😊 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.

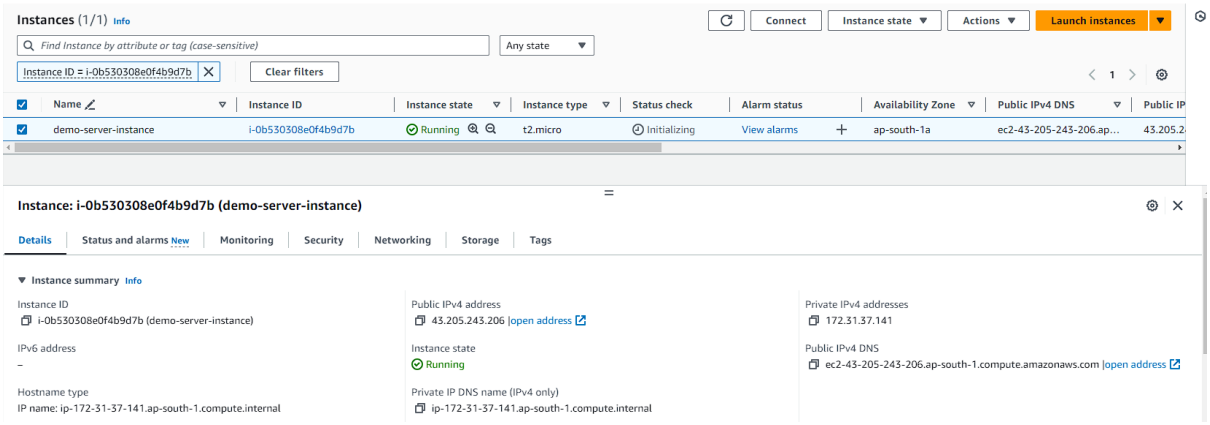
End Goal: The exercise demonstrates how to use AWS VPC Endpoint Services and PrivateLink to securely expose services across different AWS accounts, ensuring that the traffic remains within the AWS network for enhanced security and performance.

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.

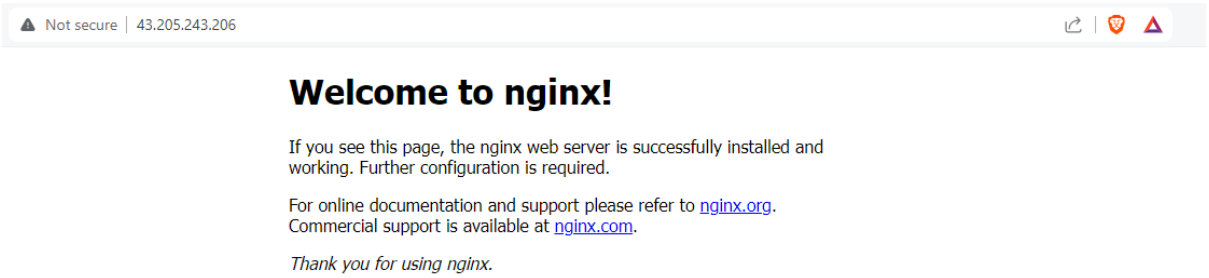
😄 Step 1:

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.



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.



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

demo-service-provider-target-group

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

TCP | 80

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.

