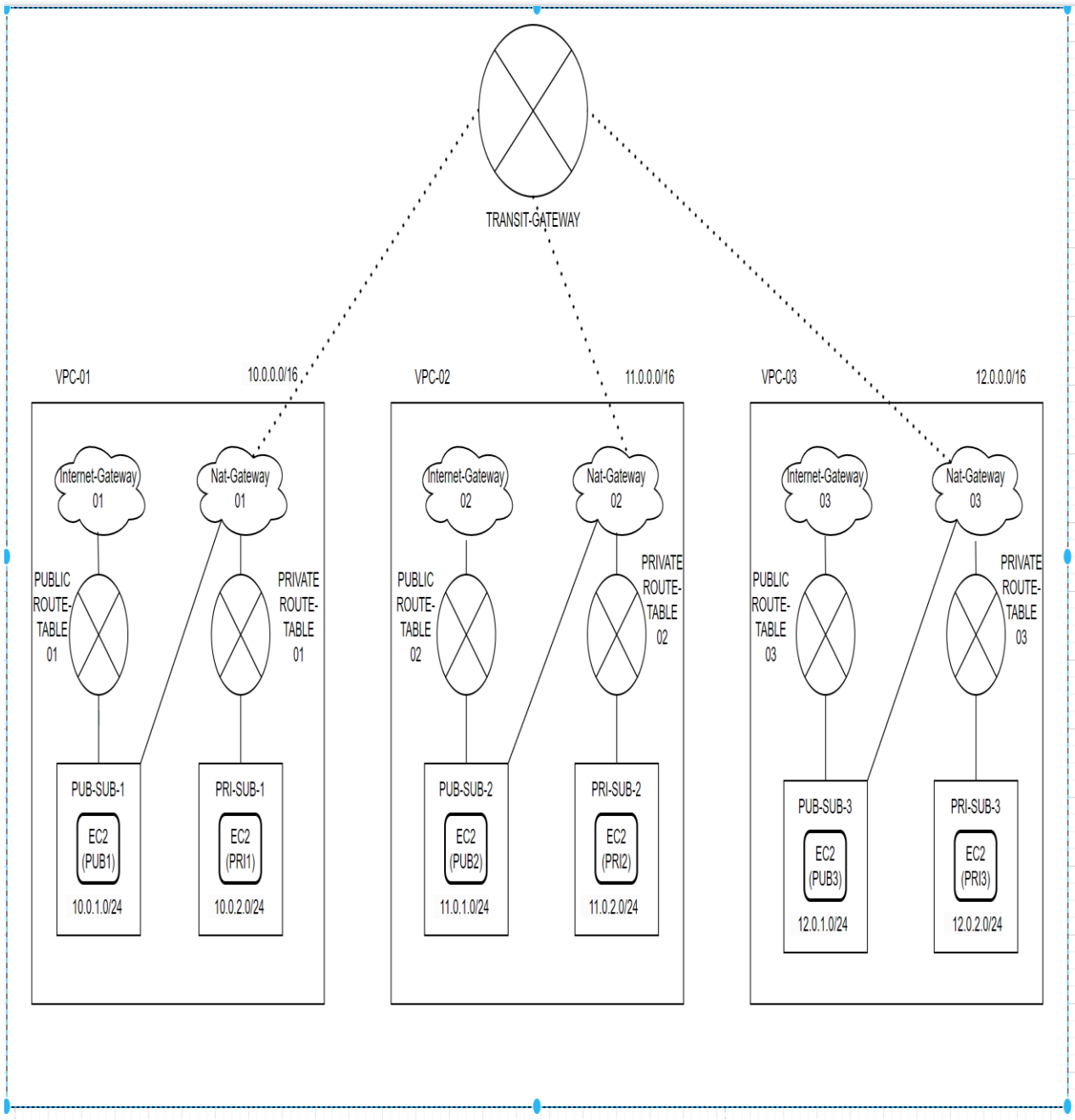
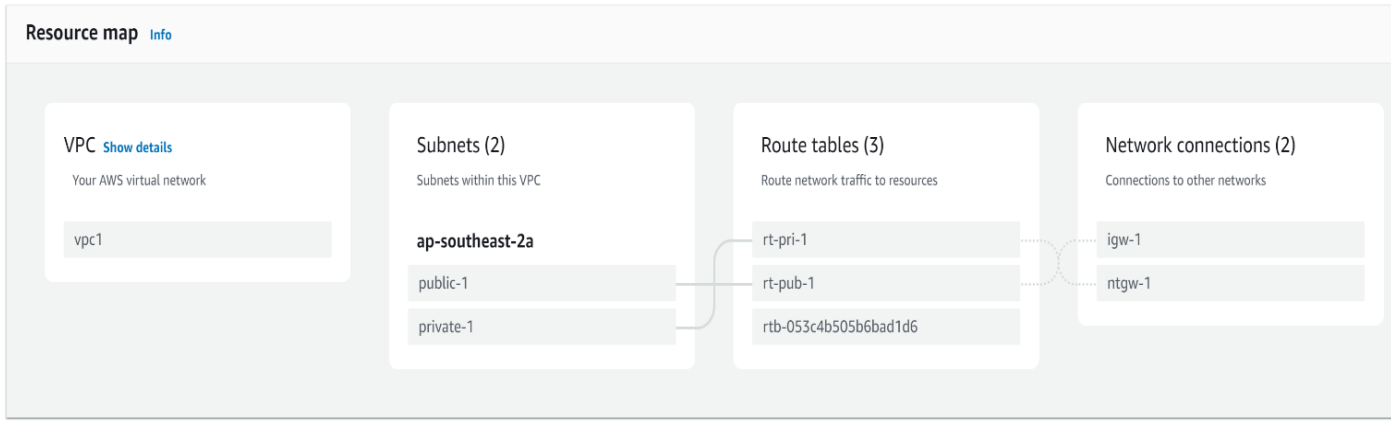
MULTI-VPC NETWORKING SETUP ON AWS WITH EC-2 INSTANCES



