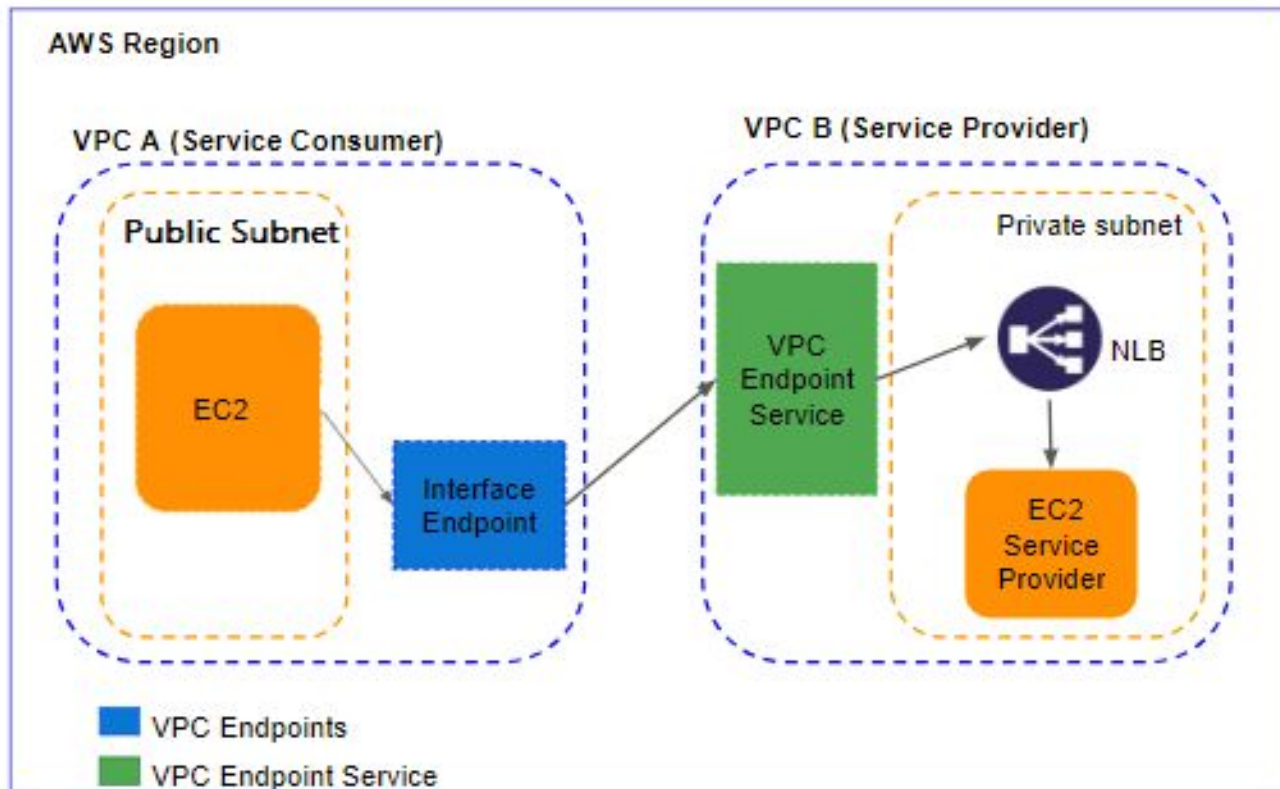
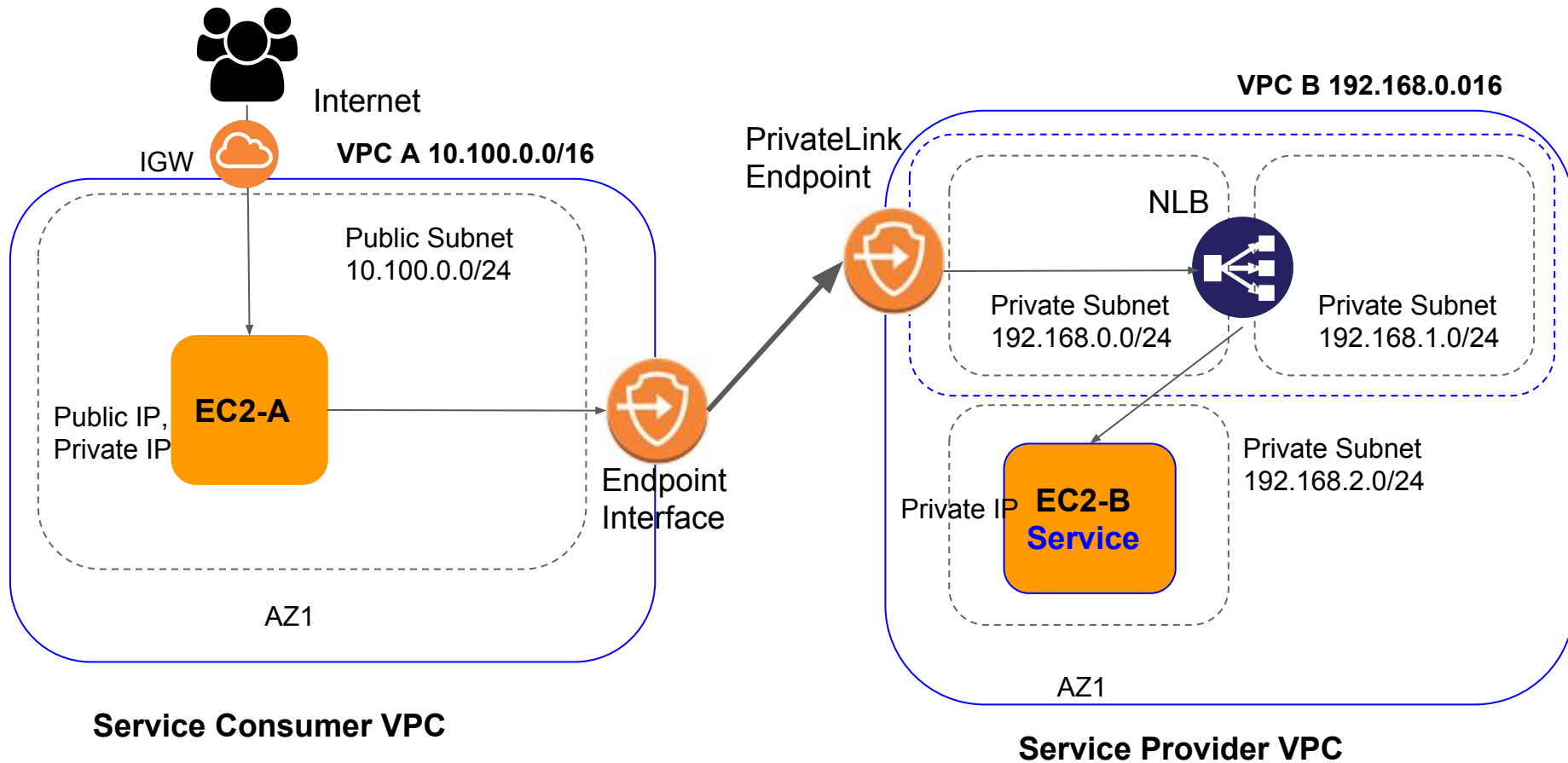


