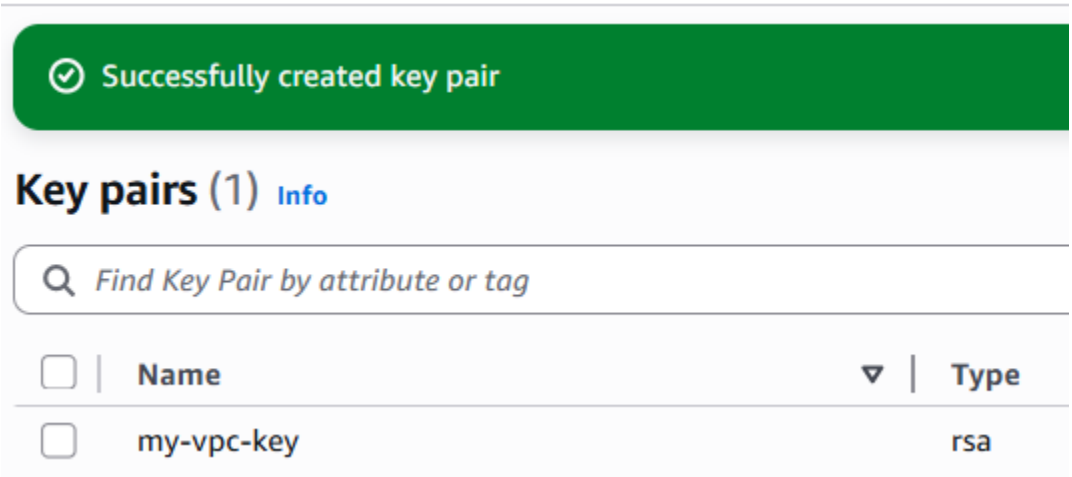
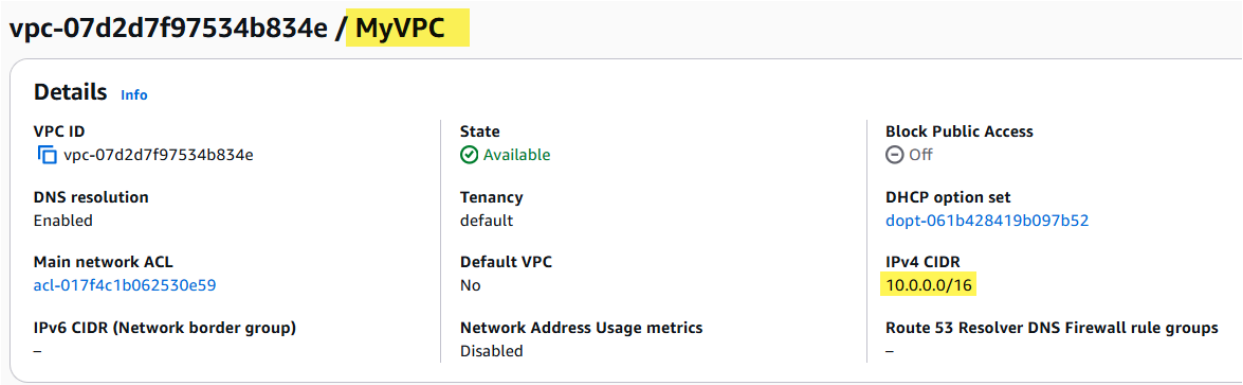
AWS VPC Assignment

Step 1: Create SSH Key Pair



Step 2: Create VPC



Step 3: Create Internet Gateway

igw-09d267a106708f221 / MyVPC-IGW

Details Info

Internet gateway ID	State	VPC ID
igw-09d267a106708f221	Attached	vpc-07d2d7f97534b834e | MyVPC

Tags

Search tags

Key	Value
Name	MyVPC-IGW

Step 4: Create Subnets

Name	Subnet ID	State	IPv4 CIDR
Public-Subnet-1	subnet-0ffc0d70540d9d5d9	Available	10.0.1.0/24
Public-Subnet-2	subnet-07cd8b20761170488	Available	10.0.2.0/24
Private-Subnet-1	subnet-008b9a552d348cdd3	Available	10.0.3.0/24
Private-Subnet-2	subnet-07b43612c78200147	Available	10.0.4.0/24