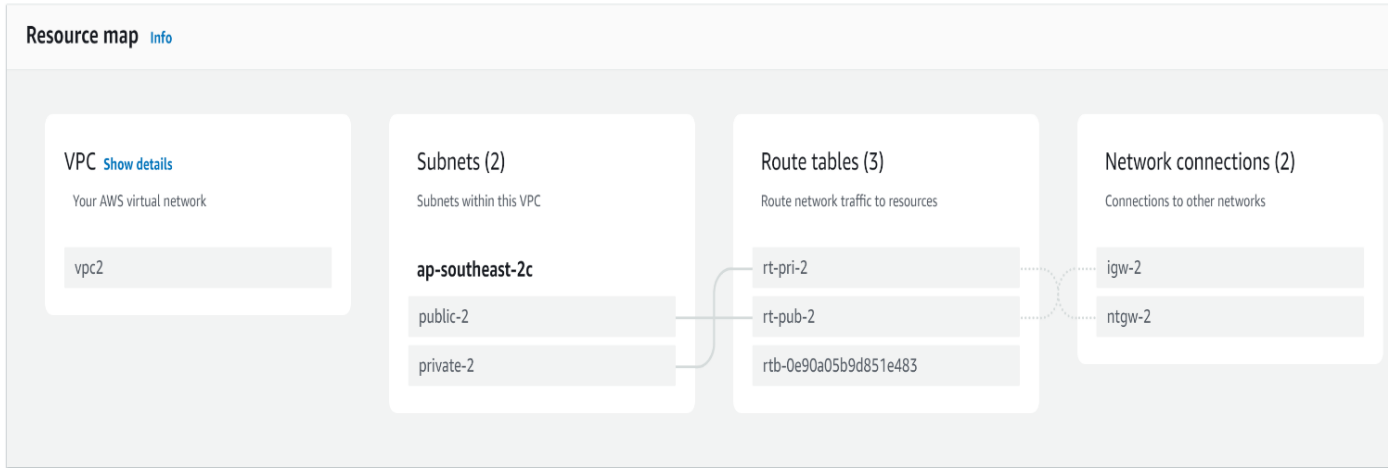
vpc-09164543bae905075 / vpc1

Details | Resource map | CIDRs | Flow logs | Tags | Integrations



