



VPC Endpoint Services

A **VPC (Virtual Private Cloud) Endpoint Service** is a service in cloud computing, particularly in Amazon Web Services (AWS), that allows you to privately connect your VPC to supported AWS services and VPC endpoint services powered by AWS PrivateLink. This means that traffic between your VPC and the services does not leave the Amazon network, providing enhanced security and privacy.

Here's a breakdown of key concepts:

1. **VPC (Virtual Private Cloud):** A virtual network dedicated to your AWS account, isolated from other networks in AWS. It allows you to launch AWS resources into a virtual network that you've defined.
2. **VPC Endpoint:** A VPC endpoint allows you to privately connect your VPC to supported AWS services and VPC endpoint services. There are two types of VPC endpoints:
 - **Interface Endpoint:** An elastic network interface with a private IP address that serves as an entry point for traffic destined for a supported service.
 - **Gateway Endpoint:** A gateway that you specify as a target for a route in your route table for traffic destined for a supported AWS service.
3. **VPC Endpoint Service:** This is a service that you create in your VPC and configure with Network Load Balancers (NLBs). This allows other AWS customers to connect to your service via their own VPCs using interface endpoints. Essentially, it's a way to expose services that you host within your VPC to other VPCs without exposing them to the public internet.

Benefits of VPC Endpoint Services

- **Security:** Traffic between your VPC and the service doesn't leave the AWS network, reducing exposure to the internet.
- **Low Latency:** As the traffic does not traverse the public internet, it can provide lower latency.
- **Scalability and Availability:** Built on top of AWS infrastructure, providing scalable and highly available connectivity.

Use Cases

- **Private API Access:** Allow clients or partners to access APIs you host within your VPC securely.
- **Hybrid Environments:** Connect on-premises data centers to AWS services privately.
- **Service Integration:** Offer managed services to your customers with secure, low-latency connectivity.

How It Works

1. **Service Provider:** The service provider creates a VPC endpoint service with one or more Network Load Balancers (NLBs) that distribute incoming traffic to the service's instances.
2. **Service Consumer:** The consumer of the service creates an interface endpoint in their VPC, specifying the endpoint service name. The interface endpoint is essentially a private link that connects the consumer's VPC to the provider's service.

VPC Endpoint Services provide a secure and efficient way to connect services across different VPCs and accounts, making it a crucial feature for building private, scalable, and high-performing cloud architectures.

What are we doing in this Lab?

In this lab, you're setting up a **VPC Endpoint Service** using AWS. The goal is to securely expose a Nginx web server, hosted in one AWS account, to another AWS account without using the public internet.

Here's a brief summary of the steps:

1. Setup in Account 1:

- Launch an EC2 instance, install Nginx, and verify its functionality.
- Create a Network Load Balancer (NLB) and a target group, associating the EC2 instance with the NLB.
- Set up a VPC Endpoint Service using the NLB.

2. Setup in Account 2:

- Launch another EC2 instance.
- Create a VPC endpoint to connect to the VPC Endpoint Service created in Account 1.

3. Verification and Cleanup:

- Verify the connection by accessing the Nginx server from the EC2 instance in Account 2.
- Finally, delete the resources to avoid unnecessary costs.

End Goal: The exercise demonstrates how to use AWS VPC Endpoint Services and PrivateLink to securely expose services across different AWS accounts, ensuring that the traffic remains within the AWS network for enhanced security and performance.

In this lab, you're setting up VPC Endpoint Services to enable private connectivity between services in your VPC and external resources without traversing the public internet. The end goal is to demonstrate how to create and configure VPC Endpoint Services across two AWS accounts, establish connectivity between them using Network Load Balancers (NLB), and then verify the connectivity by accessing a service hosted in one account from an instance

in the other account. Finally, you'll clean up by deleting the created resources to avoid any additional costs.

😊 Step 1:

1. So, the prerequisites for this lab are that you should have two AWS accounts in place.
2. Login to AWS Console in account 1 and navigate to EC2. There you have to launch an instance and allow port 80 while launching.