Step 5: Create NAT Gateway

nat-0427fa6c35746a15c / MyVPC-NAT

Details

NAT gateway ID
nat-0427fa6c35746a15c

NAT gateway ARN
arn:aws:ec2:us-east-1:307946636515:natgateway/nat-0427fa6c35746a15c

VPC
vpc-07d2d7f97534b834e / MyVPC

Connectivity type
Public

Primary public IPv4 address
44.199.88.116

Subnet
subnet-0ffc0d70540d9d5d9 / Public-Subnet-1

Step 6: Create Route Tables

rtb-0026a73ca15ad7f8b / Public-RT

Details Info

Route table ID
rtb-0026a73ca15ad7f8b

Main
No

VPC
vpc-07d2d7f97534b834e | MyVPC

Owner ID
307946636515

Routes | Subnet associations | Edge associations | Route propagation | Tags

Routes (2)

Filter routes

Destination	Target	Status
0.0.0.0/0	igw-09d267a106708f221	Active
10.0.0.0/16	local	Active

rtb-044c102a663241972 / Private-RT

Details Info

Route table ID
rtb-044c102a663241972

Main
No

VPC
vpc-07d2d7f97534b834e | MyVPC

Owner ID
307946636515

Routes | Subnet associations | Edge associations | Route propagation | Tags

Routes (2)

Filter routes

Destination	Target	Status
0.0.0.0/0	nat-0427fa6c35746a15c	Active
10.0.0.0/16	local	Active

Step 7: Associate Subnets with Route Tables

rtb-0026a73ca15ad7f8b / Public-RT

Details | Routes | Subnet associations | Edge associations | Route propagation | Tags

Explicit subnet associations (2)

Find subnet association

Name	Subnet ID	IPv4 CIDR
Public-Subnet-1	subnet-0ffc0d70540d9d5d9	10.0.1.0/24
Public-Subnet-2	subnet-07cd8b20761170488	10.0.2.0/24

rtb-044c102a663241972 / Private-RT

Details | Routes | Subnet associations | Edge associations | Route propagation | Tags

Explicit subnet associations (2)

Find subnet association

Name	Subnet ID	IPv4 CIDR
Private-Subnet-2	subnet-07b43612c78200147	10.0.4.0/24
Private-Subnet-1	subnet-008b9a552d348cdd3	10.0.3.0/24

Step 8: Create Security Groups

sg-068ad613bd09c5b52 - Bastion-SG

Details

Security group name	Security group ID	Description
Bastion-SG	sg-068ad613bd09c5b52	Bastion-SG: SSH access for bastion host
Owner	**Inbound rules count**	**Outbound rules count**
307946636515	1 Permission entry	1 Permission entry

Inbound rules | Outbound rules | Sharing - *new* | VPC associations - *new* | Tags

Inbound rules (1)

Search

Name	Security group rule ID	IP version	Type	Protocol	Port
–	sgr-097946a038b783b97	IPv4	SSH	TCP	22

sg-027ebeca9b3f94370 - Private-SG

Details

Security group name	Security group ID	Description
Private-SG	sg-027ebeca9b3f94370	Private-SG: Access for private instances
Owner	**Inbound rules count**	**Outbound rules count**
307946636515	2 Permission entries	1 Permission entry

Inbound rules | Outbound rules | Sharing - *new* | VPC associations - *new* | Tags

Inbound rules (2)

Search

Name	Security group rule ID	IP version	Type	Protocol	Port
–	sgr-080438ed22a6f8d2e	IPv4	All ICMP - IPv4	ICMP	All
–	sgr-06348703ab02891ea	–	SSH	TCP	22

Step 9: Launch EC2 Instances

Instances (2) Info

Find Instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type
Bastion-Host	i-06e68037eb3b7caa1	Running	t2.micro
Private-Instance	i-0bcc3fa827f539a2a	Running	t2.micro

