VPC Components

[Practical 04]

Name: Shivshankar Ghyar
PRN: 202201040031
Batch: CCF1

Problem Statement

Create a Virtual Private Cloud (VPC) on AWS with two public subnets and one private subnet, attach an Internet Gateway for internet access, configure route tables for proper routing, and set up a security group to allow SSH and HTTP access

---

Step 1: Set Up AWS CLI

command: aws configure

```
PS C:\Users\Shivshankar\Downloads> aws configure
AWS Access Key ID [**********************GNE3]:
AWS Secret Access Key [**********************RT5c]:
Default region name [ap-south-1]:
Default output format [json]:
```

Step 2: Create a VPC

➔ Create a VPC with a CIDR block, such as `10.0.0.0/16`

command: aws ec2 create-vpc --cidr-block 10.0.0.0/16

```
PS C:\Users\Shivshankar\Downloads> aws ec2 create-vpc --cidr-block 10.0.0.0/16
{
    "Vpc": {
        "OwnerId": "715841363442",
        "InstanceTenancy": "default",
        "Ipv6CidrBlockAssociationSet": [],
        "CidrBlockAssociationSet": [
            {
                "AssociationId": "vpc-cidr-assoc-0618ecc6f1080f6fd",
                "CidrBlock": "10.0.0.0/16",
                "CidrBlockState": {
                    "State": "associated"
                }
            }
        ],
        "IsDefault": false,
        "VpcId": "vpc-0859e45ef9bbf6475",
        "State": "pending",
        "CidrBlock": "10.0.0.0/16",
        "DhcpOptionsId": "dopt-0b3f715aec34ad700"
    }
}
```

➔ "VpcId": "vpc-0859e45ef9bbf6475"

Step 3: Create Subnets

➔ We'll create two public subnets and one private subnet within the VPC.

1. Create the first public subnet [in availability zone `ap-south-1a`]

command: `aws ec2 create-subnet --vpc-id vpc-0859e45ef9bbf6475 --cidr-block 10.0.1.0/24 --availability-zone ap-south-1a`

```
PS C:\Users\Shivshankar\Downloads> aws ec2 create-subnet --vpc-id vpc-0859e45ef9bbf6475 --cidr-block 10.0.1.0/24 --avail
ability-zone ap-south-1a
{
    "Subnet": {
        "AvailabilityZoneId": "aps1-az1",
        "OwnerId": "715841363442",
        "AssignIpv6AddressOnCreation": false,
        "Ipv6CidrBlockAssociationSet": [],
        "SubnetArn": "arn:aws:ec2:ap-south-1:715841363442:subnet/subnet-01dee052d8f3466a2",
        "EnableDns64": false,
        "Ipv6Native": false,
        "PrivateDnsNameOptionsOnLaunch": {
            "HostnameType": "ip-name",
            "EnableResourceNameDnsARecord": false,
            "EnableResourceNameDnsAAAARecord": false
        },
        "SubnetId": "subnet-01dee052d8f3466a2",
        "State": "available",
        "VpcId": "vpc-0859e45ef9bbf6475",
        "CidrBlock": "10.0.1.0/24",
        "AvailableIpAddressCount": 251,
        "AvailabilityZone": "ap-south-1a",
        "DefaultForAz": false,
        "MapPublicIpOnLaunch": false
    }
}
```

"SubnetId": "subnet-01dee052d8f3466a2"

2.Create the second public subnet [in availability zone `ap-south-1b`]