The screenshot shows the AWS EC2 Instances page. A search bar at the top has 'Instance ID = i-0b530308e0f4b9d7b' entered. Below the search bar, there are filters for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public IP. One instance is listed: 'demo-server-instance' (Instance ID: i-0b530308e0f4b9d7b), which is 'Running'. The instance type is 't2.micro' and its status is 'Initializing'. It is located in 'ap-south-1a' with a Public IPv4 DNS of 'ec2-43-205-243-206.ap...' and a Public IP of '43.205.2...'. The instance summary details show the Public IPv4 address as '43.205.243.206' and the Private IP as '172.31.37.141'. The Public IP DNS name is 'ec2-43-205-243-206.ap-south-1.compute.amazonaws.com'.

3. When it is launch quickly SSH into it then switch to root user and install nginx into it. Once nginx is installed then you have to start it as well.

systemctl start nginx

4. Once nginx is started then you have to copy the public IP address and paste it in new tab to quickly verify.

The screenshot shows a web browser window with the URL '43.205.243.206'. The page displays the 'Welcome to nginx!' message. It includes a note that the server is successfully installed and working, links to online documentation at 'nginx.org' and commercial support at 'nginx.com', and a thank you message for using nginx.

5. Now you have to navigate to Load Balancers and create network load balancer.
6. So, before launching NLB first you should define a target group. First create a target group.
7. In there choose the target type as instances and then give it a name after that choose protocol as TCP because you will be using a NLB.

Target group name

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol : Port

Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation

1-65535

8. After that include your instance and create your target group.
9. Then quickly go to Load balances and create network load balancer.
10. Now give your NLB a name and in network mapping choose your default VPC and choose all the mappings that are available.

Basic configuration

Load balancer name

Name must be unique within your AWS account and can't be changed after the load balancer is created.

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme

Scheme can't be changed after the load balancer is created.

Internet-facing

An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

Internal

An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type | [Info](#)

Select the type of IP addresses that your subnets use.

IPv4

Recommended for internal load balancers.

Dualstack

Includes IPv4 and IPv6 addresses.

Mappings

Select at least one Availability Zone and one subnet for each zone. We recommend selecting at least two Availability Zones. The load balancer will route traffic only to targets in the selected Availability Zones. Zones that are not supported by the load balancer or VPC can't be selected. Subnets can be added, but not removed, once a load balancer is created.

ap-south-1a (aps1-az1)

Subnet

IPv4 address

ap-south-1b (aps1-az3)

Subnet

IPv4 address

ap-south-1c (aps1-az2)

Subnet

IPv4 address

11. Choose the security group of your choice because you are going to change the inbound rules so that it can listen on port 80. After that choose your target group and then keep everything to default and create your NLB.

Security groups - recommended

Security groups support on Network Load Balancers can only be enabled at creation by including at least one security group. You can change security groups after creation. The security groups for your load balancer must allow it to communicate with registered targets on both the listener port and the health check port. For PrivateLink Network Load Balancers, security group rules are enforced on PrivateLink traffic; however, you can turn off inbound rule evaluation after creation within the load balancer's Security tab or using the API.

The screenshot shows the AWS Lambda function configuration page. In the 'Listeners and routing' section, there is a single listener named 'TCP:80'. The configuration details are as follows:

- Protocol:** TCP
- Port:** 80
- Default action:** Forward to 'demo-service-tg'
- Target type:** Instance, IPv4
- Actions:** Remove, Copy (C)
- Create target group:** Create target group

12. Currently your NLB is in provisioning state, wait for it. Until then go to the security group that you have assigned to it and add port 80 on it.

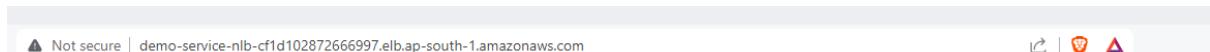
The screenshot shows the AWS Network Load Balancer (NLB) configuration page. In the 'Details' tab, the following information is displayed:

Load balancer type Network	Status Provisioning	VPC vpc-00e852ef26c39581b	IP address type IPv4
Scheme Internet-facing	Hosted zone ZVDDRBQ08TROA	Availability Zones subnet-090908b5b996470fc ap-south-1c (aps1-az2) subnet-01c162e279b989d09 ap-south-1a (aps1-az1) subnet-05b1afa053579e078 ap-south-1b (aps1-az3)	Date created February 28, 2024, 16:30 (UTC+05:30)
Load balancer ARN arn:aws:elasticloadbalancing:ap-south-1:878893308172:loadbalancer/net/demo-service-nlb/cf1d10287266699	DNS name Info demo-service-nlb-cf1d10287266699.elb.ap-south-1.amazonaws.com (A Record)		

13. Once your NLB is provisioned. Copy its DNS name and paste it in a new tab. You will see that nginx server is active and running.

