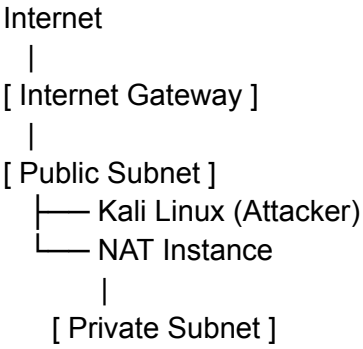
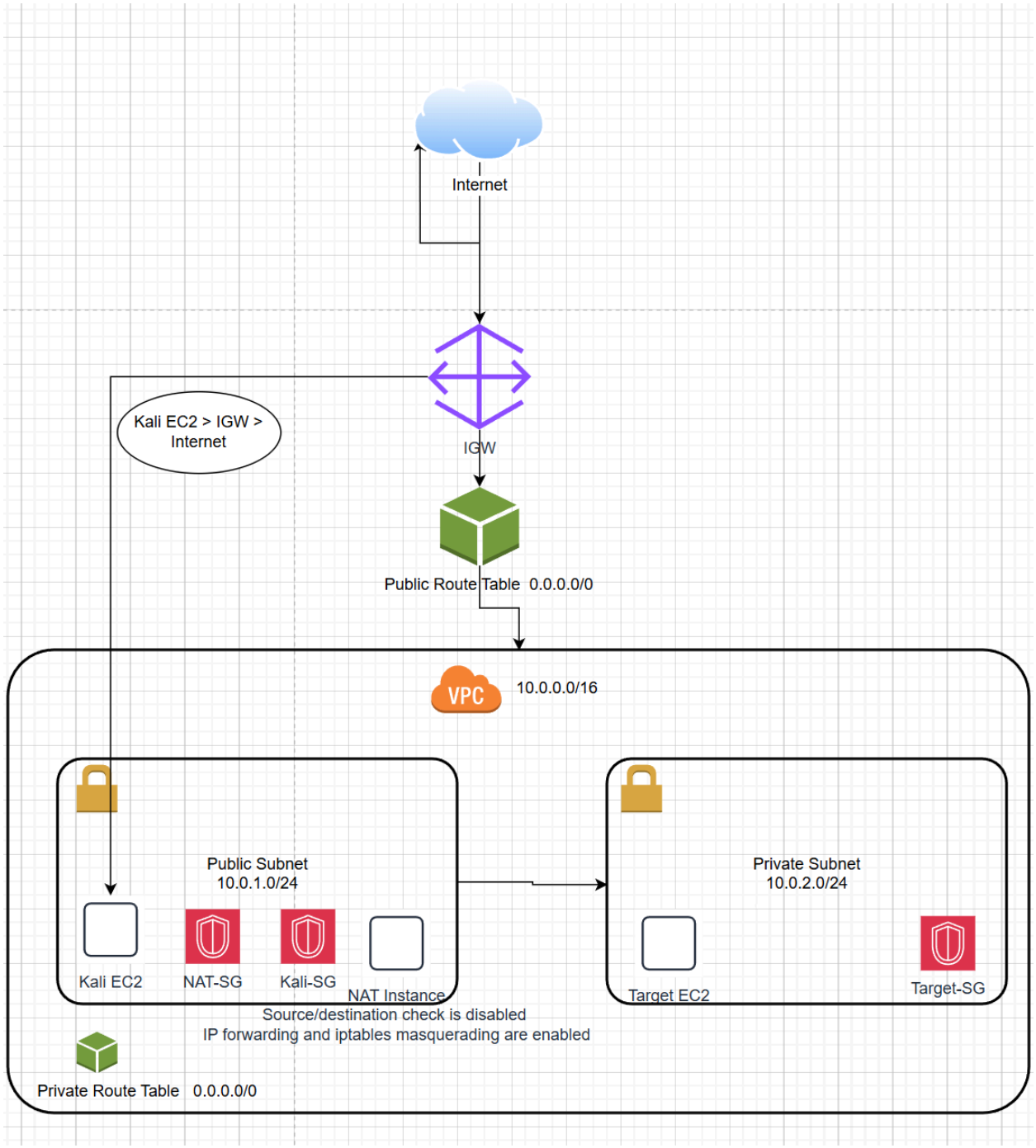


- Design the network
- Deploy Kali linux
- Create a NAT Instance (Free Tier)

This project demonstrates how to design and deploy a secure AWS VPC architecture using public and private subnets, a NAT instance, and a Kali Linux EC2 for controlled testing. It showcases cloud security fundamentals, network segmentation, Linux hardening, and secure outbound routing using free-tier resources.