command:aws ec2 create-subnet --vpc-id vpc-0859e45ef9bbf6475 --cidr-block 10.0.2.0/24 --availability-zone ap-south-1b

```
PS C:\Users\Shivshankar\Downloads> aws ec2 create-subnet --vpc-id vpc-0859e45ef9bbf6475 --cidr-block 10.0.2.0/24 --avail
ability-zone ap-south-1b
{
    "Subnet": {
        "AvailabilityZoneId": "aps1-az3",
        "OwnerId": "715841363442",
        "AssignIpv6AddressOnCreation": false,
        "Ipv6CidrBlockAssociationSet": [],
        "SubnetArn": "arn:aws:ec2:ap-south-1:715841363442:subnet/subnet-0bdc11dd3759a425a",
        "EnableDns64": false,
        "Ipv6Native": false,
        "PrivateDnsNameOptionsOnLaunch": {
            "HostnameType": "ip-name",
            "EnableResourceNameDnsARecord": false,
            "EnableResourceNameDnsAAAARecord": false
        },
        "SubnetId": "subnet-0bdc11dd3759a425a",
        "State": "available",
        "VpcId": "vpc-0859e45ef9bbf6475",
        "CidrBlock": "10.0.2.0/24",
        "AvailableIpAddressCount": 251,
        "AvailabilityZone": "ap-south-1b",
        "DefaultForAz": false,
        "MapPublicIpOnLaunch": false
    }
}
```

"SubnetId": "subnet-0bdc11dd3759a425a"

3.Create the private subnet [in availability zone ap-south-1a]

command: aws ec2 create-subnet --vpc-id vpc-0859e45ef9bbf6475 --cidr-block 10.0.3.0/24 --availability-zone ap-south-1a

```
PS C:\Users\Shivshankar\Downloads> aws ec2 create-subnet --vpc-id vpc-0859e45ef9bbf6475 --cidr-block 10.0.3.0/24 --avail
ability-zone ap-south-1a
{
    "Subnet": {
        "AvailabilityZoneId": "aps1-az1",
        "OwnerId": "715841363442",
        "AssignIpv6AddressOnCreation": false,
        "Ipv6CidrBlockAssociationSet": [],
        "SubnetArn": "arn:aws:ec2:ap-south-1:715841363442:subnet/subnet-088cf1e9e29f117a3",
        "EnableDns64": false,
        "Ipv6Native": false,
        "PrivateDnsNameOptionsOnLaunch": {
            "HostnameType": "ip-name",
            "EnableResourceNameDnsARecord": false,
            "EnableResourceNameDnsAAAARecord": false
        },
        "SubnetId": "subnet-088cf1e9e29f117a3",
        "State": "available",
        "VpcId": "vpc-0859e45ef9bbf6475",
        "CidrBlock": "10.0.3.0/24",
        "AvailableIpAddressCount": 251,
        "AvailabilityZone": "ap-south-1a",
        "DefaultForAz": false,
        "MapPublicIpOnLaunch": false
    }
}
```

"SubnetId": "subnet-088cf1e9e29f117a3"

Step 4: Create and Attach an Internet Gateway

➔ Create an Internet Gateway

command: aws ec2 create-internet-gateway

```
PS C:\Users\Shivshankar\Downloads> aws ec2 create-internet-gateway
{
    "InternetGateway": {
        "Attachments": [],
        "InternetGatewayId": "igw-0a95d4d9a0a4cc671",
        "OwnerId": "715841363442",
        "Tags": []
    }
}
```

"InternetGatewayId": "igw-0a95d4d9a0a4cc671"

➔ Attach the Internet Gateway to the VPC

command: aws ec2 attach-internet-gateway --internet-gateway-id igw-0a95d4d9a0a4cc671 --vpc-id vpc-0859e45ef9bbf6475

```
PS C:\Users\Shivshankar\Downloads> aws ec2 attach-internet-gateway --internet-gateway-id igw-0a95d4d9a0a4cc671 --vpc-id
vpc-0859e45ef9bbf6475
PS C:\Users\Shivshankar\Downloads>
```

Step 5: Create Route Tables and Set Up Routes

➔ Create a route table for the VPC:

command: aws ec2 create-route-table --vpc-id vpc-0859e45ef9bbf6475

```
PS C:\Users\Shivshankar\Downloads> aws ec2 create-route-table --vpc-id vpc-0859e45ef9bbf6475
{
    "RouteTable": {
        "Associations": [],
        "PropagatingVgws": [],
        "RouteTableId": "rtb-0e6c5c959dab015ed",
        "Routes": [
            {
                "DestinationCidrBlock": "10.0.0.0/16",
                "GatewayId": "local",
                "Origin": "CreateRouteTable",
                "State": "active"
            }
        ],
        "Tags": [],
        "VpcId": "vpc-0859e45ef9bbf6475",
        "OwnerId": "715841363442"
    },
    "ClientToken": "5c7dc4f5-2836-4041-a882-e434a214db86"
}
```