demo-service-nlb

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: subnet-01c162e279b989d09

IPv4 address: Assigned by AWS

☑ **ap-south-1b (aps1-az3)**

Subnet: subnet-05b1afa053579e078

IPv4 address: Assigned by AWS

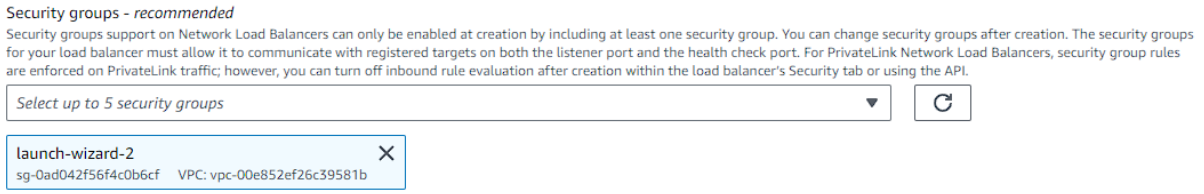
subnet-05b1afa053579e078

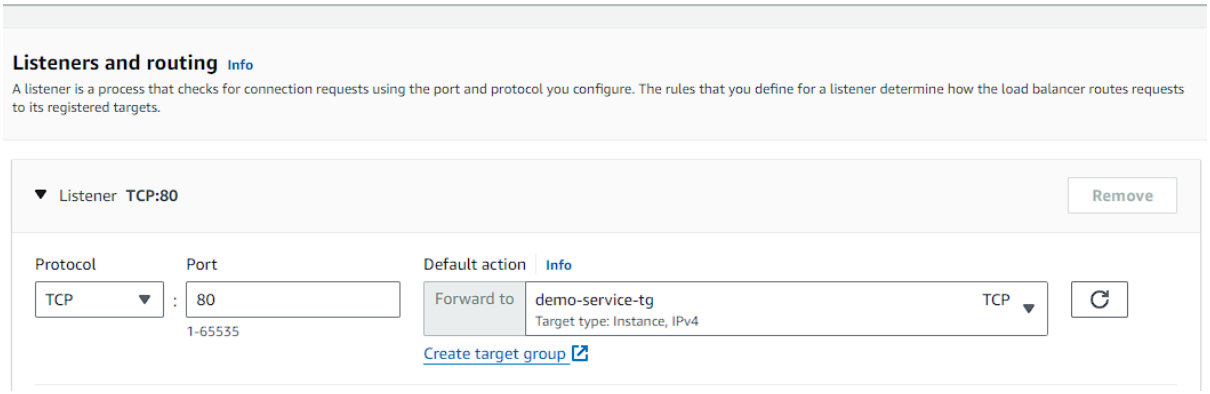
☑ **ap-south-1c (aps1-az2)**

Subnet: subnet-090908b5b996470fc

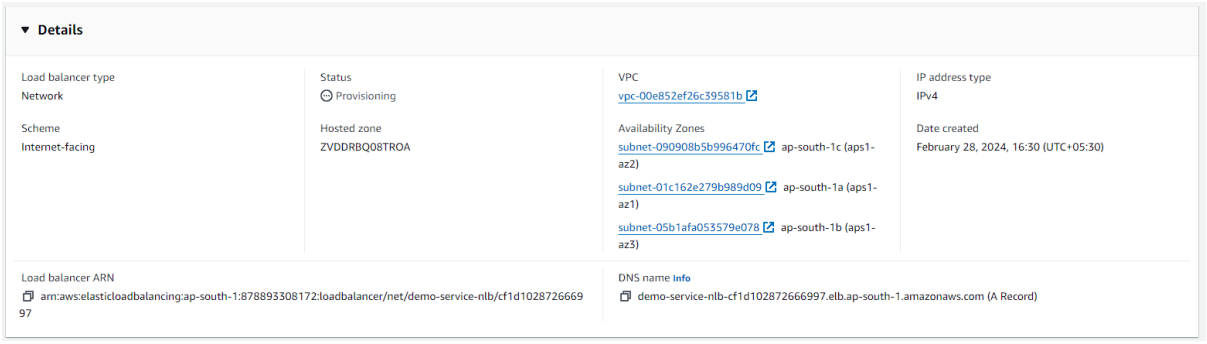
IPv4 address: Assigned by AWS

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.

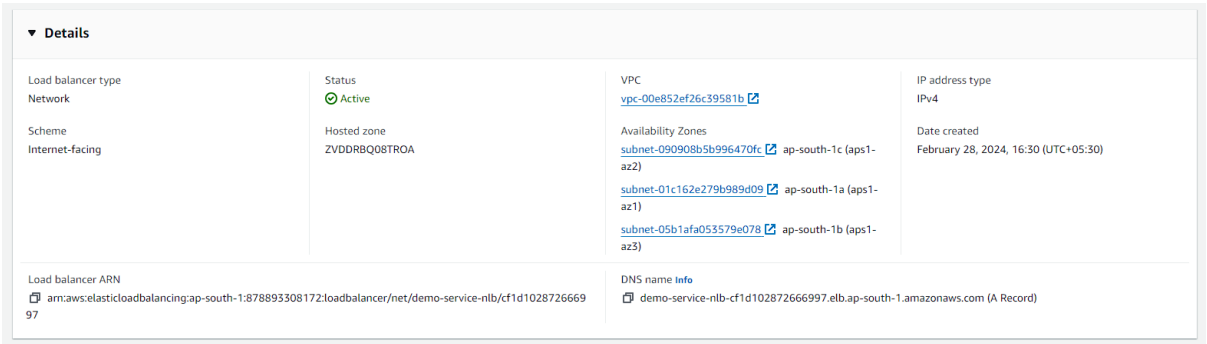


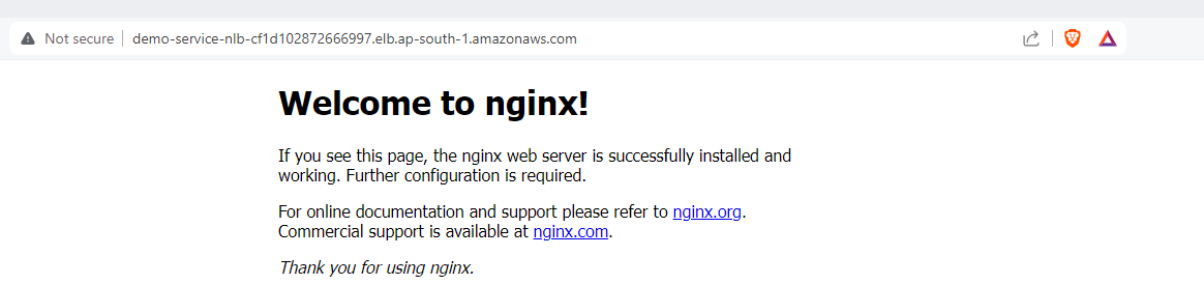


12. Currently your NLB is in provisioning state, wait for it. Until then go to the security group that you have assigned to it and add port 80 on it.