# VPC PrivateLink



# VPC PrivateLink



# Prerequisites for this exercise

**Must have pre-configured Web server AMI to launch EC2 instance in Private subnet**

Steps to create AMI

1. Launch EC2 instance (Amazon Linux) in default VPC with following Userdata

```
#!/bin/bash  
yum install httpd -y  
service httpd start  
chkconfig httpd on  
echo "This is my web server" > /var/www/html/index.html  
echo successful
```

2. Verify that Web server is accessible by hitting EC2 Public IP
3. Stop the instance and create Amazon Machine Image (AMI)
4. Terminate EC2 instance

# Steps

1. Create VPC-A in say **Mumbai (ap-south-1)** region. Create and attach IGW.
2. In VPC-A, create Public Subnet VPC-A-Public and launch EC2-A instance with Public IP. Open Security group port 22 for MyIP
3. Create VPC-B in same region
4. In VPC-B, create 2 Private subnets (VPC-B-NLB-1 and VPC-B-NLB-2) across AZs for hosting Network Load Balancer
5. In VPC-B, create one more Private subnet VPC-B-Private in AZ-1
6. In VPC-B private subnet VPC-B-Private, launch EC2-B instance from **pre-configured AMI** (web server). Open security group for HTTP port 80 for NLB subnets
7. In VPC-B, create Network Load Balancer in VPC-B-NLB-1 and VPC-B-NLB-2 subnets, create target group and add EC2-B instance to target group
8. In VPC-B, create **VPC Endpoint Service** with target as Network Load Balancer
9. In VPC-A, create **VPC Endpoint interface** for the VPC Endpoint service created above
  - Make sure you create endpoint interface in same AZ as VPC-A public subnet
  - Also create security group for VPC endpoint interface to allow HTTP traffic from VPC-A CIDR
10. Login to VPC-A EC2 instance over SSH and access VPC Endpoint DNS using curl command  
\$curl <vpc endpoint dns>
11. You should be able to get the response from EC2-B webserver