**Configuring AWS PrivateLink: Guide**

**Notes on Terminology**

The terms ‘producer’ and ‘server; as well as ‘client’ and ‘consumer’ may be used interchangeably in this document.

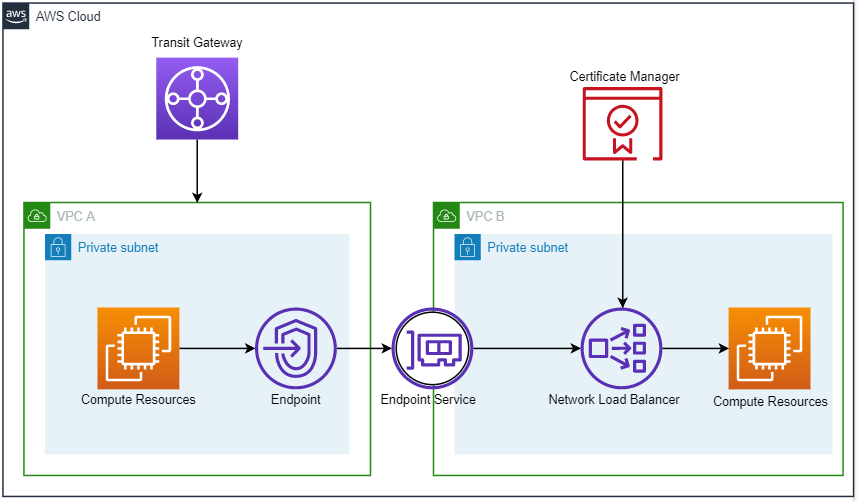
**Background­­­**

AWS [PrivateLink](https://aws.amazon.com/privatelink/) simplifies the security of data shared with cloud-based applications by eliminating the exposure of data to the public Internet. AWS PrivateLink provides private connectivity between VPCs, AWS services, and on-premises applications, securely on the Amazon network. AWS PrivateLink makes it easy to connect services across different accounts and VPCs to significantly simplify the network architecture.

**Problem**

Link resources in separate VPCs without exposure to the internet or use of VPC Peering.

**AWS PrivateLink Architecture**



The [VPC Endpoint Service](https://docs.aws.amazon.com/vpc/latest/userguide/endpoint-service.html) is AWS PrivateLink. It can be configured with multiple services as either [interface endpoints](https://docs.aws.amazon.com/vpc/latest/userguide/vpce-interface.html) or [gateway endpoints](https://docs.aws.amazon.com/vpc/latest/userguide/vpce-gateway.html). This solution is an interface endpoint because the association is between the VPC Endpoint Service and a Network Load Balancer. The Endpoint Service allows the Network Load Balancer to receive traffic from Endpoints that are associated with it.

In the diagram above, an Endpoint has been configured to forward traffic to the Endpoint Service. Endpoints can be in separate VPCs and accounts from the Endpoint Service. The same Endpoint configuration can be deployed in multiple subnets and availability zones in a VPC to achieve high availability.

For enhanced security, the Network Load Balancer can be configured to accept only TLS traffic. An X.509 certificate must be created in [AWS Certificate Manager](https://aws.amazon.com/certificate-manager/) by a [Private Certificate Authority](https://aws.amazon.com/certificate-manager/private-certificate-authority/?nc=sn&loc=6) and then associated to the Network Load Balancer [Listener](https://docs.aws.amazon.com/elasticloadbalancing/latest/network/load-balancer-listeners.html). It is important to bear in mind that NLBs **only** support TCP, UDP, and TLS traffic.

[Proxy Protocol Headers](https://docs.aws.amazon.com/elasticloadbalancing/latest/network/load-balancer-target-groups.html#proxy-protocol) can be enabled on the Network Load Balancer to preserve the source IP address on traffic.

Compute resources can be configured to send traffic to and receive traffic from the Network Load Balancer. The type compute resource depends on the use case. To connect to the compute resources in the private subnets, Systems Manager or SSH can be used through the Transit Gateway.

**CloudFormation Guide**

The CloudFormation template provided for this architecture provisions the infrastructure for AWS PrivateLink. The template will not provision or configure the client or server that communicate over PrivateLink.

**Before provisioning the infrastructure**, ensure the following prerequisites are met:

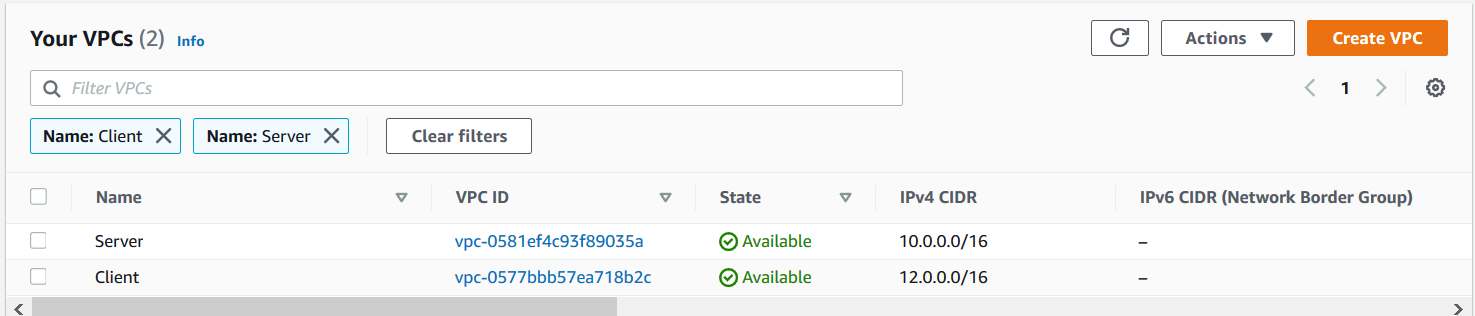
1. 2 VPCs are created with subnets that have >5 IPs available
2. A security group is created in the client/source VPC that will be associated with the client/source compute resources
3. Import or Create an X.509 Certificate with AWS Certificate Manager

