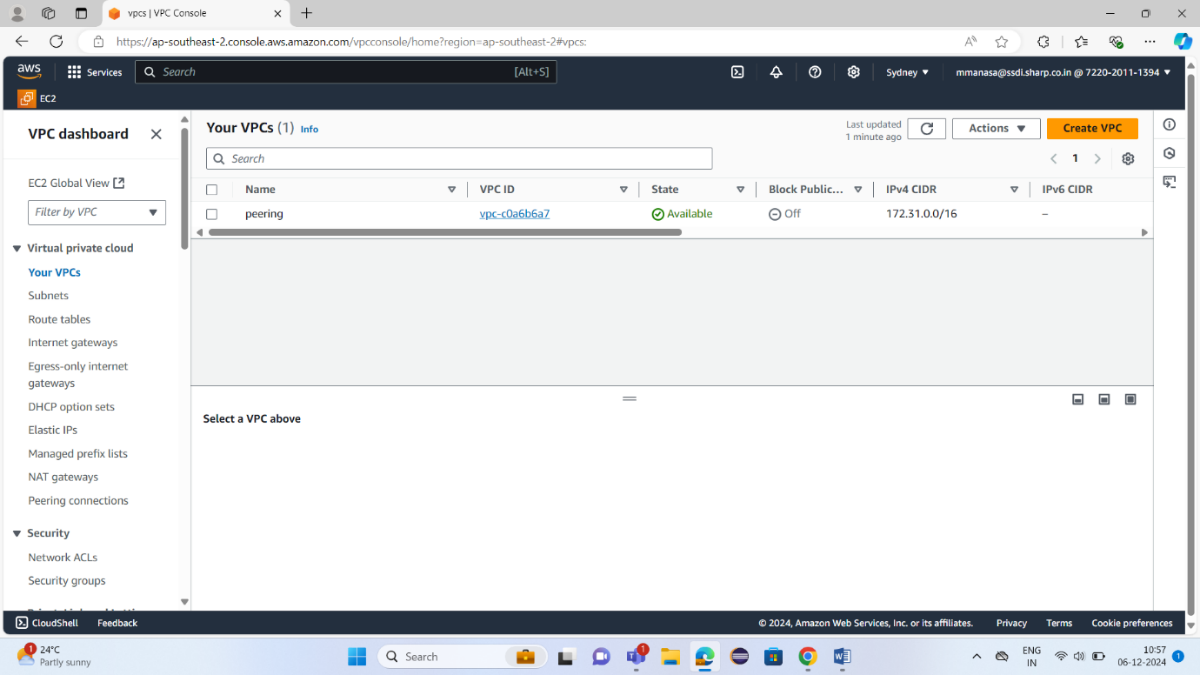
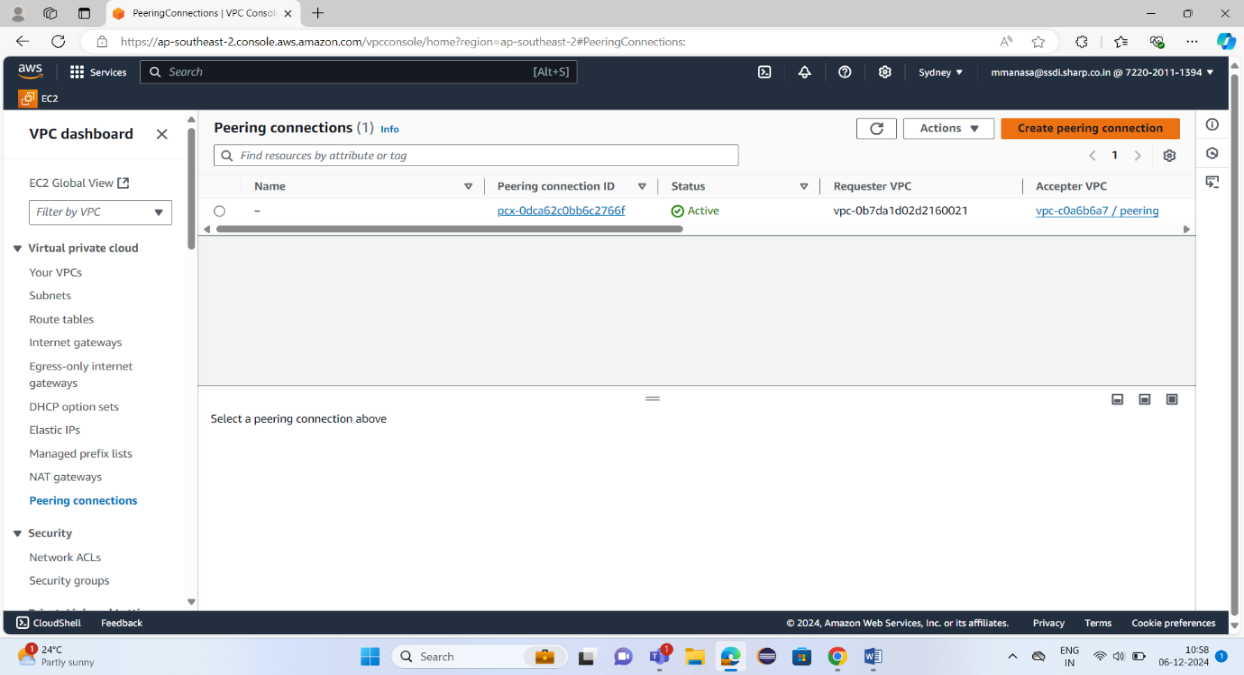
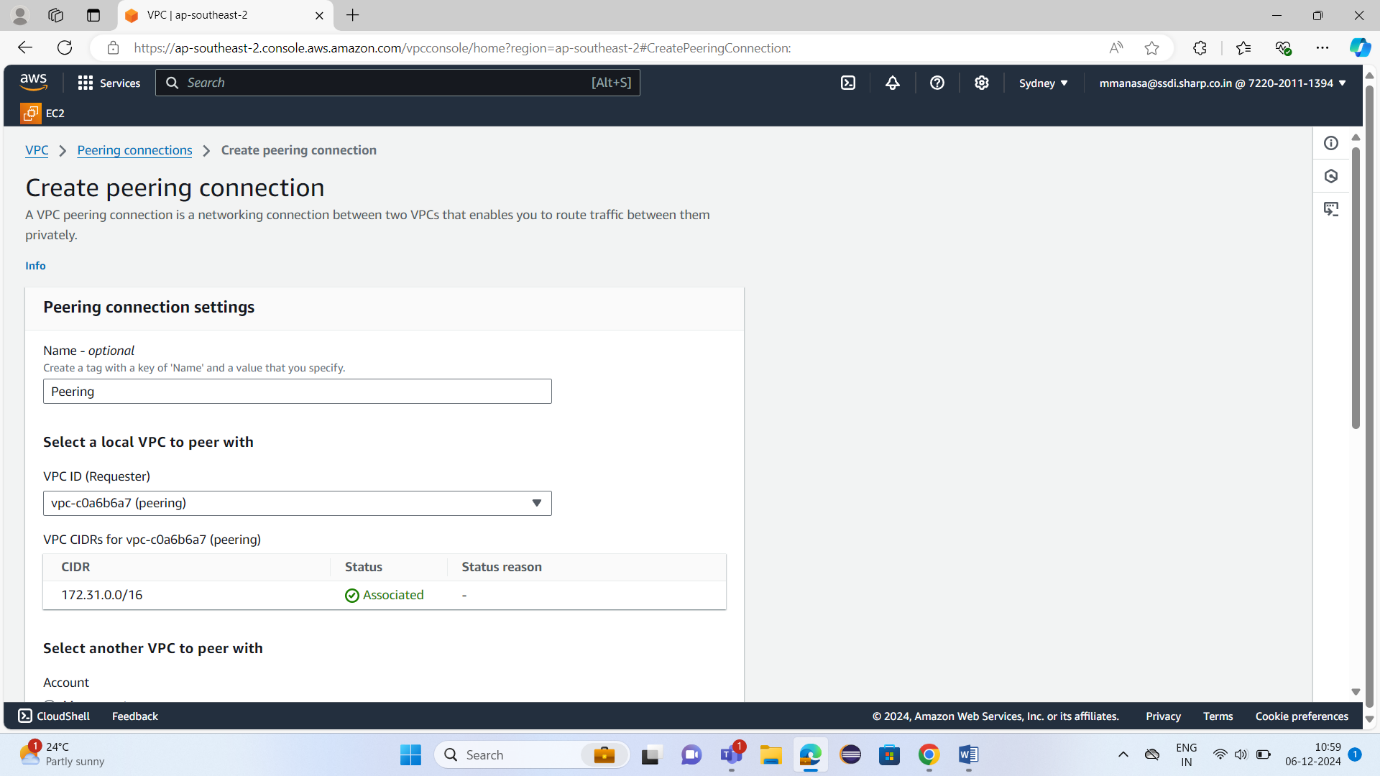
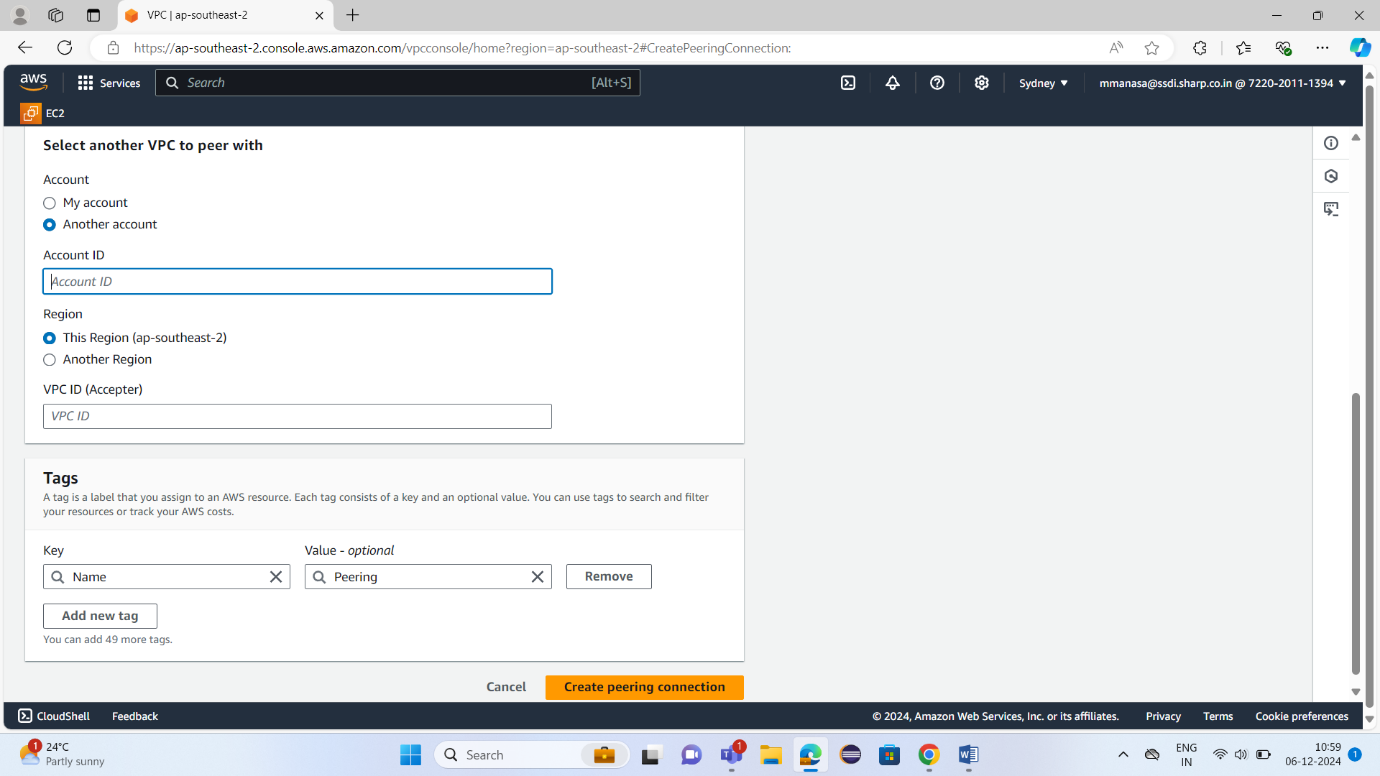
**VPC PEERING**