The screenshot shows the AWS Network Load Balancer (NLB) configuration page. In the 'Details' tab, the following information is displayed:

Load balancer type Network	Status Active	VPC vpc-00e852ef26c39581b	IP address type IPv4
Scheme Internet-facing	Hosted zone ZVDDRBQ08TROA	Availability Zones subnet-090908b5b996470fc ap-south-1c (aps1-az2) subnet-01c162e279b989d09 ap-south-1a (aps1-az1) subnet-05b1afa053579e078 ap-south-1b (aps1-az3)	Date created February 28, 2024, 16:30 (UTC+05:30)
Load balancer ARN arn:aws:elasticloadbalancing:ap-south-1:878893308172:loadbalancer/net/demo-service-nlb/cf1d10287266699	DNS name Info demo-service-nlb-cf1d10287266699.elb.ap-south-1.amazonaws.com (A Record)		



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

14. Now you have to navigate to VPC and from the left pane go to Endpoint services. There click on create endpoint service.

15. Here you have to give it a name then select network in load balancer type.

16. Then you have to select your NLB and move forward.

17. Now keep the settings to default as shown below and create your endpoint service.

Additional settings

Require acceptance for endpoint | [Info](#)
 Specify whether requests from service consumers to connect to your service through an endpoint must be accepted.

Acceptance required

Enable private DNS name
 This option allows users of endpoints to use the specified private DNS name for access the service from their VPCs.

Associate a private DNS name with the service

Supported IP address types

IPv4
 IPv6

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="demo-service-provider"/> X Remove

[Add new tag](#)

You can add 49 more tags.

[Cancel](#) [Create](#)

18. Below you can see that your endpoint service is created and it is connected with your NLB.

VPC > Endpoint services > vpce-svc-00c647f2c4a870764		vpce-svc-00c647f2c4a870764 / demo-service-provider																					
Actions ▾																							
Details																							
Service ID	vpce-svc-00c647f2c4a870764	Types	Service name																				
Network Load Balancers ARNs	aws:elasticloadbalancing:ap-south-1:878893308172:loadbalancer/net/demo-service-nlb/cf1d102872666997	Interface	com.amazonaws.vpce.ap-south-1.vpce-svc-00c647f2c4a870764																				
DNS names	vpce-svc-00c647f2c4a870764.ap-south-1.vpce.amazonaws.com	Gateway Load Balancers ARNs	Availability Zones																				
Domain verification type	Info	Private DNS name	Availability Zones																				
-	-	-	Domain verification status Info																				
			-																				
			Supported IP address type																				
			ipv4																				
Load balancers Allow principals Endpoint connections Notifications Monitoring Contributor Insights Tags																							
Load balancers (3) Info <table border="1"> <thead> <tr> <th colspan="2">Search</th> <th>C</th> <th>Associate or Disassociate load balancers</th> </tr> </thead> <tbody> <tr> <td>Availability Zone</td> <td>Load balancer names</td> <td colspan="2"> C 1 2 3 @ </td> </tr> <tr> <td>ap-south-1c (aps1-az2)</td> <td>demo-service-nlb</td> <td colspan="2"></td> </tr> <tr> <td>ap-south-1b (aps1-az3)</td> <td>demo-service-nlb</td> <td colspan="2"></td> </tr> <tr> <td>ap-south-1a (aps1-az1)</td> <td>demo-service-nlb</td> <td colspan="2"></td> </tr> </tbody> </table>				Search		C	Associate or Disassociate load balancers	Availability Zone	Load balancer names	C 1 2 3 @		ap-south-1c (aps1-az2)	demo-service-nlb			ap-south-1b (aps1-az3)	demo-service-nlb			ap-south-1a (aps1-az1)	demo-service-nlb		
Search		C	Associate or Disassociate load balancers																				
Availability Zone	Load balancer names	C 1 2 3 @																					
ap-south-1c (aps1-az2)	demo-service-nlb																						
ap-south-1b (aps1-az3)	demo-service-nlb																						
ap-south-1a (aps1-az1)	demo-service-nlb																						

Step 2:

1. Now navigate to Account 2 and open EC2 and then launch an instance.

Instances (1/1) Info

Instance ID = i-043f7e0276893c25d

Find Instance by attribute or tag (case-sensitive)