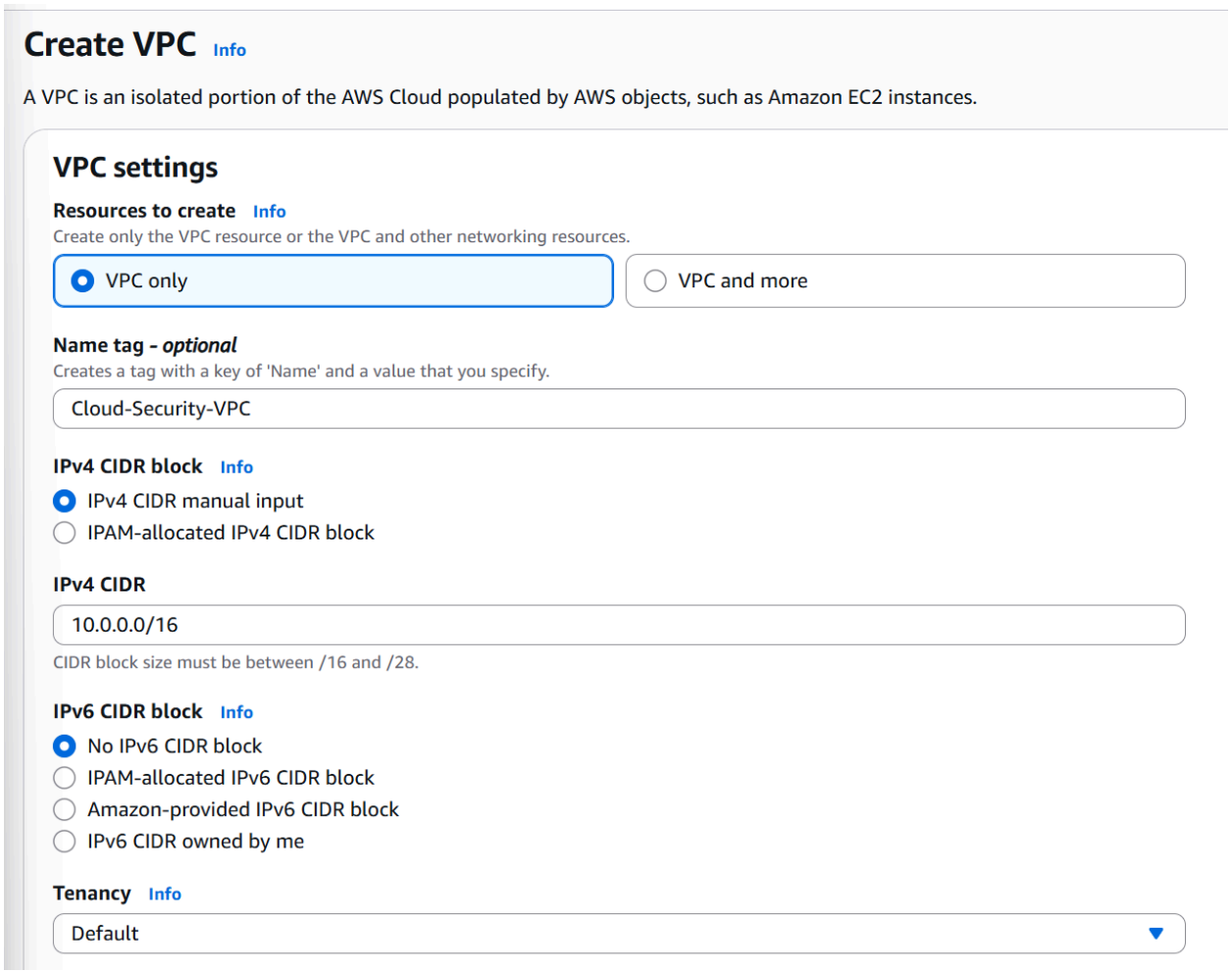

Internet
Kali EC2 > IGW > Internet
IGW
Public Route Table 0.0.0.0/0
VPC
10.0.0.0/16
Public Subnet
10.0.1.0/24
Private Subnet
10.0.2.0/24
Kali EC2
NAT-SG
Kali-SG
NAT Instance
Source/destination check is disabled
IP forwarding and iptables masquerading are enabled
Target EC2
Target-SG
Private Route Table 0.0.0.0/0

Network Foundation (AWS SAA CORE)

1. Create the VPC

- AWS Console > VPC > Create VPC

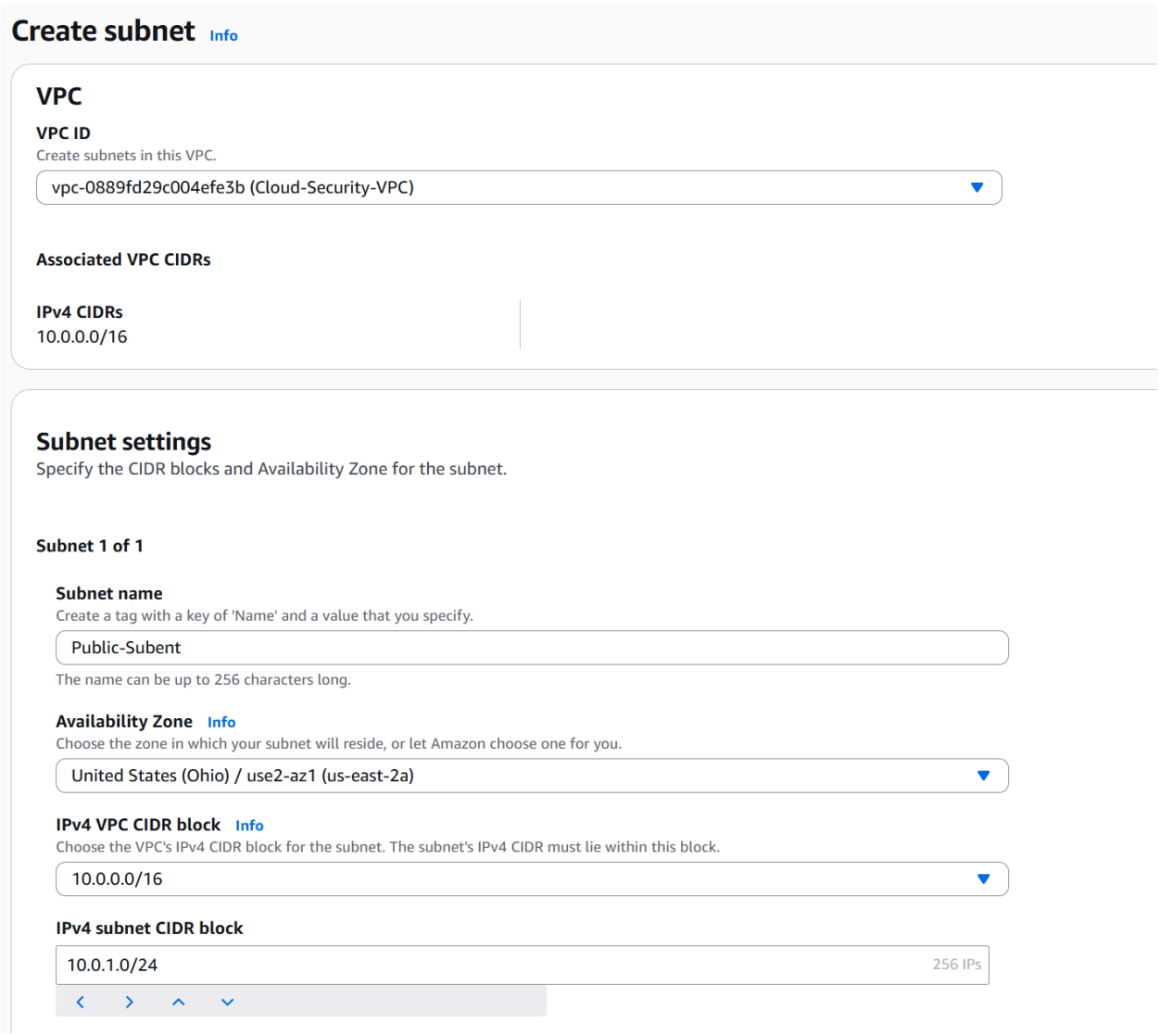
- Name: Cloud-Security-VPC
- IPv4 CIDR: 10.0.0.0/16
- Tenancy: Default