"RouteTableId": "rtb-0e6c5c959dab015ed"

➔ Create a route in the route table to direct traffic to the Internet Gateway

Command: aws ec2 create-route --route-table-id rtb-0e6c5c959dab015ed

--destination-cidr-block 0.0.0.0/0 --gateway-id igw-0a95d4d9a0a4cc671

➔ Associate public subnets with the route table

```
PS C:\Users\Shivshankar\Downloads> aws ec2 create-route --route-table-id rtb-0e6c5c959dab015ed --destination-cidr-block
0.0.0.0/0 --gateway-id igw-0a95d4d9a0a4cc671
{
    "Return": true
}
```

For the first public subnet:
Command: aws ec2 associate-route-table --route-table-id rtb-0e6c5c959dab015ed
--subnet-id subnet-01dee052d8f3466a2

```
PS C:\Users\Shivshankar\Downloads> aws ec2 associate-route-table --route-table-id rtb-0e6c5c959dab015ed --subnet-id subn
et-01dee052d8f3466a2
{
    "AssociationId": "rtbassoc-0a1c7e163853f8bfd",
    "AssociationState": {
        "State": "associated"
    }
}
```

For the second public subnet:
Command: aws ec2 associate-route-table --route-table-id rtb-0e6c5c959dab015ed
--subnet-id subnet-0bdc11dd3759a425a

```
PS C:\Users\Shivshankar\Downloads> aws ec2 associate-route-table --route-table-id rtb-0e6c5c959dab015ed --subnet-id subn
et-0bdc11dd3759a425a
{
    "AssociationId": "rtbassoc-045e21086ba3e3d54",
    "AssociationState": {
        "State": "associated"
    }
}
```

The private subnet does not need internet access, so no association is required for it.

Step 6: Modify Public Subnets to Auto-assign Public IPs

➔ Enable auto-assign public IPs for each public subnet:

For the first public subnet

command: aws ec2 modify-subnet-attribute --subnet-id subnet-01dee052d8f3466a2
--map-public-ip-on-launch

For the second public subnet:

command:aws ec2 modify-subnet-attribute --subnet-id subnet-0bdc11dd3759a425a --map-public-ip-on-launch

```
PS C:\Users\Shivshankar\Downloads> aws ec2 modify-subnet-attribute --subnet-id subnet-01dee052d8f3466a2 --map-public-ip-on-launch
PS C:\Users\Shivshankar\Downloads>
PS C:\Users\Shivshankar\Downloads> aws ec2 modify-subnet-attribute --subnet-id subnet-0bdc11dd3759a425a --map-public-ip-on-launch
PS C:\Users\Shivshankar\Downloads>
```

Step 7: Create a Security Group with Inbound and Outbound Rules

➔ Create a security group for the VPC:

```
PS C:\Users\Shivshankar\Downloads> aws ec2 create-security-group --group-name my-security1-group --description "Security
 group for public access" --vpc-id vpc-0859e45ef9bbf6475
{
    "GroupId": "sg-0b8686ce950191670"
}
```

"GroupId": "sg-0b8686ce950191670"

➔ Add inbound rules to allow SSH and HTTP access

Allow SSH access (port 22) from a specific IP (e.g., `0.0.0.0/0` for any IP)

command: aws ec2 authorize-security-group-ingress --group-id sg-0b8686ce950191670 --protocol tcp --port 22 --cidr 0.0.0.0/0

```
PS C:\Users\Shivshankar\Downloads> aws ec2 authorize-security-group-ingress --group-id sg-0b8686ce950191670 --protocol t
cp --port 22 --cidr 0.0.0.0/0
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0ef6e8436ba2d417f",
            "GroupId": "sg-0b8686ce950191670",
            "GroupOwnerId": "715841363442",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "0.0.0.0/0"
        }
    ]
}
```

