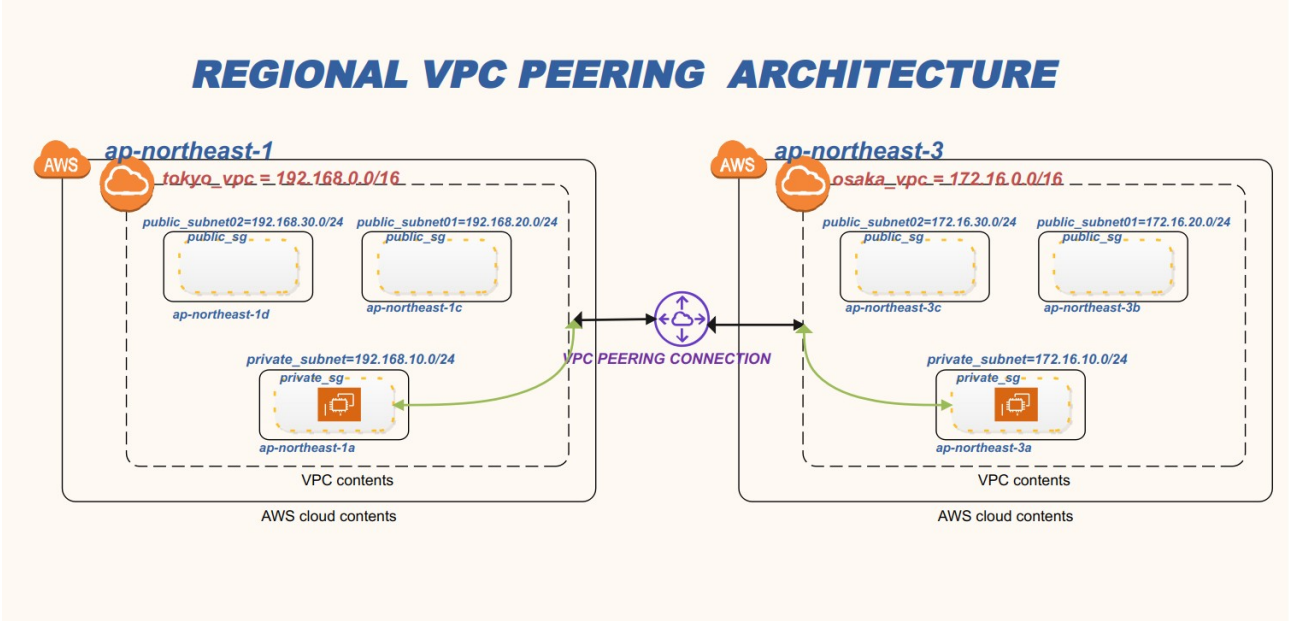
Cross-Regional VPC Peering: A Secure and Efficient Cloud Network Infrastructure with Terraform

Project Architecture:



Project Descriptions:

- This project will create the architecture above using **Terraform**

Prerequiestes:

1. AWS Account
2. AWS ACCESS KEY and SECRET ACCESS KEY with administrator ACCESS
3. Terraform Must be Installed
4. GIT Must be Installed

Instructions:

1. clone project repository :
   git clone -b terraform https://github.com/robudexIT/awsdevopsproject.git
2. cd to terraform project directory:
   cd awsdevopsproject/terraform/vpcpeering/
3. **Run these commands**
   - terraform init
   - terraform validate
   - terraform plan
   - terraform apply -auto-approve

```
(base) robudex@robudex-Dell-System-Vostro-3360:~/SBTPHPROJECTS/awsdevopsproject/terraform/vpcpeering$ terraform init

Initializing the backend...
Initializing modules...
- osaka_vpc in vpc
- tokyo_osaka_vpc_peering in vpc_peering
- tokyo_vpc in vpc

Initializing provider plugins...
- Finding hashicorp/aws versions matching "~> 5.0"...
- Installing hashicorp/aws v5.31.0...
- Installed hashicorp/aws v5.31.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.
```

```
(base) robudex@robudex-Dell-System-Vostro-3360:~/SBTPHPROJECTS/awsdevopsproject/terraform/vpcpeering$ terraform validate

  Warning: Reference to undefined provider

    on main.tf line 13, in module "osaka_vpc":
    13:     aws = aws.osaka

  There is no explicit declaration for local provider name "aws" in module.osaka_vpc, so Terraform is assuming you mean to pass a
  configuration for "hashicorp/aws".

  If you also control the child module, add a required_providers entry named "aws" with the source address "hashicorp/aws".

Success! The configuration is valid, but there were some validation warnings as shown above.
```

```
(base) robudex@robudex-Dell-System-Vostro-3360:~/SBTPHPROJECTS/awsdevopsproject/terraform/vpcpeering$ terraform plan
module.osaka_vpc.data.aws_availability_zones.available: Reading...
module.tokyo_vpc.data.aws_availability_zones.available: Reading...
module.osaka_vpc.data.aws_availability_zones.available: Read complete after 1s [id=ap-northeast-3]
module.tokyo_vpc.data.aws_availability_zones.available: Read complete after 1s [id=ap-northeast-1]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create
 <= read (data resources)

Terraform will perform the following actions:
```

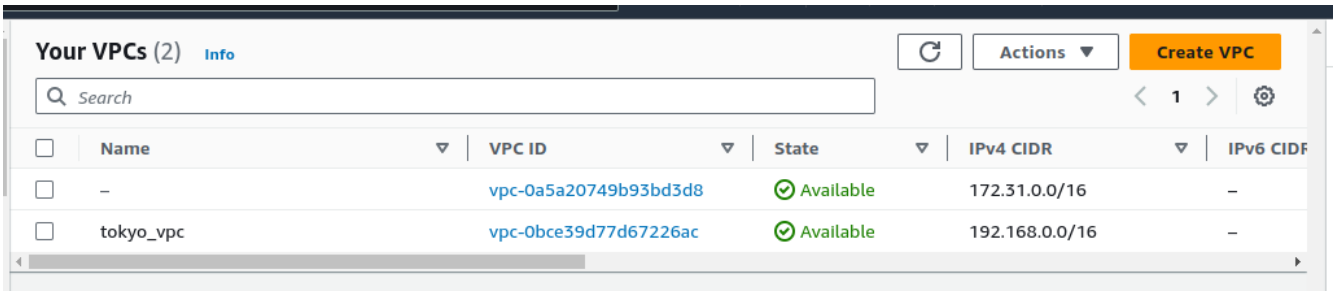
```
(base) robudex@robudex-Dell-System-Vostro-3360:~/SBTPHPROJECTS/awsdevopsproject/terraform/vpcpeering$ terraform apply -auto-approve
module.tokyo_vpc.data.aws_availability_zones.available: Reading...
module.tokyo_vpc.data.aws_availability_zones.available: Read complete after 1s [id=ap-northeast-1]
module.osaka_vpc.data.aws_availability_zones.available: Reading...
module.osaka_vpc.data.aws_availability_zones.available: Read complete after 2s [id=ap-northeast-3]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create
 <= read (data resources)

Terraform will perform the following actions:
```