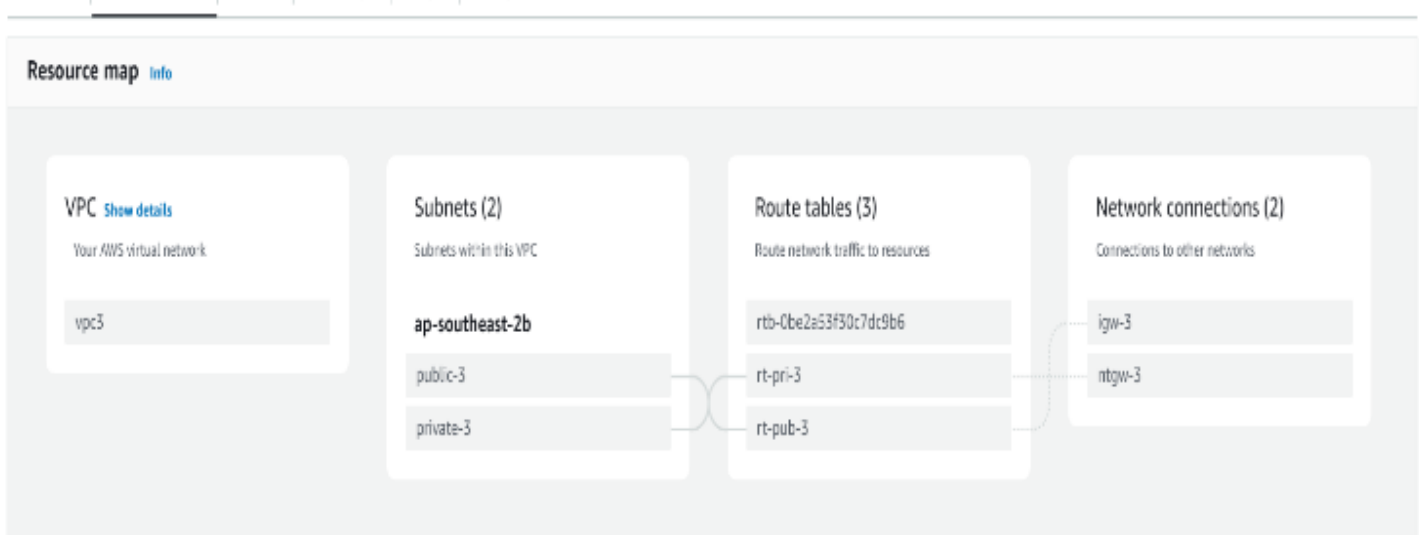
vpc-0e58b3dbc98c03c4f / vpc2

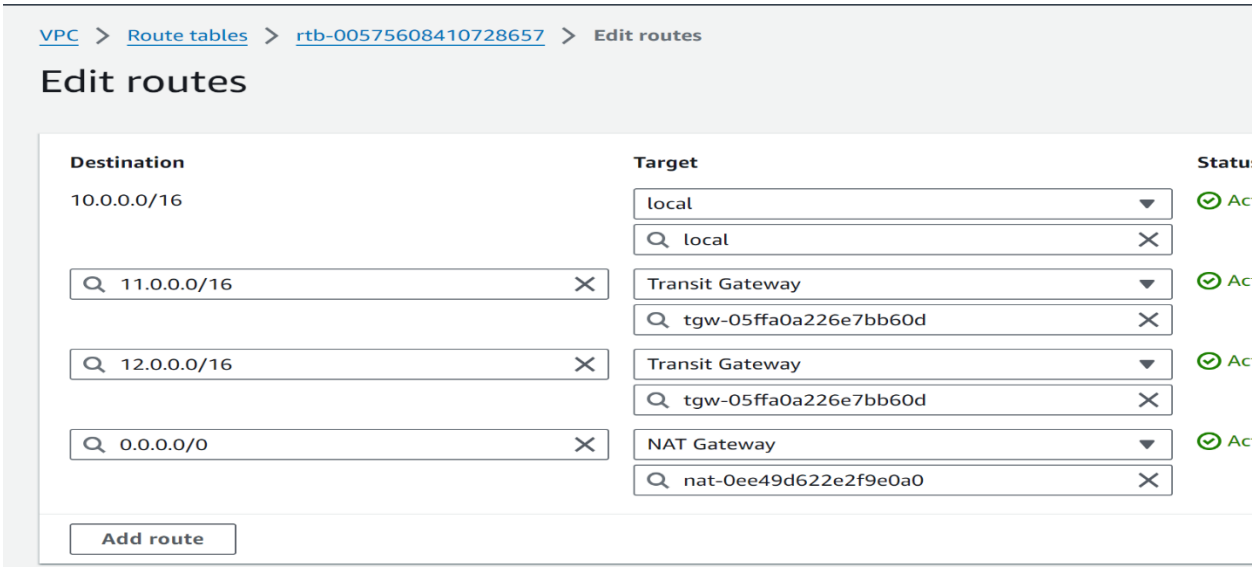
Details | Resource map | CIDRs | Flow logs | Tags | Integrations