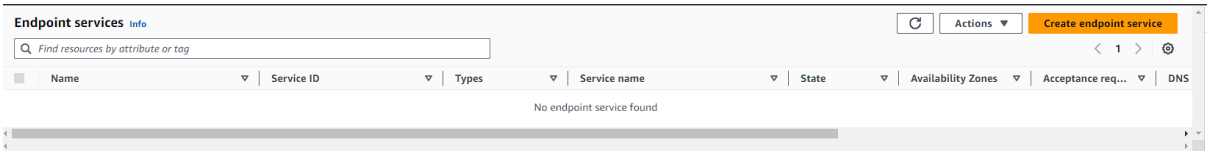


13. Once your NLB is provisioned. Copy its DNS name and paste it in a new tab. You will see that nginx server is active and running.

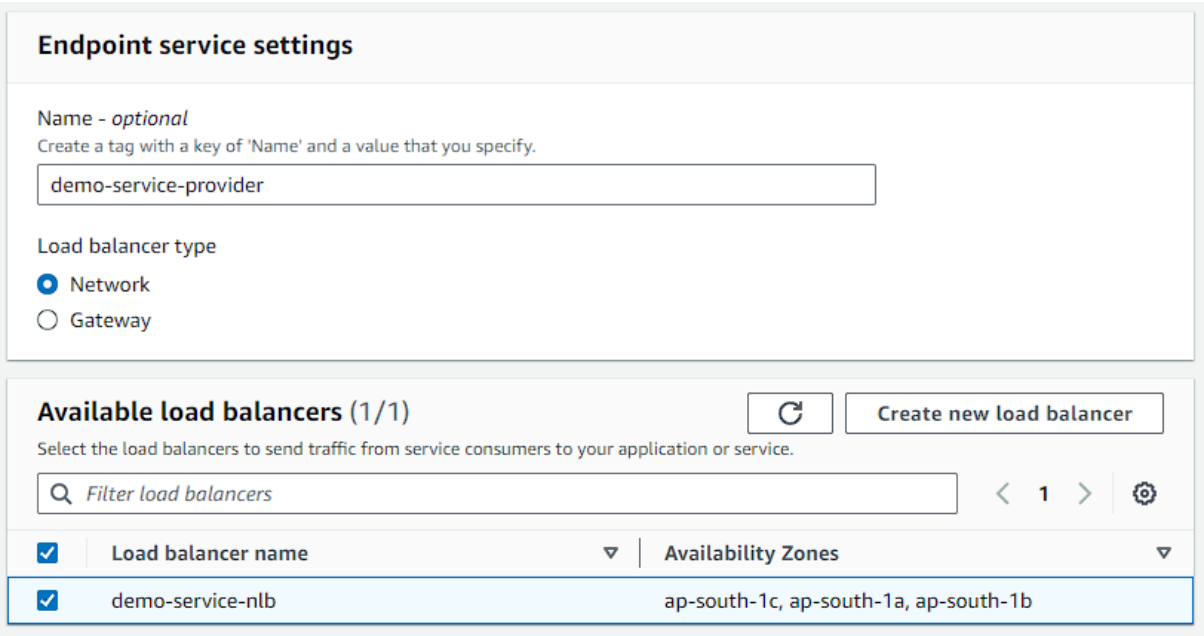




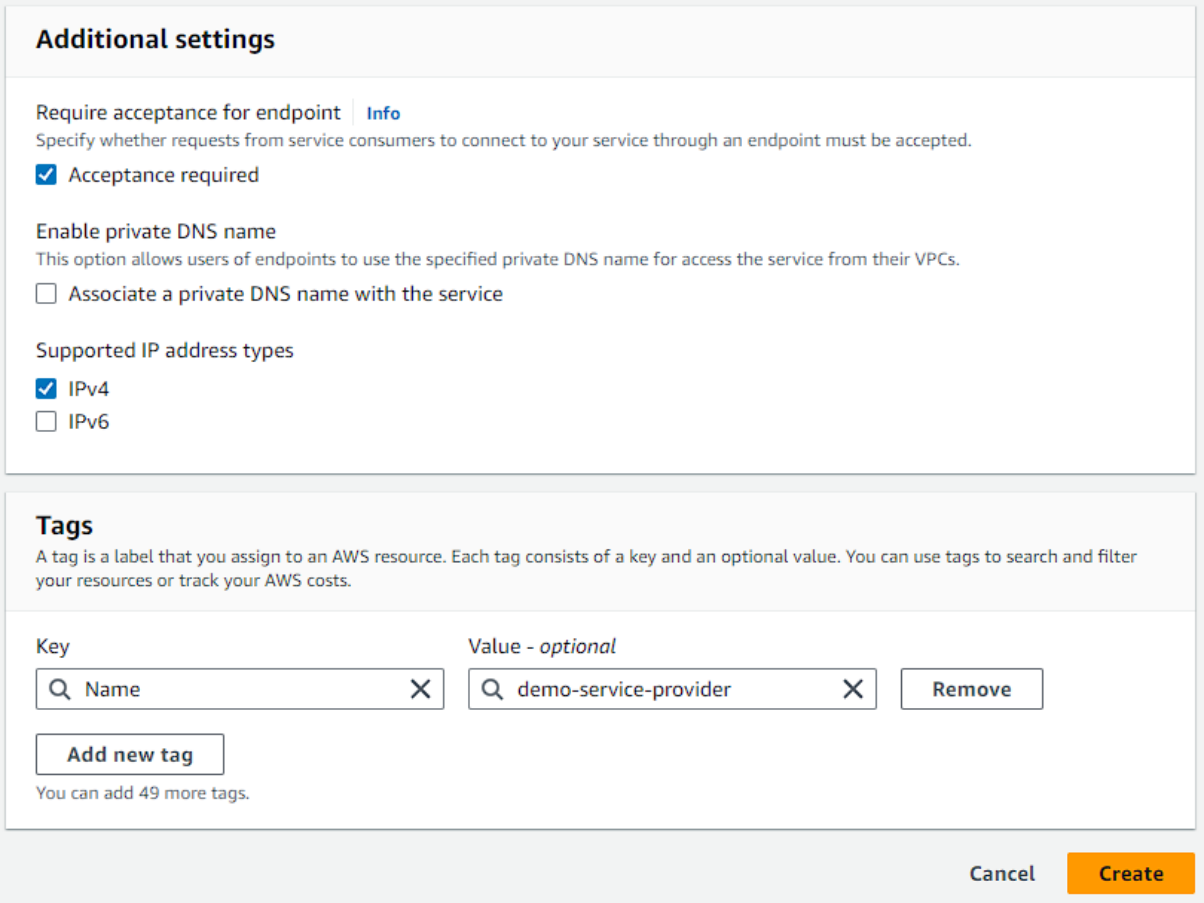
