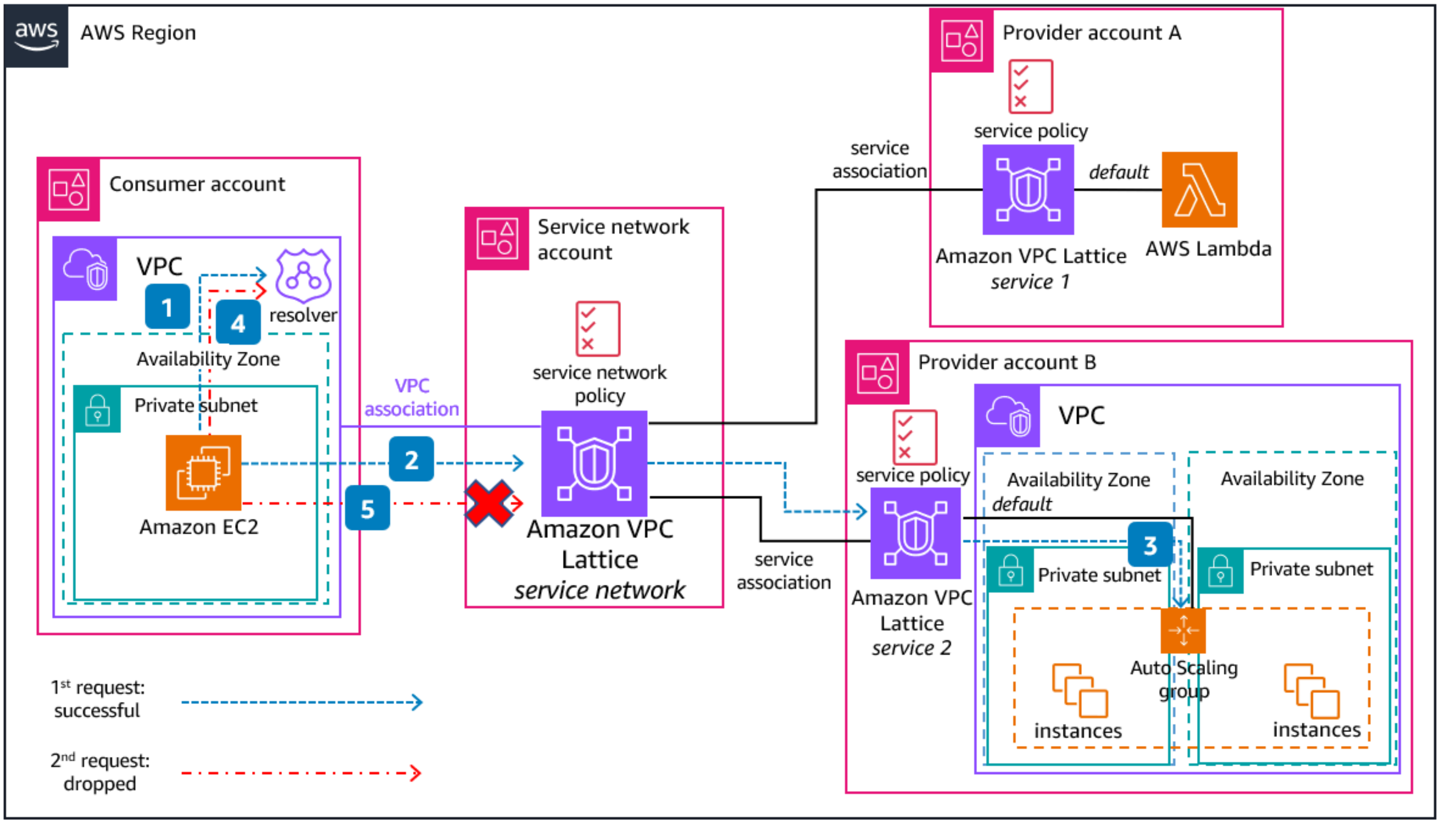


Records		
Record name	Type	Value
service1.example.com	CNAME	service1-{service_id}.{partition_id}.vpc-lattice-svcs.region.on.aws

1. To allow the mapping between your custom domain name and the **VPC Lattice**-generated domain name, create a private hosted zone associated to your consumer VPC. This hosted zone will contain the CNAME record doing this mapping.
2. Optionally, to serve HTTPS requests, you must have your own SSL/TLS certificate ready in **AWS Certificate Manager** (ACM) when setting up a custom domain name.
3. The consumer application placed in the **Amazon EC2** instance queries the VPC resolver to get domain name resolution of *service1.example.com*. The first DNS query to the VPC resolver will resolve to the **VPC Lattice**-generated domain name, as configured in the private hosted zone associated to the VPC.
4. The second DNS query – to the **VPC Lattice**-generated domain name – will determine that the traffic should be sent to the service network.
5. Because the requested **VPC Lattice** service is associated with the service network associated to the consumer VPC and the service network's policy allows communication to it, traffic will be forwarded to the specific target.
6. In this example, *service1* only has one target group (routed by default) which is an auto scaling group. Traffic will be forwarded to one of the **Amazon EC2** instances in the group.