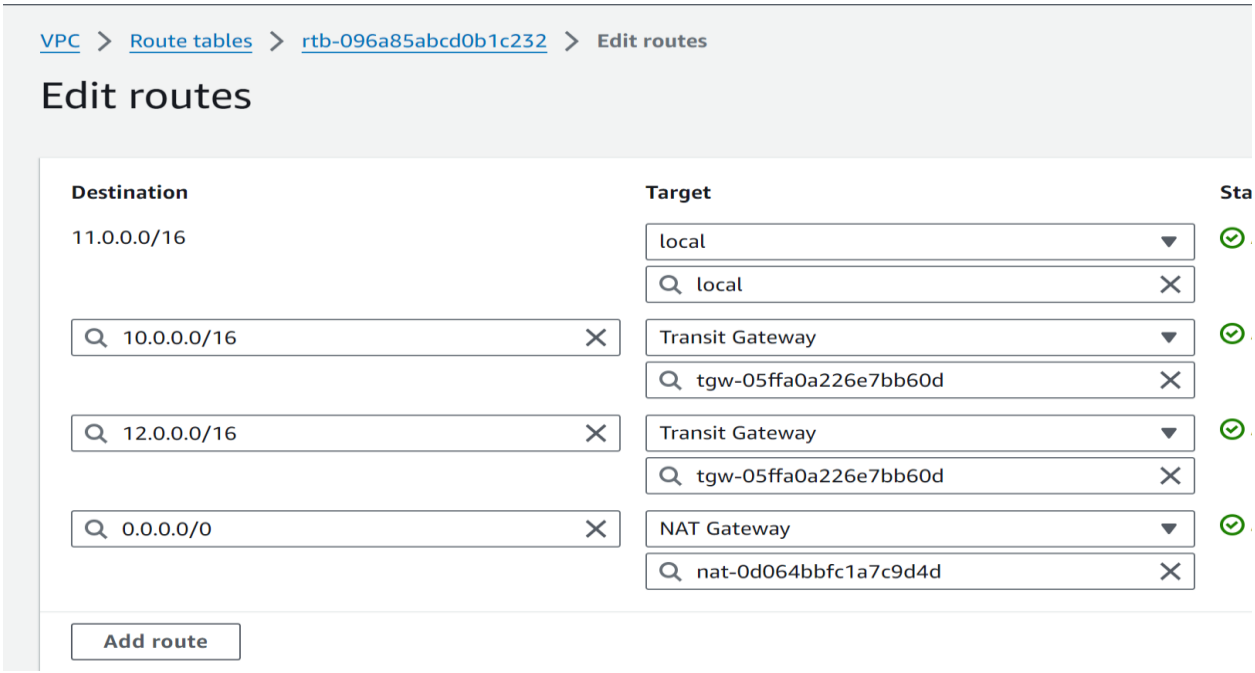


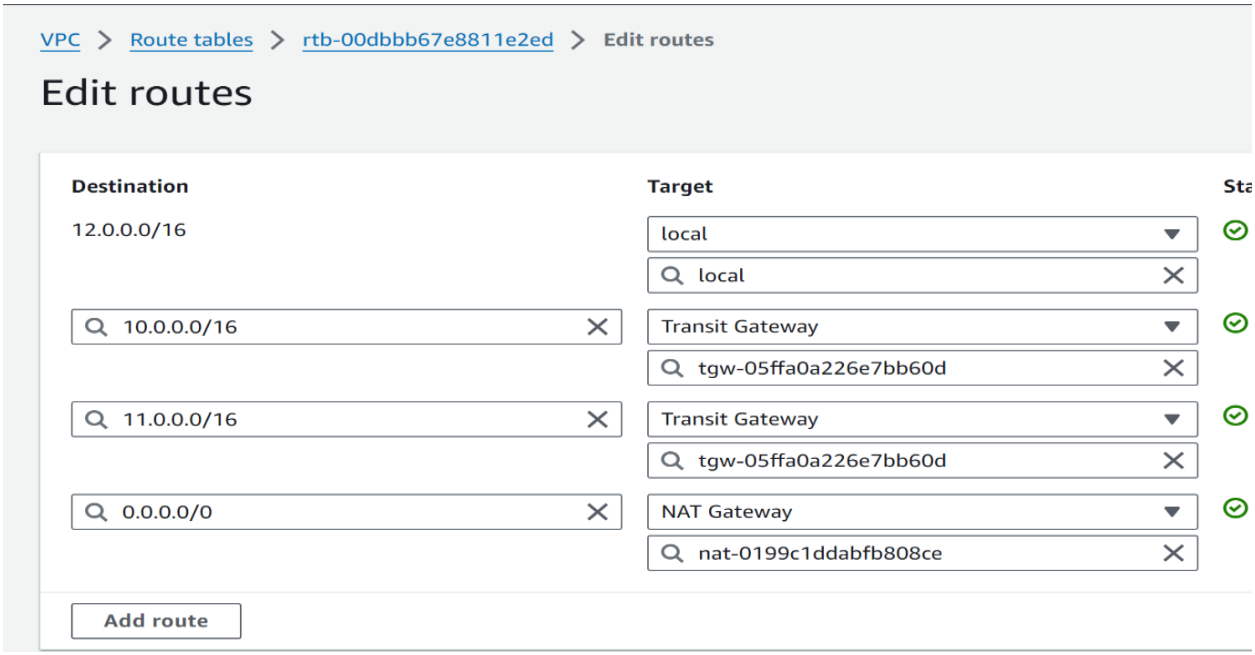
vpc-003c5913e97c7896c / vpc3

