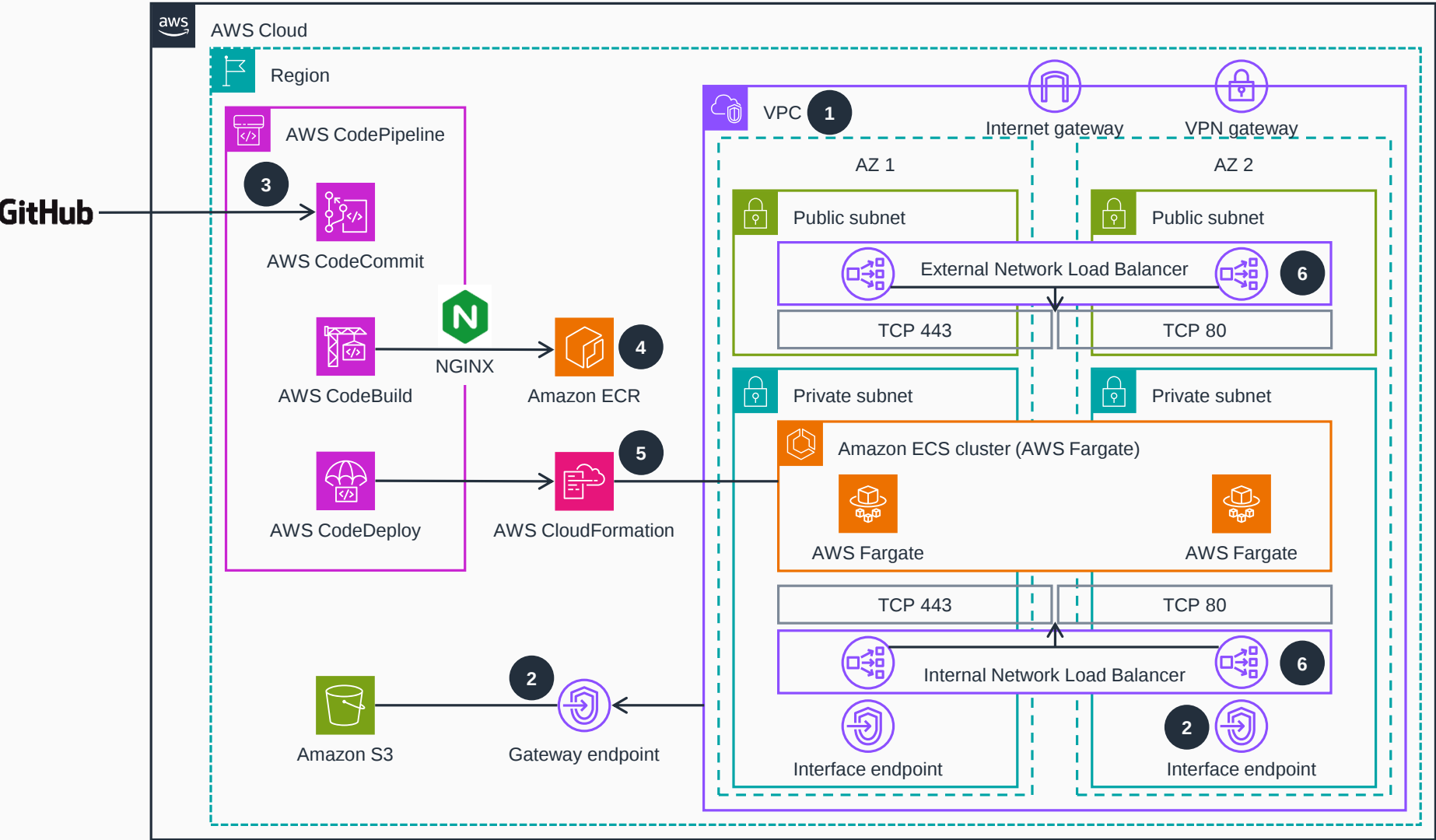


Guidance for External Connectivity to Amazon VPC Lattice

This architecture diagram shows how to configure a proxy in a virtual private cloud (VPC) to connect external services to Amazon VPC Lattice. Slides 2–4 detail three ways to use Amazon VPC Lattice for public, hybrid, or cross-Region access.