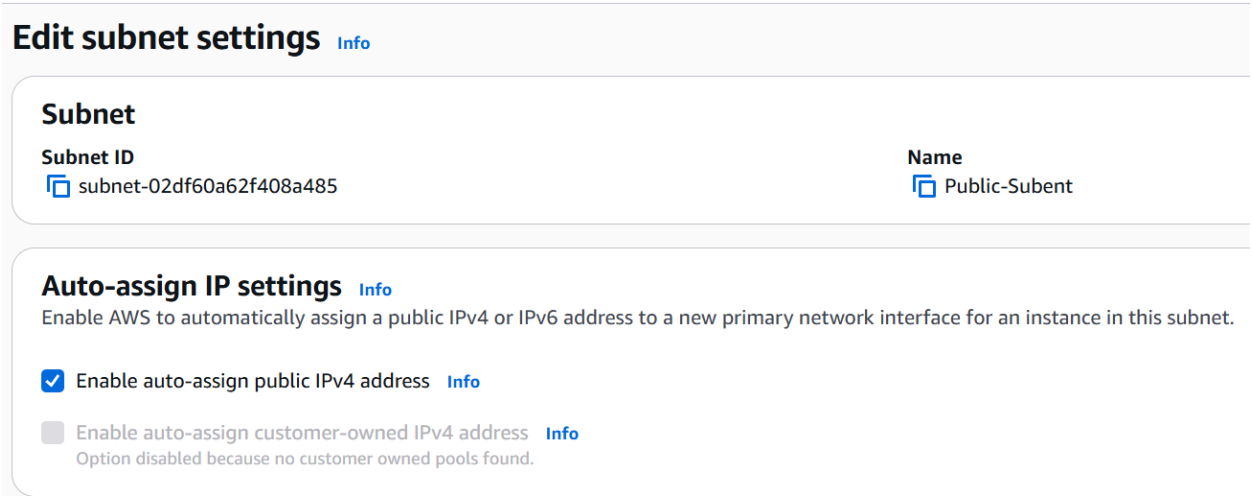


2. Create Subnets

   Public Subnet

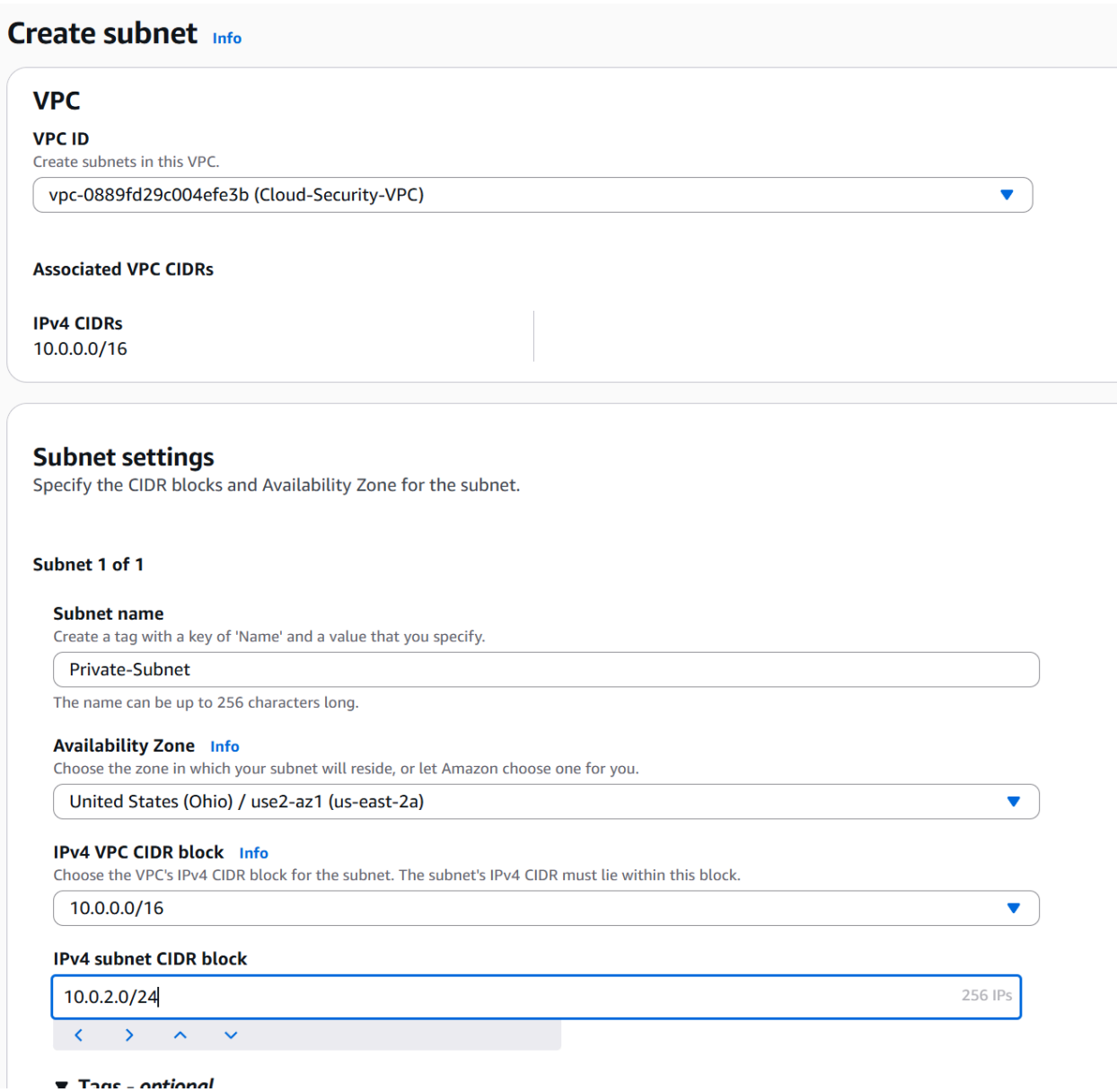
- Name: Public-Subent
- CIDR: 10.0.1.0/24
- Enable auto-assign public IP





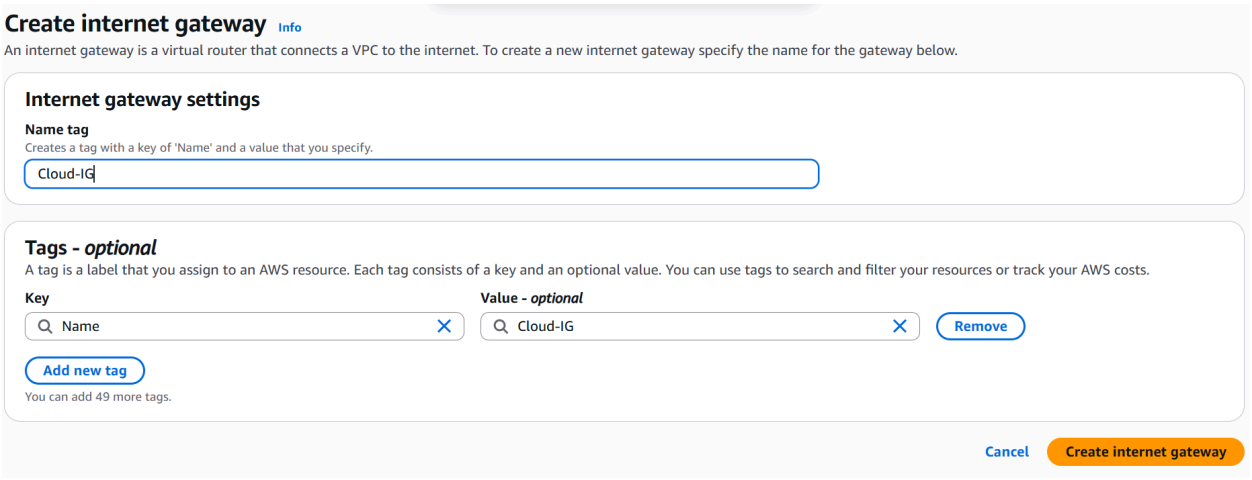
Private Subnet

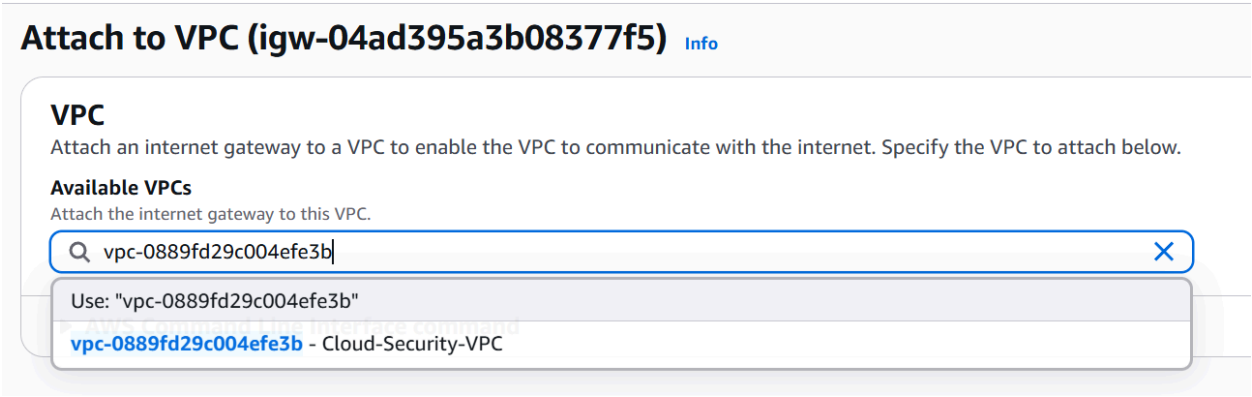
- Name: Private-Subnet
- CIDR: 10.0.2.0/24
- Auto-assign public IP: Disabled