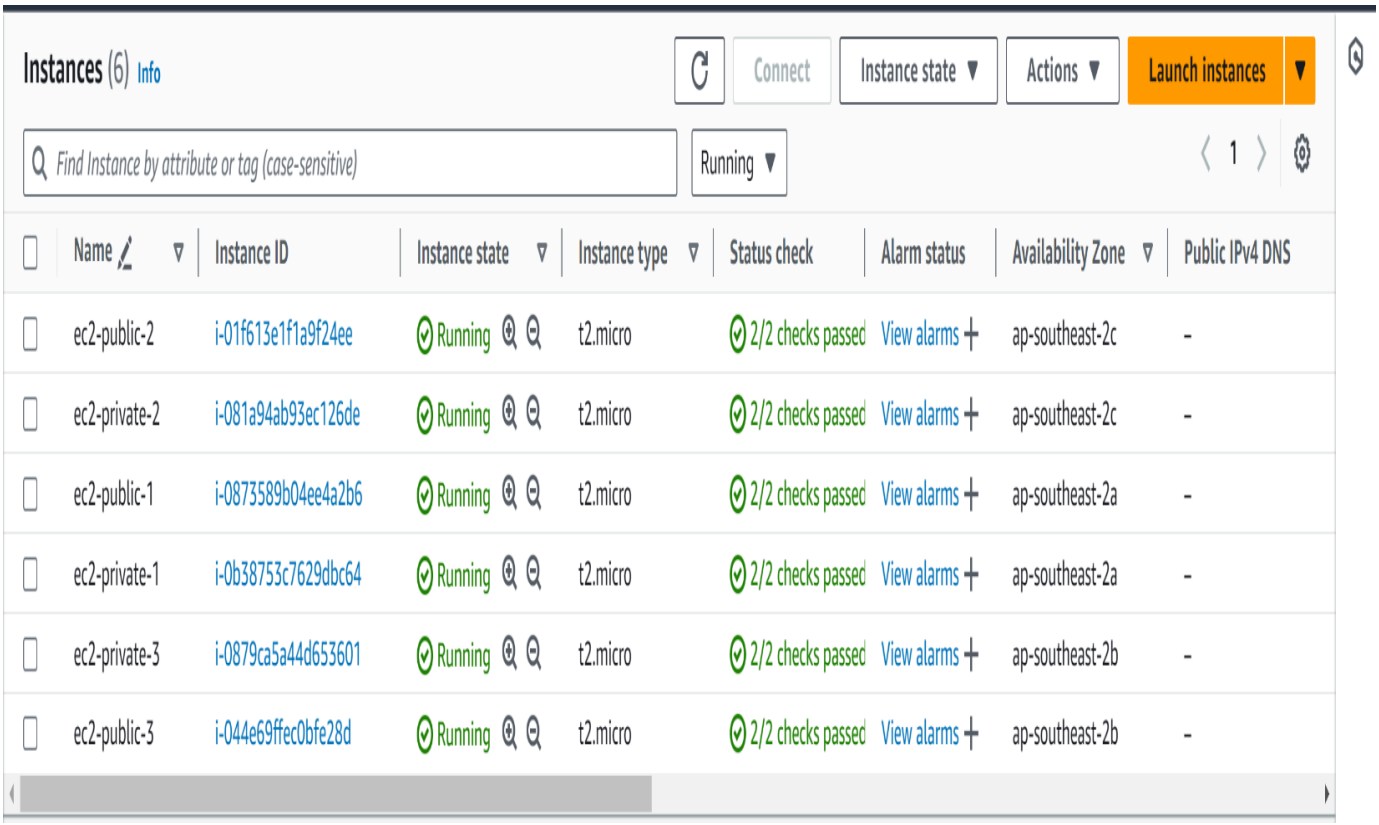
Details | Resource map | CIDRs | Flow logs | Tags | Integrations