Any state

Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IP

account2-ec2 | i-043f7e0276893c25d | Running | t2.micro | 2/2 checks passed | View alarms | ap-south-1a | ec2-13-23

Instance: i-043f7e0276893c25d (account2-ec2)

Details | Status and alarms New | Monitoring | Security | Networking | Storage | Tags

Instance summary

Instance ID: i-043f7e0276893c25d (account2-ec2)

Public IPv4 address: 13.234.117.169 [open address]

Private IPv4 addresses: 172.31.37.136

IPv6 address: -

Instance state: Running

Public IPv4 DNS: ec2-13-234-117-169.ap-south-1.compute.amazonaws.com [open address]

- Once your instance is created then you have to navigate to VPC and open Endpoint there. Then click on create endpoint.

Endpoints Info

Search

Name | VPC endpoint ID | VPC ID | Service name

No endpoint found

- In there you have to give it a name and select other endpoint services.

Name tag - optional
Creates a tag with a key of 'Name' and a value that you specify.

demo-endpoint

Service category
Select the service category

AWS services
Services provided by Amazon

PrivateLink Ready partner services
Services with an AWS Service Ready designation

AWS Marketplace services
Services that you've purchased through AWS Marketplace

EC2 Instance Connect Endpoint
An elastic network interface that allow you to connect to resources in a private subnet

Other endpoint services
Find services shared with you by service name

- Below you will see and options for service name. For it you have to go back to account one and navigate to endpoint services. There in your created endpoint service you will see the service name.

Service ID vpc-e-svc-00c647f2c4a870764	Types Interface	Service name com.amazonaws.vpce.ap-south-1.vpce-svc-00c647f2c4a870764
Network Load Balancers ARNs arn:aws:elasticloadbalancing:ap-south-1:878893308172:loadbalancer/net/demo-service-nlb/cf1d102872666997	Gateway Load Balancers ARNs -	Availability Zones 3 Availability Zones
DNS names vpc-e-svc-00c647f2c4a870764.ap-south-1.vpce.amazonaws.com	Private DNS name -	Domain verification status Info
Domain verification type Info	Domain verification value -	Supported IP address type IPv4
		State Available
		Acceptance required Yes
		Domain verification name Info

- Now copy this service name and paste it in the account 2. After pasting that you have to verify it but you will see that it is giving you an error because from your account 1 you have to allow this account first in your endpoint service.

Service settings

Service name
com.amazonaws.vpce.ap-south-1.vpce-svc-00c647f2c4a870764 Verify service

✖ Service name could not be verified.

6. Comeback to account 1 in the endpoint service. You will see an option for allow principals. Click on allow.

Load balancers Allow principals Endpoint connections Notifications Monitoring Contributor Insights Tags

Allow principals Info

Search		Actions		Allow principals
	Name	ID	Type	Service permission ID
You do not have any allow principals in this Region.				

7. Now you have to paste this ARN along with the account ID of account 2. Then click on allow principals.

arn:aws:iam::accountID:root

Allow principals Info

Summary

Endpoint service ID vpce-svc-00c647f2c4a870764	Endpoint service name tag demo-service-provider	Endpoint service name com.amazonaws.vpce.ap-south-1.vpce-svc-00c647f2c4a870764
--	---	--

Principals to add

ARN	<input style="width: 100%;" type="text" value="arn:aws:iam::463646775279:root"/>
Add principal Allow principals	

8. Once this is allowed go back to your endpoint in account 2 and now re-verify your service name you will see that it has been verified.

Service settings

Service name
com.amazonaws.vpce.ap-south-1.vpce-svc-00c647f2c4a870764 Verify service

✓ Service name verified.

9. Now choose your VPC and then the subnets along with the subnet ID. After choose a security group. Then just create your endpoint.