3. Internet Gateway

- Create IG: Cloud-IG
- Attach to Cloud-Security-VPC

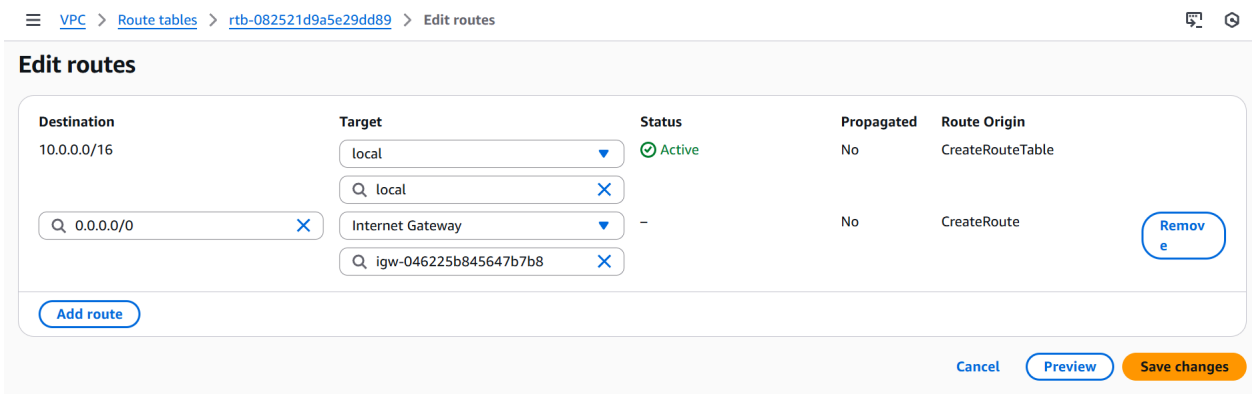


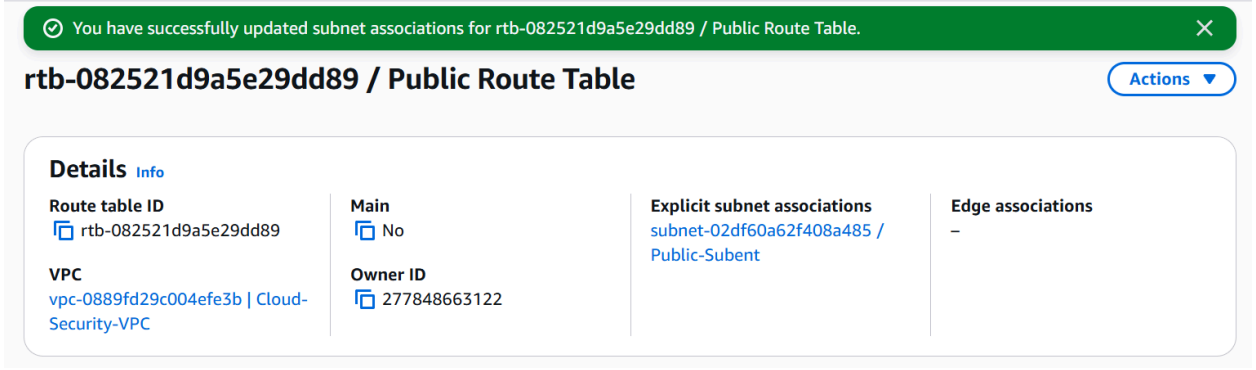


---

4. Route Tables

   Public Route Table
   Routes: 0.0.0.0/0 > IG
   Associate: Public-Subnet





Security Controls (AWS SAA CORE)

5. Security Groups
   Kali-SG
   Inbound: SSH (22) > /32
   Outbound: Allow all

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name Info

Kali-SG

Name cannot be edited after creation.

Description Info

Kali Linux

VPC Info

vpc-0889fd29c004efe3b (Cloud-Security-VPC)

Inbound rules Info

Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
SSH	TCP	22	My IP	[redacted]		Delete

Add rule

Outbound rules Info

Type Info	Protocol Info	Port range Info	Destination Info		Description - optional Info	
All traffic	All	All	Cu...	0.0.0.0/0		Delete

Add rule

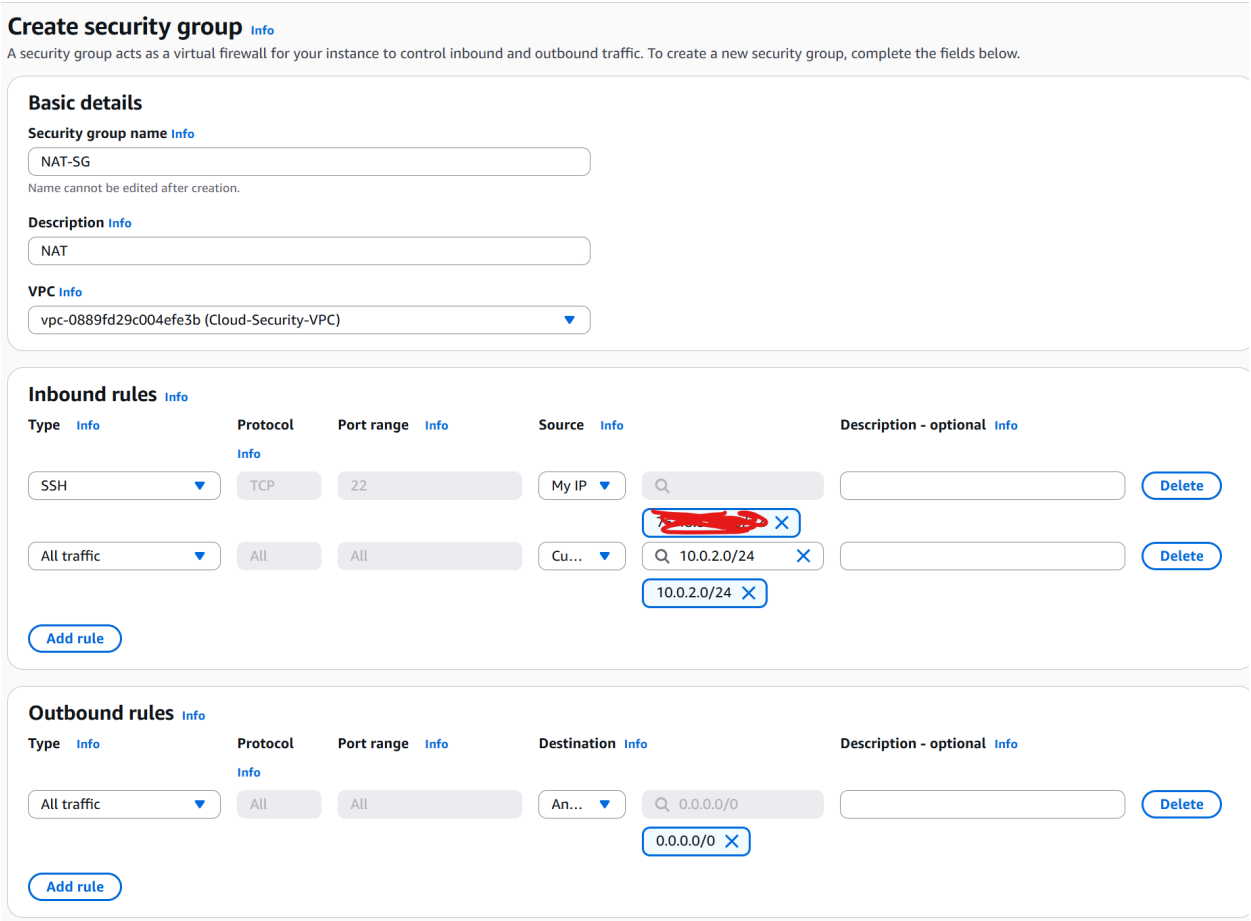
NAT-SG

Inbound:

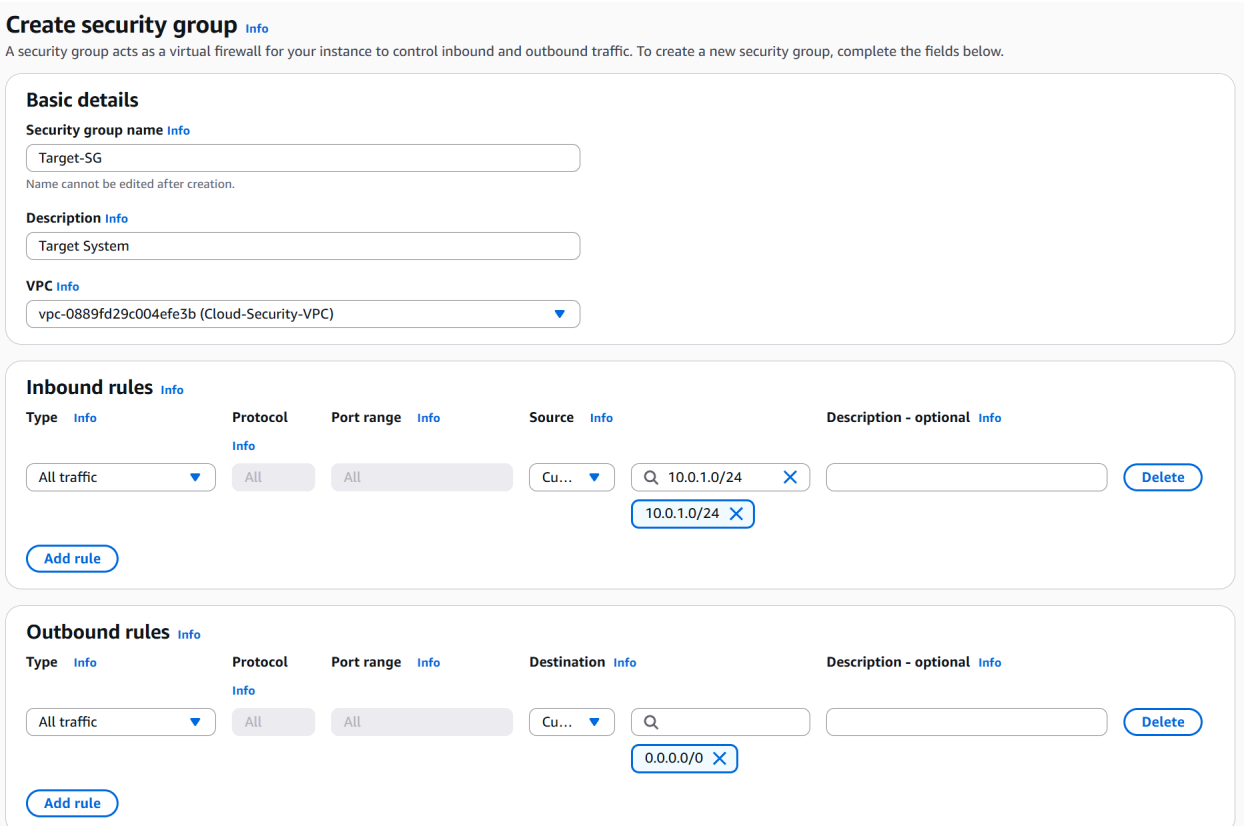
- SSH > /32
- ALL traffic > 10.0.2.0/24

Outbound:

- Allow all



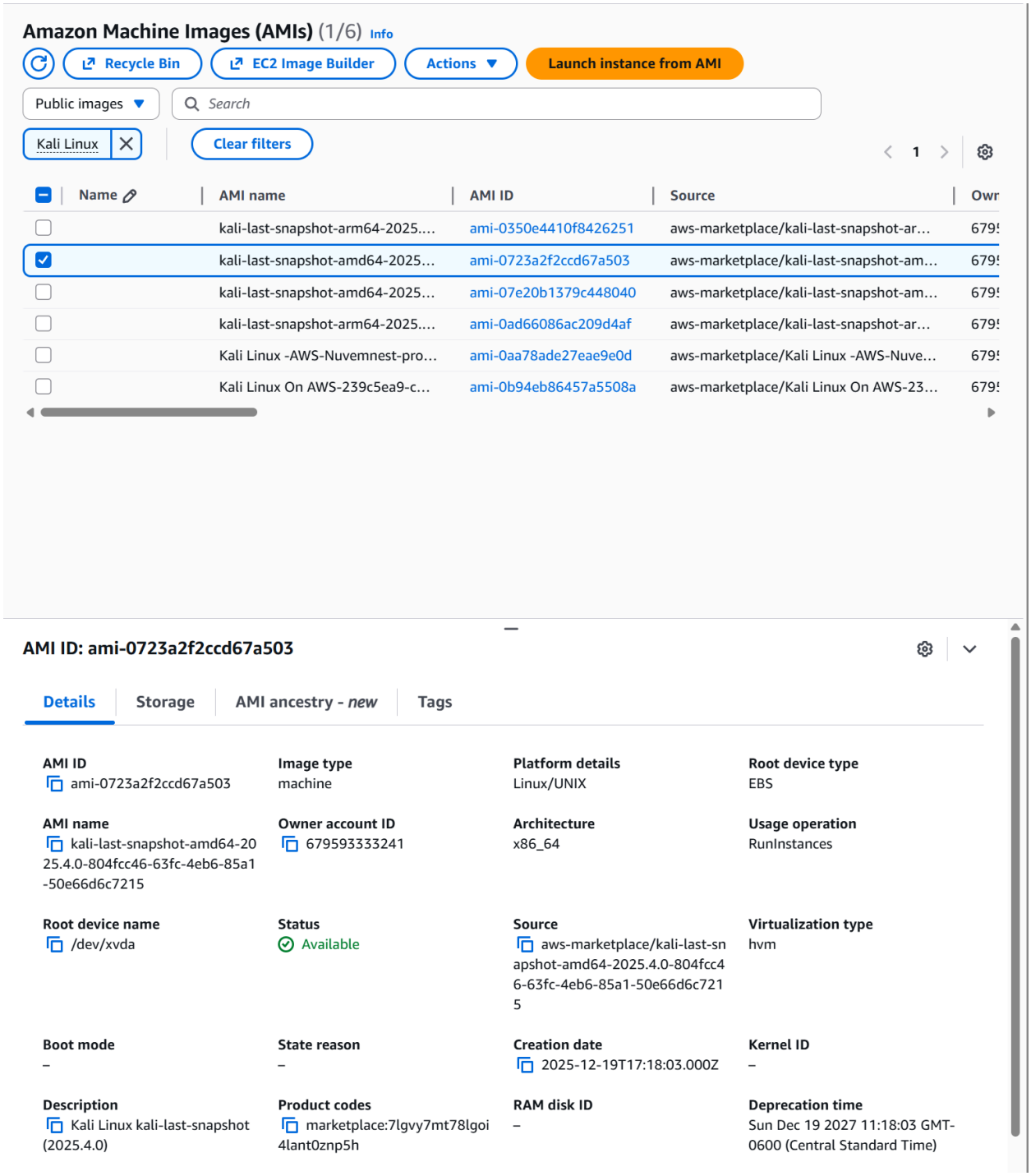
Target-SG
Inbound: All traffic > 10.0.1.0/24
Outbound: Allow all