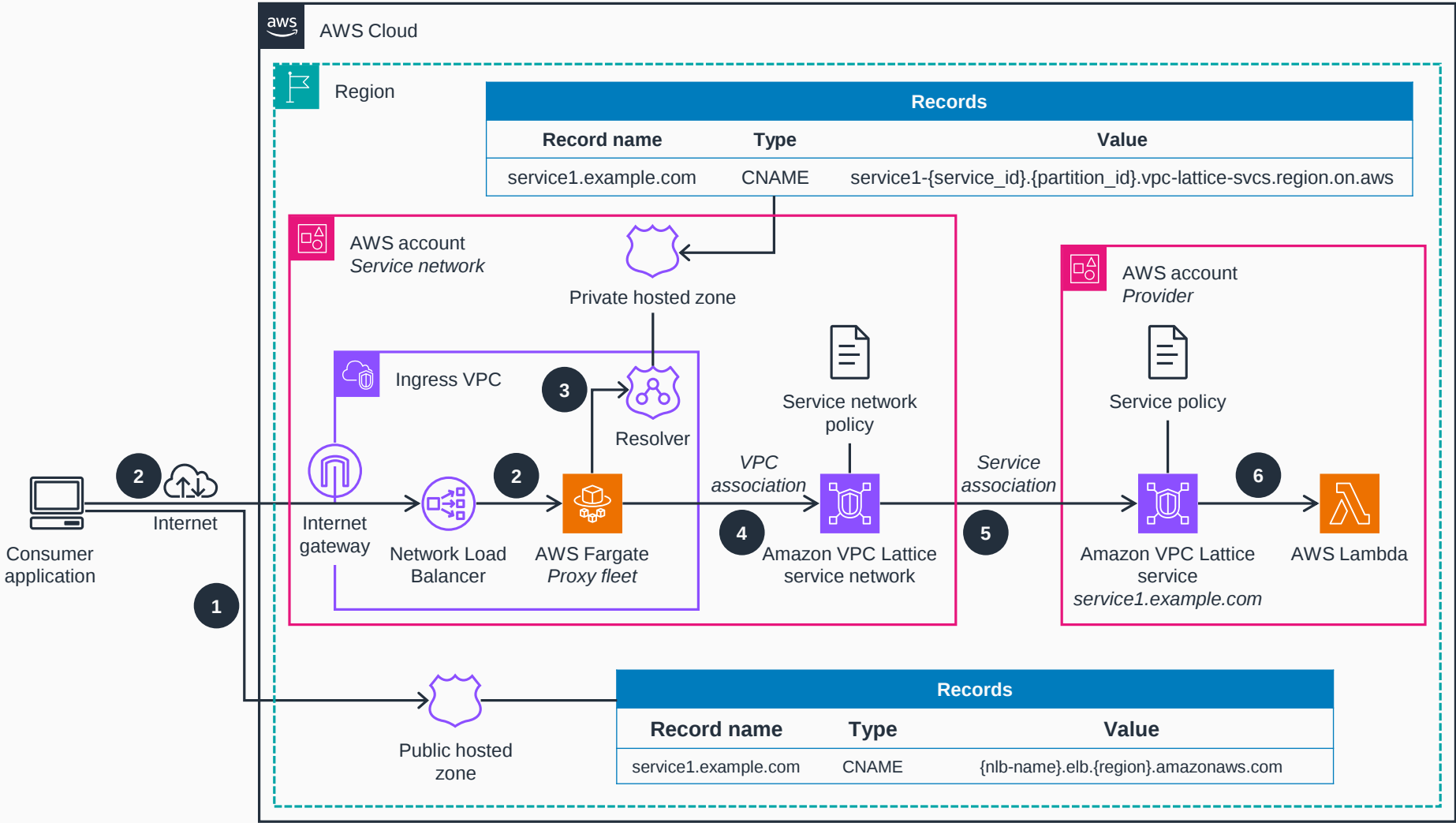


- 1 This Guidance will deploy a virtual private cloud (VPC) in multiple Availability Zones (AZs), with both public and private subnets containing internal and external Network Load Balancers.
- 2 **AWS PrivateLink** VPC endpoints (interface and gateway) are created to reach AWS services privately.
- 3 **AWS CodePipeline** orchestrates the build and delivery of this Guidance. The code is pulled from GitHub to an **AWS CodeCommit** repository.
- 4 **AWS CodeBuild** builds containers that run an open-source version of NGINX. The container image is stored in **Amazon Elastic Container Registry (Amazon ECR)**.
- 5 The deployment stage in the pipeline uses **AWS CloudFormation** to build an **Amazon Elastic Container Service (Amazon ECS)** cluster, task definition, and service, using **AWS Fargate** as the capacity provider.
- 6 Four target groups are used to pass traffic to the backend compute solution. Each Network Load Balancer configures two TCP listeners for ports 80 (HTTP) and 443 (HTTPS). The **Amazon ECS** tasks therefore service both internal and external traffic.



Guidance for External Connectivity to Amazon VPC Lattice

Public access: This architecture diagram shows how placing a proxy solution in an associated VPC enables external consumption of VPC Lattice services by adjusting the DNS resolution.