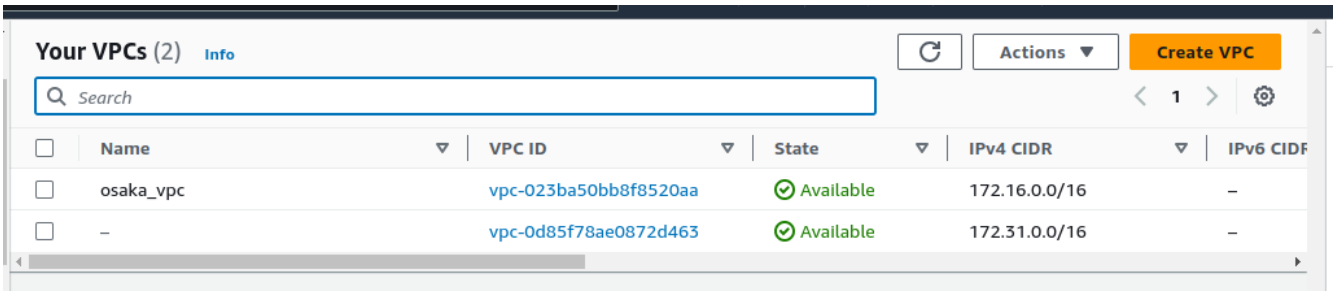
```
Apply complete! Resources: 30 added, 0 changed, 0 destroyed.

Outputs:

osaka_private_route_table_id = "rtb-0f6e1ec3eba283e95"
osaka_private_security_group_id = "sg-04963afd6ae560c35"
osaka_private_subnet_id = "subnet-031a7f11d8a479ea1"
osaka_vpc_id = "vpc-08fa9571f99350bf7"
tokyo_private_route_table_id = "rtb-0e8fd86f1a3938104"
tokyo_private_security_group_id = "sg-0e629a997d839378f"
tokyo_private_subnet_id = "subnet-0d7621ada68e30350"
tokyo_vpc_id = "vpc-0bce39d77d67226ac"
```

4. When Terraform completes its execution, it creates the following AWS services and resources based on the configurations defined in your Terraform files

- 2 VPCS one for Tokyo Region and one for Osaka Region
- **Each VPC Contains**:
    1. 3 Subnets (Two Public and One Private)
    2. One public routing table with associations to two public subnets.
    3. One public routing table with associations to one subnet.
    4. 2 Security Groups (One Public and One Private)
    5. VPC Connection between two VPCs
    6. Internet Gateway

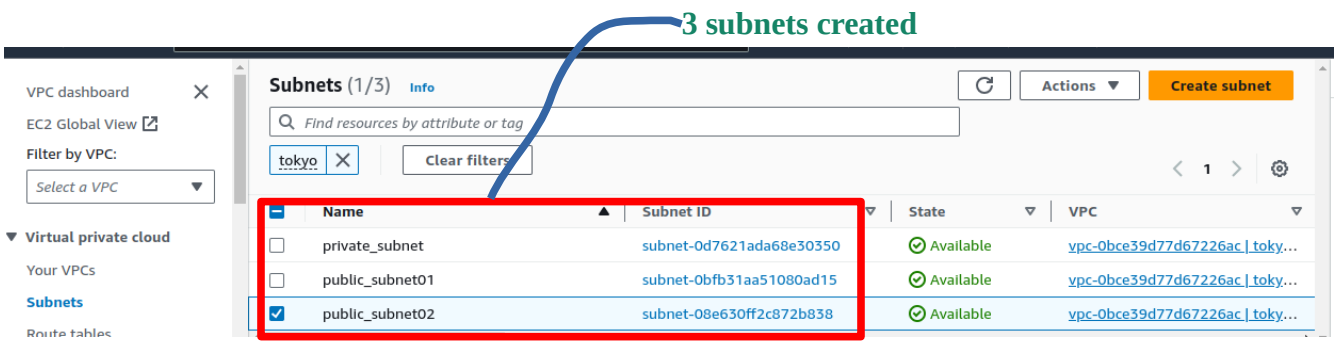
5. Open the AWS console in two separate tabs. In one tab, navigate to the **Tokyo region**, and in the other tab, navigate to the **Osaka region**. Once there, go to the VPC section in each region and verify if the VPCs have been successfully created.