The template will provision the following resources:

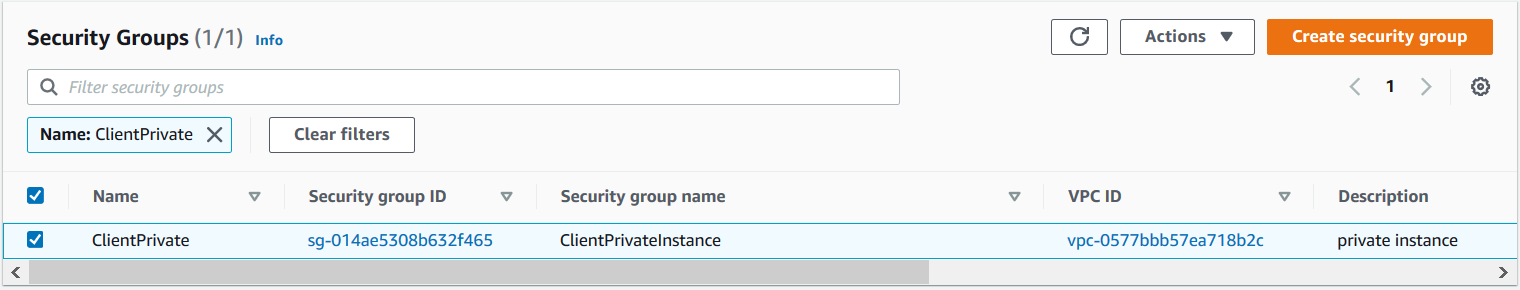
* Network Load Balancer (Server VPC)
* Network Load Balancer Listener (Server VPC)
* Network Load Balancer Target Group (Server VPC)
* VPC Endpoint Service (Server VPC)
* Security Group for VPC Endpoint (Client VPC)
* VPC Endpoint (Client VPC)

**CloudFormation Walk-through**

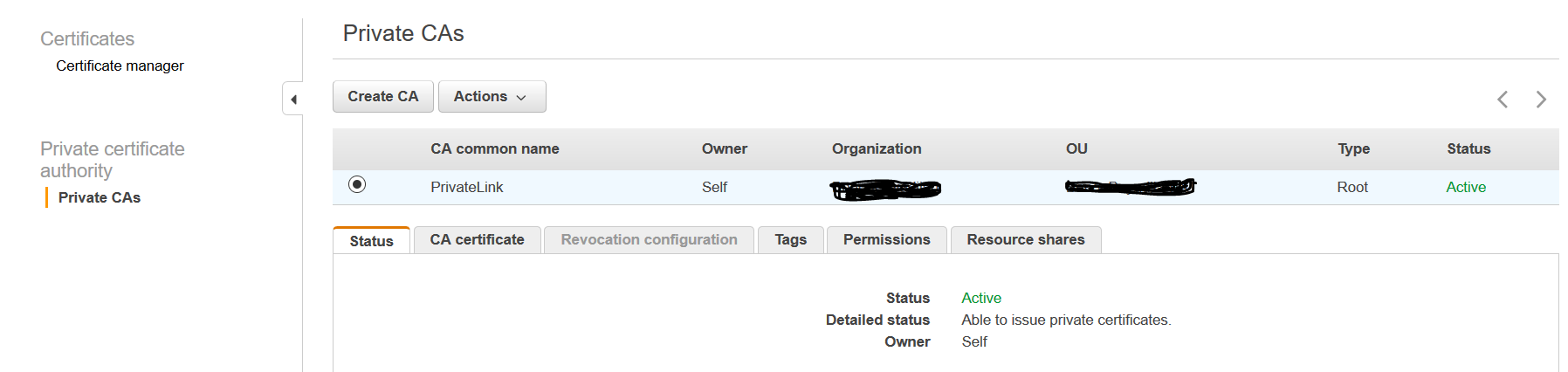
1. Both VPCs are created. They are named Server and Client in this example. The Server VPC will contain the Network Load Balancer and the Client VPC will contain the VPC Endpoint.