VPC
Select the VPC in which to create the endpoint

The VPC in which to create your endpoint:
vpce-072c4acd761c3b942 (default-vpc)

► Additional settings

Subnets (3/3) Info

Availability Zone	Subnet ID	Designate IP addresses	IPv4 address	IPv6 address
ap-south-1a (aps1-az1)	subnet-04ffa8e5e9c79ad3d	<input type="checkbox"/>		
ap-south-1b (aps1-az3)	subnet-0bc545e3030c41ef4	<input type="checkbox"/>		
ap-south-1c (aps1-az2)	subnet-0f5b8a66b67a97c62	<input type="checkbox"/>		

IP address type
 IPv4
 IPv6
 Dualstack

10. Once it is created you will see that it is pending acceptance.

Endpoints (1/1) Info

Name	VPC endpoint ID	VPC ID	Service name
demo-endpoint	vpce-0e1d7dfc33d72e8de	vpce-072c4acd761c3b942 default-vpc	com.amazonaws.vpce.ap-south-1.vpce-svc-00c647f2c4...

vpce-0e1d7dfc33d72e8de / demo-endpoint

Details | Subnets | Security Groups | Notification | Monitoring | Tags

Details

Endpoint ID vpce-0e1d7dfc33d72e8de	Status Pending acceptance	Creation time Wednesday, February 28, 2024 at 17:05:33 GMT+5:30	Endpoint type Interface
VPC ID vpce-072c4acd761c3b942 (default-vpc)	Status message -	Service name com.amazonaws.vpce.ap-south-1.vpce-svc-00c647f2c4a870764	Private DNS names enabled No
DNS record IP type ipv4	IP address type ipv4	DNS names vpce-0e1d7dfc33d72e8de-ot4plr05.vpce-svc-00c647f2c4a870764.ap-south-1.vpce.amazonaws.com	

11. Now go back to endpoint service in account 1. There in the endpoint connections you will see the request. Select it and accept it.

Load balancers | Allow principals | **Endpoint connections** | Notifications | Monitoring | Contributors

Endpoint connections (1/1) Info

Name	Endpoint ID	Owner	State	Create
-	vpce-0e1d7dfc33d72e8de	463646775279	Pending acceptance	Wednesday, February 28, 2024 at 17:05:33 GMT+5:30

Accept endpoint connection request
Reject endpoint connection request
Manage tags
Actions ▲ Create endpoint connection

12. Now if you comeback to endpoint in account 2 you will see that the status is available.

The screenshot shows the AWS VPC Endpoints console. At the top, there's a search bar and a 'Create endpoint' button. Below it is a table with columns: Name, VPC endpoint ID, VPC ID, Service name, and Actions. One row is selected, showing 'demo-endpoint', 'vpce-0e1d7dfc33d72e8de', 'vpce-0e1d7dfc33d72e8de | default-vpc', 'com.amazonaws.vpce.ap-south-1.vpce-svc-00c647f2c4...', and an 'Int' icon. Below the table is a breadcrumb path 'vpce-0e1d7dfc33d72e8de / demo-endpoint'. Underneath are tabs for Details, Subnets, Security Groups, Notification, Monitoring, and Tags. The 'Details' tab is selected, displaying endpoint information: Endpoint ID (vpce-0e1d7dfc33d72e8de), Status (Available), Creation time (Wednesday, February 28, 2024 at 17:05:33 GMT+5:30), Service name (com.amazonaws.vpce.ap-south-1.vpce-svc-00c647f2c4a870764), and Endpoint type (Interface). It also shows that Private DNS names are enabled (No). A red box highlights the 'DNS names' section, which lists 'vpce-0e1d7dfc33d72e8de-ot4plr05.vpce-svc-00c647f2c4a870764.ap-south-1.vpce.amazonaws.com'.

13. Now one thing more the security group you attached with endpoint should have inbound rule for port 80.
14. Afterwards if you go to EC2 and SSH into your instance. Now run the curl command and paste the DNS name which is provided to you by endpoint.
15. You will see the nginx page which means that our connectivity is successful.
16. It also means that your entire setup for VPC endpoint services is working.

```

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-172-31-37-136 ~]$ curl vpce-0e1d7dfc33d72e8de-ot4plr05.vpce-svc-00c647f2c4a870764.ap-south-1.vpce.amazonaws.com
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto; font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>if you see this page, the nginx web server is successfully installed and working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">http://nginx.org/</p>
<p>Commercial support is available at
<a href="http://nginx.com/">http://nginx.com/</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
[ec2-user@ip-172-31-37-136 ~]$ 

```

😊 Step 3: Deleting Resources

1. Go to endpoint in account 2 and delete your endpoint.
2. Once it is deleted then go to endpoint services in account 1 and delete that.
3. Now terminate your NLB.
4. After that terminate your instance.
5. Don't forget to delete the target group.