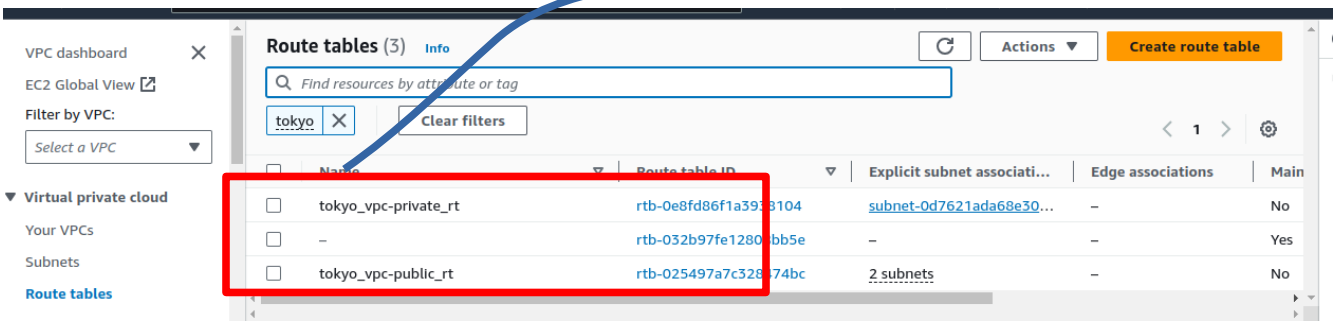


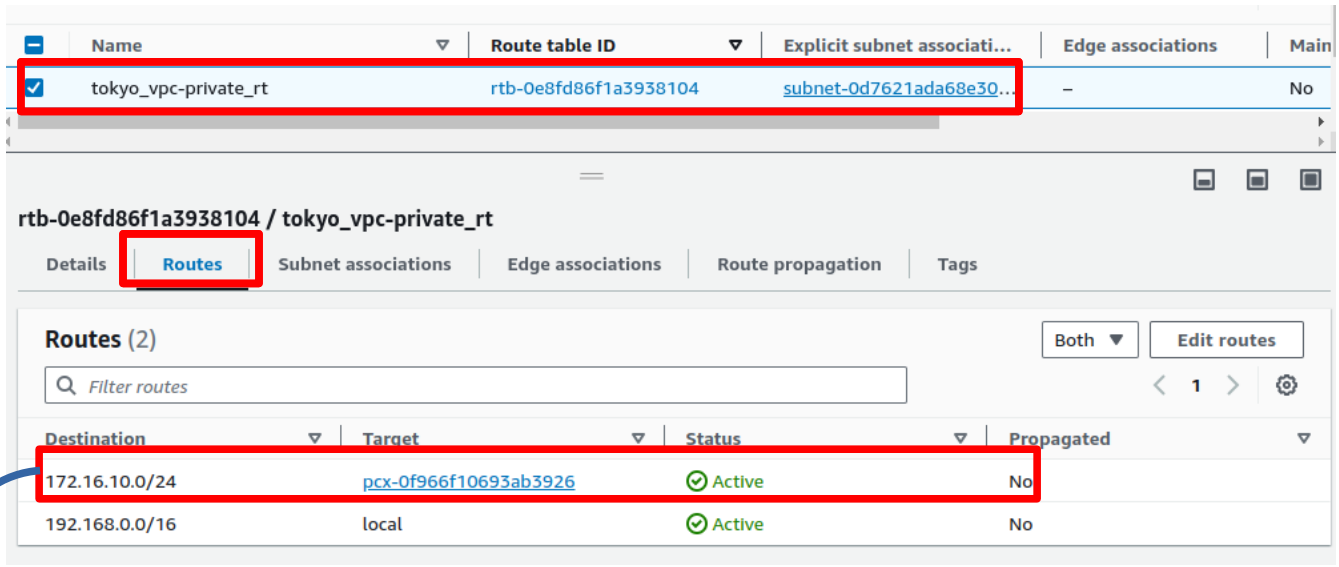
6. Verify the Following resources in each region.