Allow HTTP access (port 80) from any IP:

command: aws ec2 authorize-security-group-ingress --group-id sg-0b8686ce950191670 --protocol tcp --port 80 --cidr 0.0.0.0/0

```
PS C:\Users\Shivshankar\Downloads> aws ec2 authorize-security-group-ingress --group-id sg-0b8686ce950191670 --protocol t
cp --port 80 --cidr 0.0.0.0/0
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-07b8e3a6644410410",
            "GroupId": "sg-0b8686ce950191670",
            "GroupOwnerId": "715841363442",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "CidrIpv4": "0.0.0.0/0"
        }
    ]
}
```

➔ Add outbound rules to allow all outbound traffic:

command: aws ec2 authorize-security-group-egress --group-id sg-0b8686ce950191670 --protocol -1 --port all --cidr 0.0.0.0/0

Step 8: Verify Your Setup

➔ List VPCs:

command: aws ec2 describe-vpcs

```
PS C:\Users\Shivshankar\Downloads> aws ec2 describe-vpcs
{
    "Vpcs": [
        {
            "OwnerId": "715841363442",
            "InstanceTenancy": "default",
            "CidrBlockAssociationSet": [
                {
                    "AssociationId": "vpc-cidr-assoc-060f0c22831d9918f",
                    "CidrBlock": "10.0.0.0/16",
                    "CidrBlockState": {
                        "State": "associated"
                    }
                }
            ],
            "IsDefault": false,
            "VpcId": "vpc-01ff1002f178eeaa9",
            "State": "available",
            "CidrBlock": "10.0.0.0/16",
            "DhcpOptionsId": "dopt-0b3f715aec34ad700"
        },
        {
            "OwnerId": "715841363442",
            "InstanceTenancy": "default",
            "CidrBlockAssociationSet": [
                {
                    "AssociationId": "vpc-cidr-assoc-0db247df4d06dc896",
                    "CidrBlock": "172.31.0.0/16",
                    "CidrBlockState": {
```

➔ List Subnets:

command : aws ec2 describe-subnets --filters "Name=vpc-id,Values=vpc-0859e45ef9bbf6475"

```
PS C:\Users\Shivshankar\Downloads> aws ec2 describe-subnets --filters "Name=vpc-id,Values=vpc-0859e45ef9bbf6475"
{
    "Subnets": [
        {
            "AvailabilityZoneId": "aps1-az3",
            "MapCustomerOwnedIpOnLaunch": false,
            "OwnerId": "715841363442",
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "SubnetArn": "arn:aws:ec2:ap-south-1:715841363442:subnet/subnet-0bdc11dd3759a425a",
            "EnableDns64": false,
            "Ipv6Native": false,
            "PrivateDnsNameOptionsOnLaunch": {
                "HostnameType": "ip-name",
                "EnableResourceNameDnsARecord": false,
                "EnableResourceNameDnsAAAARecord": false
            },
            "SubnetId": "subnet-0bdc11dd3759a425a",
            "State": "available",
            "VpcId": "vpc-0859e45ef9bbf6475",
            "CidrBlock": "10.0.2.0/24",
            "AvailableIpAddressCount": 251,
            "AvailabilityZone": "ap-south-1b",
            "DefaultForAz": false,
            "MapPublicIpOnLaunch": true
        },
        {
            "AvailabilityZoneId": "aps1-az1",
```

➔ List Route Tables:

command : aws ec2 describe-route-tables --filters "Name=vpc-id,Values=vpc-0859e45ef9bbf6475"