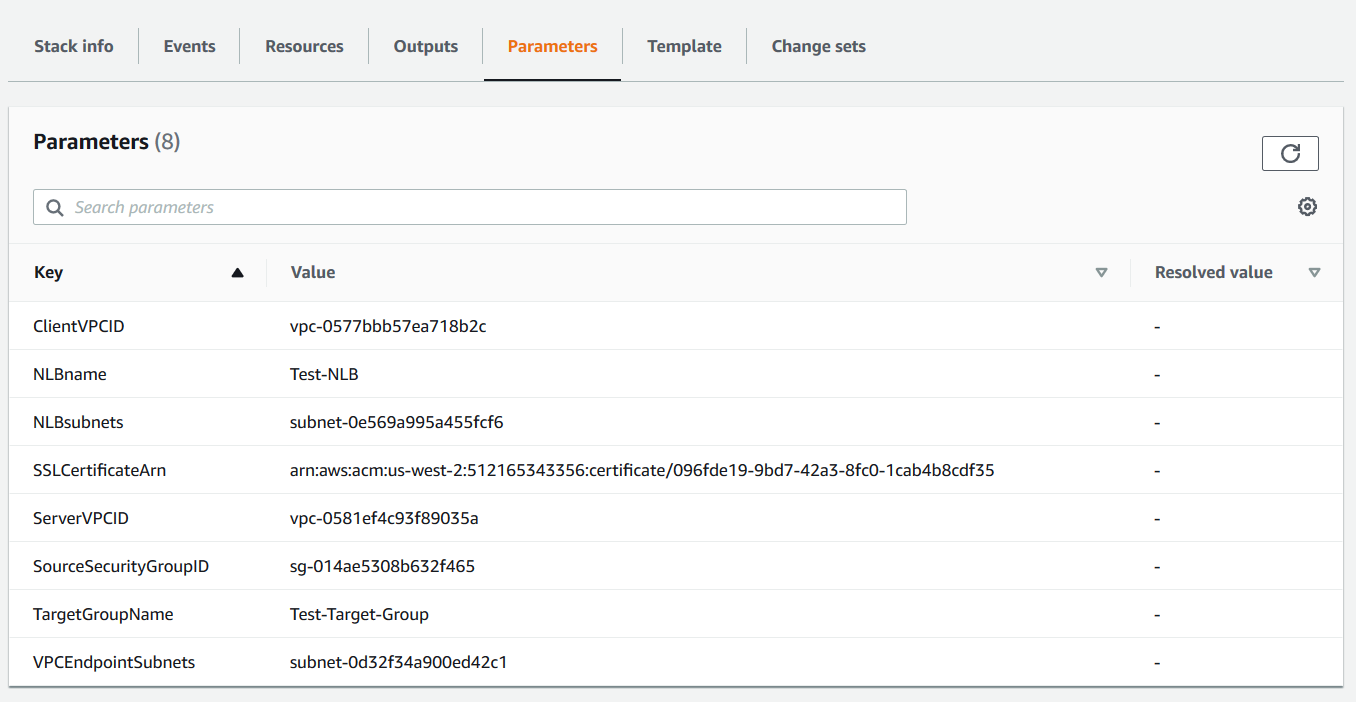
1. A Security Group must be created in the Client VPC. This Security Group will be associated with the Compute Resources that make requests to the Network Load Balancer. This is an essential step because the CloudFormation template takes this Security Group as a parameter. This Security Group will be allowed to send traffic to the VPC Endpoint.



1. If TLS traffic will be used, a Private Certificate Authority must create an X.509 Certificate for the Network Load Balancer. Choose to create a Private Certificate Authority, use an existing CA, or import a certificate from an external source. Ensure that the private key, certificate chain, and certificate are saved for use described by the “Configuration Guide” section in this document.