Step 10: Connect to Instances

```
Admin@DESKTOP-AFSF9SU MINGW64 /d/devops_codes
$ chmod 400 my-vpc-key.pem

Admin@DESKTOP-AFSF9SU MINGW64 /d/devops_codes
$ ls -l my-vpc-key.pem
-r--r--r-- 1 Admin 197121 1679 Sep 10 10:49 my-vpc-key.pem
```

```
Admin@DESKTOP-AFSF9SU MINGW64 /d/devops_codes
$ ssh -i my-vpc-key.pem ec2-user@3.237.101.48
       #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       /m/'
[ec2-user@bastion-host ~]$
```

```
Admin@DESKTOP-AFSF9SU MINGW64 /d/devops_codes
$ scp -i my-vpc-key.pem  my-vpc-key.pem ec2-user@3.237.101.48:~/
my-vpc-key.pem

Admin@DESKTOP-AFSF9SU MINGW64 /d/devops_codes
$
```

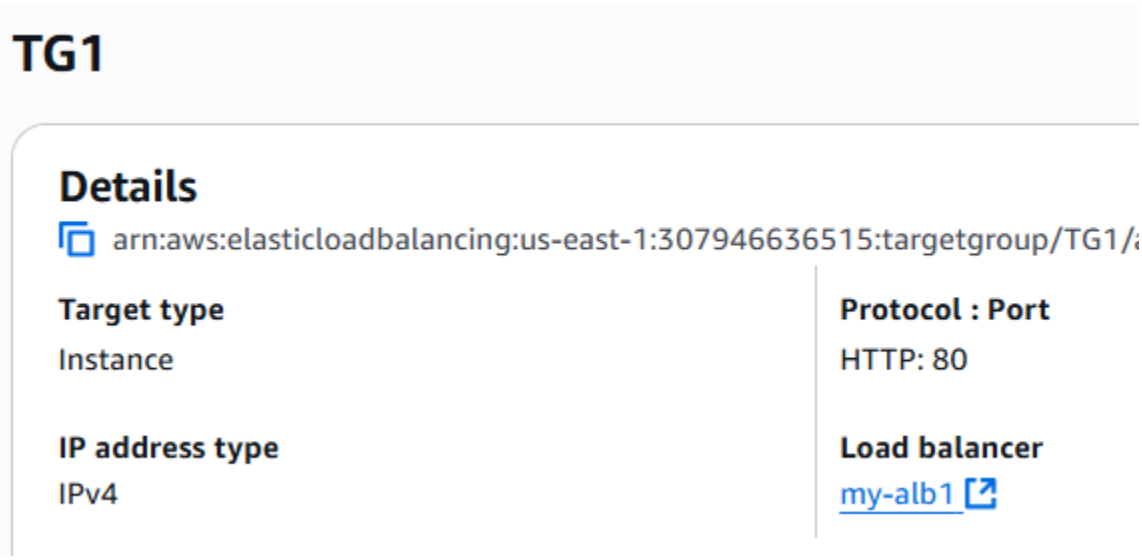
```
ec2-user@bastion-host:~
[ec2-user@bastion-host ~]$ ls -l
total 4
-r--r--r--. 1 ec2-user ec2-user 1679 Sep 10 05:26 my-vpc-key.pem
[ec2-user@bastion-host ~]$
```