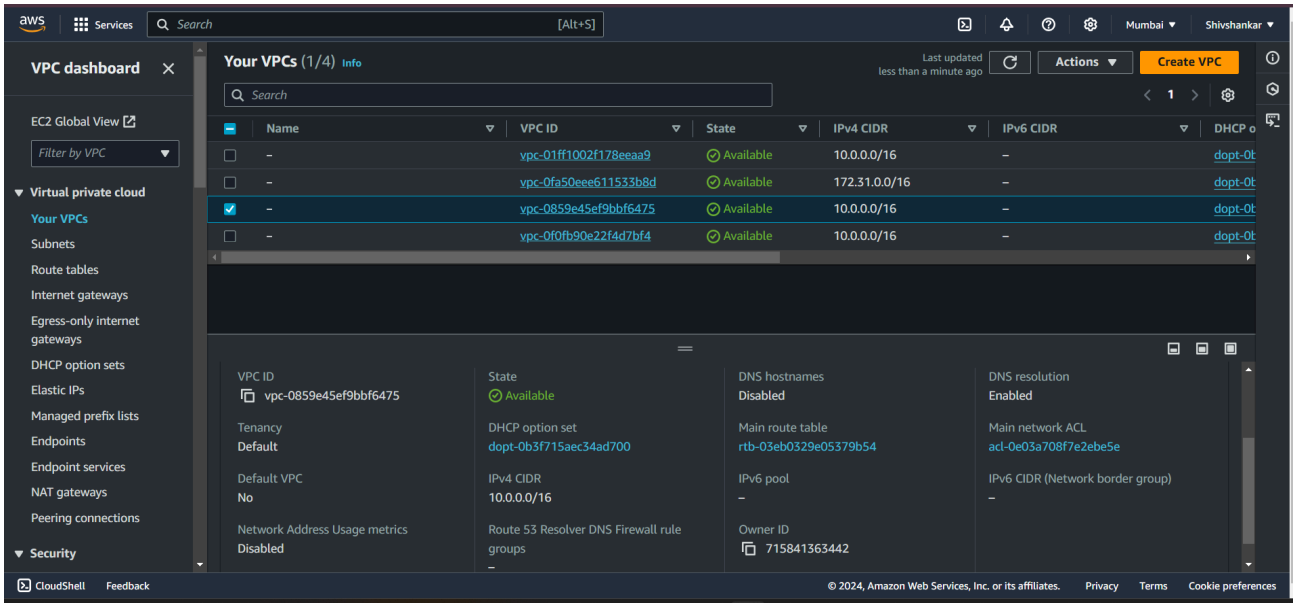
```
PS C:\Users\Shivshankar\Downloads> aws ec2 describe-route-tables --filters "Name=vpc-id,Values=vpc-0859e45ef9bbf6475"
{
    "RouteTables": [
        {
            "Associations": [
                {
                    "Main": false,
                    "RouteTableAssociationId": "rtbassoc-045e21086ba3e3d54",
                    "RouteTableId": "rtb-0e6c5c959dab015ed",
                    "SubnetId": "subnet-0bdc11dd3759a425a",
                    "AssociationState": {
                        "State": "associated"
                    }
                },
                {
                    "Main": false,
                    "RouteTableAssociationId": "rtbassoc-0a1c7e163853f8bfd",
                    "RouteTableId": "rtb-0e6c5c959dab015ed",
                    "SubnetId": "subnet-01dee052d8f3466a2",
                    "AssociationState": {
                        "State": "associated"
                    }
                }
            ],
            "PropagatingVgws": [],
            "RouteTableId": "rtb-0e6c5c959dab015ed",
            "Routes": [
                {
                    "DestinationCidrBlock": "10.0.0.0/16",
                    "GatewayId": "local",
```

