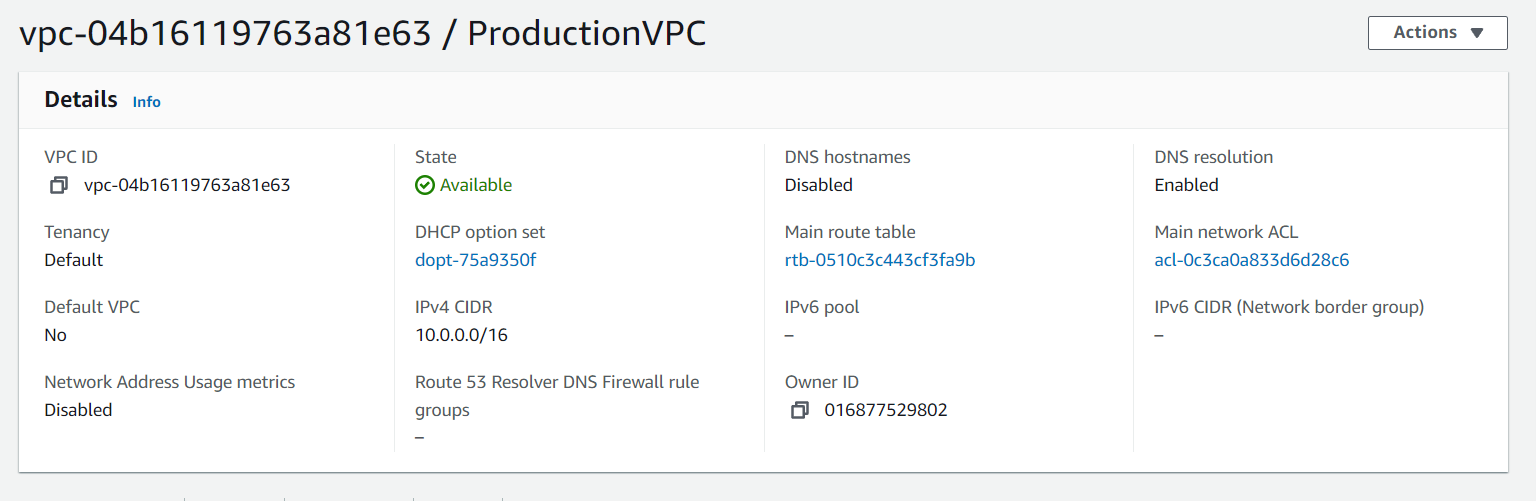
**Production Network:**

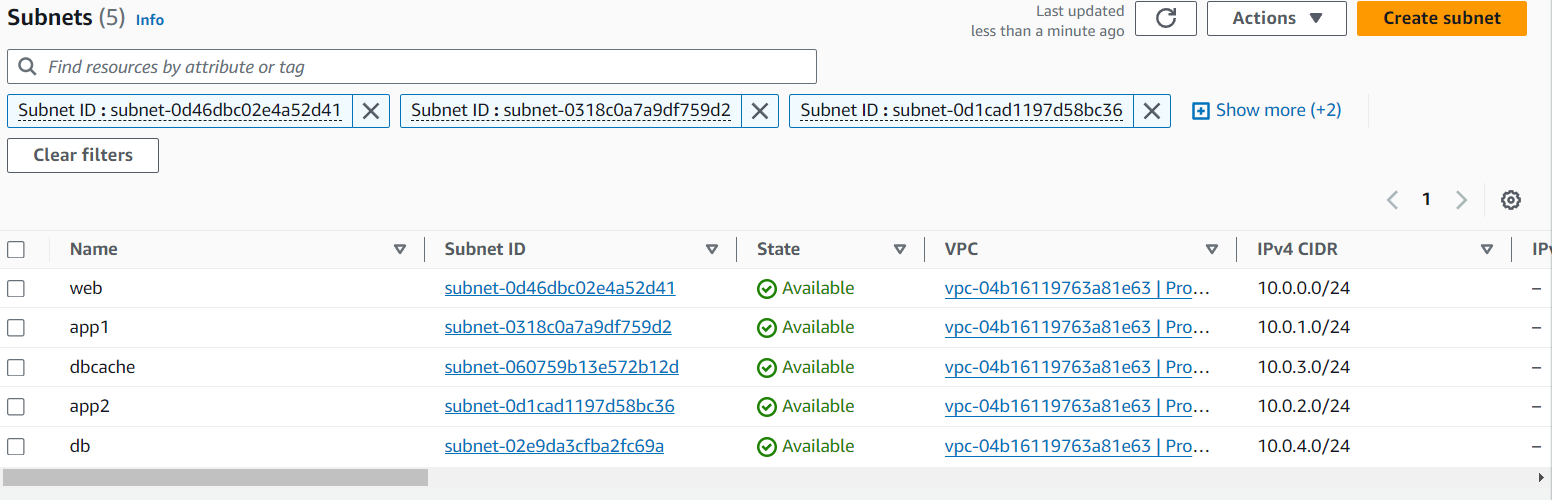
**1. Design and Build a 4-Tier Architecture:**

* **VPC Creation:**
  + Create a VPC named ProductionVPC.
  + Choose an appropriate CIDR block (e.g., 10.0.0.0/16).



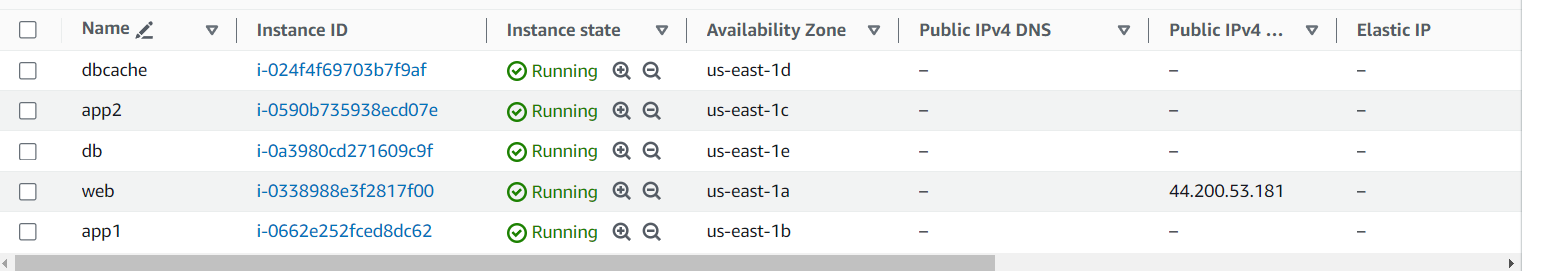
**2. Create Subnets:**

* **Public Subnet:**
  + Create a public subnet named web in ProductionVPC.
  + Example CIDR: 10.0.0.0/24.
* **Private Subnets:**
  + Create four private subnets:
    - app1 with CIDR 10.0.1.0/24.
    - app2 with CIDR 10.0.2.0/24.
    - dbcache with CIDR 10.0.3.0/24.
    - db with CIDR 10.0.4.0/24.



