NAT Gateway and VPC Peering (Cross-Region) Setup in AWS

Step 1: Create a NAT Gateway

1.1 Allocate an Elastic IP (EIP)

- Go to the **AWS Management Console**.
- Navigate to **EC2 Dashboard**.
- In the left-hand menu, select **Elastic IPs**.
- Click **Allocate Elastic IP address**.
- Choose **Amazon's pool of IPv4 addresses**.
- Click **Allocate** and note the **Elastic IP**.

1.2 Create a NAT Gateway

- Navigate to **VPC Dashboard**.
- In the left-hand menu, select **NAT Gateways**.
- Click **Create NAT Gateway**.
- Select the **public subnet** where the NAT Gateway will be deployed.
- Attach the **Elastic IP** that was allocated earlier.
- Click **Create NAT Gateway** and wait for it to become available.

1.3 Update the Route Table for Private Subnet

- Navigate to **VPC Dashboard > Route Tables**.
- Identify the **Route Table** associated with your **private subnet**.
- Click on the **Routes** tab and then **Edit Routes**.
- Click **Add Route** and enter:
  - **Destination**: `0.0.0.0/0`
  - **Target**: Select the **NAT Gateway** created earlier.
- Click **Save Routes**.

1.4 Verify NAT Gateway Functionality

- Launch a private EC2 instance in the **private subnet**.
- SSH into the private instance using a bastion host or AWS Systems Manager Session Manager.
- Run the following command to check internet access:

  ```
  curl -I https://www.google.com
  ```

  - If the response includes **HTTP/1.1 200 OK**, NAT Gateway is working correctly.
  - If the request fails, ensure that the correct **route table** is associated with the private subnet and **security groups** allow outbound traffic.

Step 2: Create a VPC Peering Connection (Cross-Region)

2.1 Create VPC Peering Request (Requester VPC)

- Navigate to **VPC Dashboard** in **Region A**.
- Click **Peering Connections** in the left-hand menu.
- Click **Create Peering Connection**.
- Provide the following details:
  - **Peering connection name**: `Peering-RegionA-RegionB`
  - **VPC (Requester)**: Select the **VPC in Region A**.
  - **VPC (Accepter)**: Choose **Another account** if the target VPC is in a different AWS account, otherwise choose **My account**.
  - **Region**: Select **Region B**.
  - Enter the **VPC ID** of the target VPC in Region B.
- Click **Create Peering Connection**.
- Wait for the status to show **Pending Acceptance**.

2.2 Accept Peering Request (Accepter VPC)

- Navigate to **VPC Dashboard** in **Region B**.
- Click **Peering Connections**.
- Locate the **Peering Connection** with **Pending Acceptance**.
- Select the peering request and click **Accept Request**.
- The status should now change to **Active**.

2.3 Update Route Tables for Peering

In Region A (Requester VPC)

- Navigate to **VPC Dashboard > Route Tables**.
- Select the **Route Table** associated with the **subnets** that need access to the other VPC.
- Click **Edit Routes**.
- Click **Add Route** and enter:
  - **Destination**: CIDR block of the **Accepter VPC**.
  - **Target**: Select the **Peering Connection** created earlier.
- Click **Save Routes**.

In Region B (Accepter VPC)

- Navigate to **VPC Dashboard > Route Tables**.
- Select the **Route Table** associated with the **subnets** in Region B.
- Click **Edit Routes**.
- Click **Add Route** and enter:
  - **Destination**: CIDR block of the **Requester VPC**.
  - **Target**: Select the **Peering Connection**.
- Click **Save Routes**.

2.4 Verify VPC Peering Connectivity

- Launch an **EC2 instance** in both VPCs (one in Region A and one in Region B).
- Ensure both instances have **Security Groups** that allow ICMP (ping) and SSH from each other.
- Connect to the instance in **Region A** and try pinging the private IP of the instance in **Region B**:

```
ping <PRIVATE_IP_OF_INSTANCE_IN_REGION_B>
```

- If the ping succeeds, VPC Peering is configured correctly.
- If it fails, check:
  - **Route Tables** have the correct entries.
  - **Security Groups** allow inbound ICMP traffic.
  - **Network ACLs** are not blocking traffic.