14. Now you have to navigate to VPC and from the left pane go to Endpoint services. There click on create endpoint service.



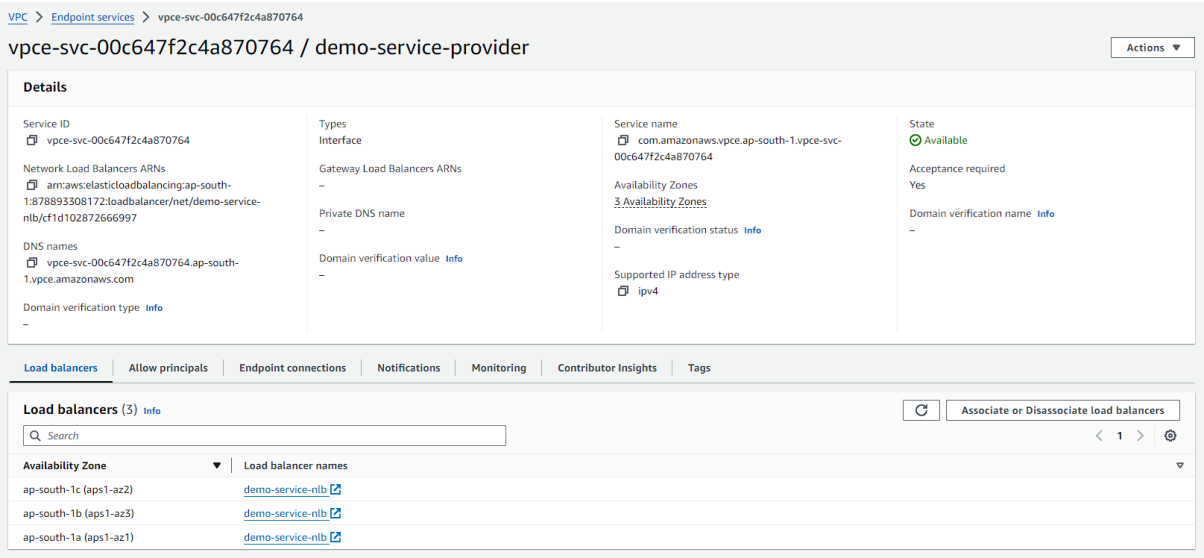
15. Here you have to give it a name then select network in load balancer type.
16. Then you have to select your NLB and move forward.