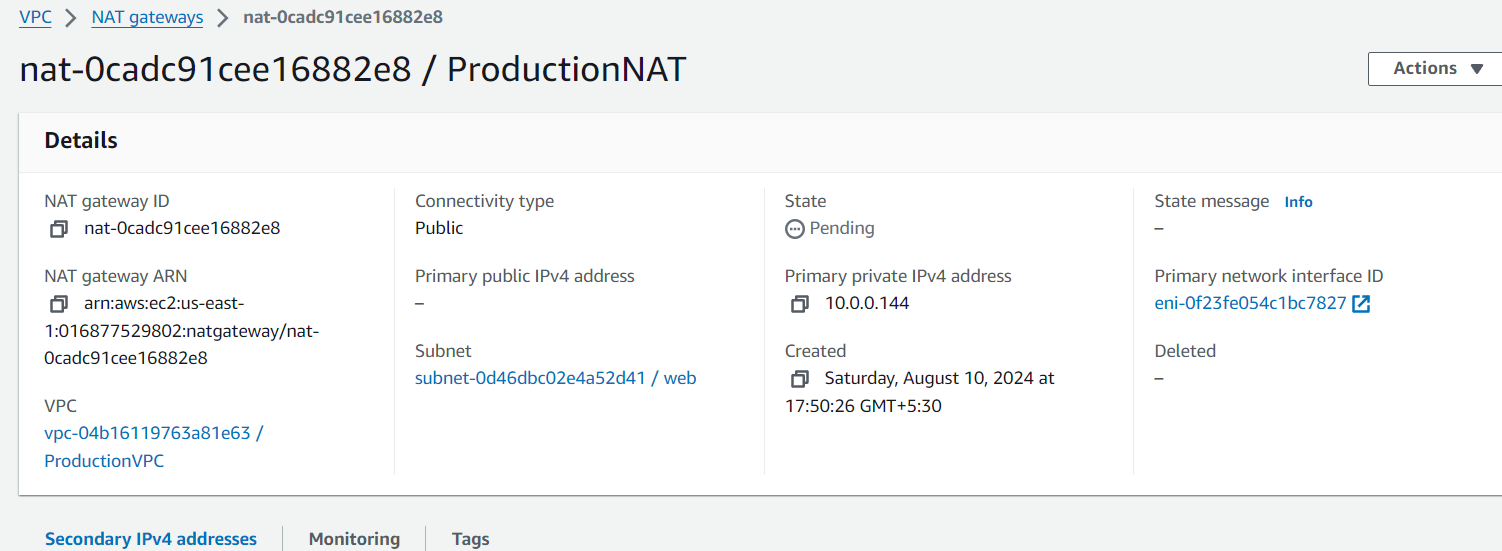
**3. Launch Instances:**

* Launch EC2 instances in each subnet:
  + Name them according to their respective subnets (web, app1, app2, dbcache, db).
* Ensure that instances in private subnets do not have public IPs.