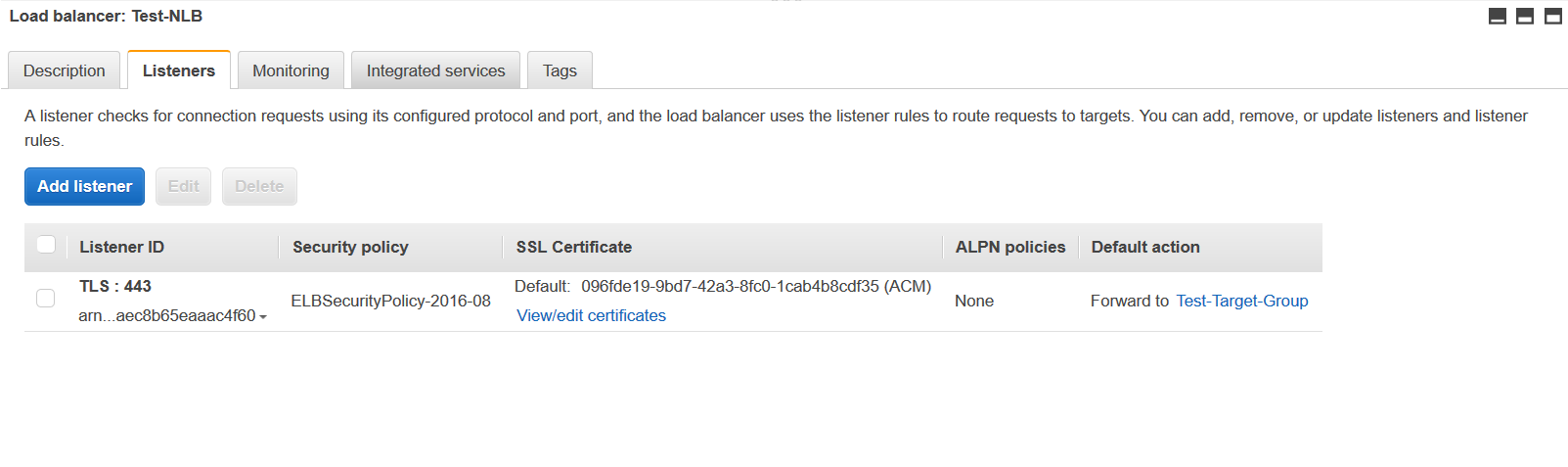


1. Create the AWS PrivateLink infrastructure with CloudFormation. Note the parameters below.



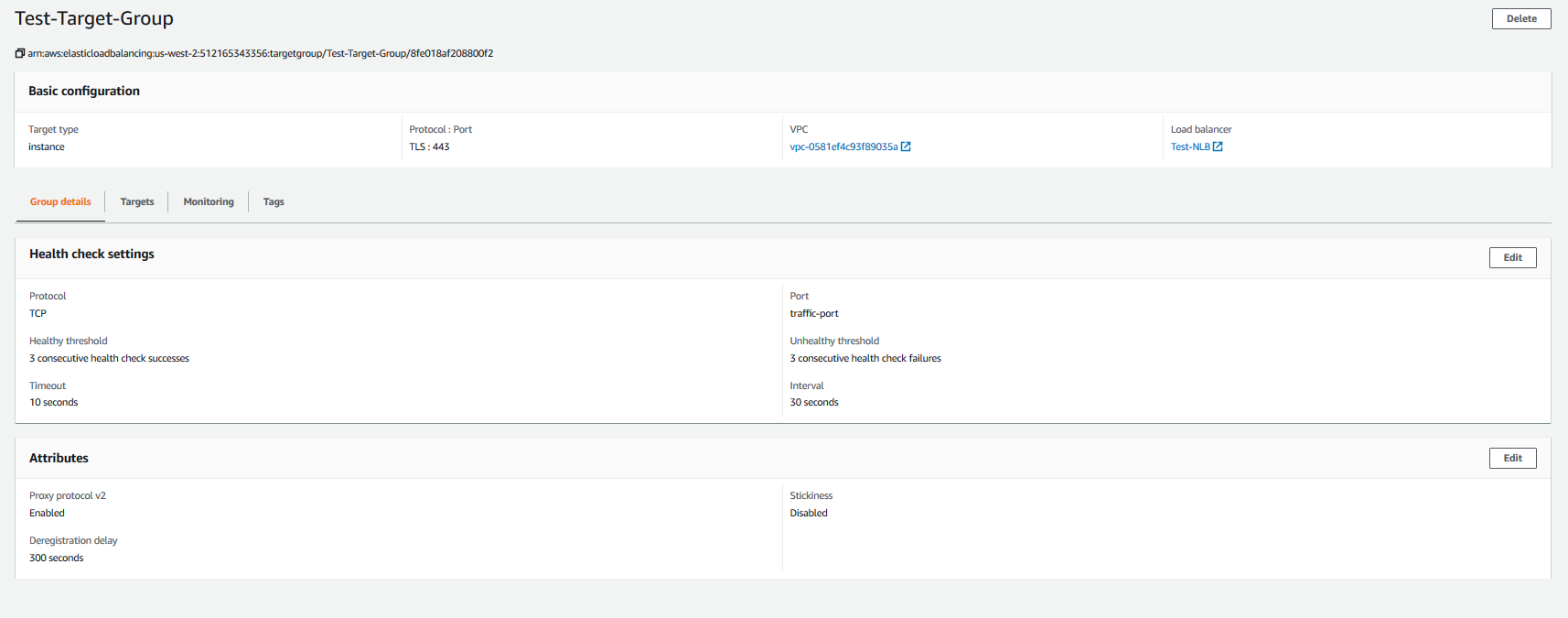
The two VPCs, Source Security Group, and SSL Certificate ARN are all referenced from the prerequisite steps. Additionally, The Subnets where the VPC Endpoints and NLB are provisioned are required. The VPC Endpoints should be provisioned in the subnets where the client (compute resources) will be provisioned. The NLB should be provisioned in the “Server VPC.”

1. Once the architecture is provisioned, Targets can be configured. Go to the NLB that has been provisioned.



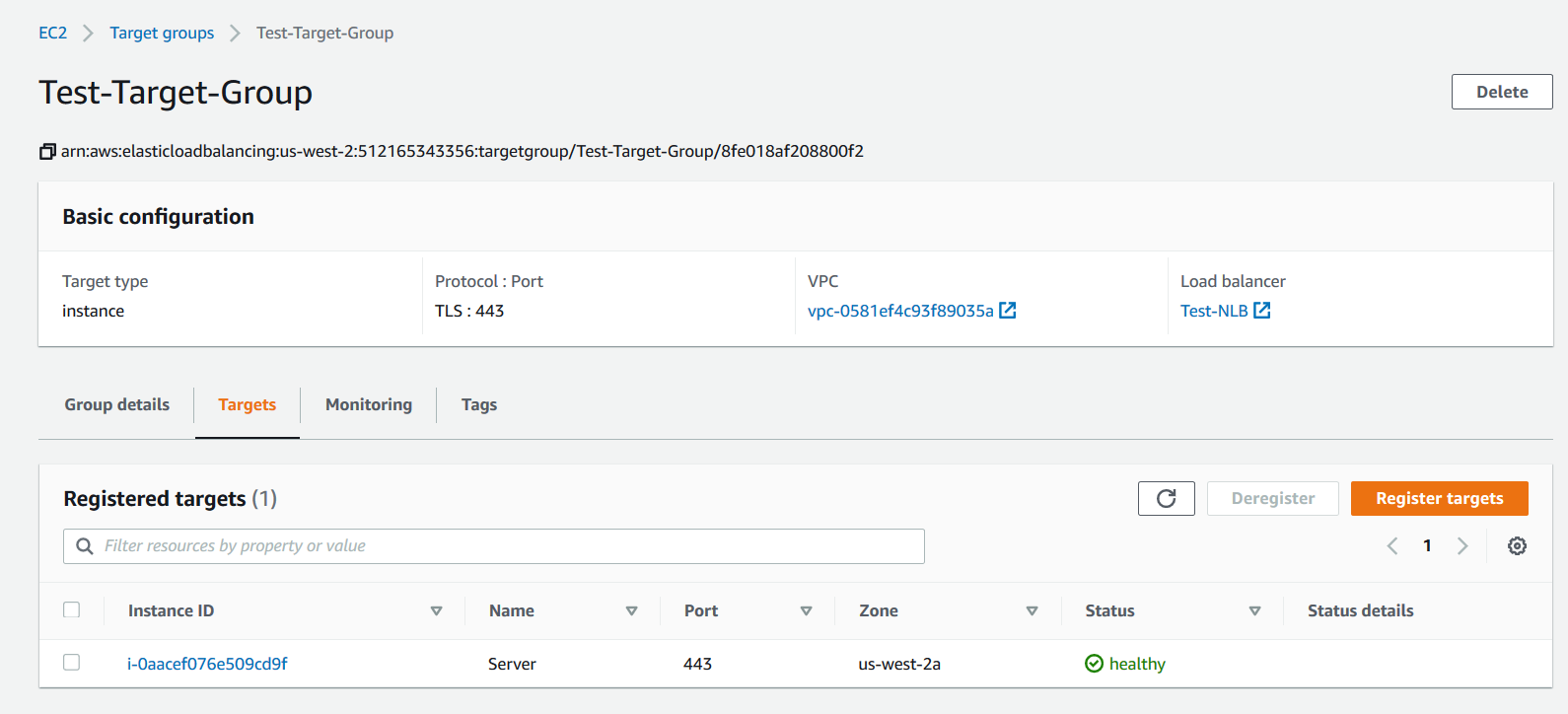
Notice that the Listener is for TLS traffic over port 443 and uses the SSL Certificate created in the prerequisites. The default action for this listener is to forward traffic to the Target Group that was created by CloudFormation.