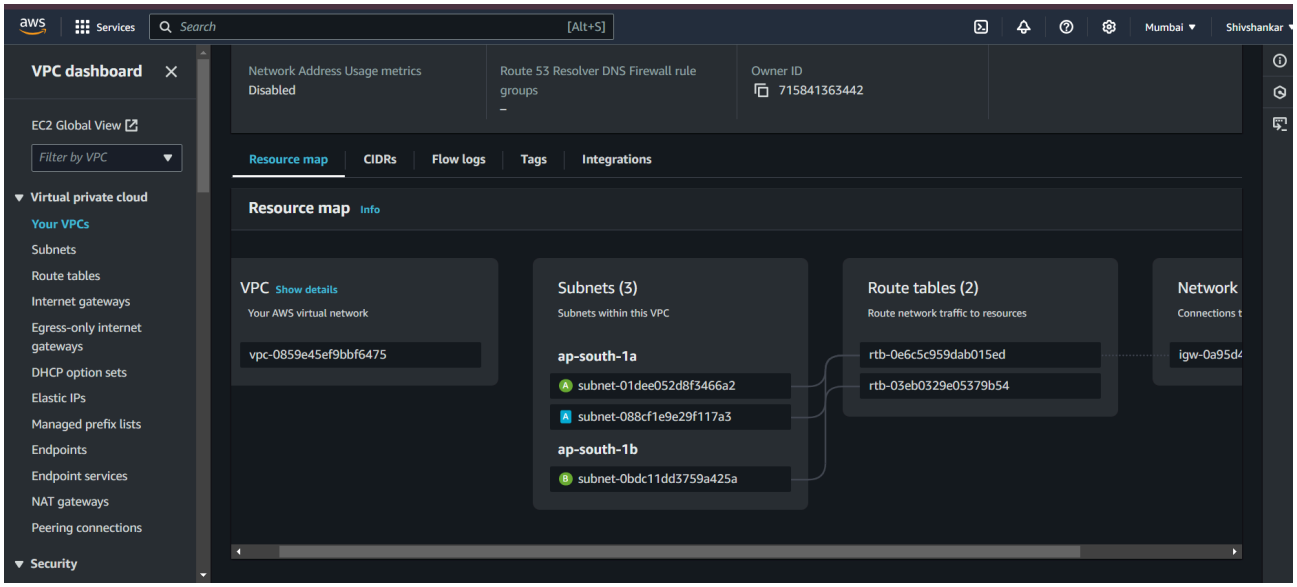
➔ List Security Groups:

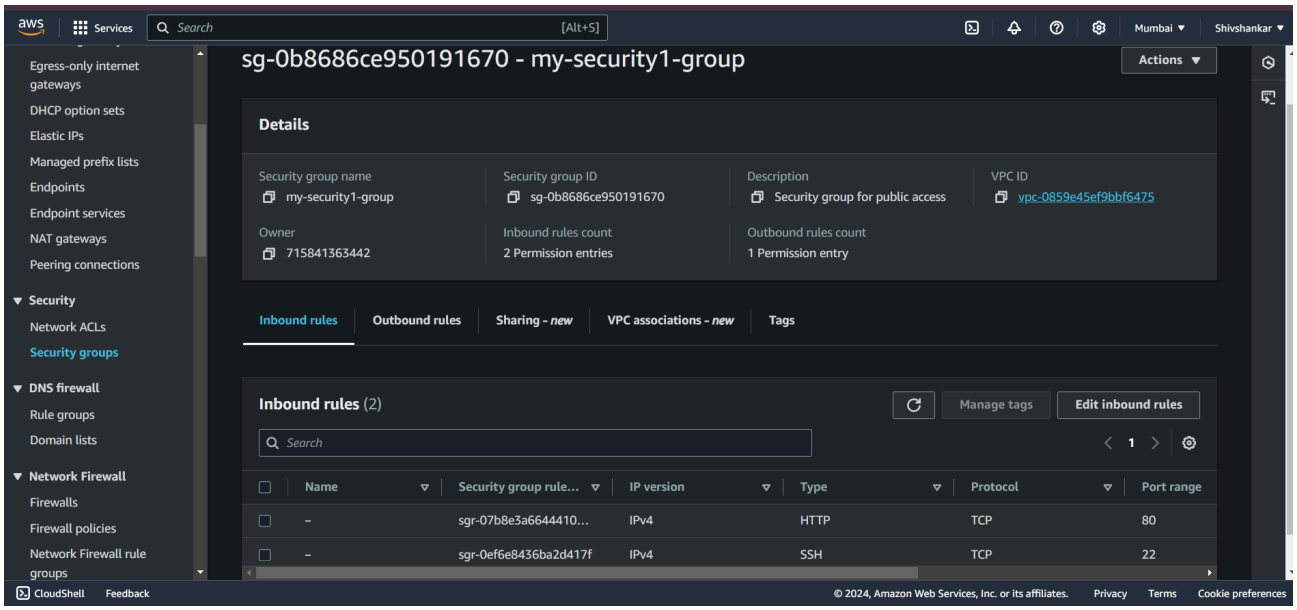
command: aws ec2 describe-security-groups --filters "Name=vpc-id,Values=vpc-0859e45ef9bbf6475"