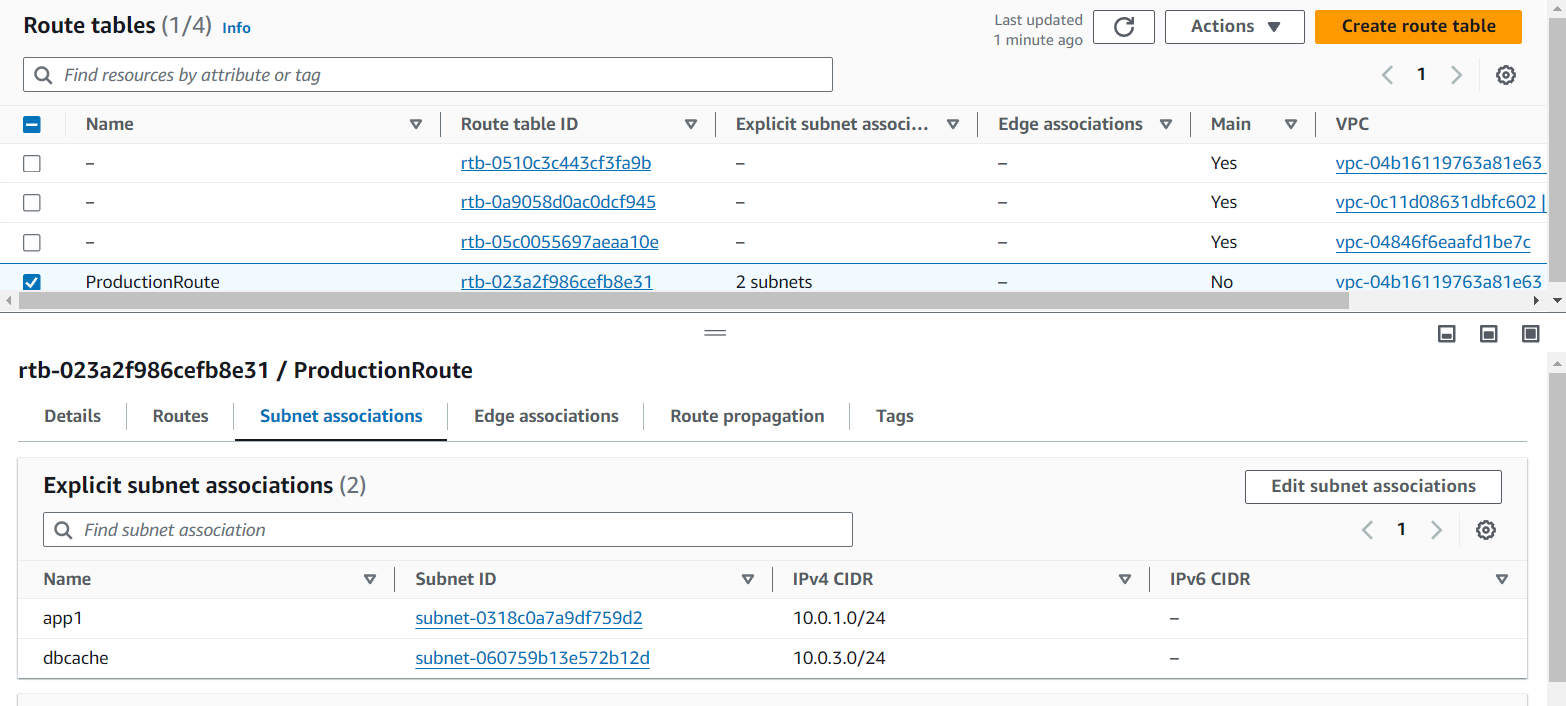


**4. Internet Access:**

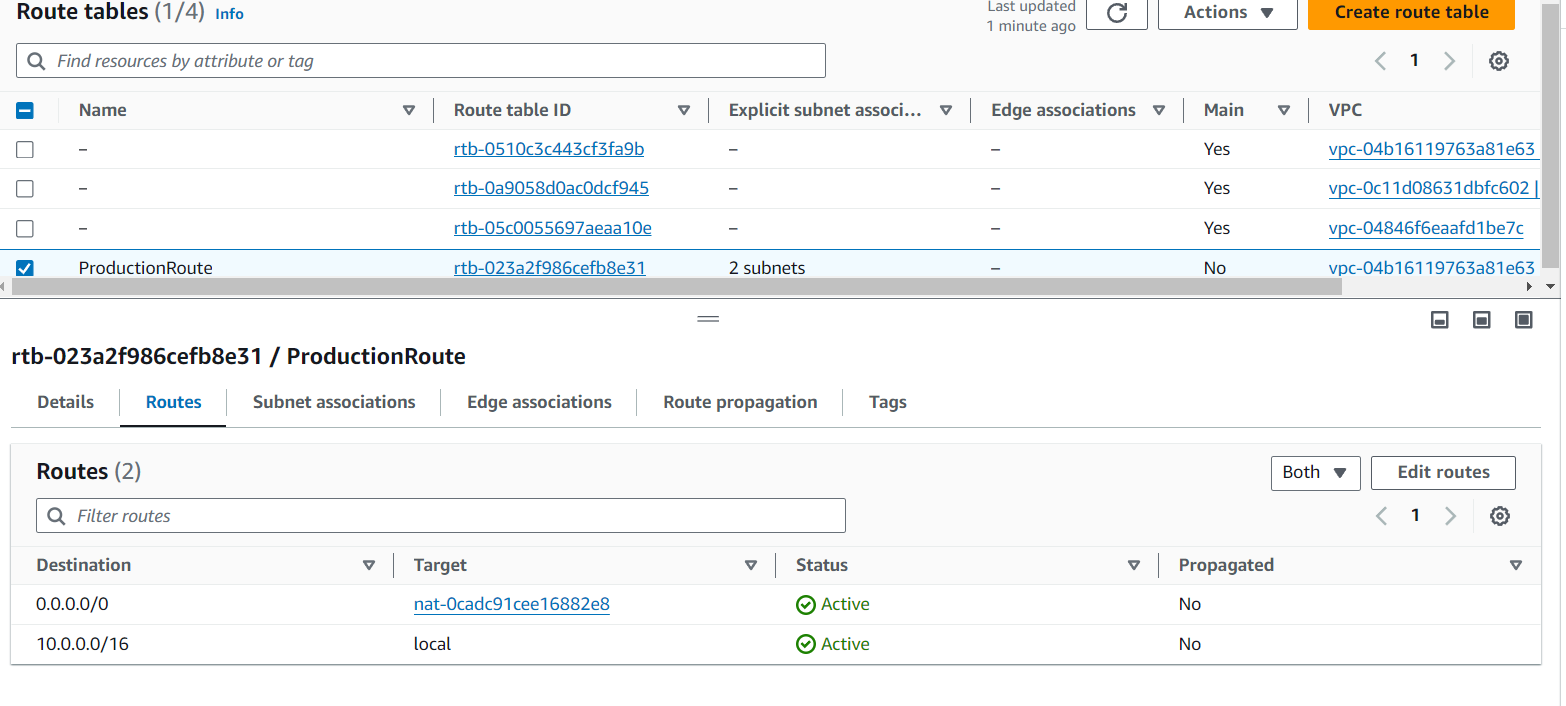
* **NAT Gateway:**
  + Create a NAT Gateway in the web subnet.



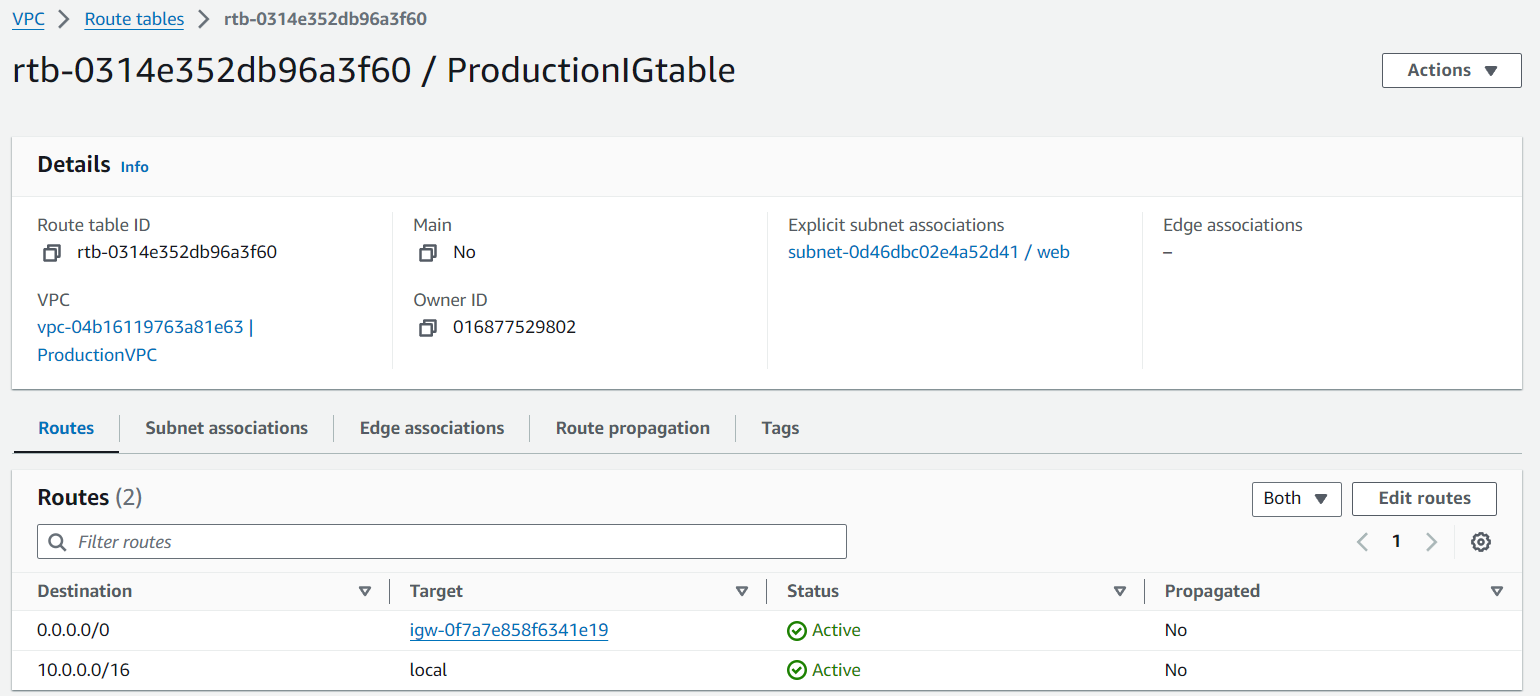
* + Associate the NAT Gateway with the route tables of the app1 and dbcache subnets.