**Prerequisites:** Ensure both VPCs use non-overlapping CIDR blocks

Go to VPC dashboard and check ipv4 CIDR if both account have same CIDR range consider changing any one CIDR.

**Step 1: Create a VPC Peering Connection**

1. **Log into Account A**
2. Go to the **VPC Dashboard** you can see the list of VPC’s in your account
3. Under the **Peering Connections** section, click **Create Peering Connection**.
4. Fill in the details:
   * **Peering connection name**: Enter a descriptive name (e.g., VPC-A-to-VPC-B).
   * **VPC ID (Requester)**: Select the VPC in Account A. CIDR will be automatically updated.
   * **Account**: Select **Another account**.
   * **Account ID**: Enter the **Account ID of Account B** (the account you're peering with).
   * **VPC ID (Accepter)**: Go to VPC dashboard of account B to get the VPC ID.
5. Click **Create Peering Connection**.

**Step 2: Accept the Peering Request in Account B**

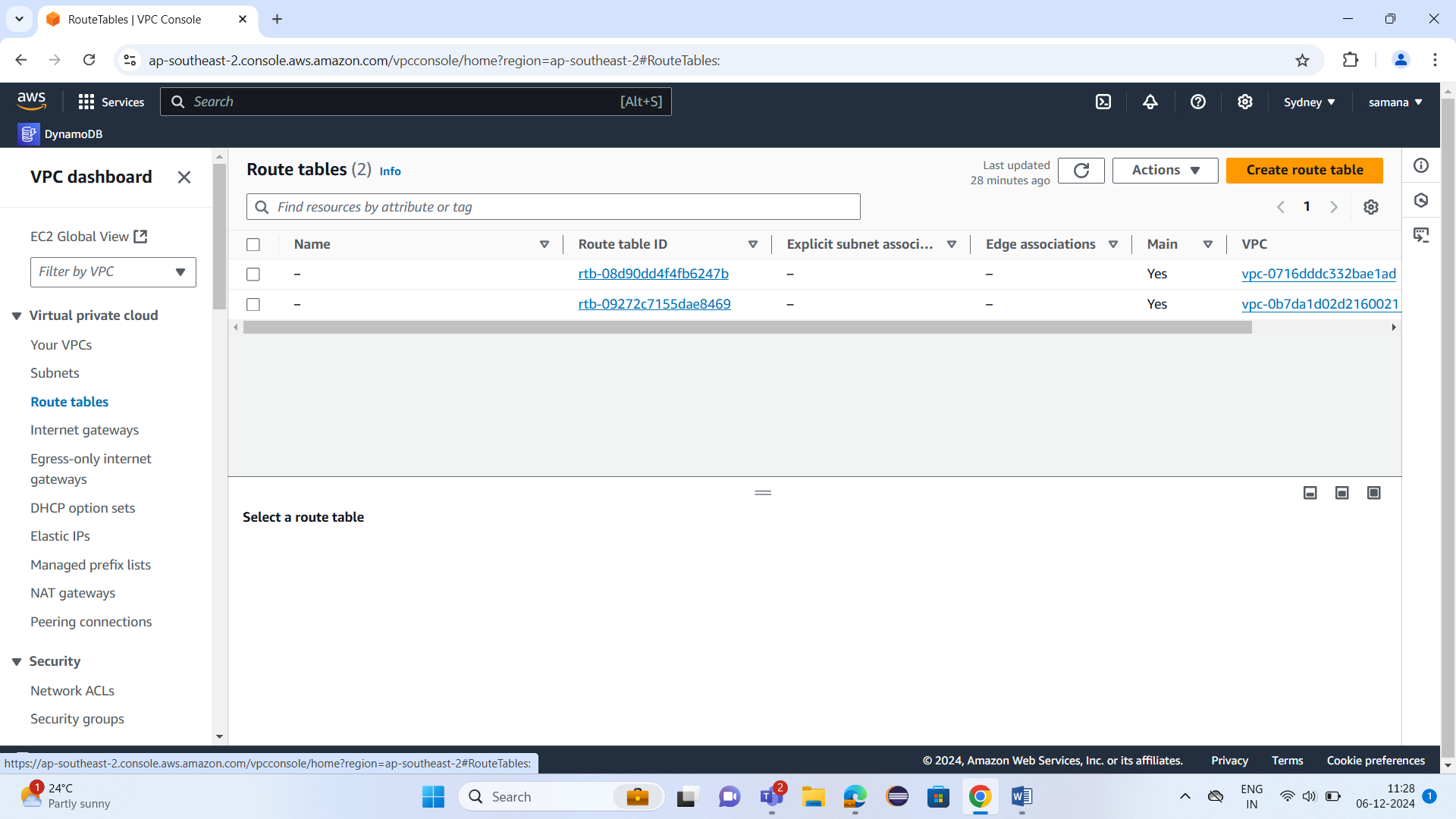
1. **Log into Account B** (the account that will accept the peering request).
2. Go to the **VPC Dashboard**.
3. In the left-hand navigation pane, click on **Peering Connections**.
4. You’ll see the pending peering request from Account A. Select it.
5. Click on **Actions**, then choose **Accept Request**.