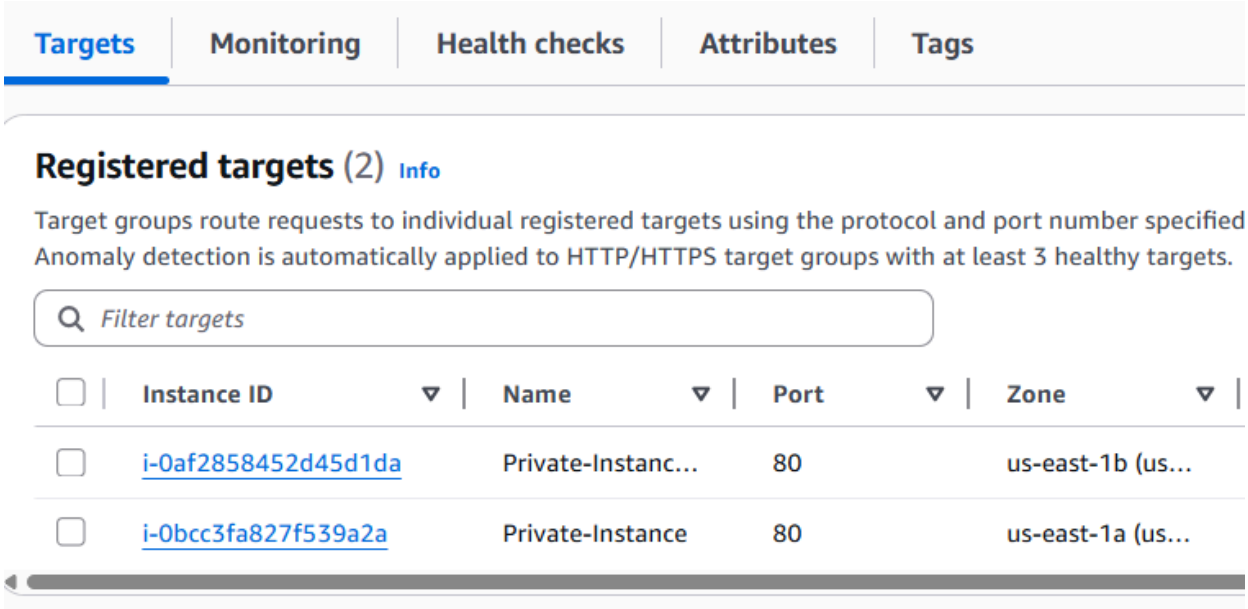
```
[ec2-user@bastion-host ~]$ sudo chmod 400 my-vpc-key.pem
[ec2-user@bastion-host ~]$ ls -l
total 4
-r--------. 1 ec2-user ec2-user 1679 Sep 10 05:26 my-vpc-key.pem
[ec2-user@bastion-host ~]$ ssh -i my-vpc-key.pem ec2-user@10.0.3.23
       #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@private-instance ~]$ |
```

```
[ec2-user@private-instance ~]$ ping google.com
PING google.com (142.251.179.113) 56(84) bytes of data.
64 bytes from pd-in-f113.1e100.net (142.251.179.113): icmp_seq=1 ttl=105 time=3.04 ms
64 bytes from pd-in-f113.1e100.net (142.251.179.113): icmp_seq=2 ttl=105 time=2.42 ms
64 bytes from pd-in-f113.1e100.net (142.251.179.113): icmp_seq=3 ttl=105 time=2.39 ms
64 bytes from pd-in-f113.1e100.net (142.251.179.113): icmp_seq=4 ttl=105 time=2.42 ms
64 bytes from pd-in-f113.1e100.net (142.251.179.113): icmp_seq=5 ttl=105 time=2.44 ms
^C
--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 2.387/2.542/3.041/0.249 ms
[ec2-user@private-instance ~]$ exit
exit
Connection to 10.0.3.23 closed.
[ec2-user@bastion-host ~]$ ping google.com
PING google.com (172.253.63.138) 56(84) bytes of data.
64 bytes from bi-in-f138.1e100.net (172.253.63.138): icmp_seq=1 ttl=106 time=2.38 ms
64 bytes from bi-in-f138.1e100.net (172.253.63.138): icmp_seq=2 ttl=106 time=2.42 ms
64 bytes from bi-in-f138.1e100.net (172.253.63.138): icmp_seq=3 ttl=106 time=2.53 ms
64 bytes from bi-in-f138.1e100.net (172.253.63.138): icmp_seq=4 ttl=106 time=2.46 ms
64 bytes from bi-in-f138.1e100.net (172.253.63.138): icmp_seq=5 ttl=106 time=2.42 ms
^C
```