* **Route Tables:**
  + Update route tables:
    - app1 and dbcache should have a route to the NAT Gateway.

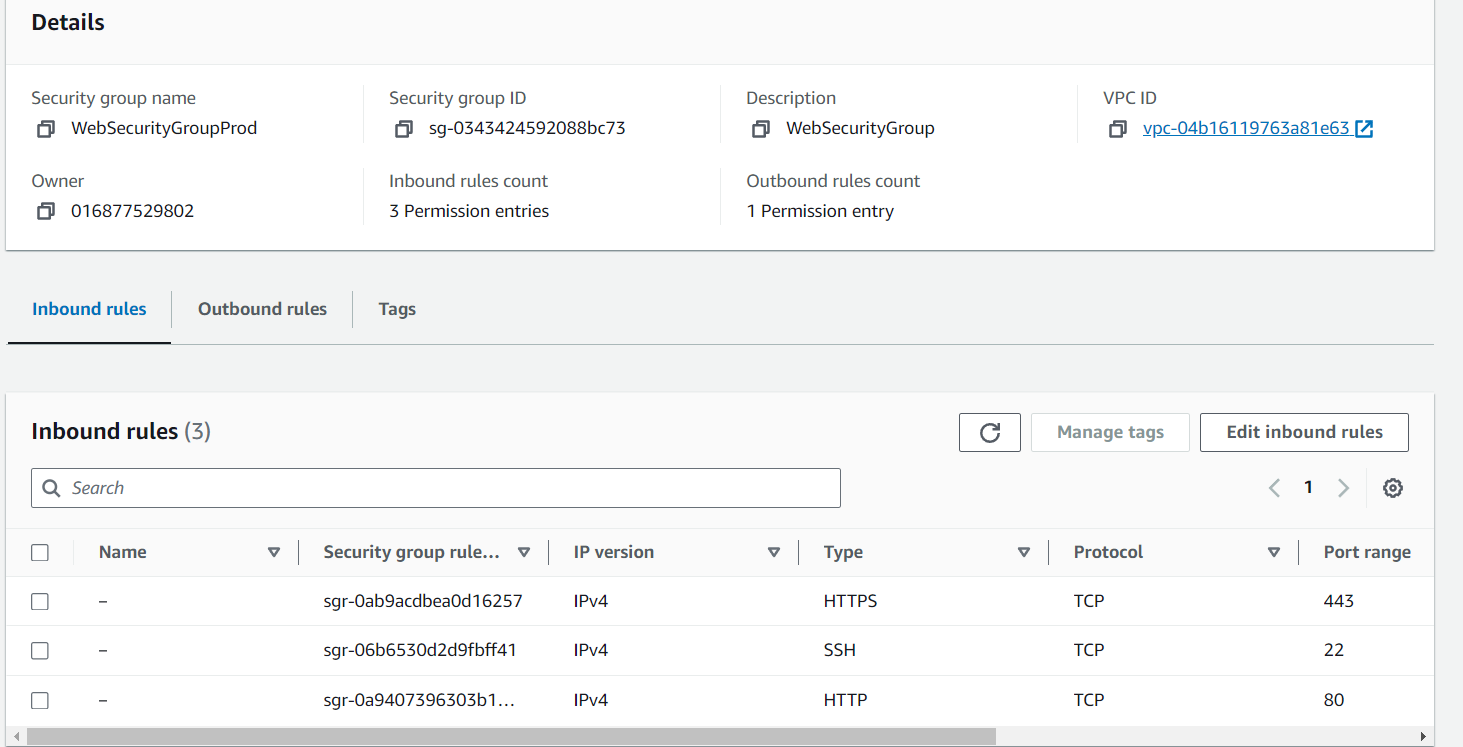


* + - web should have a route to the Internet Gateway.