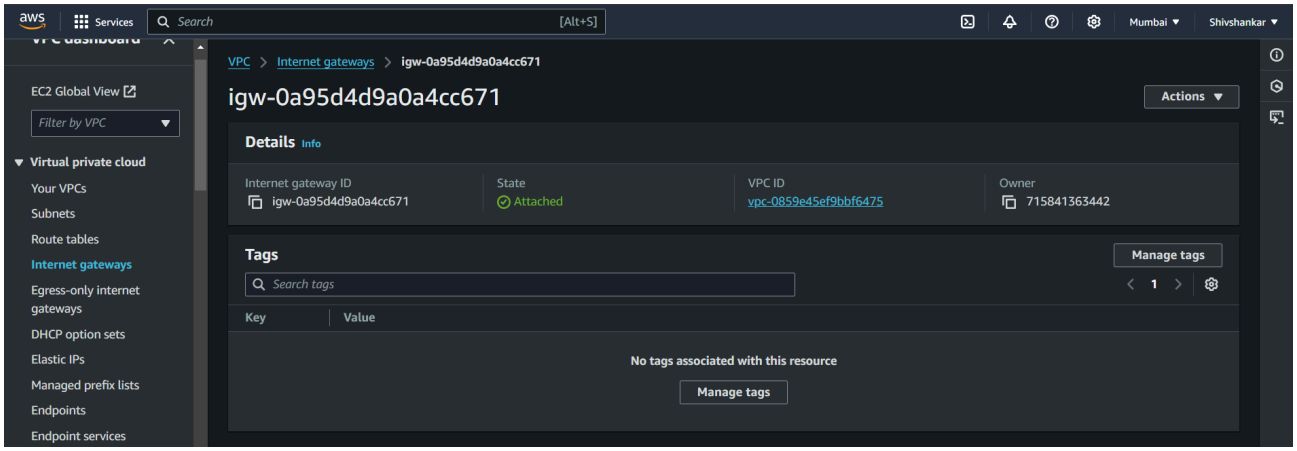
```
PS C:\Users\Shivshankar\Downloads> aws ec2 describe-security-groups --filters "Name=vpc-id,Values=vpc-0859e45ef9bbf6475"
{
    "SecurityGroups": [
        {
            "GroupId": "sg-06b6cf6fd798c5d37",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-0859e45ef9bbf6475",
            "OwnerId": "715841363442",
            "GroupName": "default",
            "Description": "default VPC security group",
            "IpPermissions": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [
                        {
                            "UserId": "715841363442",
                            "GroupId": "sg-06b6cf6fd798c5d37"
                        }
                    ],
                    "IpRanges": [],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ]
```