**Step 3: Update Route Tables in Both VPCs**

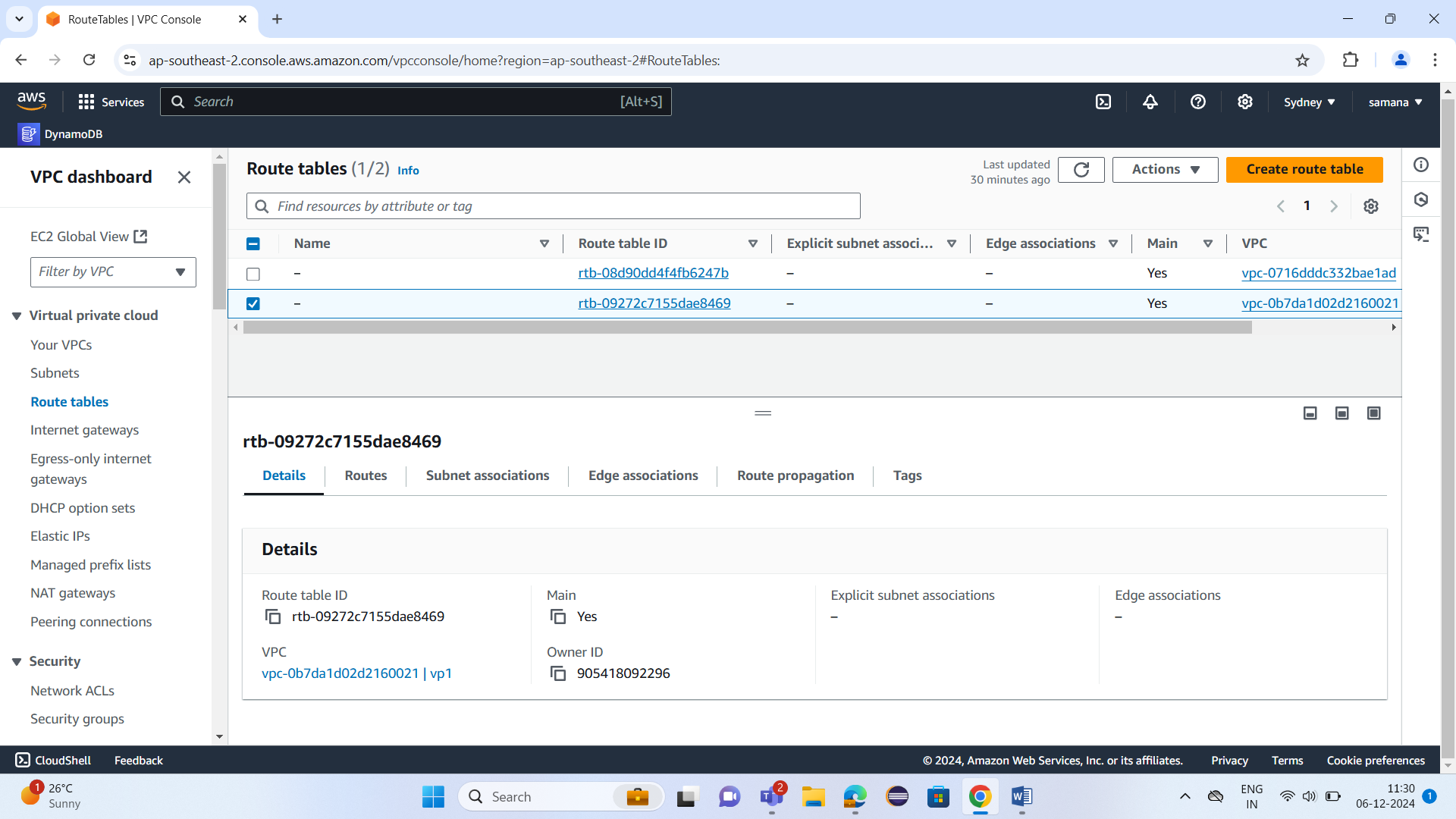
Once the peering connection is accepted, you need to modify the route tables of both VPCs to allow traffic between them.

**In Account A:**

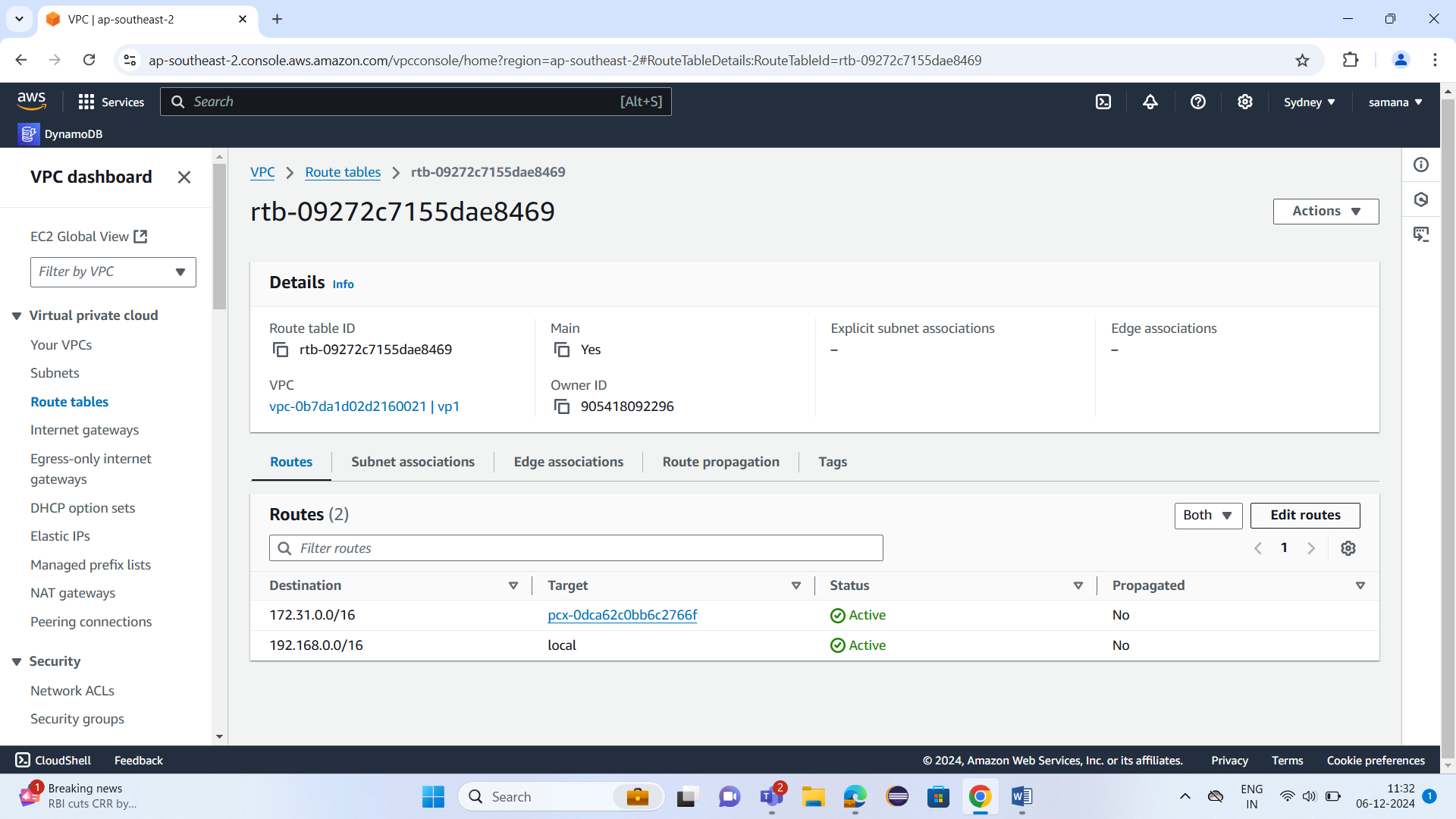
1. Go to the **VPC Dashboard**.
2. In the left pane, click on **Route Tables**.