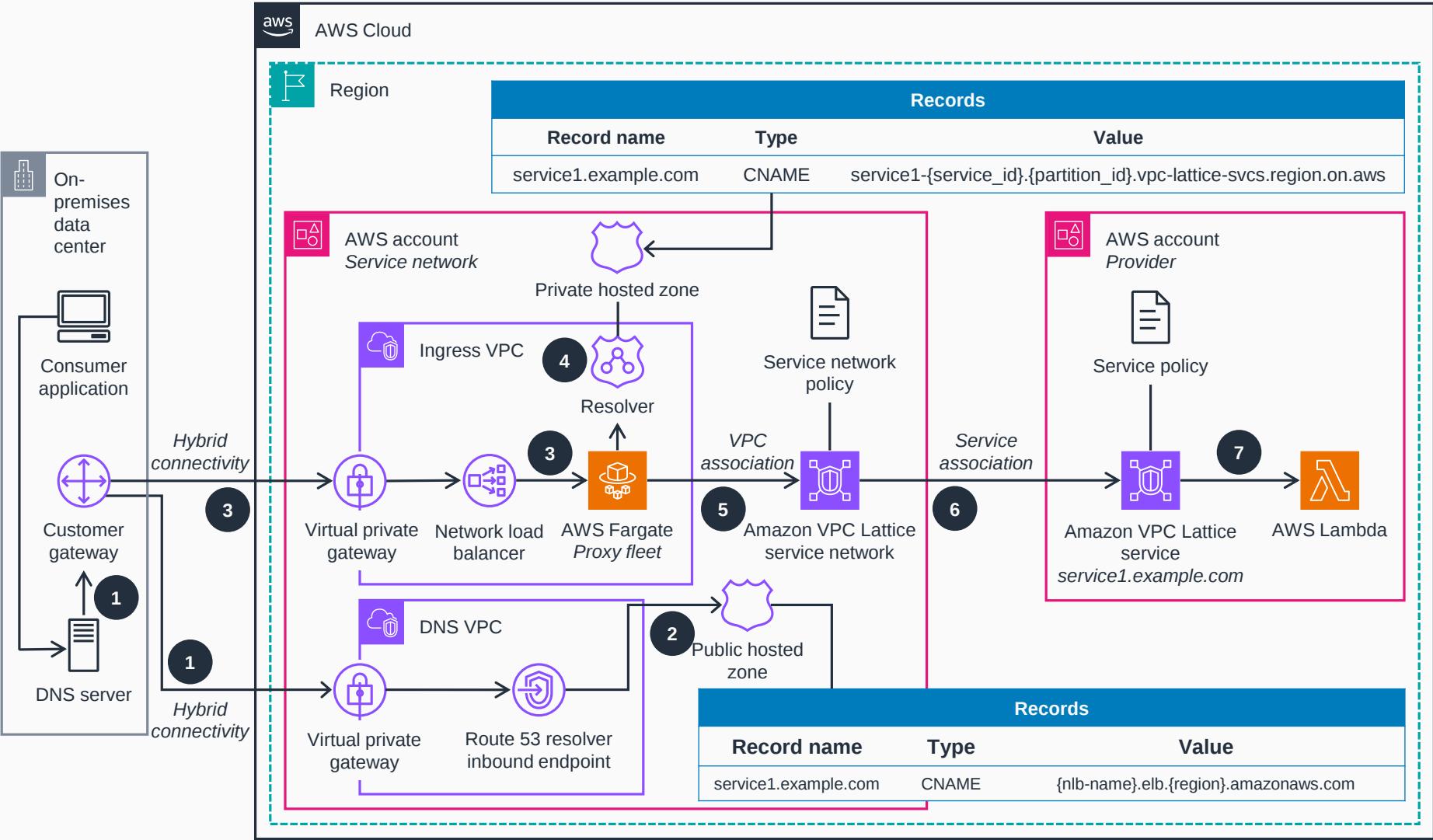


- 1 The consumer application located outside AWS tries to resolve service1's domain name publicly. An **Amazon Route 53** public hosted zone resolves to the Network Load Balancer domain name.
- 2 Traffic is sent to the Network Load Balancer public IPs (obtained after the DNS resolution), and the request is forwarded to the **Fargate** proxy fleet.
- 3 Inside the ingress VPC, the proxy fleet resolves service1's domain name by using the VPC DNS resolver. A **Route 53** private hosted zone is used to map the custom domain name with the domain name generated by **Amazon VPC Lattice**.
- 4 The DNS resolution provides **VPC Lattice** with link-local addresses. Traffic is sent using the **VPC Lattice** VPC association.
- 5 A service policy allows traffic between the AWS service network account and the AWS provider account if there is an association between the **VPC Lattice** service and the **VPC Lattice** service network. This can then be associated with the ingress VPC.
- 6 This request is redirected to an **AWS Lambda** function.



Guidance for External Connectivity to Amazon VPC Lattice

Hybrid access: This architecture diagram shows how placing a proxy solution in an associated VPC enables on-premises applications to have external consumption of VPC Lattice services by adjusting the hybrid DNS resolution.