FOR TOKYO REGION

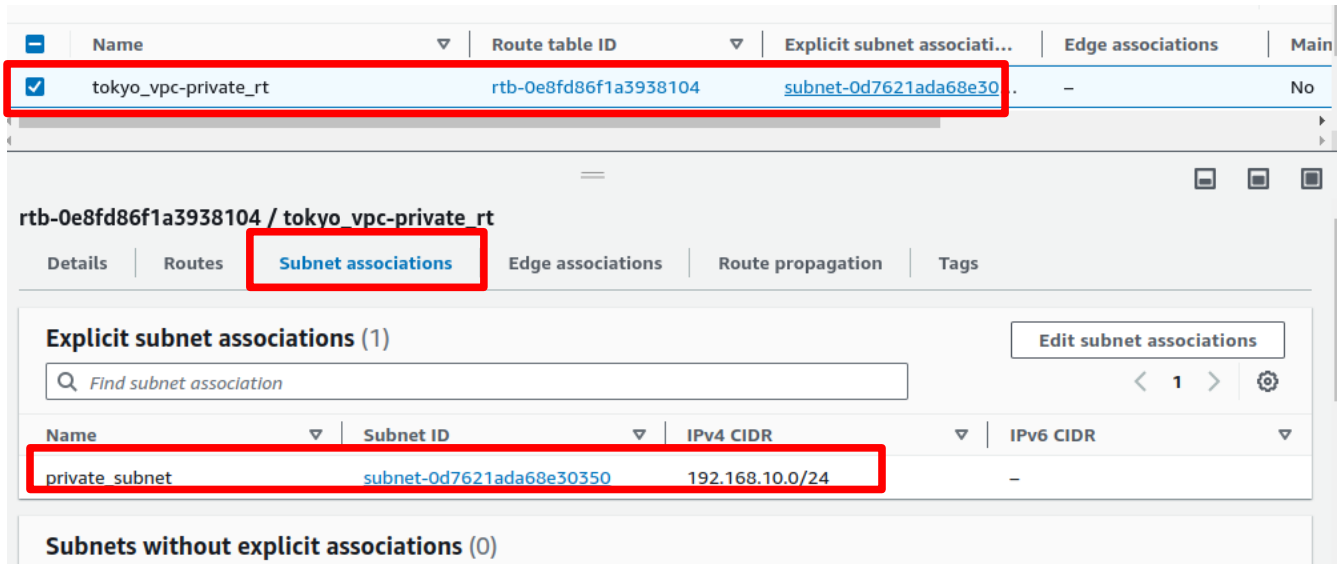


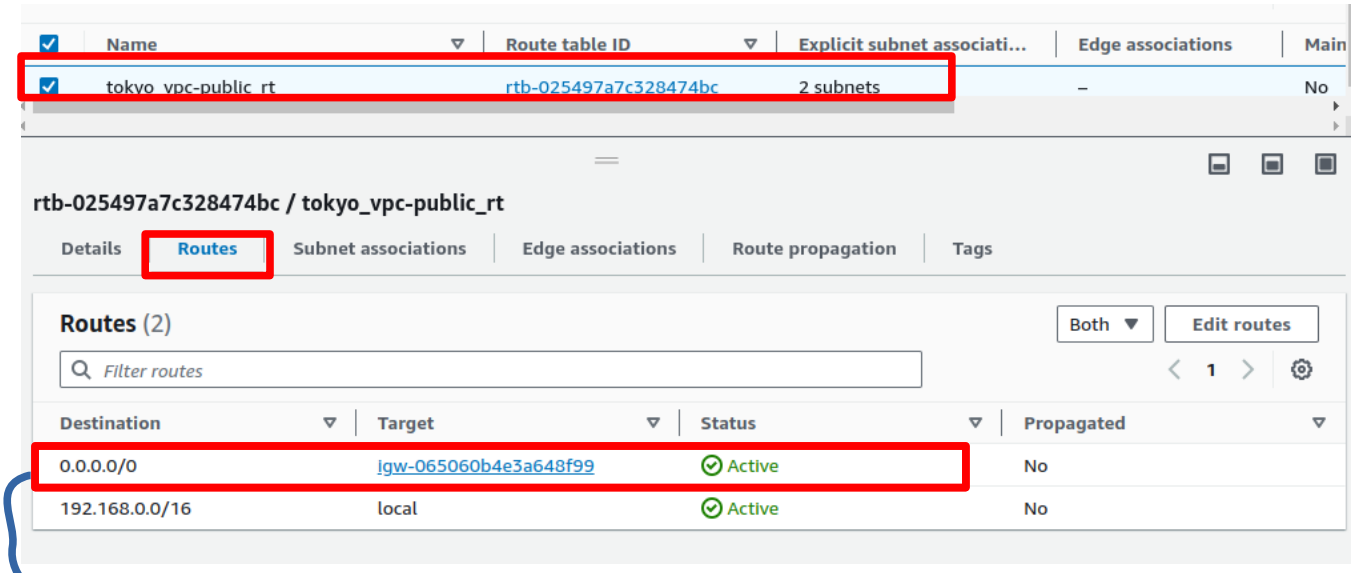
Public and Private routing table created