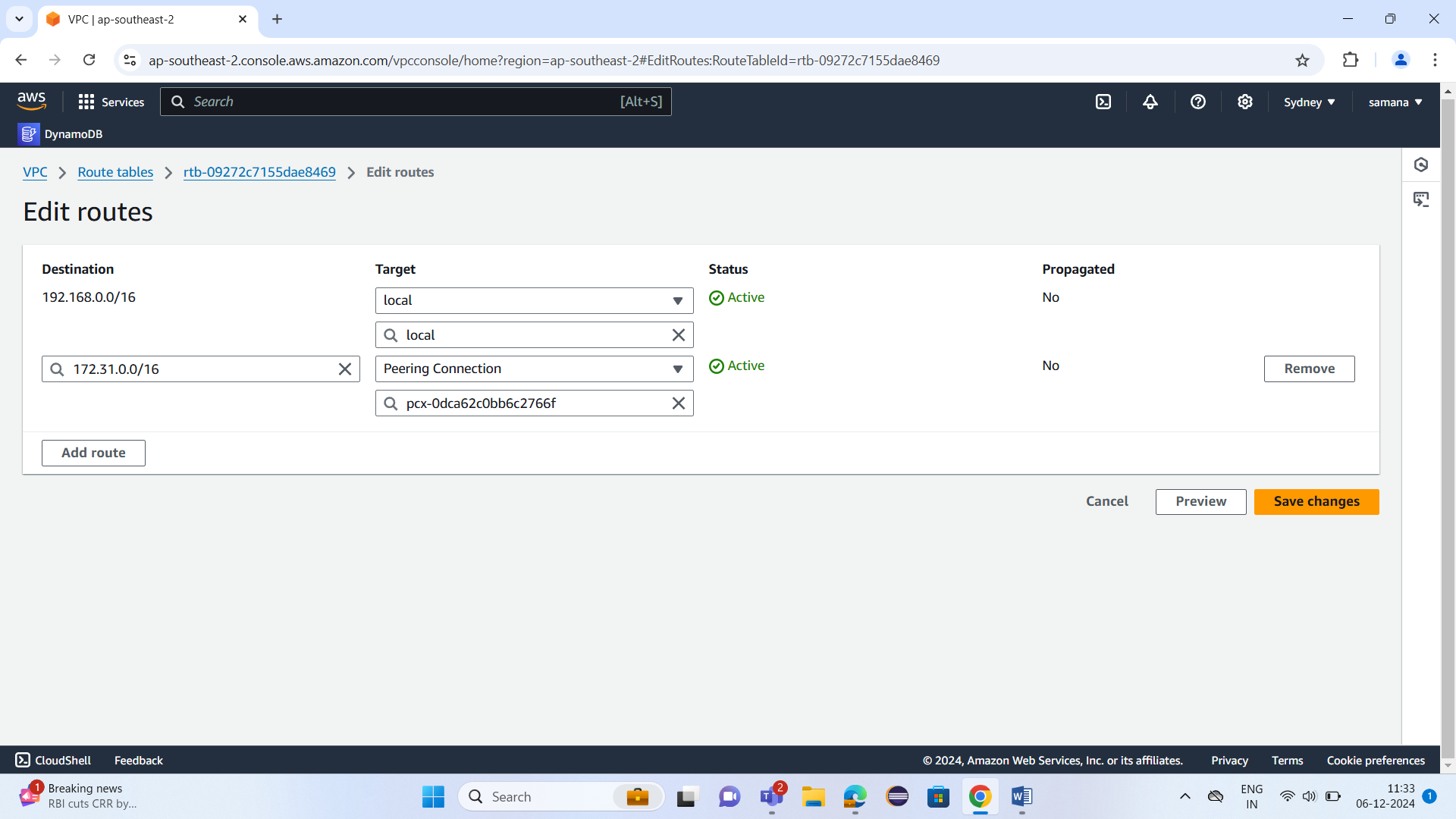
1. Find the route table associated with your VPC in Account A



1. Select the route table and click **Edit routes**.



1. Add a new route:
   * **Destination**: Enter the CIDR block of VPC B (e.g., 172.31.0.0/16).
   * **Target**: Choose the peering connection as target and below that select peering connection id you just created.



1. Click **Save changes**.

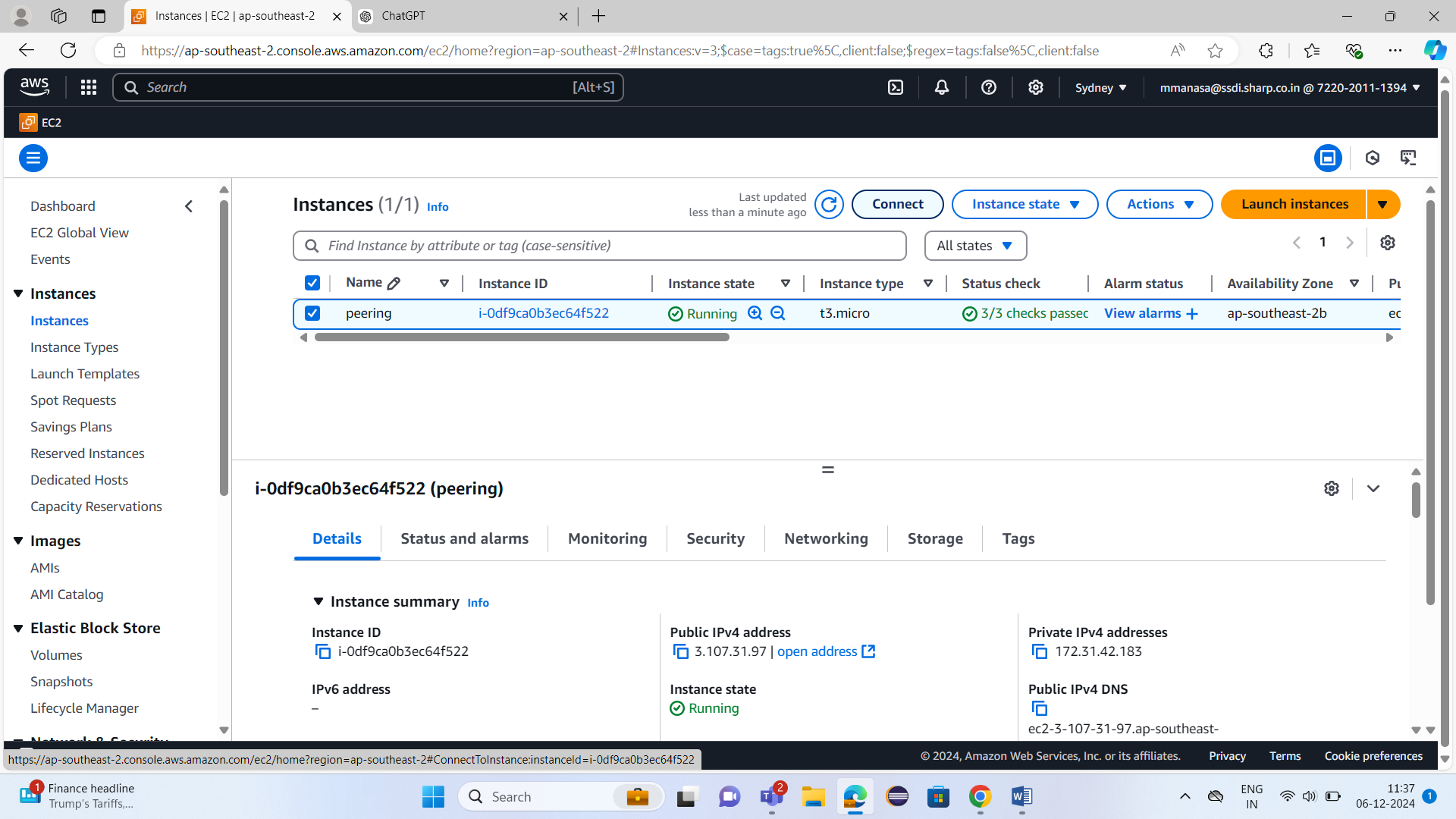
**In Account B:**

1. Go to the **VPC Dashboard** in Account B.
2. In the left pane, click on **Route Tables**.
3. Find the route table associated with your VPC in Account B.
4. Select the route table and click **Edit routes**.
5. Add a new route:
   * **Destination**: Enter the CIDR block of VPC A.
   * **Target**: Choose the peering connection.
6. Click **Save routes**.