1. Click into the Target Group.



Notice the Target Group uses Proxy Protocol v2. This is to maintain the source IP of requests to the load balancer. This can be disabled by changing the CloudFormation template “proxy\_protocol\_v2.enabled” value to “false.”

1. To add compute resources to the Target Group, click on the Targets tab.



Register targets to the Target Group to receive traffic from the load balancer.

**Configuration Guide**

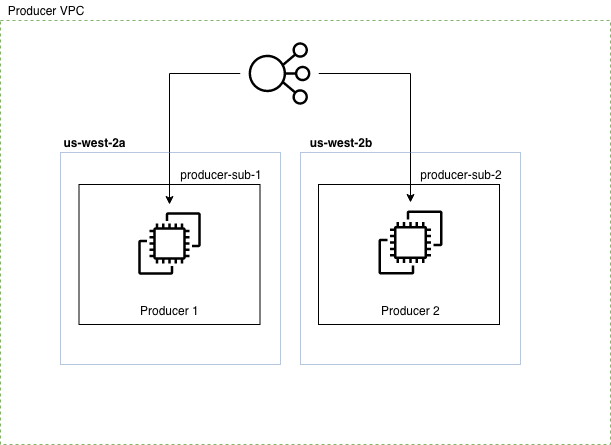
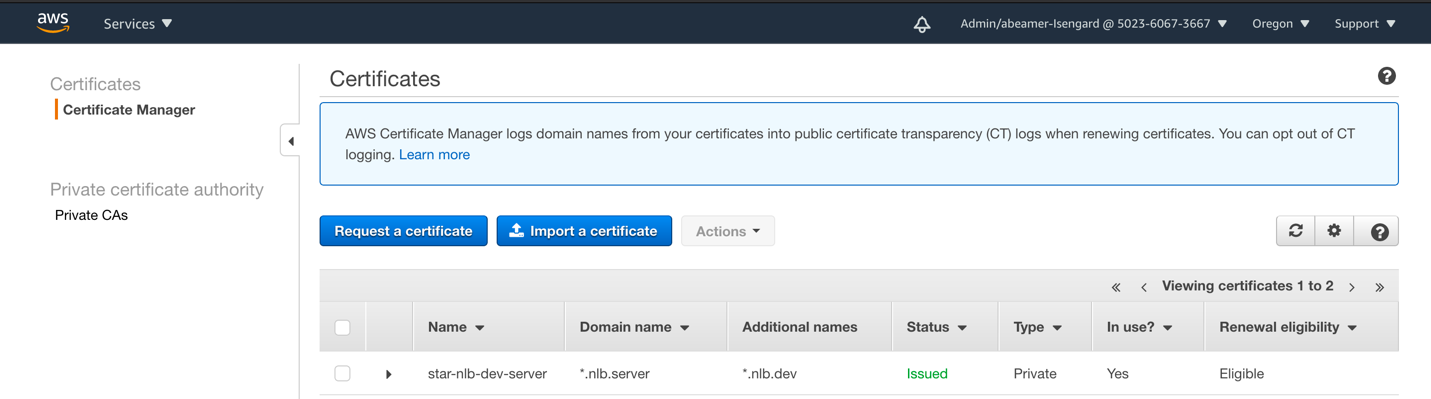
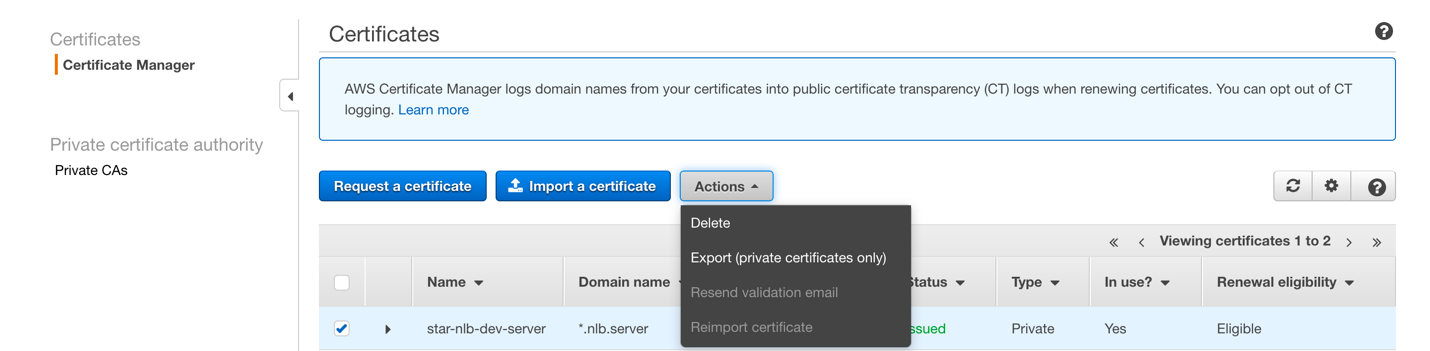
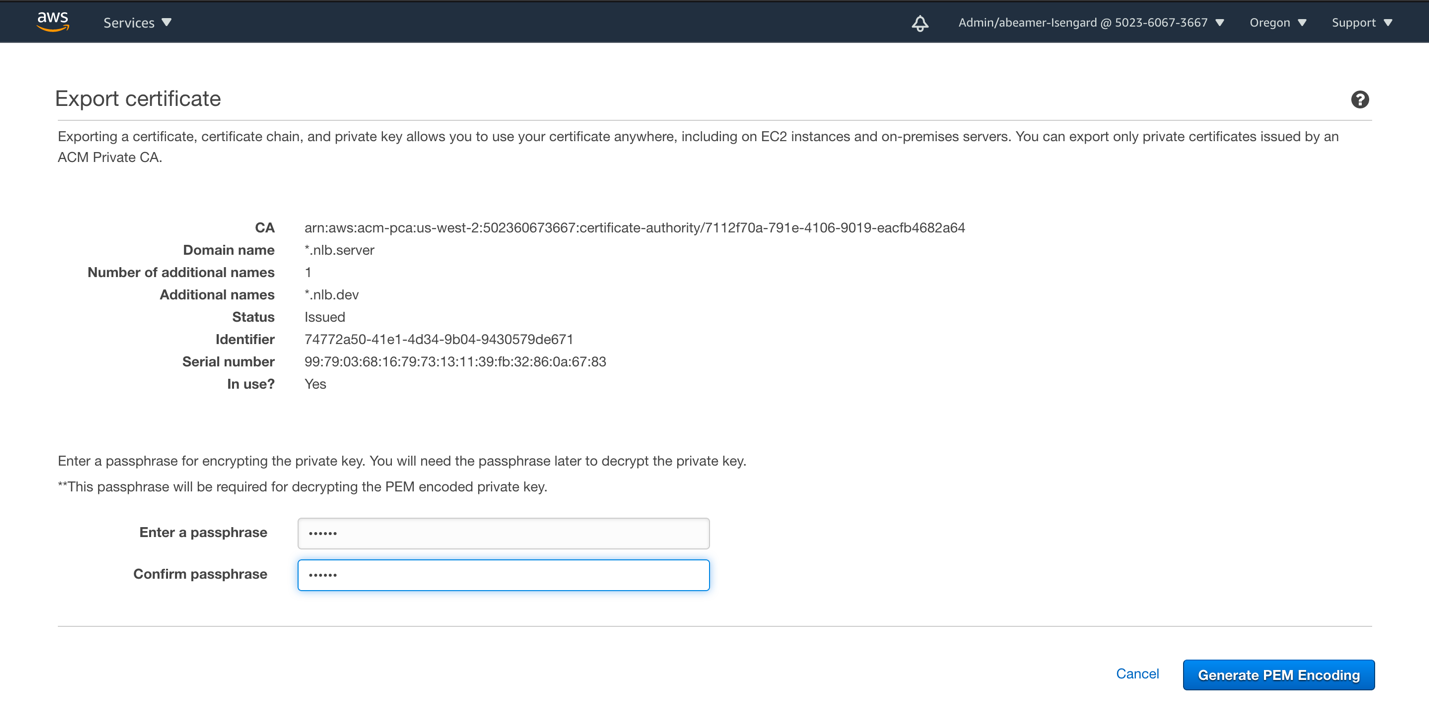
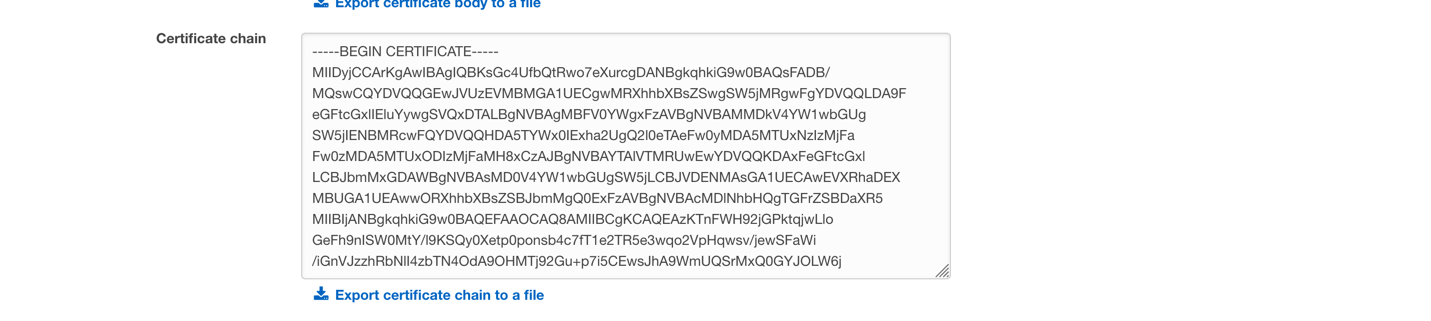
Though the architecture can be automated, there are application configurations that must be considered. Compute resources must be registered to the Target Group of the Network Load Balancer. These targets can be Instances, IPs, and lambda functions.

On the server side compute resources, the application must be able to receive TLS traffic over port 443. The application must also have access to the .crt and .key files associated with the X.509 certificate. These are configuration guides for [apache](https://httpd.apache.org/docs/2.4/ssl/ssl_howto.html) and [nginx](http://nginx.org/en/docs/http/configuring_https_servers.html). Additionally, the client must make requests using HTTPS.