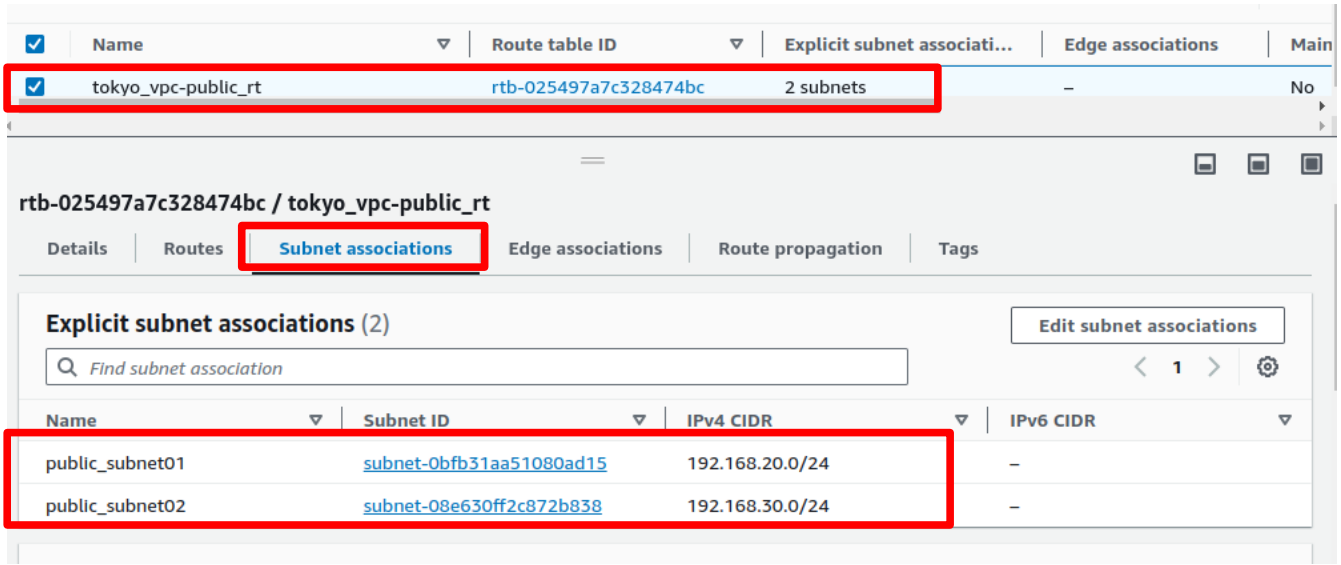


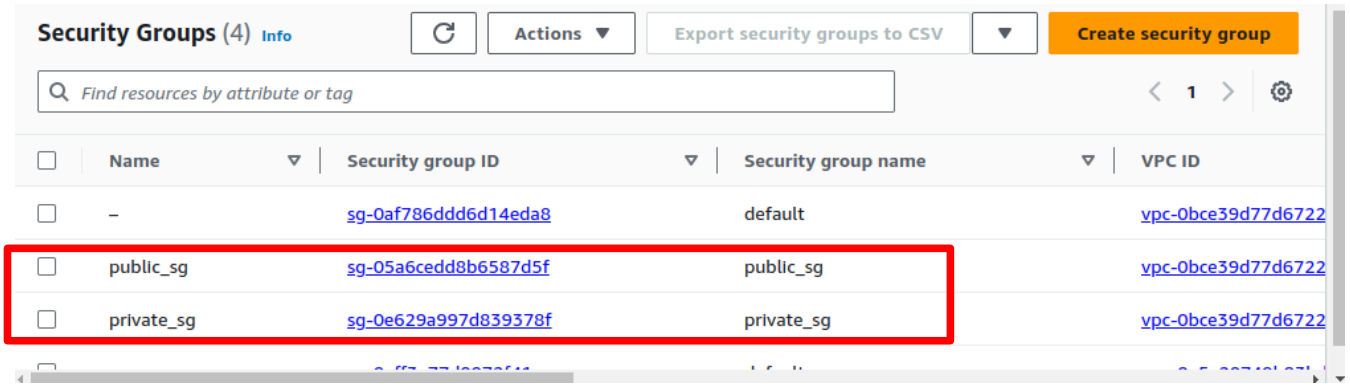
Private Routing Table has Destination route of 172.16.10.0/24 with VPC Peering