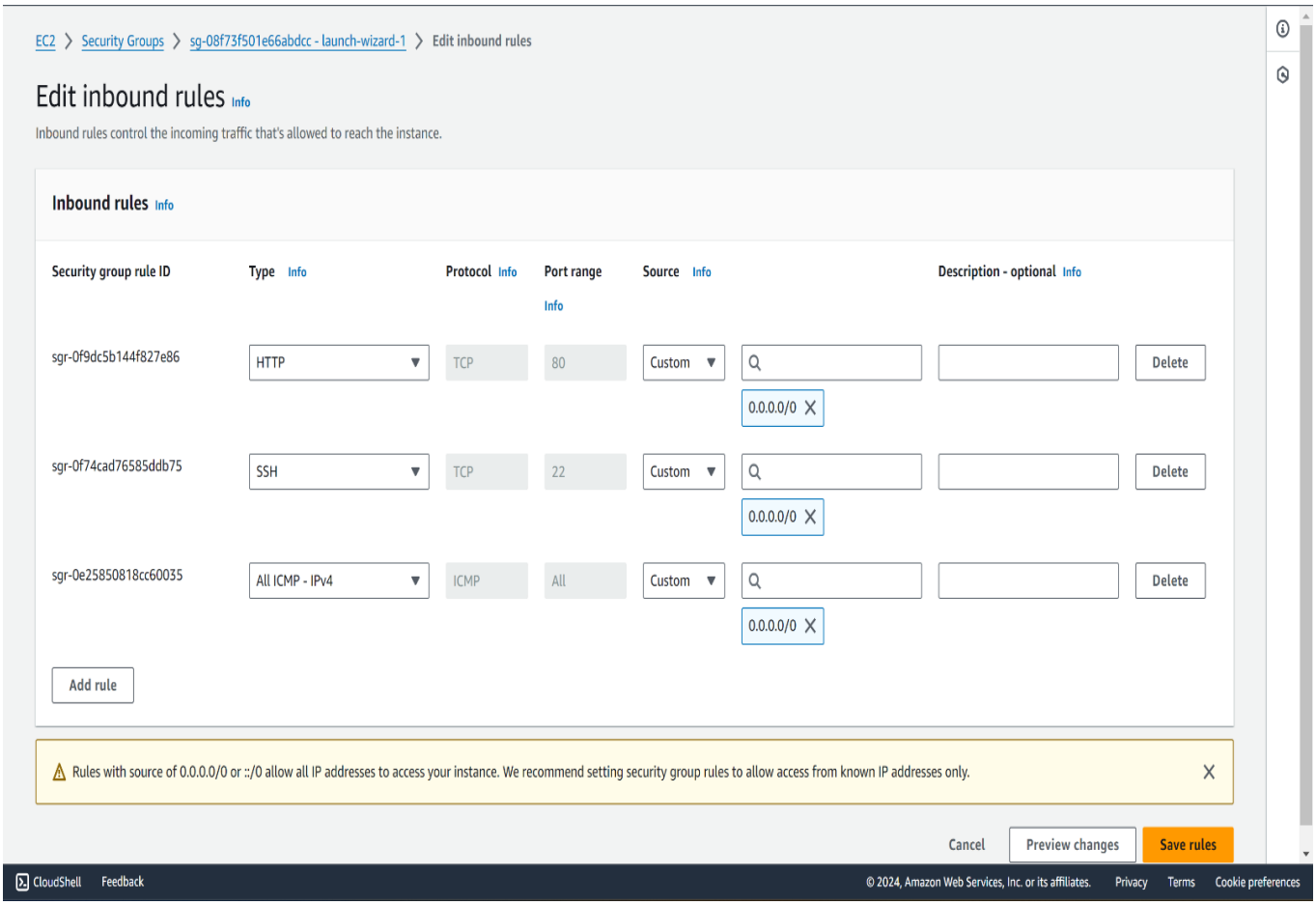



VPC > Route tables > rtb-00575608410728657 > Edit routes
Edit routes
Destination
Target
Statu
10.0.0.0/16
local
local
11.0.0.0/16
Transit Gateway
tgw-05ffa0a226e7bb60d
12.0.0.0/16
Transit Gateway
tgw-05ffa0a226e7bb60d
0.0.0.0/0
NAT Gateway
nat-0ee49d622e2f9e0a0
Add route


VPC > Route tables > rtb-096a85abcd0b1c232 > Edit routes
Edit routes
Destination
Target
Sta
11.0.0.0/16
local
local
10.0.0.0/16
Transit Gateway
tgw-05ffa0a226e7bb60d
12.0.0.0/16
Transit Gateway
tgw-05ffa0a226e7bb60d
0.0.0.0/0
NAT Gateway
nat-0d064bbfc1a7c9d4d
Add route


VPC > Route tables > rtb-00dbbb67e8811e2ed > Edit routes
Edit routes
Destination
Target
Sta
12.0.0.0/16
local
local
10.0.0.0/16
Transit Gateway
tgw-05ffa0a226e7bb60d
11.0.0.0/16
Transit Gateway
tgw-05ffa0a226e7bb60d
0.0.0.0/0
NAT Gateway
nat-0199c1ddabfb808ce
Add route


Instances (6) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
Running
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
ec2-public-2 i-01f613e1f1a9f24ee Running t2.micro 2/2 checks passed View alarms ap-southeast-2c -
ec2-private-2 i-081a94ab93ec126de Running t2.micro 2/2 checks passed View alarms ap-southeast-2c -
ec2-public-1 i-0873589b04ee4a2b6 Running t2.micro 2/2 checks passed View alarms ap-southeast-2a -
ec2-private-1 i-0b38753c7629dbc64 Running t2.micro 2/2 checks passed View alarms ap-southeast-2a -
ec2-private-3 i-0879ca5a44d653601 Running t2.micro 2/2 checks passed View alarms ap-southeast-2b -
ec2-public-3 i-044e69ffec0bfe28d Running t2.micro 2/2 checks passed View alarms ap-southeast-2b -