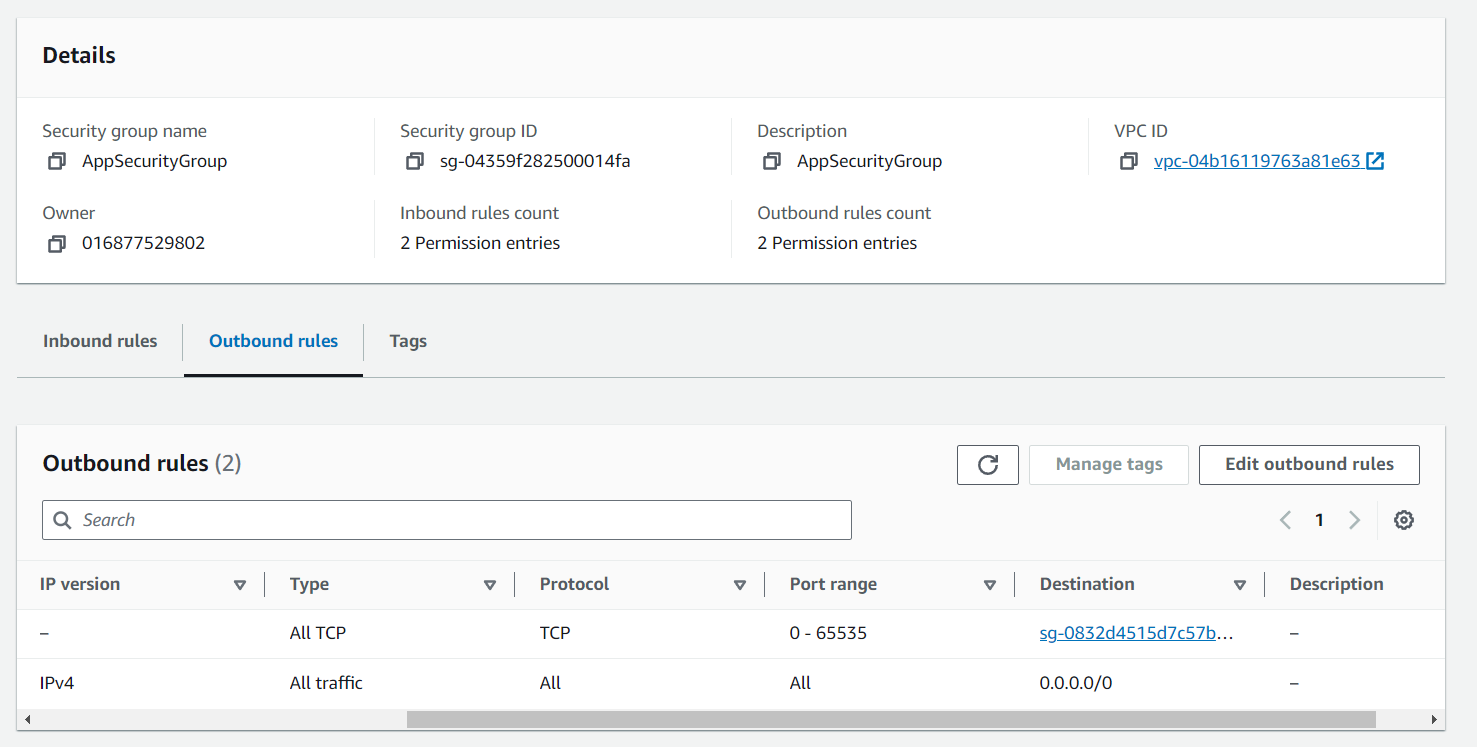


**5. Manage Security Groups and NACLs:**

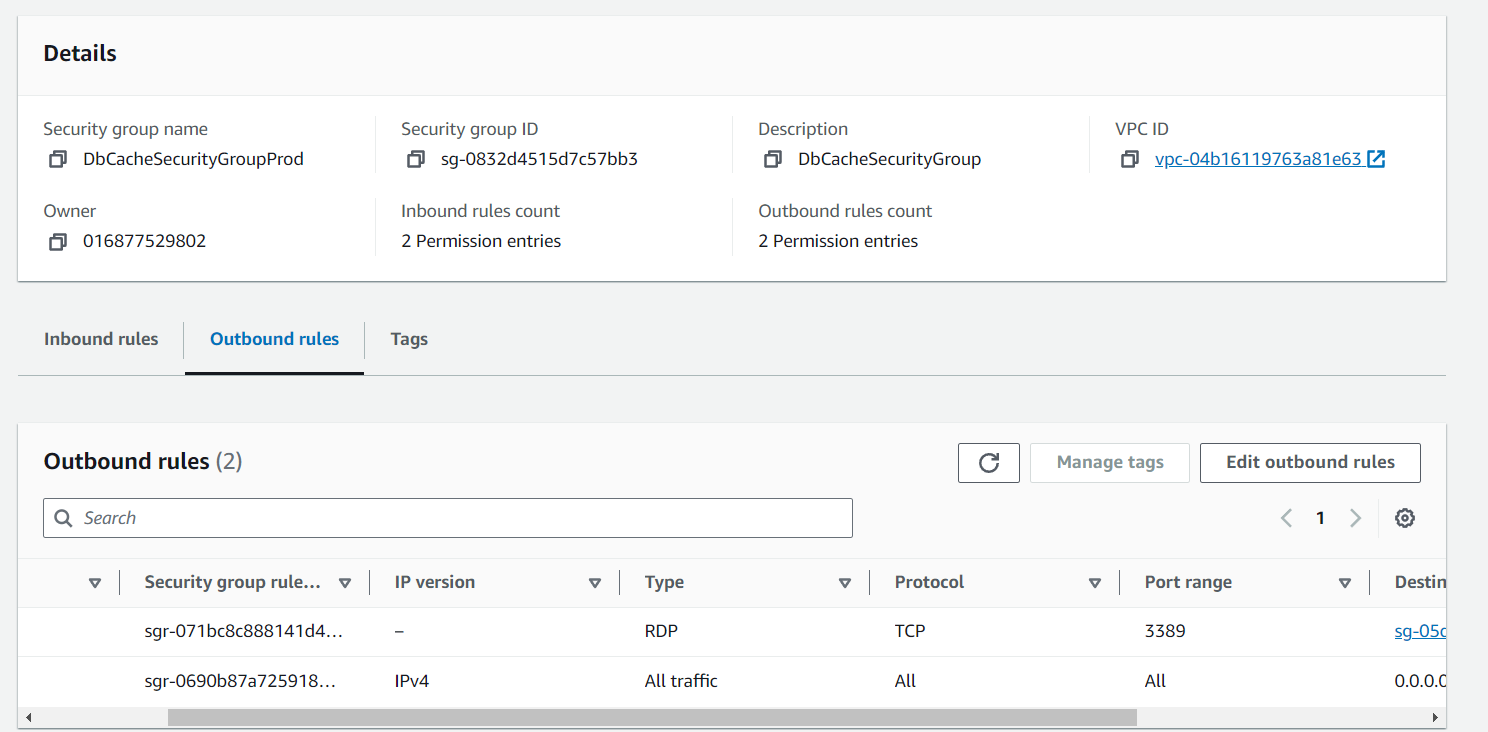
* **Security Groups:**
  + web: Allow inbound HTTP/HTTPS, and SSH access. Restrict outbound to necessary ports.