Connection ID as Target



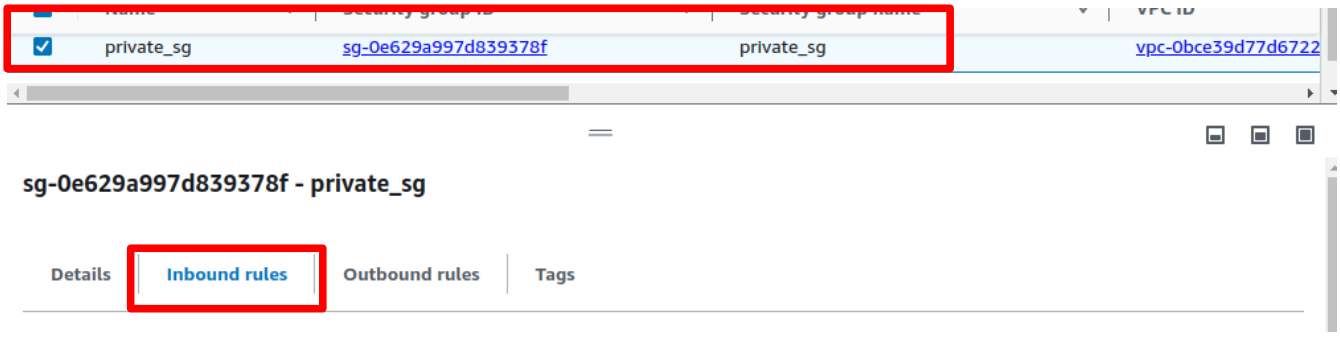


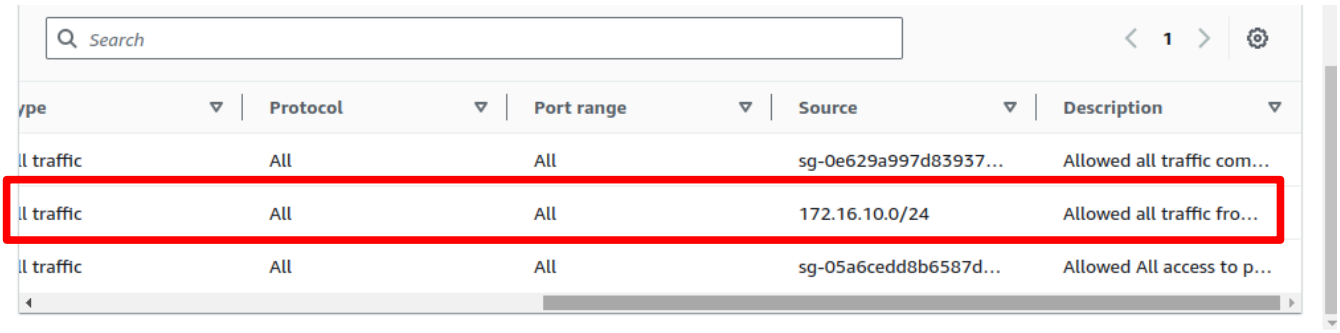
Public Routing Table with a route to the internet