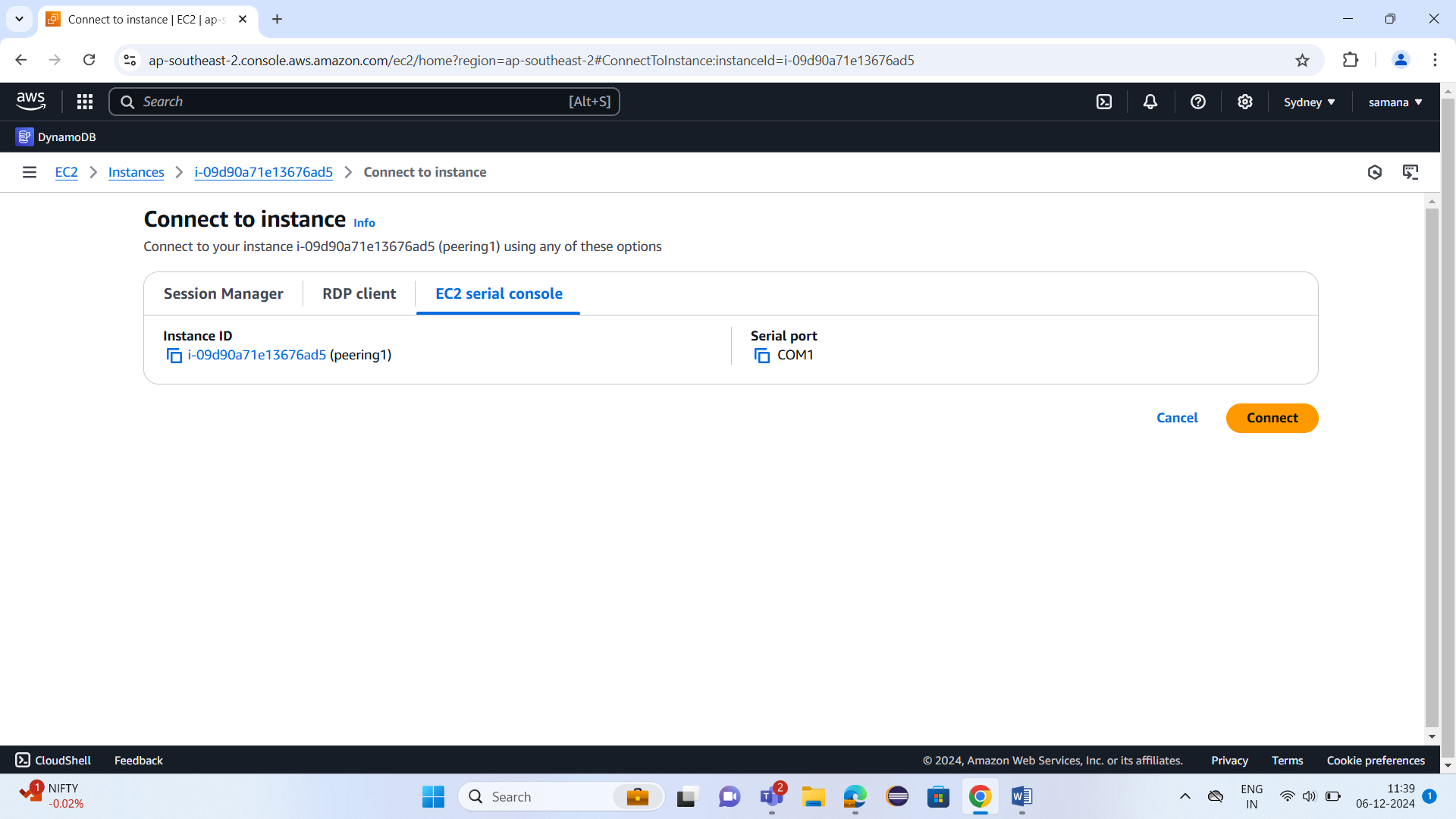
**Step 5: Test the Peering Connection**

Now that the route tables are configured, you should test the peering connection:

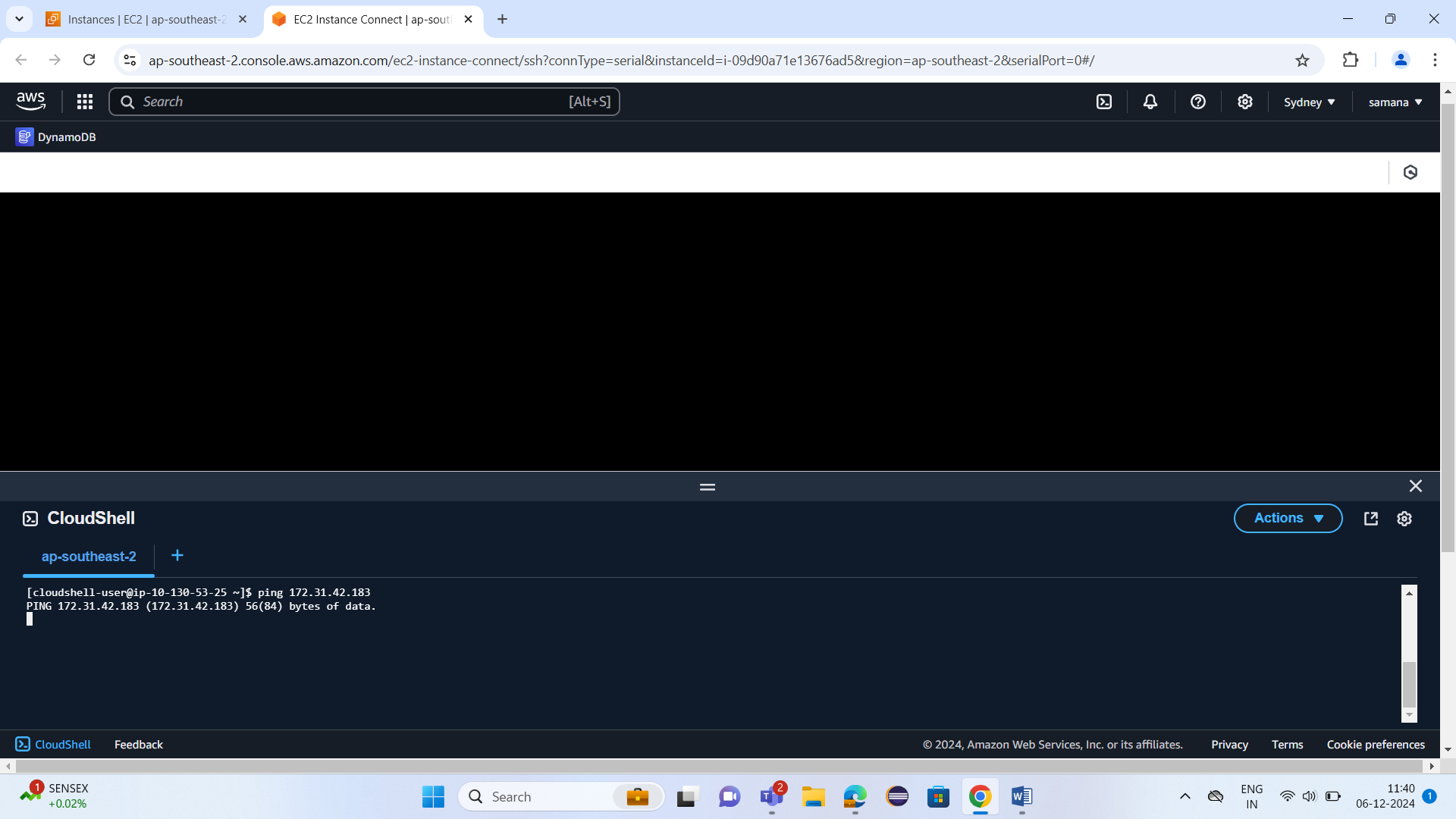
1. Go to the **EC2 instance** and select that EC2 instance you can see connect option.