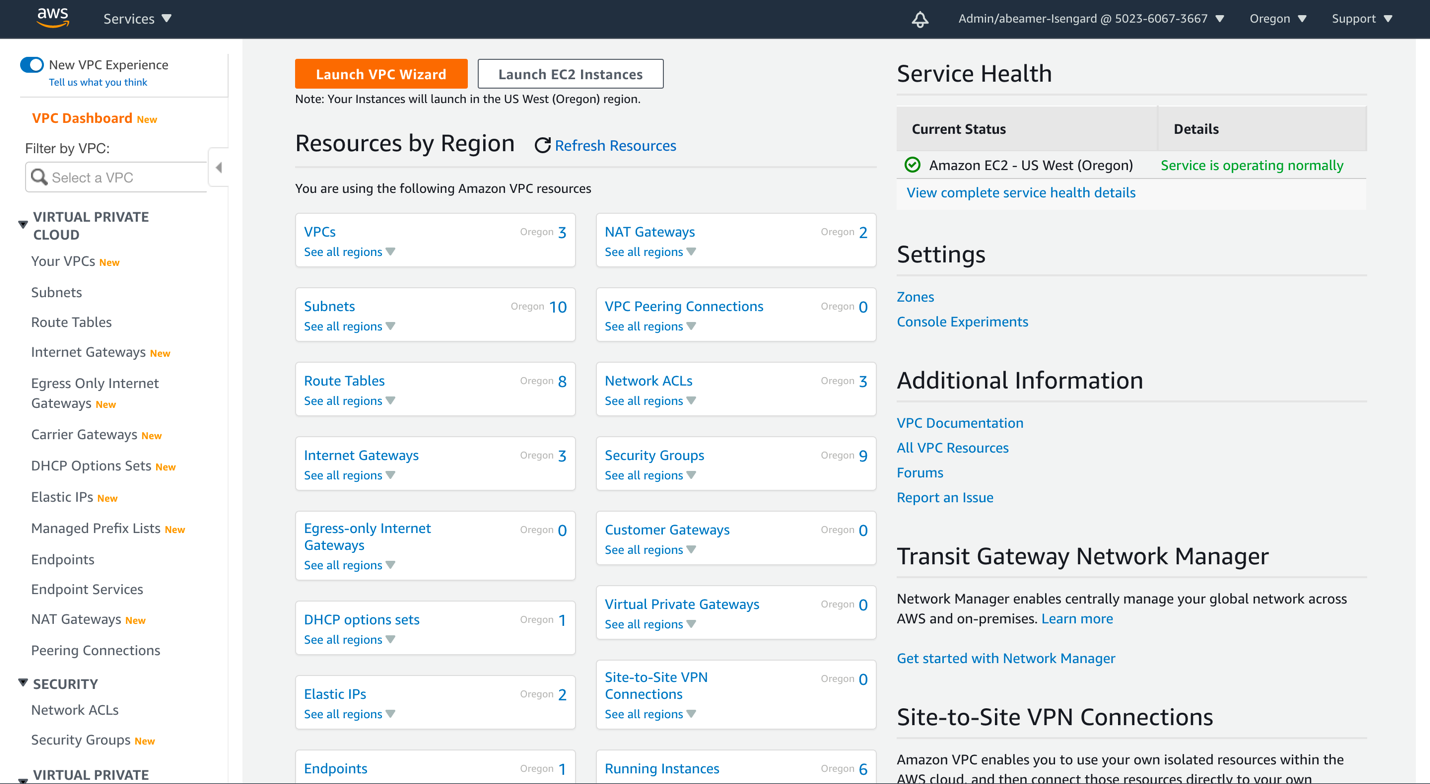
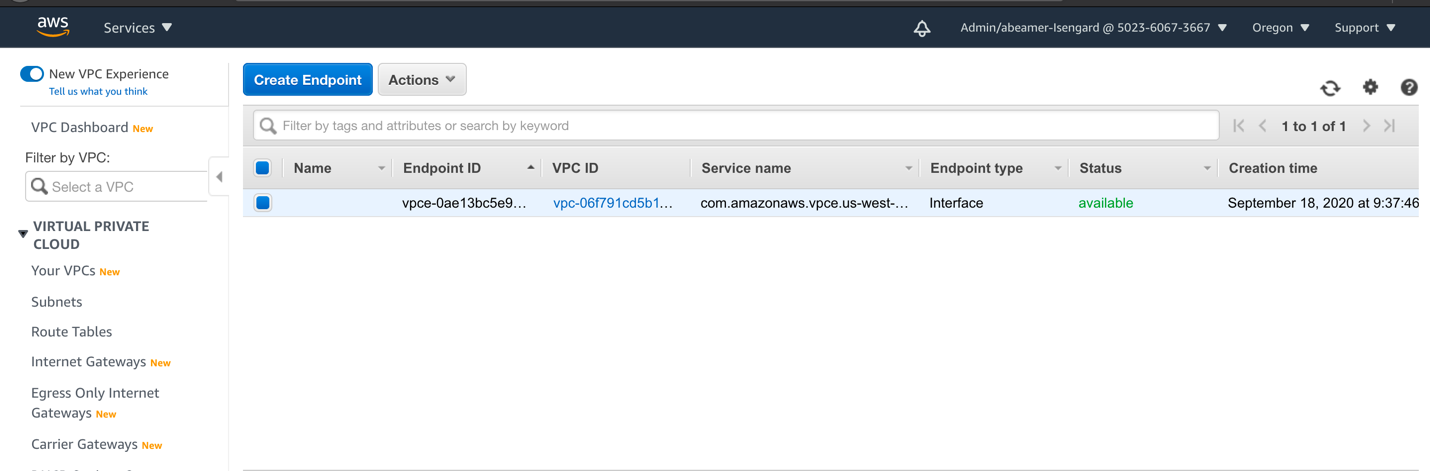
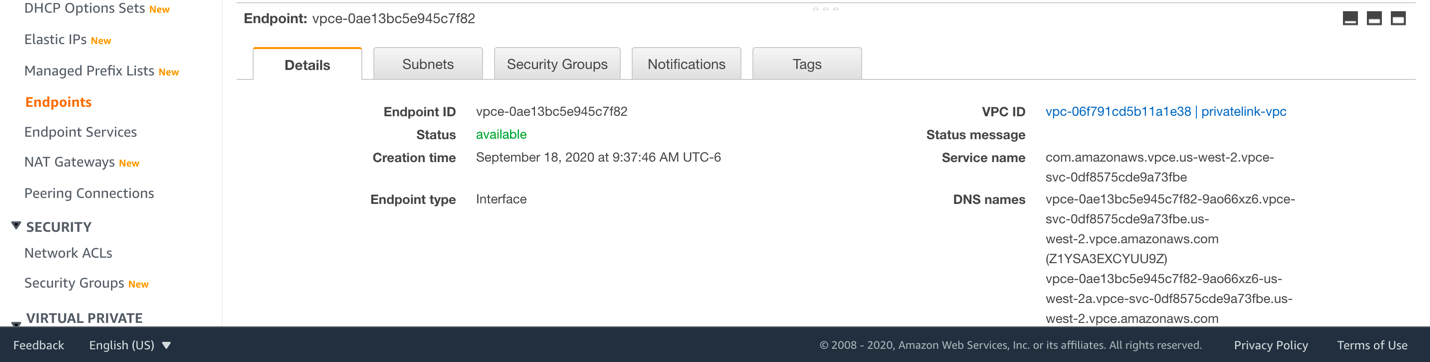
If Proxy Protocol Headers are a requirement, it can be enabled when creating the [Target Group](https://docs.aws.amazon.com/elasticloadbalancing/latest/network/load-balancer-target-groups.html). The server must be configured to accept Proxy Protocol Headers. These are configuration guidelines for Proxy Protocol Headers on [apache](https://docs.trafficserver.apache.org/en/8.0.x/admin-guide/configuration/proxy-protocol.en.html) and [nginx](https://docs.nginx.com/nginx/admin-guide/load-balancer/using-proxy-protocol/). To test if Proxy Protocol is working, send GET requests from the client and view the server-side access logs to ensure the client IP address is preserved.