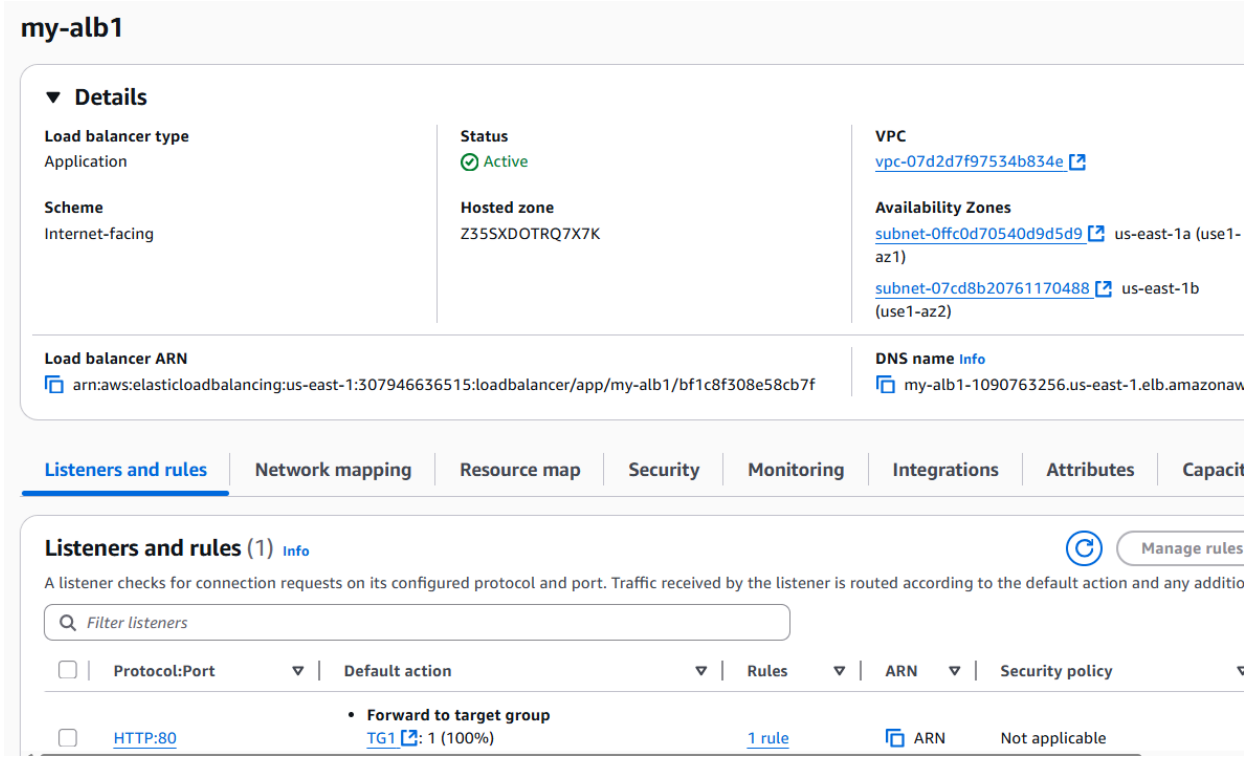
Step 10: Load Balancer Setup

Create Target Group





Create ALB



Security groups — important

- ALB SG (sg-alb): inbound 80 (or 443) from 0.0.0.0/0. Outbound to anywhere (default).

- App EC2 SG (sg-app): DO NOT open port 80 to 0.0.0.0/0. Instead add an inbound rule allowing the ALB SG as the source:
- Type: Custom TCP (port 80)
- Source: sg-alb (use the ALB security group id)
- This ensures only ALB can talk to your private instances on the app port.
- Console path: EC2 → Security Groups → Select sg-app → Edit inbound rules → Add rule → choose “Custom” and enter the ALB SG id in “Source”.

Test the ALB

http://my-alb1-1090763256.us-east-1.elb.amazonaws.com/

---

private-instance-1

---

private-instance2