EC2 > Security Groups > sg-08f73f501e66abdcc - launch-wizard-1 > Edit inbound rules
Edit inbound rules Info
Inbound rules control the incoming traffic that's allowed to reach the instance.
Inbound rules Info
Security group rule ID
Type Info
Protocol Info
Port range Info
Source Info
Description - optional Info
sgr-0f9dc5b144f827e86 HTTP TCP 80 Custom 0.0.0.0/0 Delete
sgr-0f74cad76585ddb75 SSH TCP 22 Custom 0.0.0.0/0 Delete
sgr-0e25850818cc60035 All ICMP - IPv4 ICMP All Custom 0.0.0.0/0 Delete
Add rule
Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.
Cancel
Preview changes
Save rules
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

```
ubuntu@ip-10-0-2-58:~$ ping 11.0.2.149
PING 11.0.2.149 (11.0.2.149) 56(84) bytes of data.
64 bytes from 11.0.2.149: icmp_seq=1 ttl=63 time=1.37 ms
64 bytes from 11.0.2.149: icmp_seq=2 ttl=63 time=1.24 ms
64 bytes from 11.0.2.149: icmp_seq=3 ttl=63 time=1.37 ms
64 bytes from 11.0.2.149: icmp_seq=4 ttl=63 time=1.25 ms
64 bytes from 11.0.2.149: icmp_seq=5 ttl=63 time=1.24 ms
64 bytes from 11.0.2.149: icmp_seq=6 ttl=63 time=1.26 ms
64 bytes from 11.0.2.149: icmp_seq=7 ttl=63 time=1.31 ms
^C
--- 11.0.2.149 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6010ms
rtt min/avg/max/mdev = 1.237/1.290/1.373/0.054 ms
ubuntu@ip-10-0-2-58:~$ curl 11.0.2.149
<h1> This is vm2</h1>
ubuntu@ip-10-0-2-58:~$
ubuntu@ip-10-0-2-58:~$
ubuntu@ip-10-0-2-58:~$
ubuntu@ip-10-0-2-58:~$ ping 12.0.2.191
PING 12.0.2.191 (12.0.2.191) 56(84) bytes of data.
64 bytes from 12.0.2.191: icmp_seq=1 ttl=63 time=1.29 ms
64 bytes from 12.0.2.191: icmp_seq=2 ttl=63 time=1.20 ms
64 bytes from 12.0.2.191: icmp_seq=3 ttl=63 time=1.20 ms
64 bytes from 12.0.2.191: icmp_seq=4 ttl=63 time=1.12 ms
64 bytes from 12.0.2.191: icmp_seq=5 ttl=63 time=1.16 ms
64 bytes from 12.0.2.191: icmp_seq=6 ttl=63 time=1.10 ms
64 bytes from 12.0.2.191: icmp_seq=7 ttl=63 time=1.26 ms
^C
--- 12.0.2.191 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6009ms
rtt min/avg/max/mdev = 1.099/1.189/1.293/0.064 ms
ubuntu@ip-10-0-2-58:~$ curl 12.0.2.191
<h1> This is vm3</h1>
ubuntu@ip-10-0-2-58:~$
```

i-0873589b04ee4a2b6 (public-1)