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

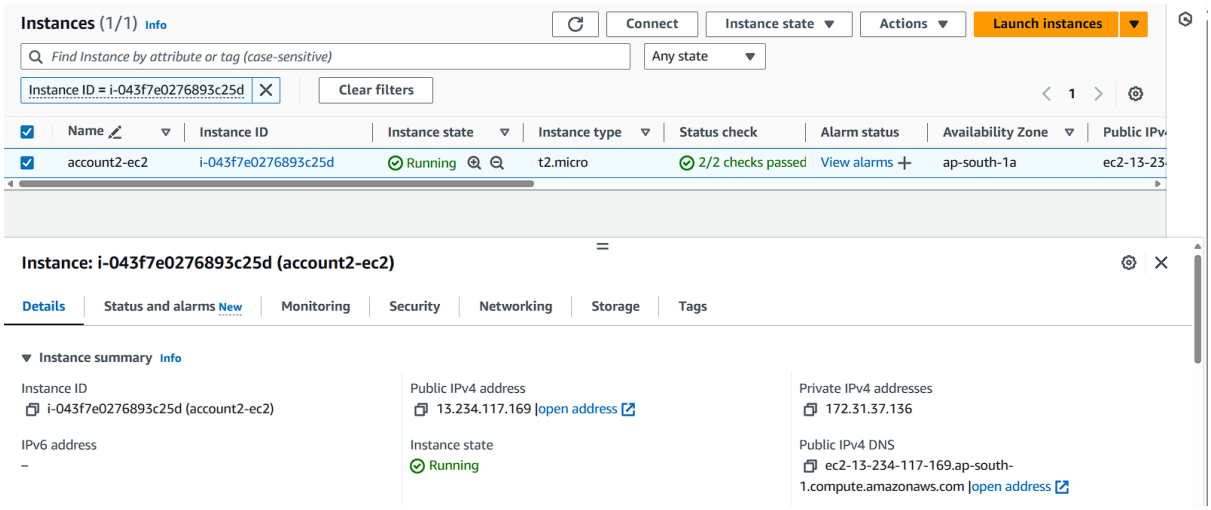


18. Below you can see that your endpoint service is created and it is connected with your NLB.

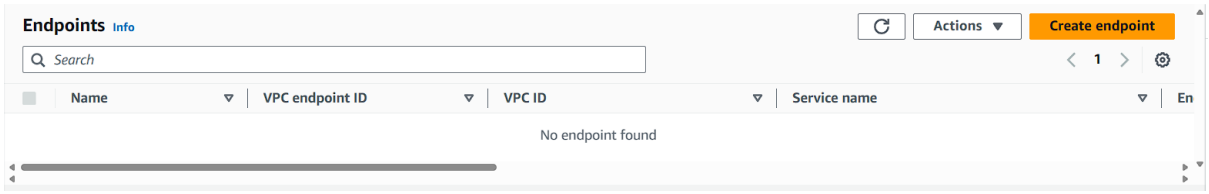


😄 Step 2:

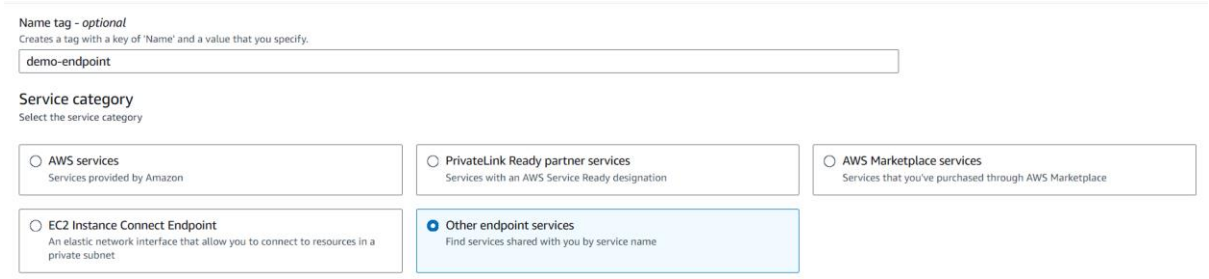
1. Now navigate to Account 2 and open EC2 and then launch an instance.



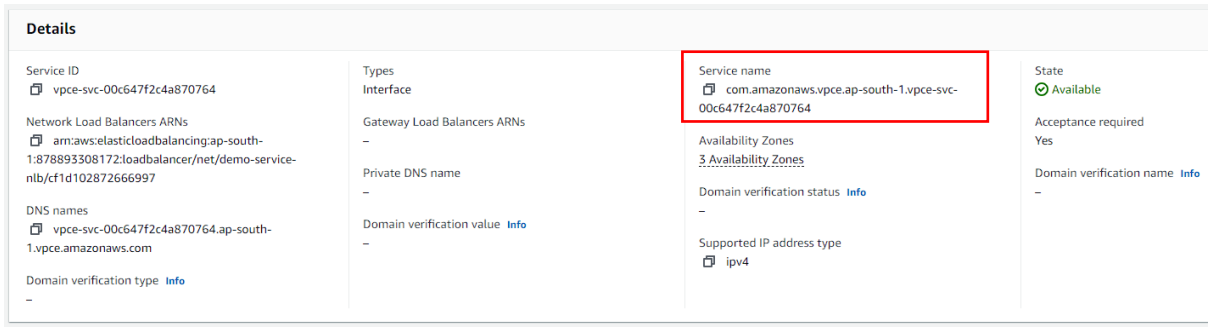
2. Once your instance is created then you have to navigate to VPC and open Endpoint there. Then click on create endpoint.