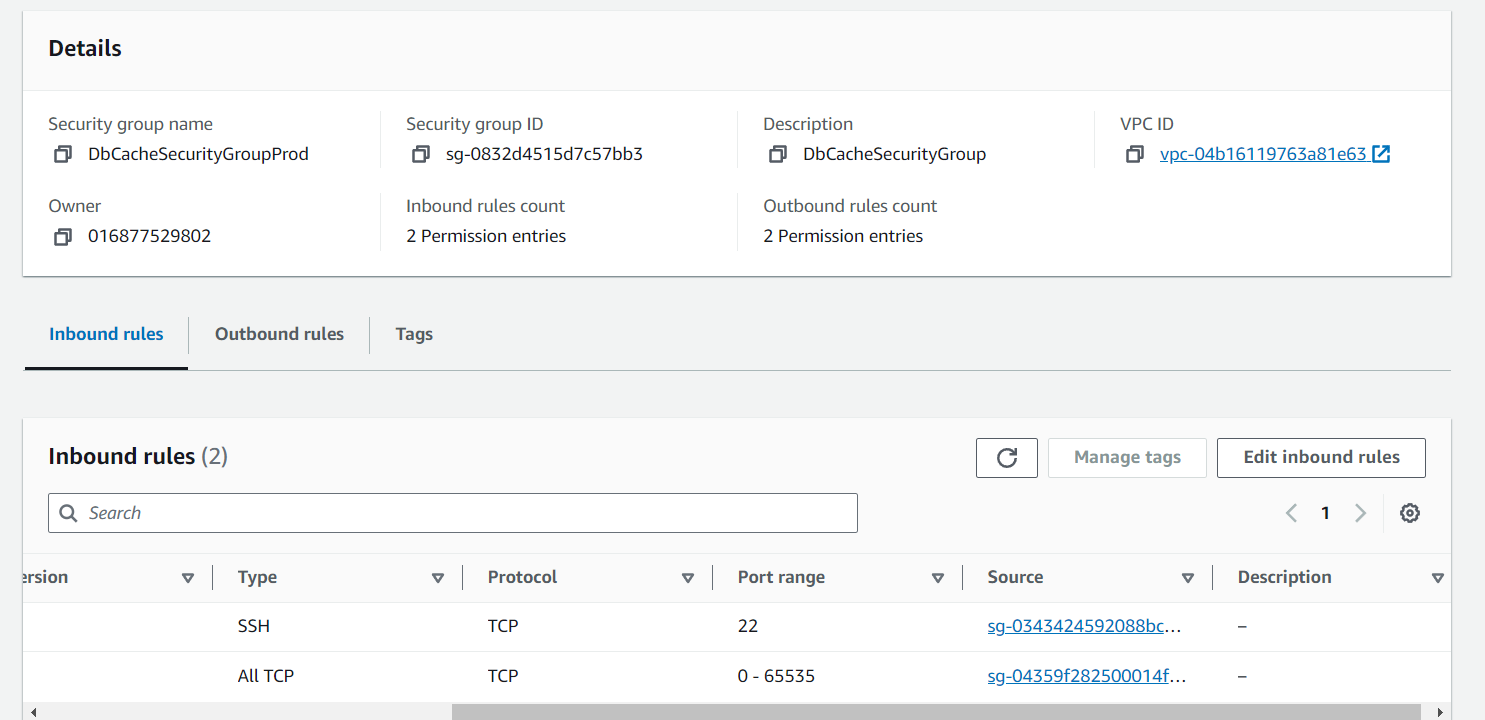


* + app1 and app2: Allow inbound traffic only from web and between themselves. Allow outbound to dbcache.

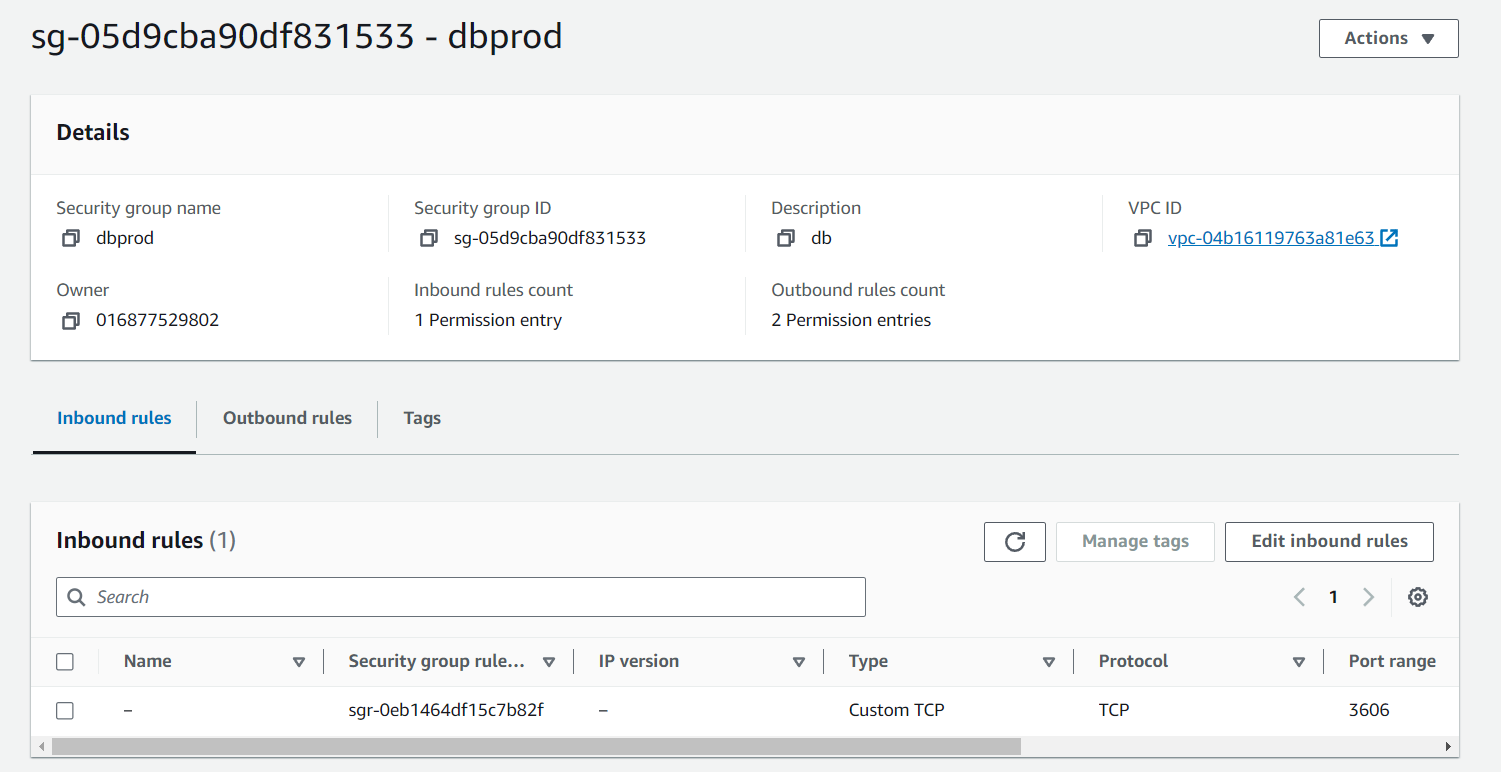


* + dbcache: Allow inbound traffic from app1 and app2. Allow outbound to the internet and db.