PublicIPs: 13.210.241.35 PrivateIPs: 10.0.1.12

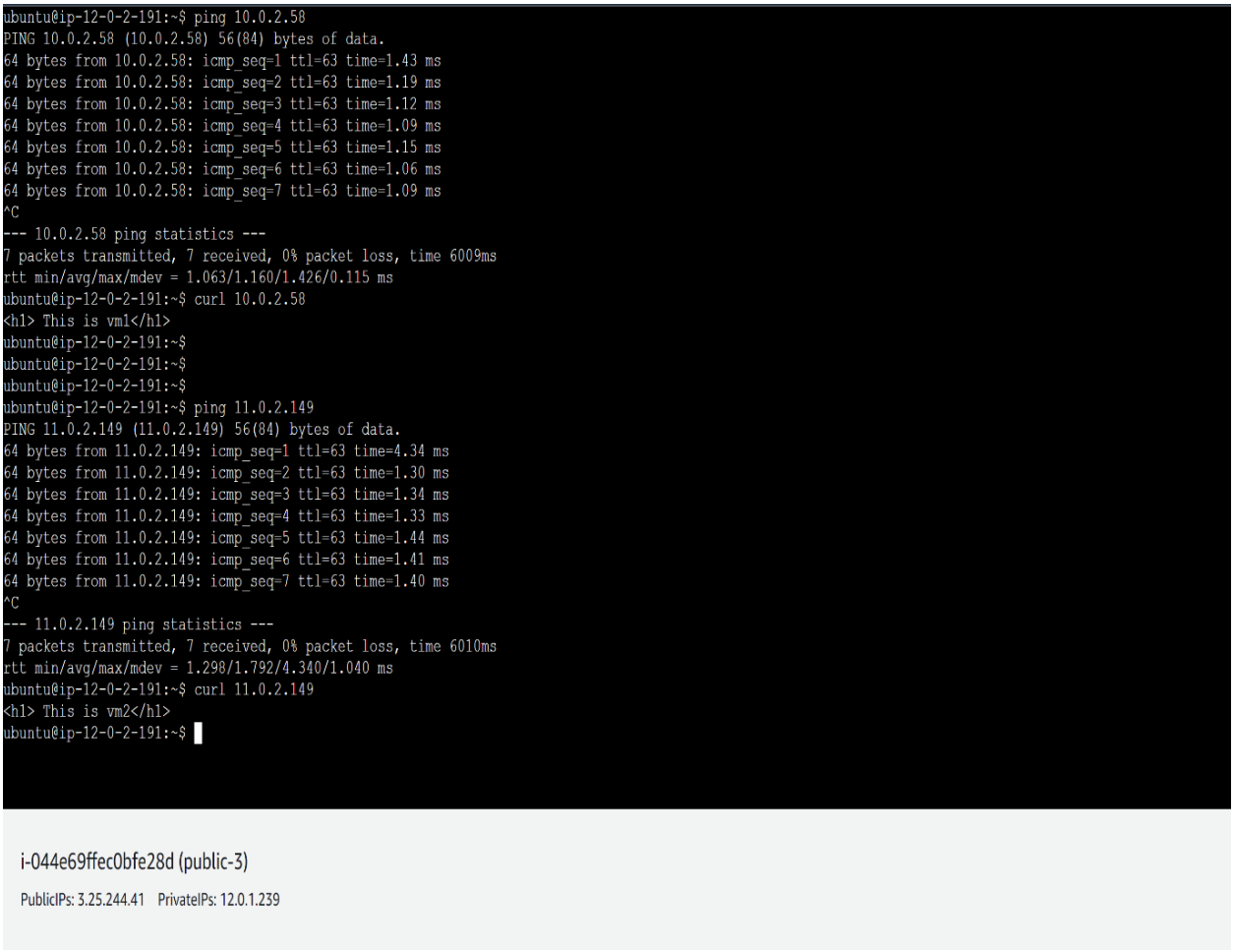
X

```
ubuntu@ip-11-0-2-149:~$ ping 10.0.2.58
PING 10.0.2.58 (10.0.2.58) 56(84) bytes of data.
64 bytes from 10.0.2.58: icmp_seq=1 ttl=63 time=1.97 ms
64 bytes from 10.0.2.58: icmp_seq=2 ttl=63 time=1.28 ms
64 bytes from 10.0.2.58: icmp_seq=3 ttl=63 time=1.25 ms
64 bytes from 10.0.2.58: icmp_seq=4 ttl=63 time=1.37 ms
64 bytes from 10.0.2.58: icmp_seq=5 ttl=63 time=1.27 ms
64 bytes from 10.0.2.58: icmp_seq=6 ttl=63 time=1.25 ms
64 bytes from 10.0.2.58: icmp_seq=7 ttl=63 time=1.27 ms
^C
--- 10.0.2.58 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6009ms
rtt min/avg/max/mdev = 1.251/1.380/1.974/0.245 ms
ubuntu@ip-11-0-2-149:~$ curl 10.0.2.58
<h1> This is vm1</h1>
ubuntu@ip-11-0-2-149:~$
ubuntu@ip-11-0-2-149:~$
ubuntu@ip-11-0-2-149:~$
ubuntu@ip-11-0-2-149:~$ ping 12.0.2.191
PING 12.0.2.191 (12.0.2.191) 56(84) bytes of data.
64 bytes from 12.0.2.191: icmp_seq=1 ttl=63 time=1.48 ms
64 bytes from 12.0.2.191: icmp_seq=2 ttl=63 time=1.38 ms
64 bytes from 12.0.2.191: icmp_seq=3 ttl=63 time=1.38 ms
64 bytes from 12.0.2.191: icmp_seq=4 ttl=63 time=1.39 ms
64 bytes from 12.0.2.191: icmp_seq=5 ttl=63 time=1.34 ms
64 bytes from 12.0.2.191: icmp_seq=6 ttl=63 time=1.44 ms
64 bytes from 12.0.2.191: icmp_seq=7 ttl=63 time=1.38 ms
^C
--- 12.0.2.191 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6010ms
rtt min/avg/max/mdev = 1.342/1.400/1.482/0.043 ms
ubuntu@ip-11-0-2-149:~$ curl 12.0.2.191
<h1> This is vm3</h1>
ubuntu@ip-11-0-2-149:~$
```

i-01f613e1f1a9f24ee (public-2)

PublicIPs: 52.63.213.62 PrivateIPs: 11.0.1.213

X



1.Designed and Implemented Multi-VPC Architecture:

- Successfully created three Virtual Private Clouds (VPCs) with public and private subnets.
- Configured route tables, internet gateways, and NAT gateways to ensure secure communication and internet access.
- Demonstrated expertise in network segmentation and isolation.

2.Deployed EC2 Instances Across Subnets:

- Provisioned EC2 instances in both public and private subnets.
- Hosted a dummy website on private instances.
- Validated connectivity by pinging and curling between instances.

3.Transit Gateway for Inter-VPC Communication:

- Set up a transit gateway to enable seamless communication between private subnets across VPCs.
- Achieved efficient and scalable inter-VPC connectivity.
- Demonstrated proficiency in AWS networking services.