Multi-Account Centralized Single Service Network Diagram

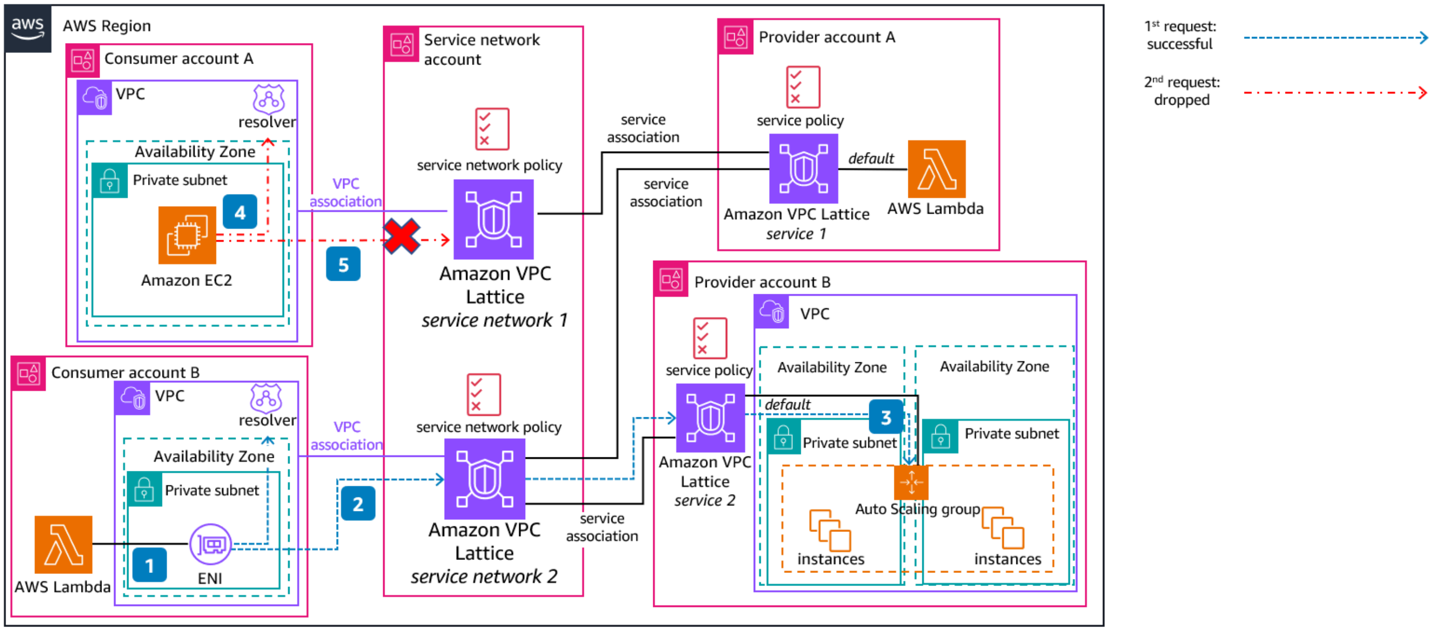
A central account can have ownership of the service network, which is shared (using AWS Resource Access Manager) to other AWS accounts inside the same or different AWS Organizations. Additionally, the provider accounts can also shared their services using AWS Resource Access Manager.



1. The consumer application placed in the **Amazon EC2** instance queries the VPC resolver to get domain name resolution of *service2*.
2. DNS resolution determines that the traffic should be sent to the service network.
3. In this example, both policies at the service network and service level allow traffic to the target (Auto Scaling group) from the consumer, so the application in the **Amazon EC2** instance can consume *service2*.
4. The consumer application gets the domain name resolution of *service1*.
5. The VPC resolver will resolve with the service network as next hop. However, in the example, either the policies at the service network or service level don't allow the consumer to consume service1 (**Lambda** function), so the traffic is dropped.