* + db: Allow inbound traffic only from dbcache. Restrict outbound as needed.



* **NACLs:**
  + Implement network ACLs to provide an additional layer of security, restricting inbound/outbound traffic according to your architecture.

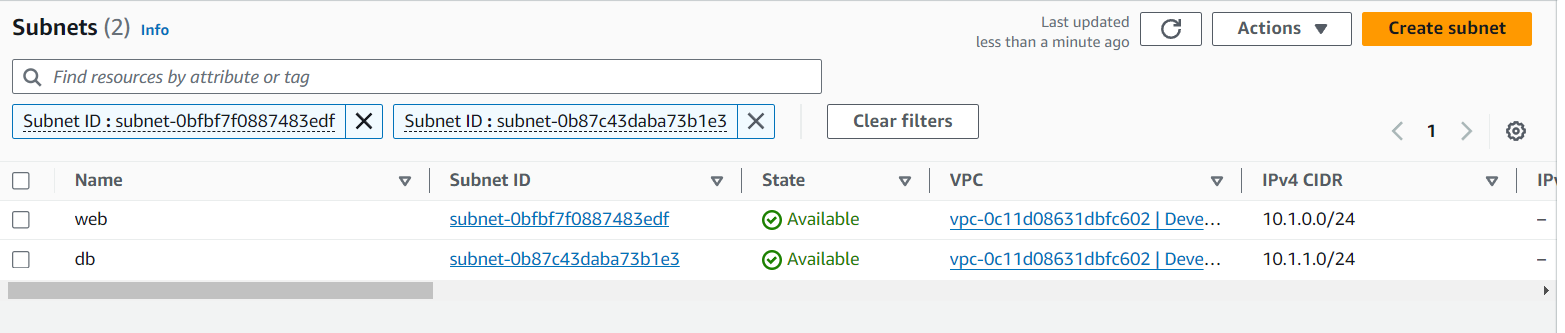
**Development Network:**

**1. Design and Build a 2-Tier Architecture:**

* **VPC Creation:**
  + Create a VPC named DevelopmentVPC.
  + Choose an appropriate CIDR block (e.g., 10.1.0.0/16).

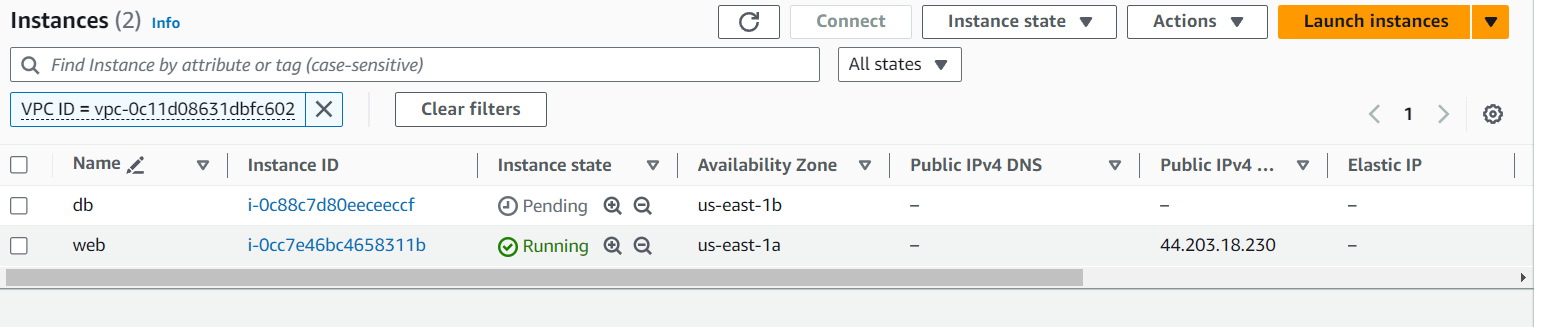
**2. Create Subnets:**

* **Web Subnet:**
  + Create a public subnet named web with CIDR 10.1.0.0/24.
* **DB Subnet:**
  + Create a private subnet named db with CIDR 10.1.1.0/24.



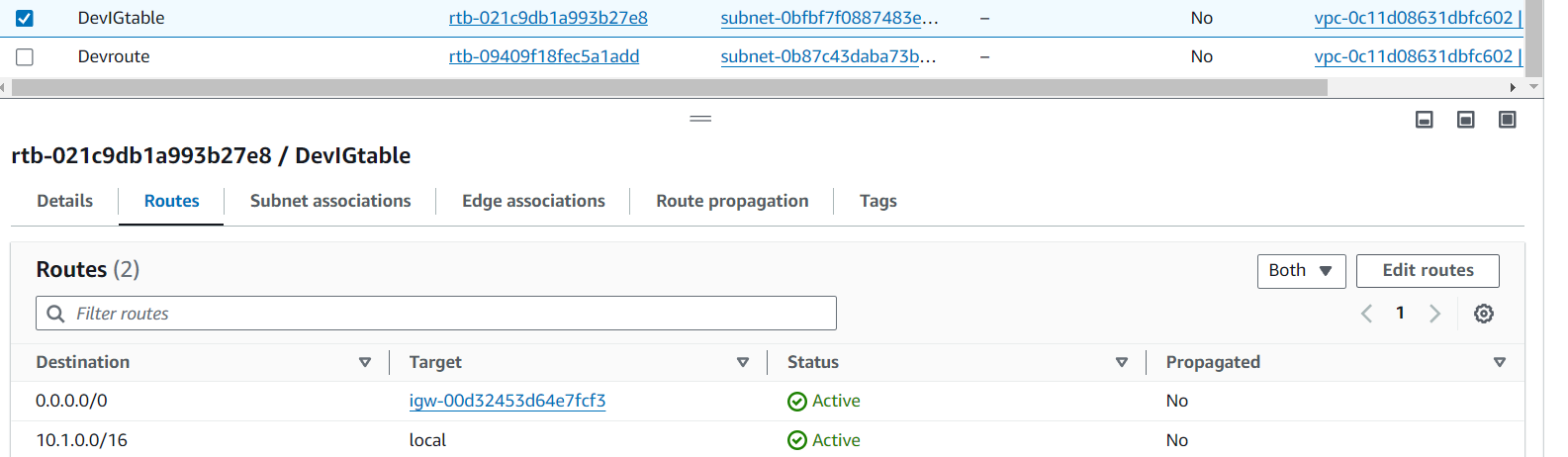
**3. Launch Instances:**

* Launch EC2 instances in both subnets and name them web and db.