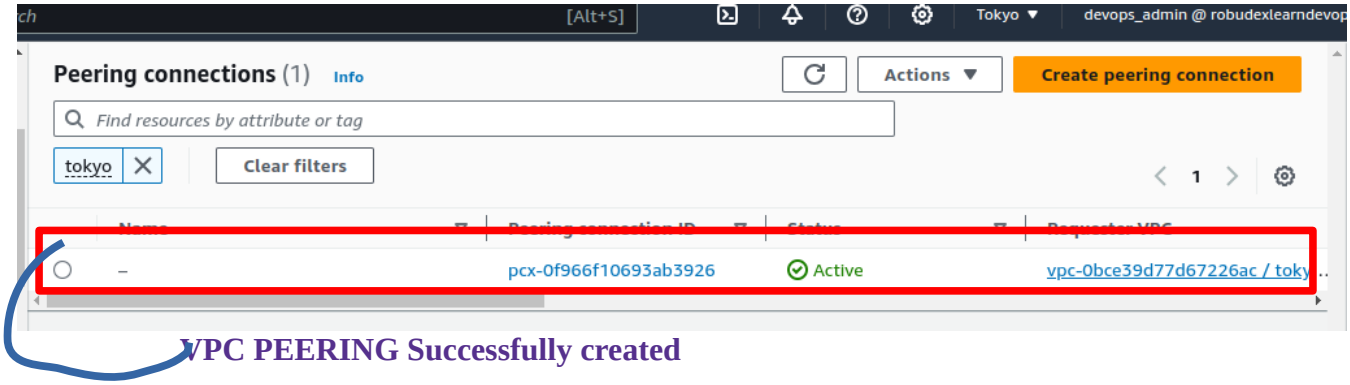




Select the '**private_sg**,' then click on '**Inbound rules**' to ensure that a rule allowing traffic from the **subnet** in the **OSAKA** region has been added.



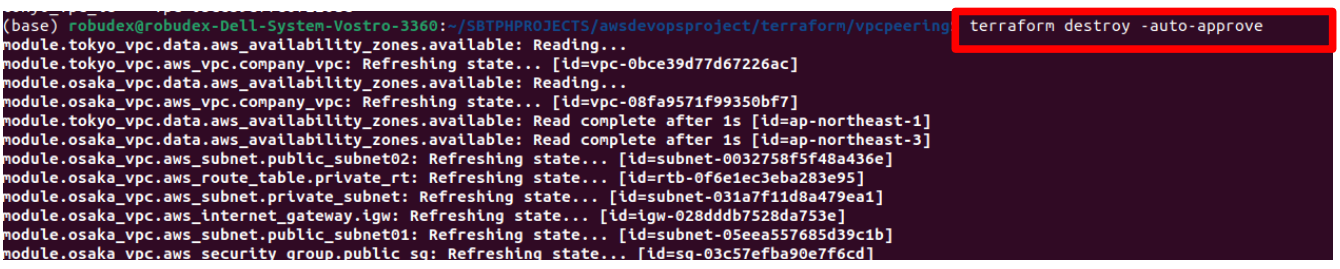




VPC PEERING Successfully created

7. Do the same on the **OSAKA REGION**

8. Once you've done, dont forget to delete your project. Goto the terminal and type the command:
   - **terraform destroy -auto-approve**




Warning: Reference to undefined provider
on main.tf line 13, in module "osaka_vpc":
13: aws = aws.osaka
There is no explicit declaration for local provider name "aws" in module.osaka_vpc, so Terraform is assuming you mean to pass a configuration for "hashicorp/aws".
If you also control the child module, add a required_providers entry named "aws" with the source address "hashicorp/aws".
estroy complete! Resources: 30 destroyed.