Multi-Account Centralized Multiple Service Networks Diagram

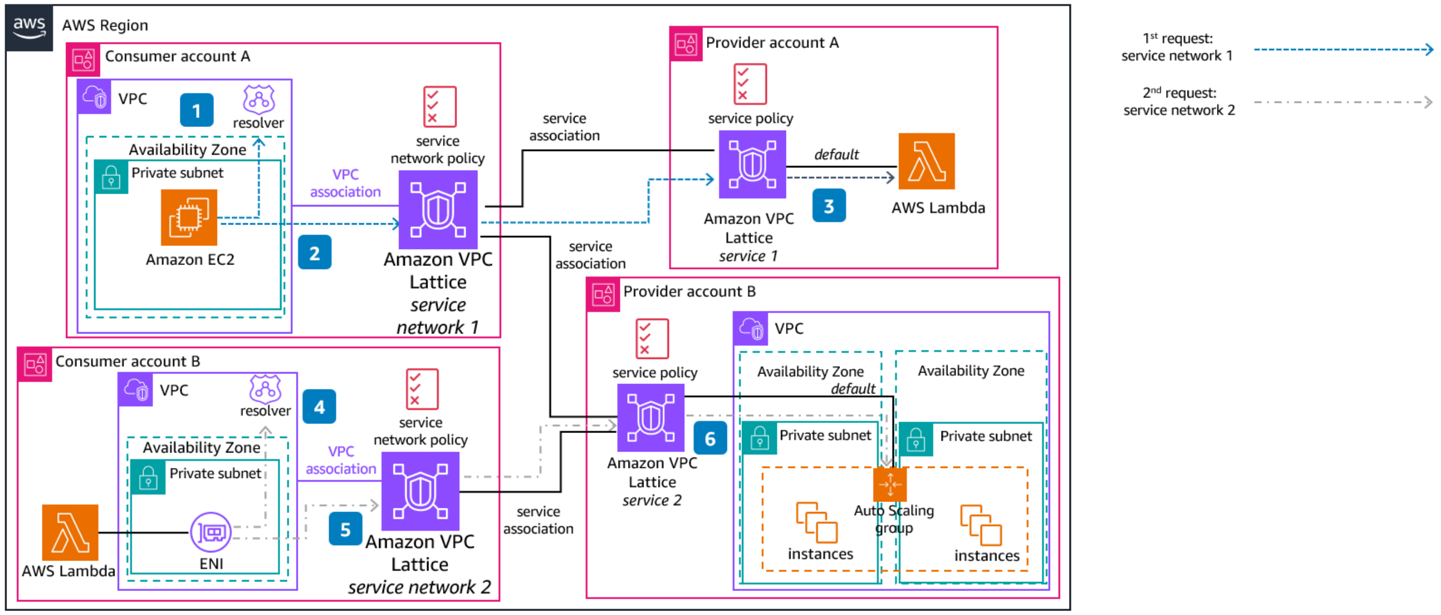
A central account can have ownership of several service networks, which are shared (using AWS Resource Access Manager) to other AWS accounts, inside the same or different AWS Organizations (same as services). The use of several service networks allow for a segmentation of services, which can also be achieved or complemented with the use of policies.



1. An **AWS Lambda** function placed in a VPC can access a service network after the VPC is associated to it. In this case, the consumer service located in the **Lambda** function queries the VPC resolver to resolve the domain name of *service1*.
2. DNS resolution determines that the traffic should be sent to service network 2.
3. Both policies at the service network and service level allow traffic to the target (Auto Scaling group) from the consumer, so the consumer application can consume the service.
4. The consumer application placed in an **Amazon EC2** instance queries the VPC resolver to resolve the domain name of the *service2*. This resolves in the **VPC Lattice** link-local addresses sending the traffic to the service network.
5. Because *service2* is not associated to service network 1, the traffic is dropped.

Multi-Account Distributed Service Networks Diagram

Each provider AWS account owns its own services, and shares them with other AWS accounts, inside the same or different AWS Organizations. Because a VPC can only be associated with one service network, consumers can own its own service network and choose which services they want to consume.



1. The consumer application – placed in an **Amazon EC2** instance – queries the VPC resolver for DNS resolution of *service1*.
2. DNS resolution determines that the traffic should be sent to service network 1.
3. Both policies at the service network and service level allow traffic to the target (a **Lambda** function) from the consumer, so the consumer application can consume service1.
4. A **Lambda** function placed in a VPC can access a service network after the VPC is associated to it. In this case, the consumer service located in the **Lambda** function queries the VPC resolver for DNS resolution of *service2*.
5. DNS resolution determines that the traffic should be sent to service network 2.
6. Both policies at the service network and service level allow traffic to the target (Auto Scaling group) from the consumer, so the consumer application can consume *service2*.

Download editable diagram

To customize this reference architecture diagram based on your business needs, download the ZIP file which contains an editable PowerPoint.

Create a free AWS account

Sign up now

Sign up for an AWS account. New accounts include 12 months of AWS Free Tier access, including the use of Amazon EC2, Amazon S3, and Amazon DynamoDB.

Further reading

For additional information, refer to

- AWS Architecture Icons
- AWS Architecture Center
- AWS Well-Architected

Contributors

Contributors to this reference architecture diagram include:

- Pablo Sánchez Carmona, Specialist Solutions Architect, Amazon Web Services
- Adam Palmer, Senior Specialist Solutions Architect, Amazon Web Services

Diagram history

To be notified about updates to this reference architecture diagram, subscribe to the RSS feed.

Change	Description	Date
Diagram updated	Updated diagram and added new use cases.	October 16, 2023
Initial publication	Reference architecture diagram first published.	April 17, 2023

Note

To subscribe to RSS updates, you must have an RSS plugin enabled for the browser you are using.