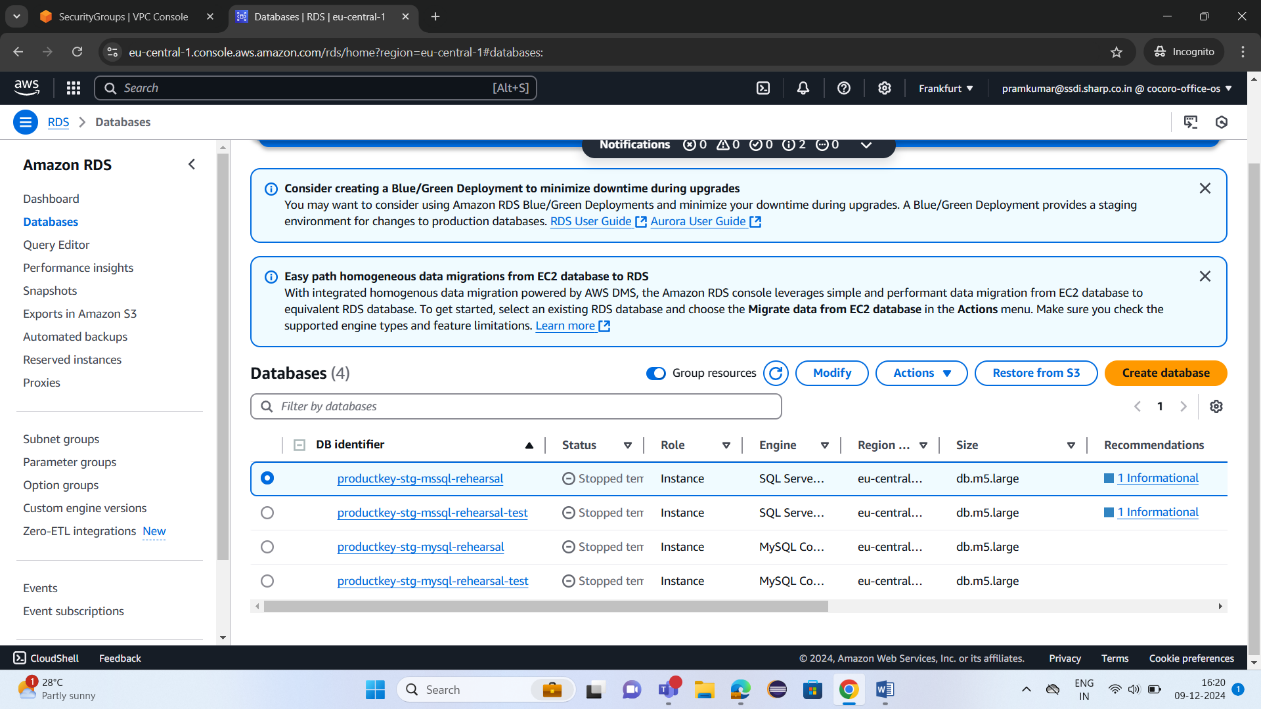
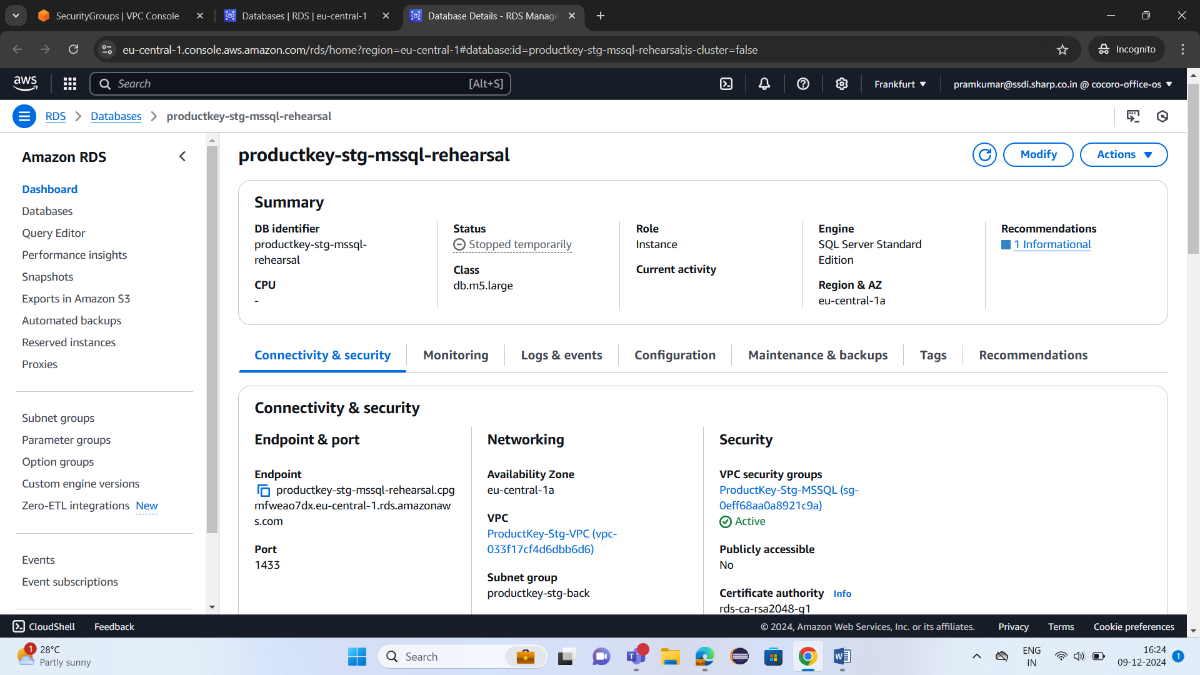
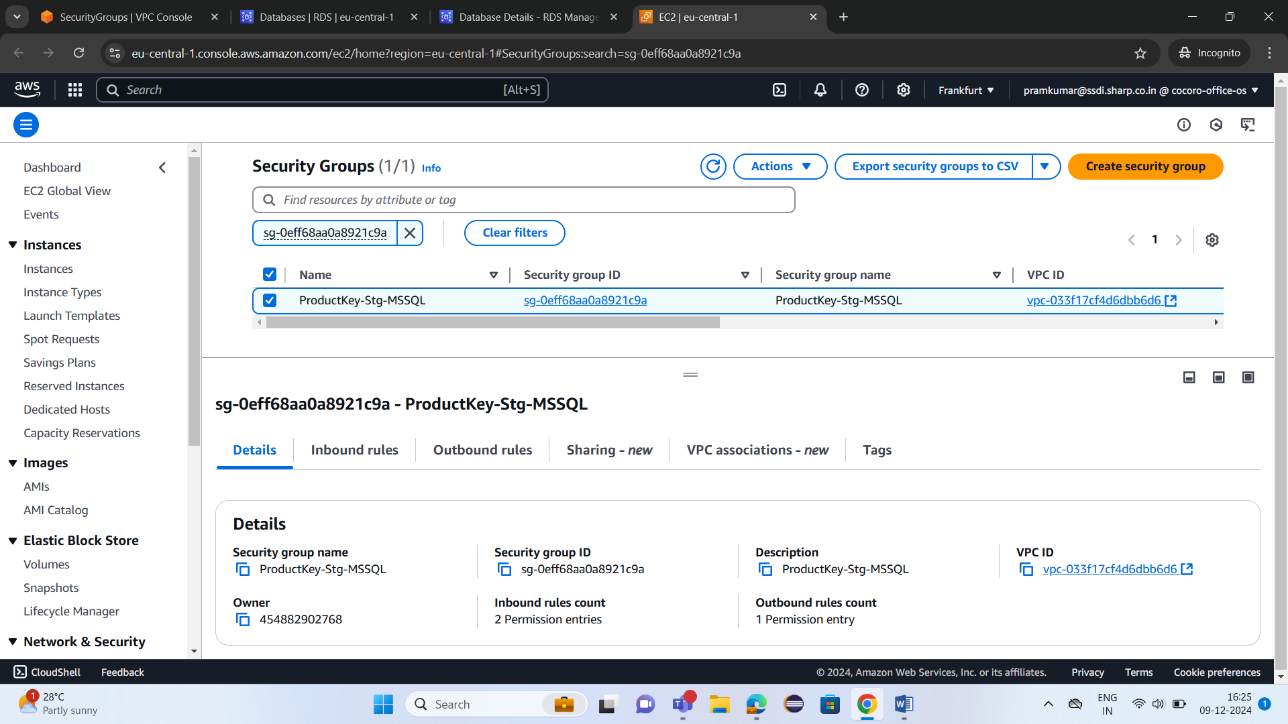
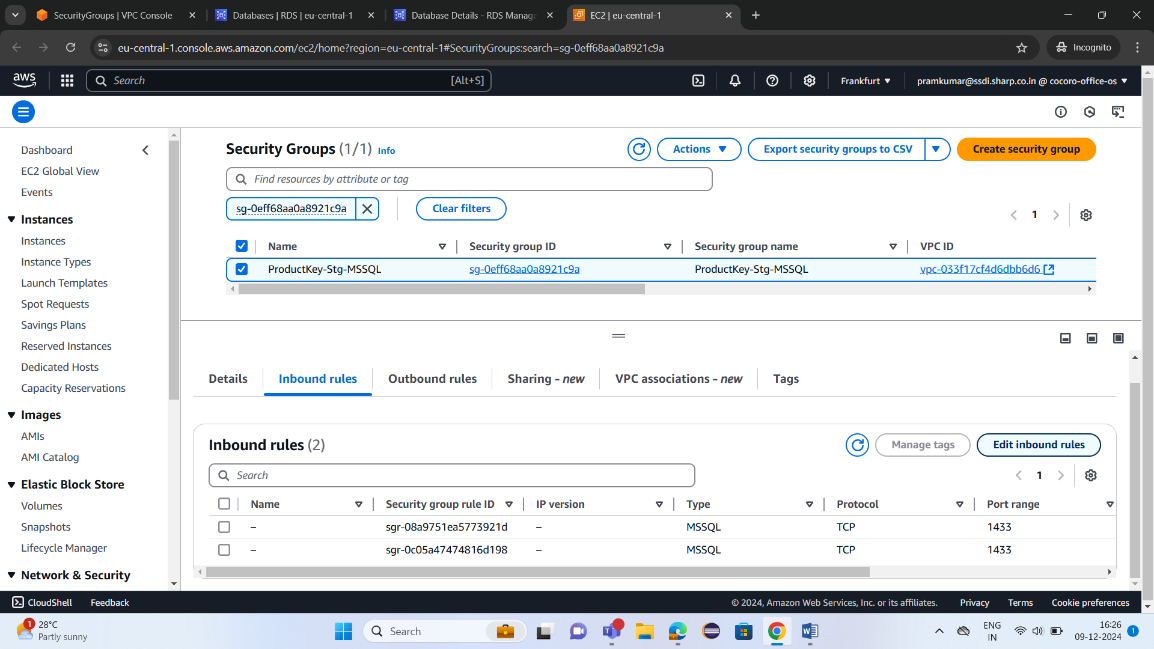
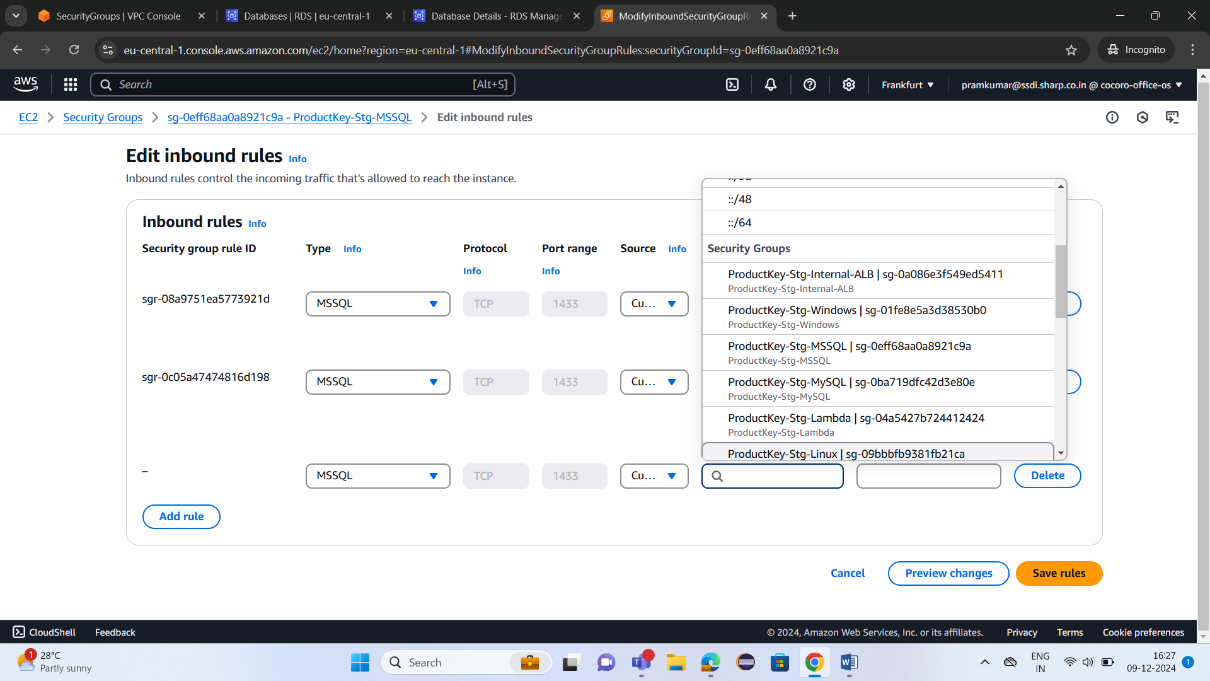
1. When you click on connect you will see the Option of EC2 serial console and only if peering connection is there you can see that connection below. Once you click on connect you can see the console