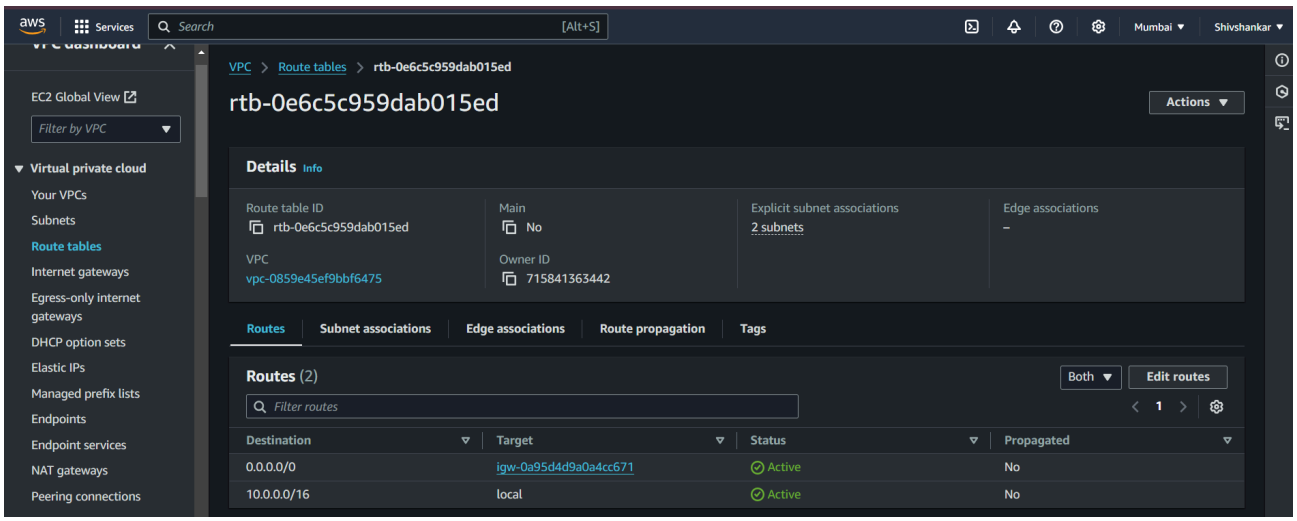
Verification through console






aws
Services
Search
[Alt+S]
Mumbai
Shivshankar
Egress-only internet gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
NAT gateways
Peering connections
Security
Network ACLs
Security groups
DNS firewall
Rule groups
Domain lists
Network Firewall
Firewalls
Firewall policies
Network Firewall rule groups
sg-0b8686ce950191670 - my-security1-group
Actions
Details
Security group name
my-security1-group
Security group ID
sg-0b8686ce950191670
Description
Security group for public access
VPC ID
vpc-0859e45ef9bbf6475
Owner
715841363442
Inbound rules count
2 Permission entries
Outbound rules count
1 Permission entry
Inbound rules
Outbound rules
Sharing - new
VPC associations - new
Tags
Inbound rules (2)
Manage tags
Edit inbound rules
Search
Name
Security group rule...
IP version
Type
Protocol
Port range
sgr-07b8e3a6644410...
IPv4
HTTP
TCP
80
sgr-0ef6e8436ba2d417f
IPv4
SSH
TCP
22
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


aws
Services
Search
[Alt+S]
Mumbai
Shivshankar
EC2 Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
VPC > Internet gateways > igw-0a95d4d9a0a4cc671
igw-0a95d4d9a0a4cc671
Actions
Details Info
Internet gateway ID
igw-0a95d4d9a0a4cc671
State
Attached
VPC ID
vpc-0859e45ef9bbf6475
Owner
715841363442
Tags
Manage tags
Search tags
Key
Value
No tags associated with this resource
Manage tags


aws
Services
Search
[Alt+S]
Mumbai
Shivshankar
EC2 Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
NAT gateways
Peering connections
VPC > Route tables > rtb-0e6c5c959dab015ed
rtb-0e6c5c959dab015ed
Actions
Details Info
Route table ID
rtb-0e6c5c959dab015ed
Main
No
Explicit subnet associations
2 subnets
Edge associations
–
VPC
vpc-0859e45ef9bbf6475
Owner ID
715841363442
Routes
Subnet associations
Edge associations
Route propagation
Tags
Routes (2)
Both
Edit routes
Filter routes
Destination
Target
Status
Propagated
0.0.0.0/0
igw-0a95d4d9a0a4cc671
Active
No
10.0.0.0/16
local
Active
No