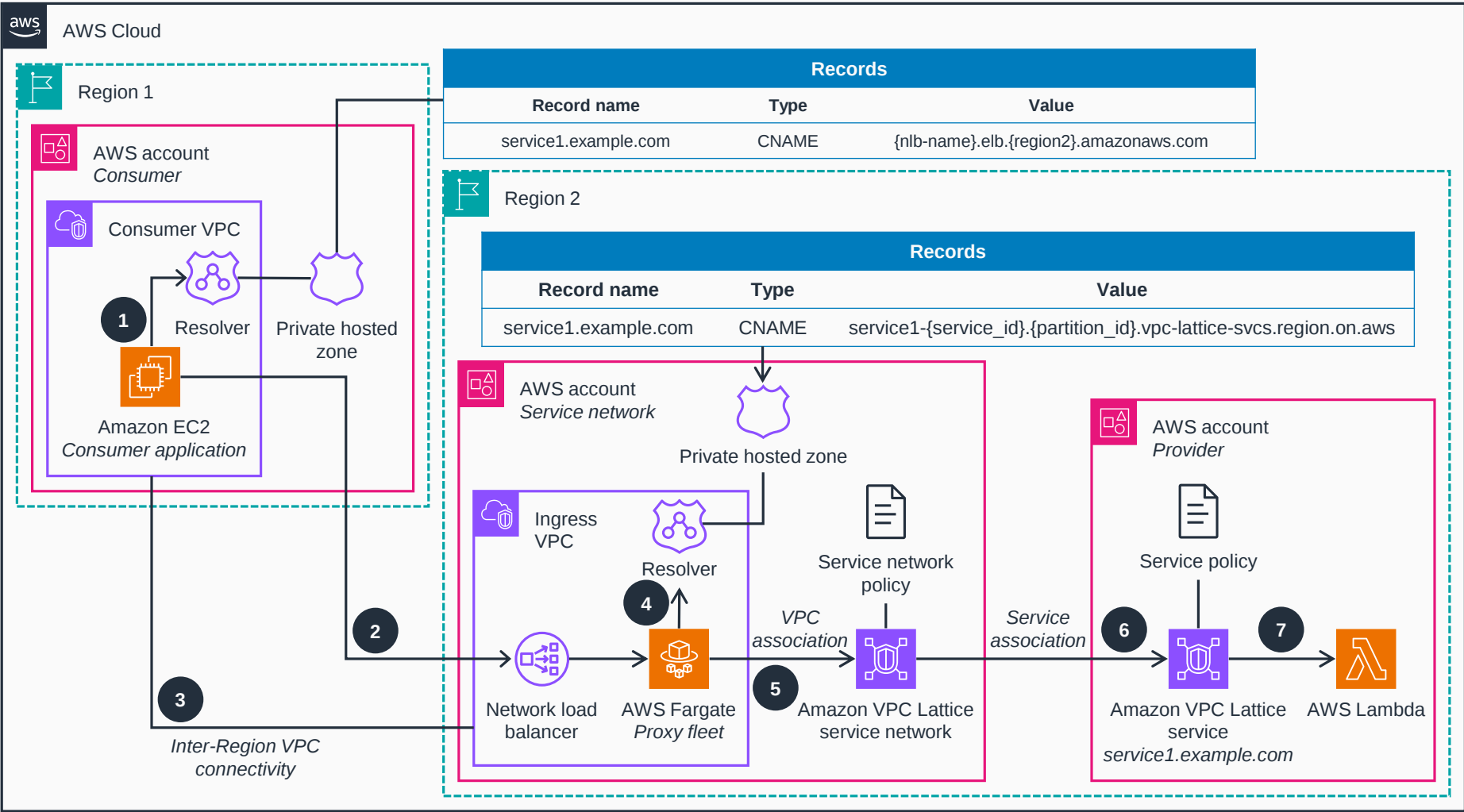


- 1 The on-premises consumer application tries to resolve service1's domain name locally. The on-premises DNS server forwards the DNS request to a **Route 53** resolver inbound endpoint, located on AWS. You can make use of any hybrid connectivity solution with AWS.
- 2 The **Route 53** resolver inbound endpoint queries a **Route 53** private hosted zone to resolve the Network Load Balancer domain name.
- 3 A hybrid connectivity solution can be used for the connectivity between on-premises applications and AWS. Traffic is sent to the Network Load Balancer private IPs (obtained after the DNS resolution), and the request is forwarded to the **Fargate** proxy fleet.
- 4 Inside the ingress VPC, the proxy fleet resolves service1's domain name by using the VPC DNS resolver. A **Route 53** private hosted zone can be used to map the custom domain name with the domain name generated by **VPC Lattice**.
- 5 The DNS resolution provides **VPC Lattice** with link-local addresses. Traffic will be sent using the **VPC Lattice** VPC association.
- 6 A service auth policy allows traffic between the AWS service network account and the AWS provider account if there is an association between the **VPC Lattice** service and the **VPC Lattice** service network. This can then be associated with the ingress VPC.
- 7 This request is redirected to a **Lambda** function.



Guidance for External Connectivity to Amazon VPC Lattice