Deploy Kali Linux (Red Team)

6. Launch Kali Linux EC2

- AMI: Official Kali Linux
- Instance Type: t2.mirco
- Subnet: Public-Subnet
- Auto-assign public IP: Enabled
- Security Group: Kali-SG
- Storage: 20-30 GB


Amazon Machine Images (AMIs) (1/6) Info
Recycle Bin
EC2 Image Builder
Actions
Launch instance from AMI
Public images
Search
Kali Linux
Clear filters
1
Name
AMI name
AMI ID
Source
Owr
kali-last-snapshot-arm64-2025....
ami-0350e4410f8426251
aws-marketplace/kali-last-snapshot-ar...
679
kali-last-snapshot-amd64-2025...
ami-0723a2f2ccd67a503
aws-marketplace/kali-last-snapshot-am...
679
kali-last-snapshot-amd64-2025...
ami-07e20b1379c448040
aws-marketplace/kali-last-snapshot-am...
679
kali-last-snapshot-arm64-2025....
ami-0ad66086ac209d4af
aws-marketplace/kali-last-snapshot-ar...
679
Kali Linux -AWS-Nuvemnest-pro...
ami-0aa78ade27eae9e0d
aws-marketplace/Kali Linux -AWS-Nuve...
679
Kali Linux On AWS-239c5ea9-c...
ami-0b94eb86457a5508a
aws-marketplace/Kali Linux On AWS-23...
679
AMI ID: ami-0723a2f2ccd67a503
Details
Storage
AMI ancestry - new
Tags
AMI ID
ami-0723a2f2ccd67a503
Image type
machine
Platform details
Linux/UNIX
Root device type
EBS
AMI name
kali-last-snapshot-amd64-2025.4.0-804fcc46-63fc-4eb6-85a1-50e66d6c7215
Owner account ID
679593333241
Architecture
x86_64
Usage operation
RunInstances
Root device name
/dev/xvda
Status
Available
Source
aws-marketplace/kali-last-snapshot-amd64-2025.4.0-804fcc46-63fc-4eb6-85a1-50e66d6c7215
Virtualization type
hvm
Boot mode
–
State reason
–
Creation date
2025-12-19T17:18:03.000Z
Kernel ID
–
Description
Kali Linux kali-last-snapshot (2025.4.0)
Product codes
marketplace:7lgvy7mt78lgoi4lant0znp5h
RAM disk ID
–
Deprecation time
Sun Dec 19 2027 11:18:03 GMT-0600 (Central Standard Time)

AMI from catalog | Quick Start

Name

kali-last-snapshot-amd64-2025.4.0-804fcc46-63fc-4eb6-85a1-50e66d6c7215

Verified provider | Free tier eligible

Description

Kali Linux kali-last-snapshot (2025.4.0)

Image ID

ami-0723a2f2ccd67a503

Username

root (Check with the AMI provider.)

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Catalog	Published	Architecture	Virtualization	Root device type	ENA Enabled
AWS Marketplace AMIs	2025-12-19T17:18:03.000Z	x86_64	hvm	ebs	Yes

If you have an existing license entitlement to use this software, then you can launch this software without creating a new subscription. If you do not have an existing entitlement, then by launching this software, you will be subscribed to this software and agree that your use of this software is subject to the pricing terms and the seller's End User License Agreement

Instance type Info | Get advice

Instance type

t3.micro — Free tier eligible

Family: t3 2 vCPU 1 GiB Memory Current generation: true

All generations

Compare instance types

The AMI vendor recommends using a t2.medium instance (or larger) for the best experience with this product.

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

cloud-security-kali-key

Create new key pair

Network settings Info

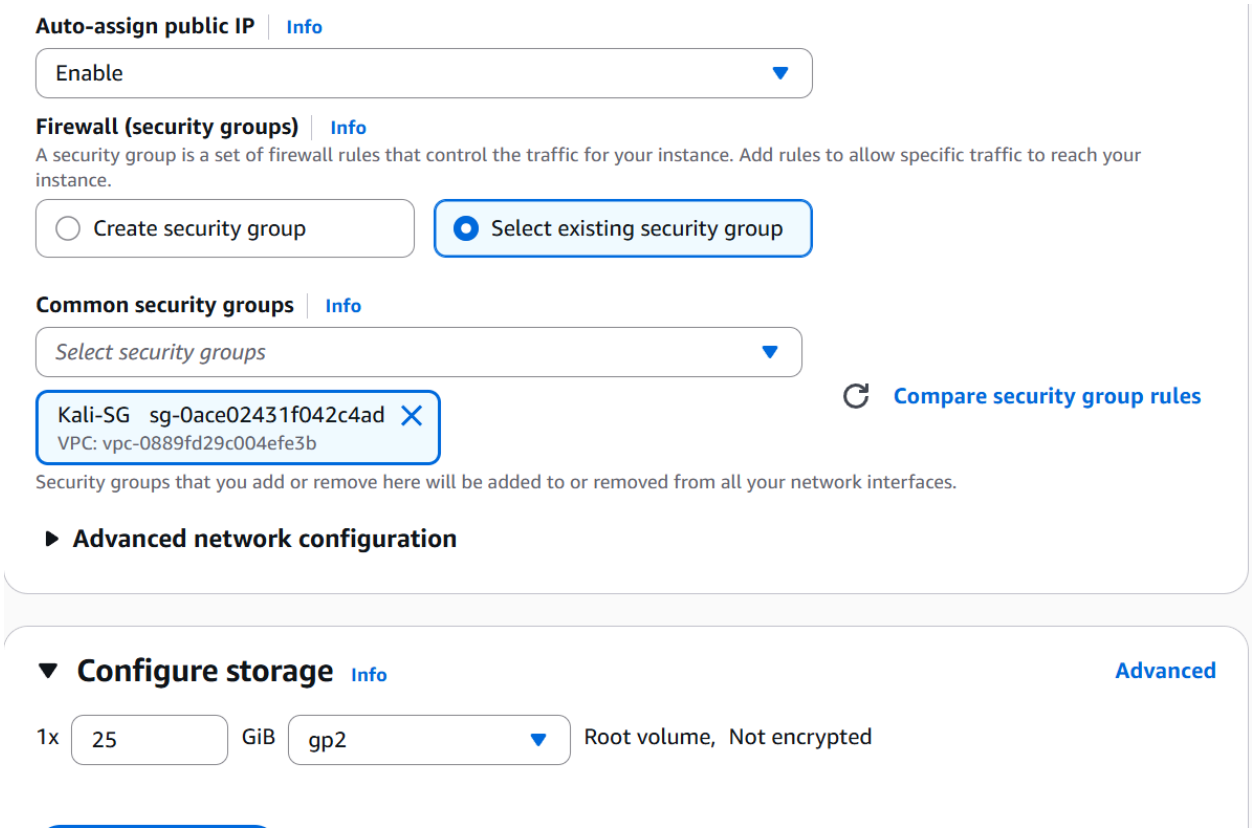
VPC - *required* Info

vpc-0889fd29c004efe3b (Cloud-Security-VPC)
10.0.0.0/16

Subnet Info

subnet-02df60a62f408a485 Public-Subent
VPC: vpc-0889fd29c004efe3b Owner: 277848663122
Availability Zone: us-east-2a (use2-az1) Zone type: Availability Zone
IP addresses available: 251 CIDR: 10.0.1.0/24)