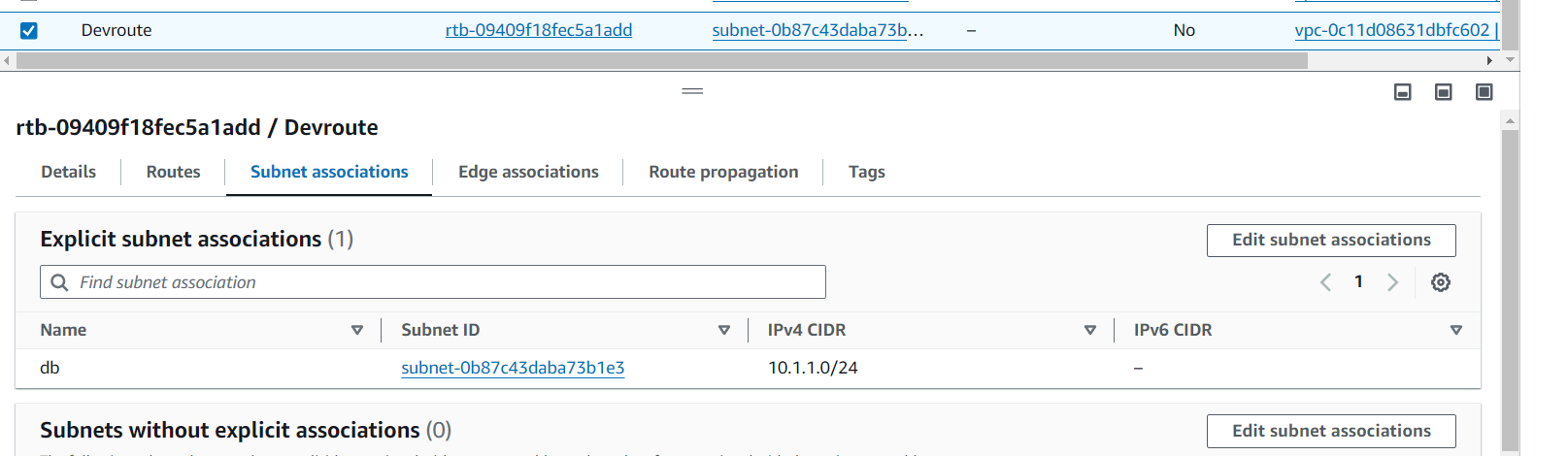


**4. Internet Access:**

* **NAT Gateway:**
  + Create a NAT Gateway in the web subnet.
  + Associate the NAT Gateway with the route table of the db subnet.
* **Route Tables:**
  + Update route tables:
    - web should have a route to the Internet Gateway.



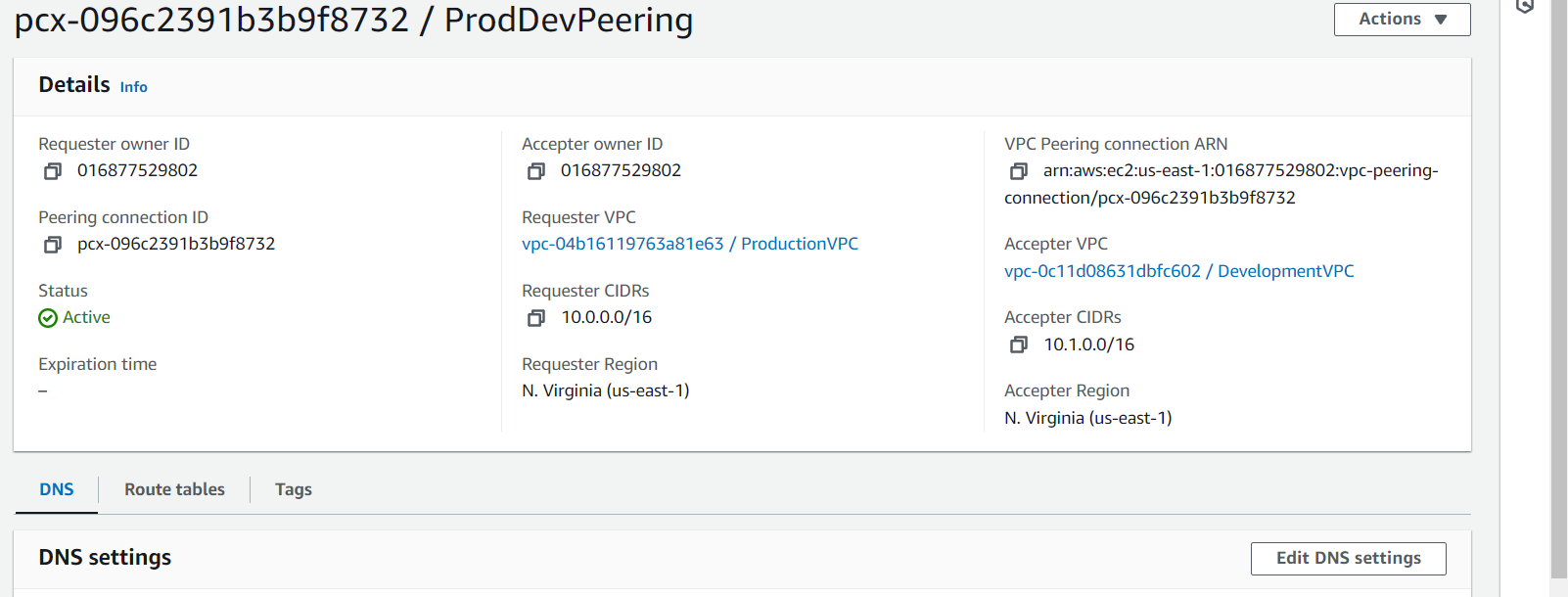
* + - db should have a route to the NAT Gateway.



**VPC Peering and Interconnectivity:**

**1. Peering Connection:**

* Create a VPC peering connection between ProductionVPC and DevelopmentVPC.
* Update the route tables in both VPCs to allow traffic between them.



**2. DB Subnets Interconnection:**

* Configure the security groups and route tables to allow communication between the db subnets in both VPCs.
* Ensure that traffic is restricted to only the required ports for database communication.