Cross-Region access: This architecture diagram shows how placing a proxy solution in an associated VPC enables cross-Region consumption of VPC Lattice services by adjusting the hybrid DNS resolution.



- 1 Consumer applications in the consumer VPC from AWS Region 1 use their local DNS VPC resolver for service1's domain name resolution by using a **Route 53** private hosted zone.
- 2 Configure the DNS resolution to point to the proxy solution in the ingress VPC in Region 2.
- 3 Any inter-Region connectivity option* enables communication between the consumer VPC in Region 1 and the ingress VPC in Region 2.
- 4 Inside the ingress VPC, the **Fargate** proxy fleet will resolve service1's domain name by using the VPC DNS resolver. A **Route 53** private hosted zone can be used to map the custom domain name to the domain name generated by **VPC Lattice**.
- 5 DNS resolution will provide **VPC Lattice** with link-local addresses. Traffic will be sent using the **VPC Lattice** VPC association.
- 6 A service auth policy allows traffic between the AWS service network account and the AWS provider account if there is an association between the **VPC Lattice** service and the **VPC Lattice** service network. This can then be associated with the ingress VPC.
- 7 This request is redirected to a **Lambda** function.



*You can check the Amazon Virtual Private Cloud Connectivity Options whitepaper for more information about inter-Region connectivity options.