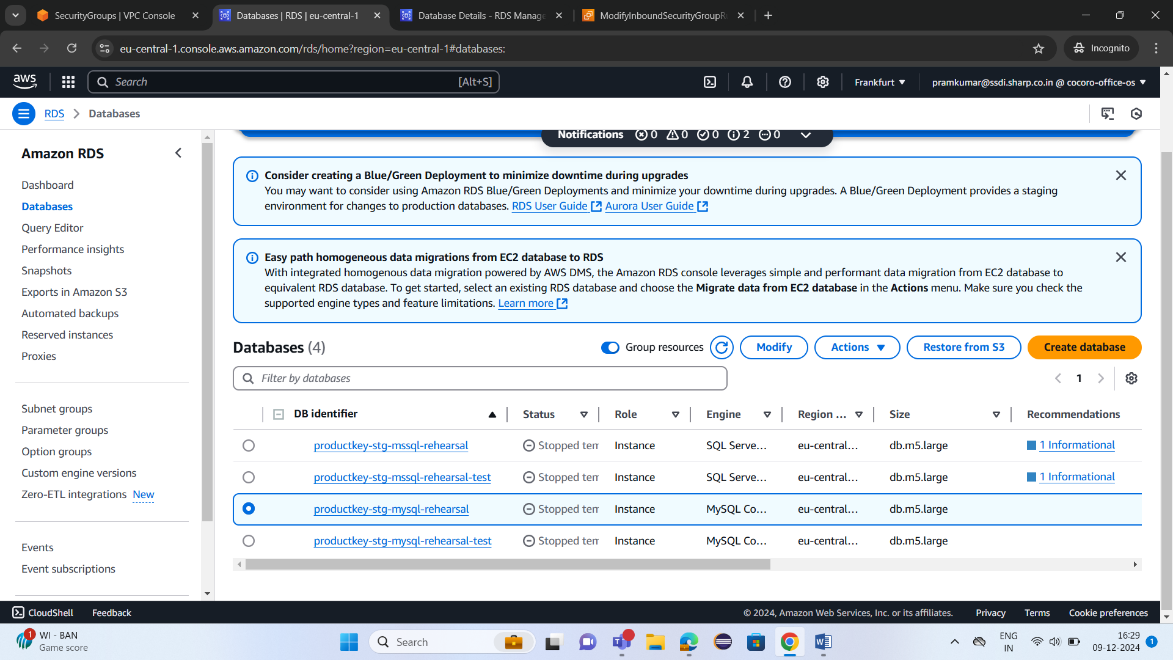
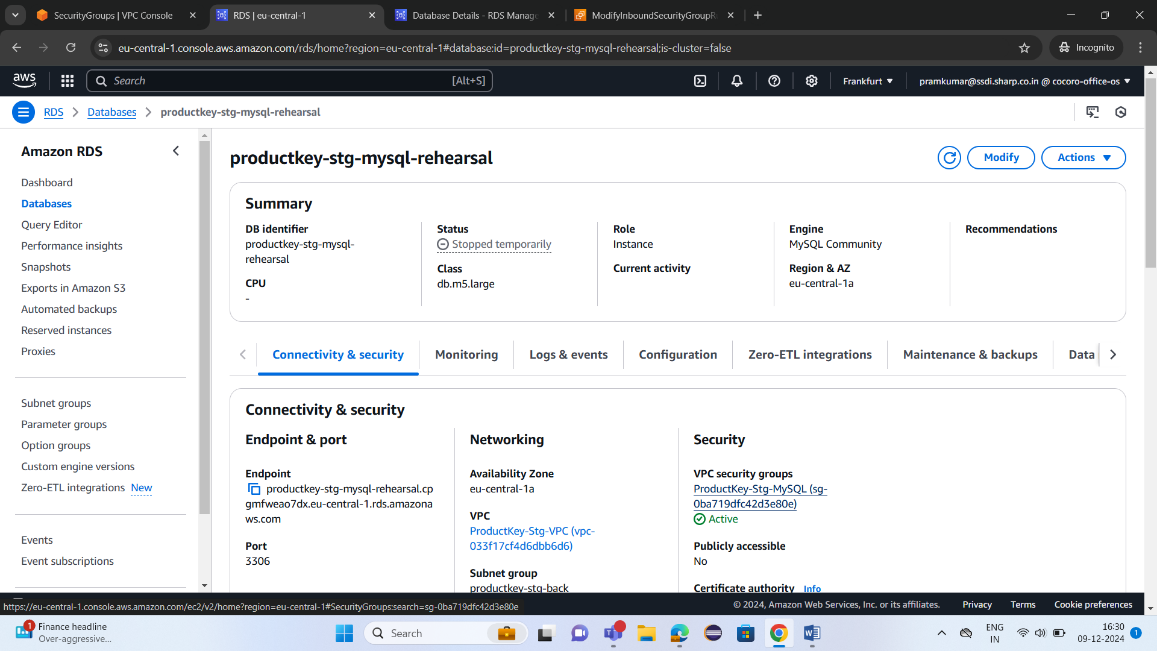
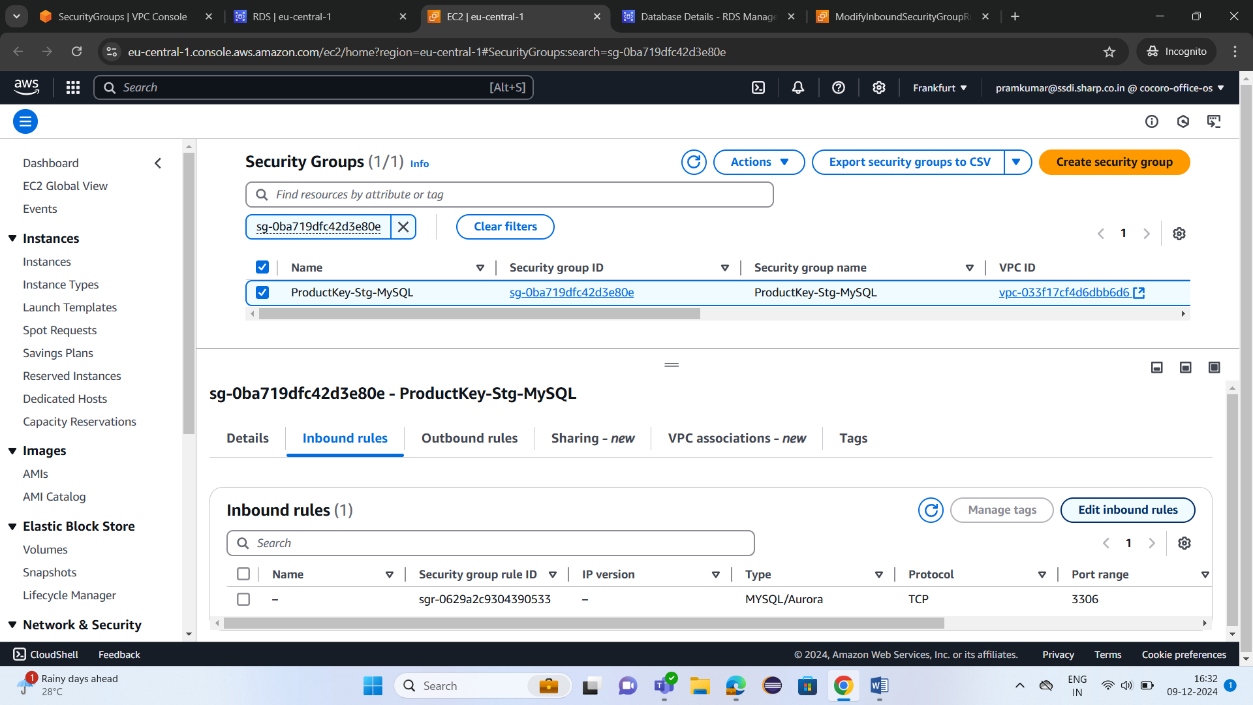
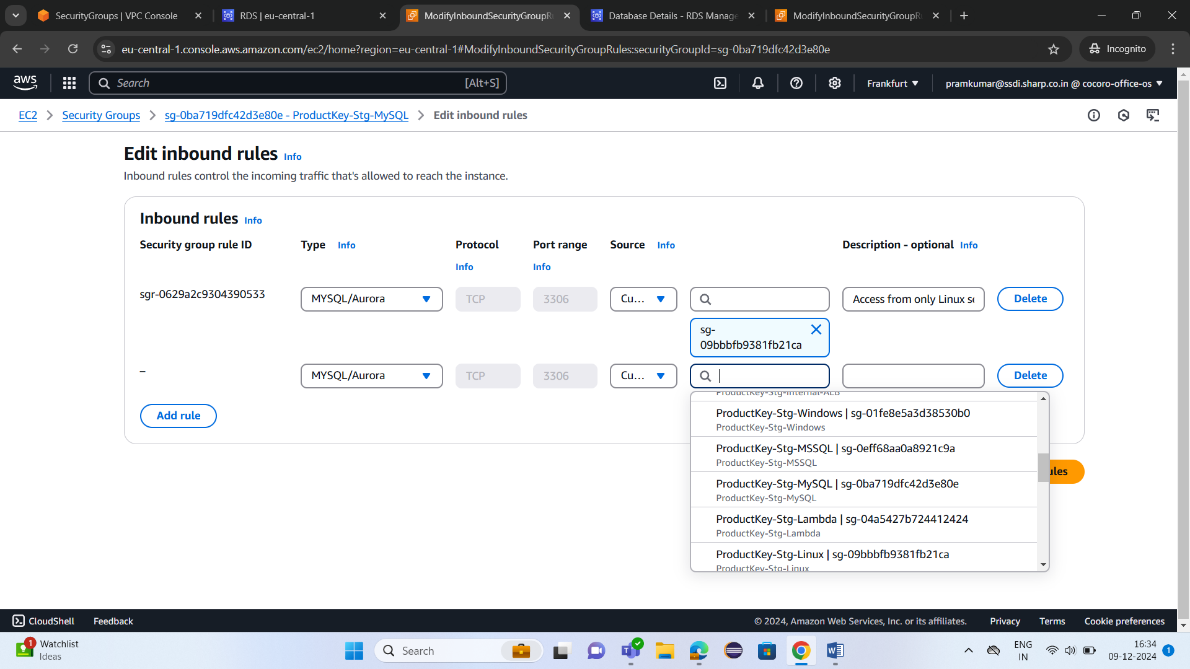
1. Here if you type ping private ip address of the instance you want to connect you can see 56(84) bytes of data if the connection is successful else it shows connection timed out.