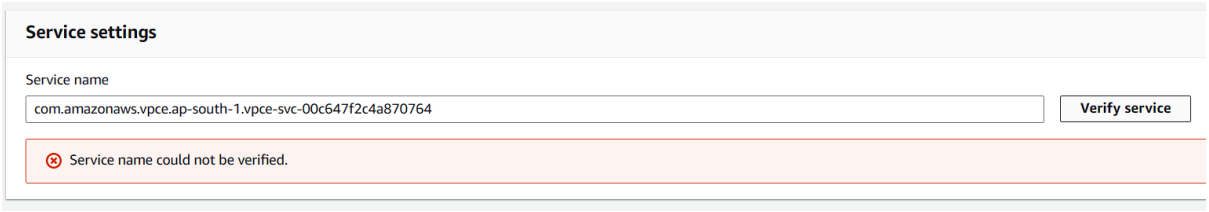
3. In there you have to give it a name and select other endpoint services.



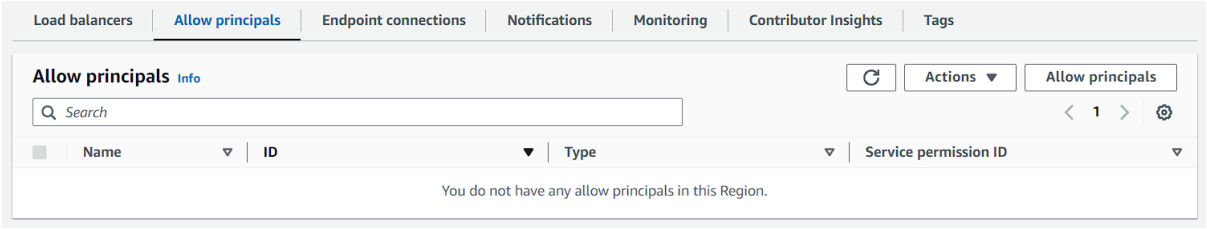
4. Below you will see and options for service name. For it you have to go back to account one and navigate to endpoint services. There in your created endpoint service you will see the service name.



5. 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.