**Testing and Validation**

To validate this architecture, accomplish the following:

1. Provision an ec2 instance in the client VPC. Associate it with the subnet, security group and route table that give access to the VPC PrivateLink endpoint.
2. Provision two ec2 instances in two separate Availability Zones in the server VPC. This is important, because an ELB (ALB or NLB) must be associated with at least two subnets. These subnets must be in two different availability zones. Consult the diagram below:
3. Ensure that the producer instances are registered in the NLB target group.
4. On the producer instances, configure a Nginx web server. Refer to the aforementioned Nginx guide. Currently, Proxy Protocol is only in preliminary support for Apache.
5. Export the certificate/key bundle from ACM. To do this accomplish the following:
6. In the AWS Console, go to the ACM page:
7. Select the Certificate you wish to export and click ‘Actions’. Under actions, select ‘Export (private certificates only)
8. Click ‘Export (private certificates only)’ The Export Certificate Page opens. Enter a passphrase to encrypt the PEM key file. Click ‘Generate PEM Encoding’4
9. The Exported Certificate Page is opened. Scroll down and copy the ‘Certificate chain’ certificate. This is the **only** partthat needs to be copied. Or you may click ‘Export certificate chain as a file’. 
10. Give the certificate chain file a descriptive name. ca-name-domain.crt is advisable, where ‘ca-name’ is name given to your CA (certificate authority) and ‘domain’ is the domain for the which the certificate is issued.
11. In order for browsers, services, utilities etc. to trust the NLB certificate, they must trust the certificate chain. The process for trusting certificates is beyond the scope of this, but is well documented for Windows, various browsers and other operating systems. In the case of using `curl`, a command-line oriented utility for invoking web requests, complete the following:

`curl –cacerrt /path/to/certfile.crt https://example.com`

1. In any case, distribute the certificate chain file to all consumers that will be accessing the VPC endpoint.
2. To ensure that there are no certificate errors because of mismatching domains. The following steps need to be accomplished:
3. Obtain the DNS record for the VPC endpoint. To do this, navigate to the VPC page in the AWS console.
4. Click the `Endpoints` icon to view the endpoints. Check the radio button the on the endpoint you want to access. 
5. Observe the DNS names under the ‘Details’ tab
6. Note that there are two DNS names. These both resolved to the same endpoint, but one includes the region name. Select either one.
7. Create a CNAME record in your DNS service that points to the VPC endpoint DNS address. This CNAME record should match the domain name pattern chosen in the certificate domain and/or Subject Alternate Name. For example: if the domain `private.example.com` was selected for the cert, then the CNAME should be `private.example.com

7. With the CNAME now created, clients and services in the consumer VPC should now have no trouble accessing the VPC endpoint service in the producer VPC.

**Summary**

Ensure that the cert-chain certificate is distributed to all services and clients. Automation is highly recommended for this. Ensure that there exists an alias record or CNAME record pointing to the VPC endpoint DNS record. This ensures the DNS record matches the Common Name and/or Subject Alternate Name for the NLB certificate.

Proxy Protocol 2 must be enabled on the NLB and the NLB must be configured for TLS. HTTPS does not exist on NLBs. Refer the aforementioned Nginx docs for configuring Nginx for proxy protocol.