**Steps to Allow Communication Between RDS in Account A and EC2 in Account B:**

1. **Access the RDS Dashboard in Account A**
   * Go to the **AWS Management Console** and navigate to **RDS**.
   * Under **Databases**, click on the RDS instance you want to modify (productkey-stg-mssql-rehearsal).
   * This will bring up the dashboard for that RDS instance.
2. **Navigate to Security Groups**
   * In the **RDS Dashboard**, scroll down to the **Security** section.
   * You will see an option for **VPC Security Groups**.
   * Click on the **Security Group ID** listed here. This will redirect you to the **EC2 Security Groups** page.
3. **Edit Inbound Rules of Security Group**
   * In the **Security Groups** tab, select the relevant security group.
   * Go to the **Inbound Rules** tab.
   * Click on **Edit inbound rules**.
4. **Add New Inbound Rule for MSSQL**
   * Click on **Add Rule**.
   * Set the **Type** to **MSSQL**. The **Protocol** and **Port Range** will be automatically set to **TCP** and **1433**.
   * For **Source**, select **Custom** and enter the **CIDR block or Security Group ID** of the VPC in Account B that you want to allow traffic from. If VPC peering is correctly set up, the Security Group associated with the EC2 instance in Account B should be available in the dropdown.
5. **Save the Inbound Rule**
   * Add a description to the rule for clarity
   * Click **Save rules** to apply the changes.
6. **Verify Connectivity**
   * Once the inbound rule is updated, you should be able to access the **MSSQL** database running in the RDS instance from an EC2 instance in **Account B**.

**Steps to Allow Communication Between RDS MySQL in Account A and EC2 in Account B:**

1. **Access the RDS Dashboard in Account A**
   * Log in to the **AWS Management Console** and navigate to **RDS**.
   * Under **Databases**, locate and select the RDS instance for MySQL you want to modify (productkey-stg-mysql-rehearsal).
   * This will open the dashboard for that RDS instance.
2. **Navigate to the RDS Security Groups**
   * In the **RDS Dashboard**, find the **Security** section.
   * You will see a field labeled **VPC Security Groups**. Click on the **Security Group ID** here. This will redirect you to the **EC2 Security Groups** page.
3. **Edit Inbound Rules of the Security Group**
   * In the **Security Groups** tab, find and select the security group associated with your MySQL RDS instance.
   * Go to the **Inbound Rules** tab.
   * Click on **Edit inbound rules**.
4. **Add New Inbound Rule for MySQL**
   * Click on **Add Rule**.
   * Set the **Type** to **MySQL/Aurora**. This will automatically select the correct **Protocol** (TCP) and **Port Range** (3306, which is the default port for MySQL).
   * In the **Source** field, select **Custom** and then specify the **CIDR block or Security Group ID** of the VPC in Account B that you want to allow traffic from. If VPC peering has been successfully set up, the Security Group associated with the EC2 instance in Account B will be available in the dropdown list.
5. **Save the Inbound Rule**
   * Add a **Description** for clarity.
   * Click **Save rules** to apply the changes.
6. **Verify Connectivity**
   * After the inbound rule is updated, verify that the **EC2 instance in Account B** can successfully connect to the **MySQL database** in the RDS instance in Account A.