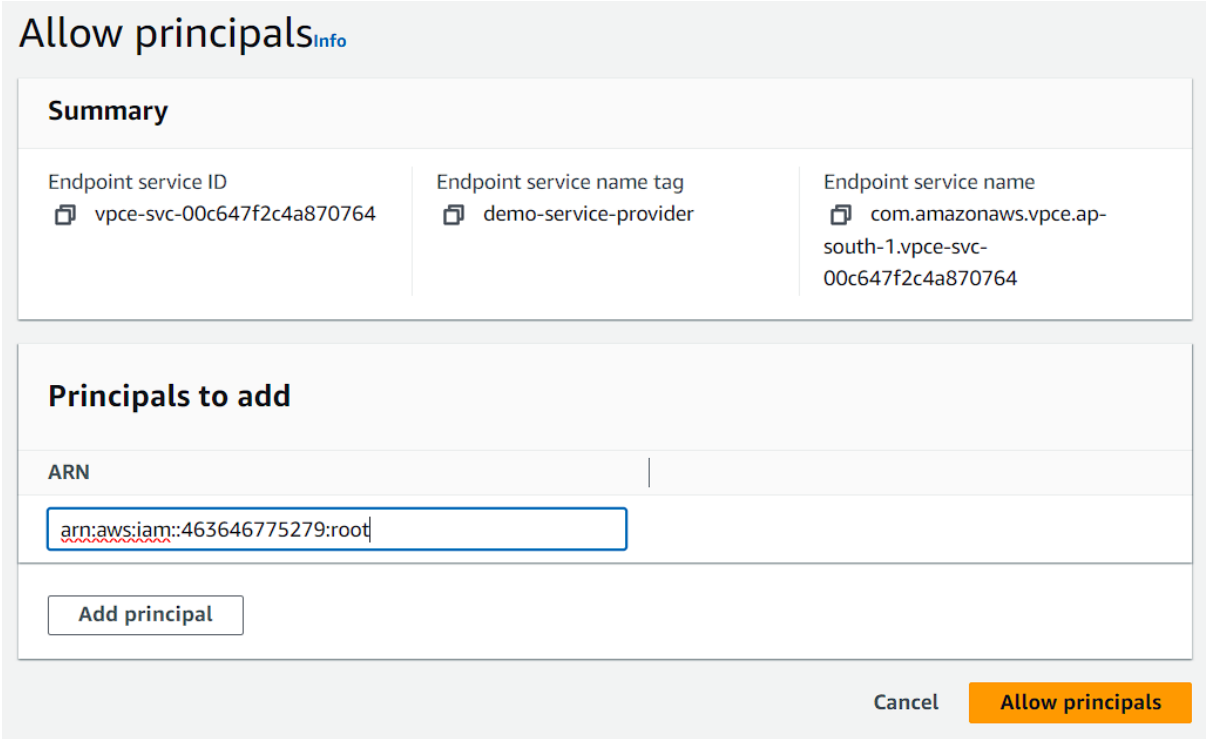


6. Comeback to account 1 in the endpoint service. You will see an option for allow principals. Click on allow.

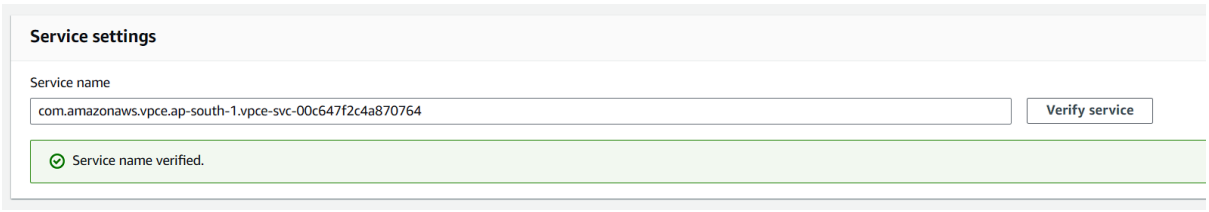


7. Now you have to paste this ARN along with the account ID of account 2. Then click on allow principals.

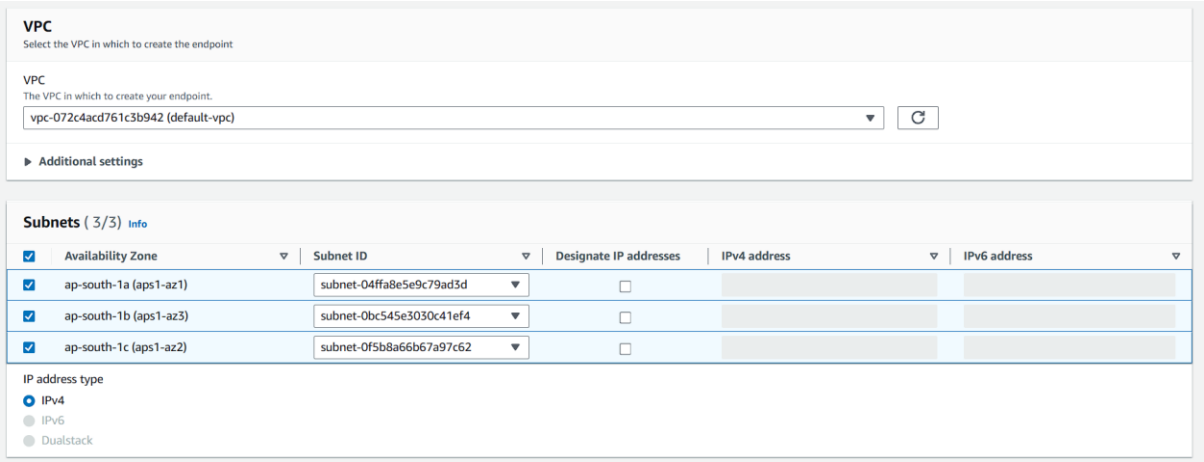
   arn:aws:iam::accountID:root



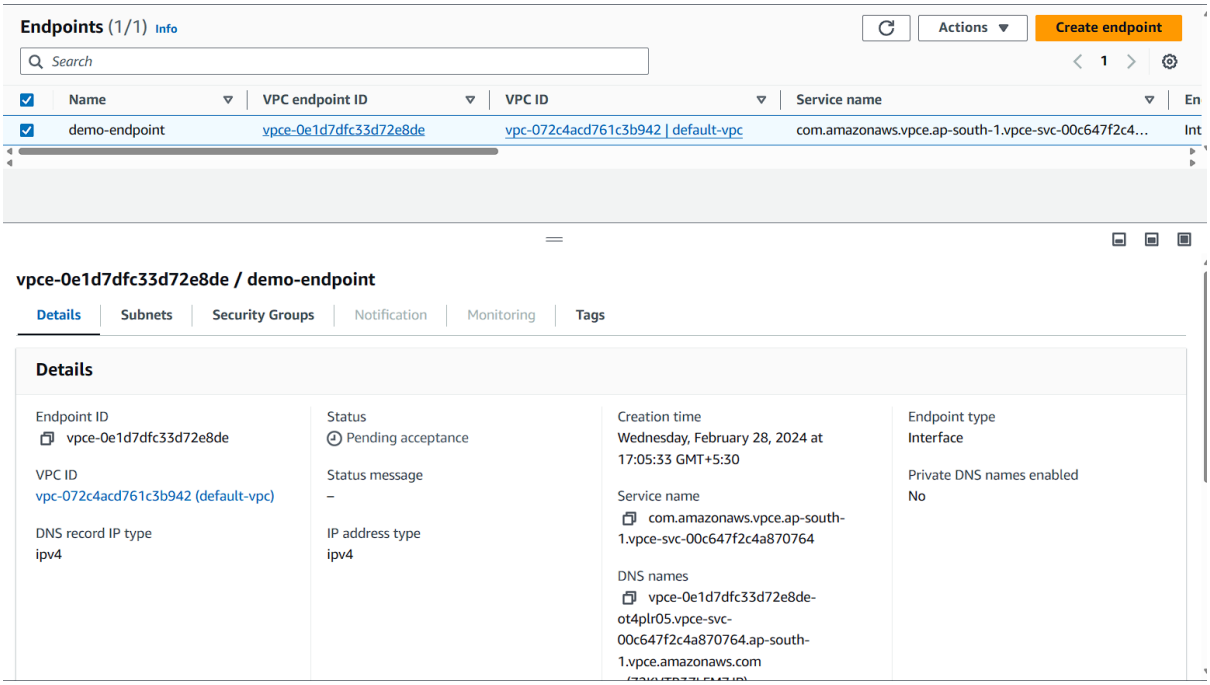
8. Once this is allowed go back to your endpoint in account 2 and now re-verify your service name you will see that it has been verified.