Create new subnet



SSH: ssh -i kali.pem kali@<Public-IP>

```
C:\[redacted]>ssh -i C:\[redacted]\Downloads\cloud-security-kali-key.pem kali@18.227.111.248
The authenticity of host '18.227.111.248 (18.227.111.248)' can't be established.
ED25519 key fingerprint is SHA256:ABye8cCWUwBR+V0sNdmSsiaNizkODvVXDqSOCGPRe2Q.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?  yes
Warning: Permanently added '18.227.111.248' (ED25519) to the list of known hosts.
Linux kali 6.16.8+kali-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.16.8-1kali1 (2025-09-24) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
┌─(Message from Kali developers)
│
│  This is a minimal installation of Kali Linux, you likely
│  want to install supplementary tools. Learn how:
│  ⇒ https://www.kali.org/docs/troubleshooting/common-minimum-setup/
│
│  This is a cloud installation of Kali Linux. Learn more about
│  the specificities of the various cloud images:
│  ⇒ https://www.kali.org/docs/troubleshooting/common-cloud-setup/
│
└─(Run: "touch ~/.hushlogin" to hide this message)
┌──(kali㉿kali)-[~]
└─$
```

Verify:
Ping google.com

```
┌──(kali㉿kali)-[~]
└─$ ping google.com
PING google.com (142.250.191.142) 56(84) bytes of data.
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=1 ttl=117 time=8.69 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=2 ttl=117 time=8.79 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=3 ttl=117 time=8.70 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=4 ttl=117 time=8.71 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=5 ttl=117 time=8.70 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=6 ttl=117 time=8.69 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=7 ttl=117 time=8.72 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=8 ttl=117 time=8.78 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=9 ttl=117 time=8.72 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=10 ttl=117 time=8.80 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=11 ttl=117 time=8.72 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=12 ttl=117 time=8.70 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=13 ttl=117 time=8.71 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=14 ttl=117 time=8.72 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=15 ttl=117 time=8.71 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=16 ttl=117 time=8.72 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=17 ttl=117 time=8.73 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=18 ttl=117 time=8.69 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=19 ttl=117 time=8.69 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=20 ttl=117 time=8.72 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=21 ttl=117 time=8.71 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=22 ttl=117 time=8.70 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=23 ttl=117 time=8.73 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=24 ttl=117 time=8.70 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=25 ttl=117 time=8.71 ms
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=26 ttl=117 time=8.70 ms
^C
--- google.com ping statistics ---
61 packets transmitted, 61 received, 0% packet loss, time 60113ms
rtt min/avg/max/mdev = 8.685/8.714/8.797/0.022 ms
```

NAT Instance (FREE TIER MAGIC)

7. Launch NAT Instance

- AMI: Amazon Linux 2
- Instance Type: t2.micro
- Subnet: Public-subnet
- Auto-assign Public IP: Enabled
- Security Groups: NAT-SG

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

nat-instance-public

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose **Browse more AMIs**.

Search our full catalog including 1000s of application and OS images

Recents | Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linu

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 kernel-6.1 AMI
ami-06f1fc9ae5ae7f31e (64-bit (x86), uefi-preferred) / ami-058e74ab207ed2b33 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 (kernel-6.1) is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.10.20260105.0 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID	Publish Date	Username
64-bit ...	uefi-preferred	ami-06f1fc9ae5ae7f31e	2026-01-02	ec2-user

Verified provider

Instance type Info | Get advice

Instance type

t3.micro — Free tier eligible
Family: t3 2 vCPU 1 GiB Memory Current generation: true
On-Demand RHEL base pricing: 0.0392 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0139 USD per Hour
On-Demand Windows base pricing: 0.0196 USD per Hour
On-Demand SUSE base pricing: 0.0104 USD per Hour
On-Demand Linux base pricing: 0.0104 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

cloud-security-kali-key

Create new key pair

Network settings Info

VPC - *required* Info

vpc-0889fd29c004efe3b (Cloud-Security-VPC)
10.0.0.0/16

Subnet Info

subnet-02df60a62f408a485 Public-Subent
VPC: vpc-0889fd29c004efe3b Owner: 277848663122
Availability Zone: us-east-2a (use2-az1) Zone type: Availability Zone
IP addresses available: 250 CIDR: 10.0.1.0/24)