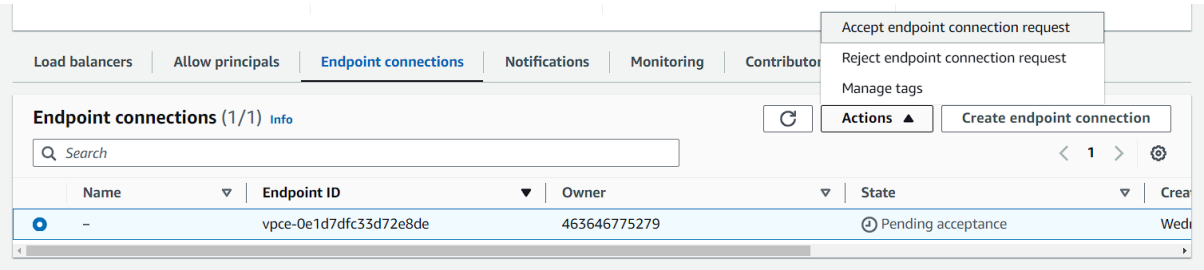
9. Now choose your VPC and then the subnets along with the subnet ID. After choose a security group. Then just create your endpoint.



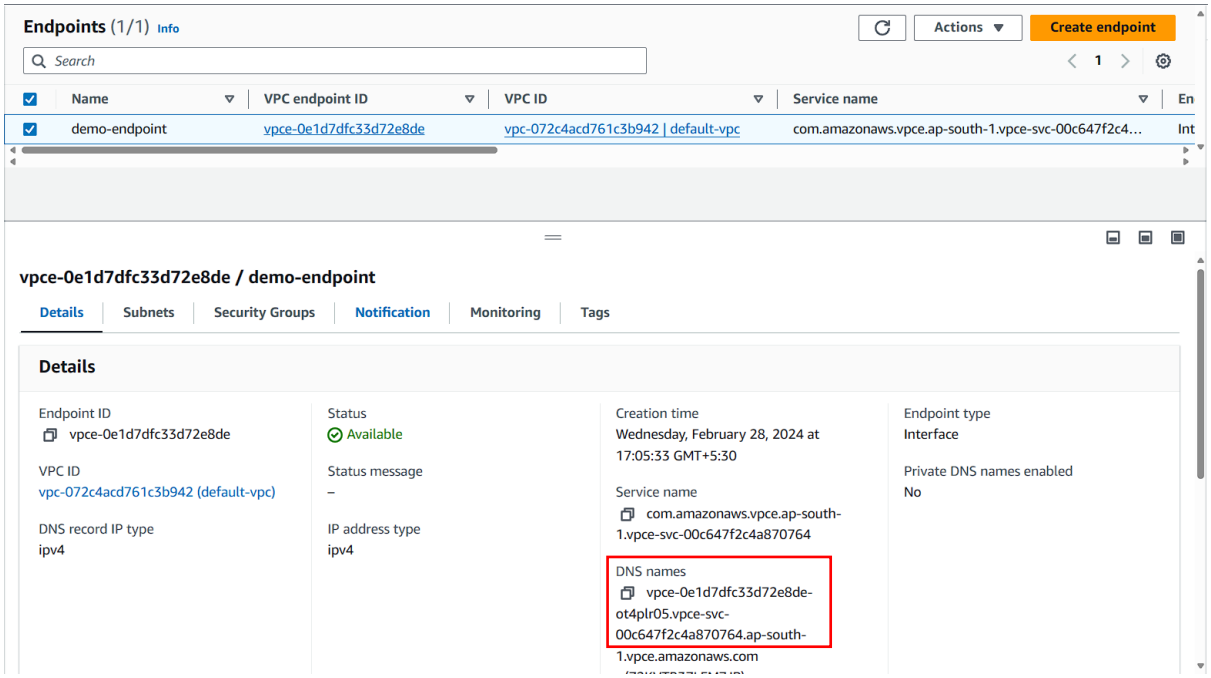
10. Once it is created you will see that it is pending acceptance.



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.



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



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.

😄 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.