Create new subnet

Auto-assign public IP Info

Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

- ○ Create security group
- ◉ Select existing security group

Common security groups Info

Select security groups

NAT-SG sg-01d741fc8ae3c57a3
VPC: vpc-0889fd29c004efe3b

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

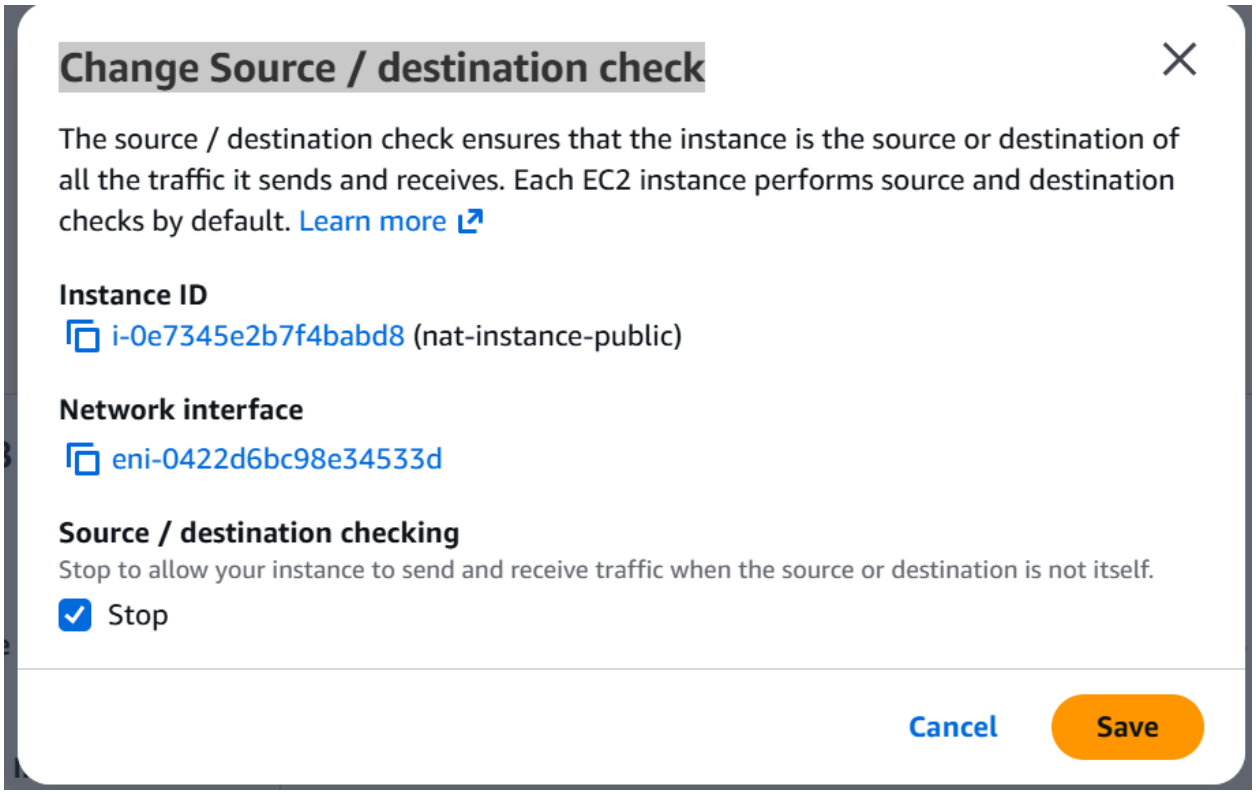
▸ **Advanced network configuration**

Configure storage Info

Advanced

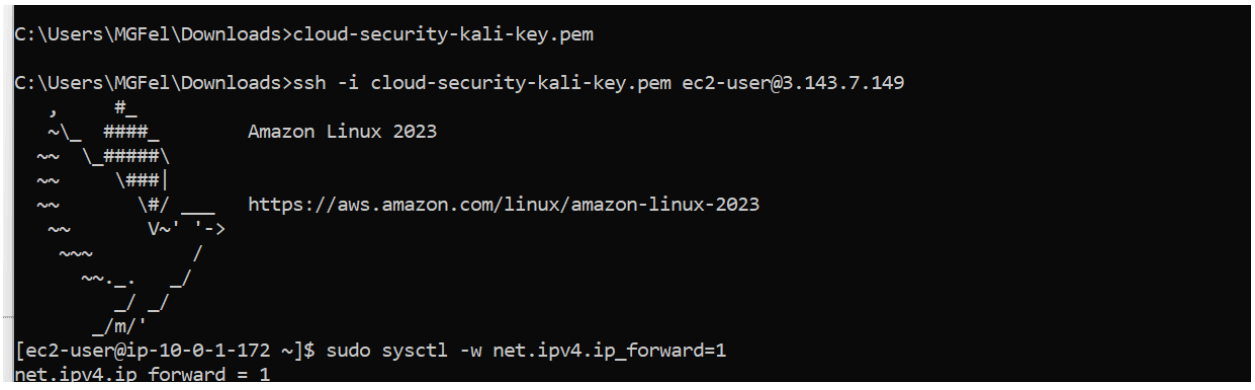
1x 8 GiB gp3 Root volume, 3000 IOPS, Not encrypted

Disable Source/Destination Check
EC2 > Instance > Networking > Disable



Enable Fowarding
sudo sysctl -w net.ipv4.ip_forward=1

Persist:
sudo nano /etc/sysctl.conf



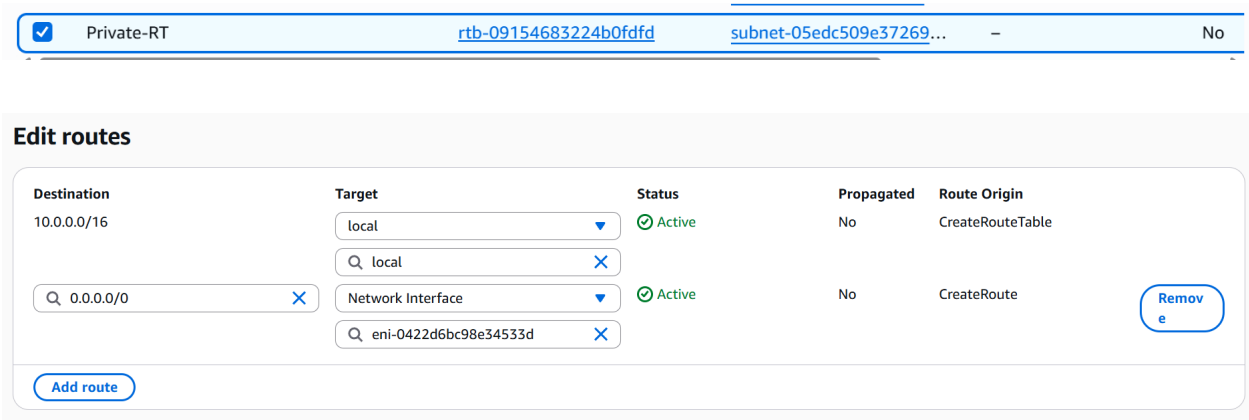
Add:
net.ipv4.ip_forward=1

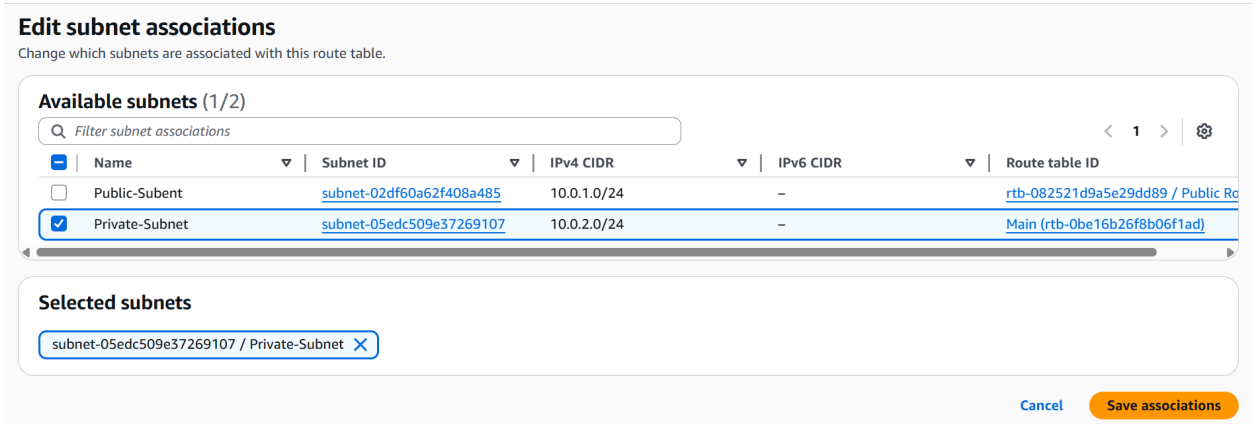
Configure iptables
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo yum install iptables-services -y
sudo service iptables save

```
sudo: iptables: command not found
[ec2-user@ip-10-0-1-172 ~]$ sudo yum install iptables-services -y
Amazon Linux 2023 Kernel Livepatch repository                                    248 kB/s |  30 kB     00:00
Last metadata expiration check: 0:00:01 ago on Thu Jan  8 02:47:22 2026.
Dependencies resolved.
================================================================================================================
 Package                    Architecture       Version                        Repository                Size
================================================================================================================
Installing:
 iptables-services          noarch             1.8.8-3.amzn2023.0.2           amazonlinux               18 k
Installing dependencies:
 iptables-libs              x86_64             1.8.8-3.amzn2023.0.2           amazonlinux              401 k
 iptables-nft               x86_64             1.8.8-3.amzn2023.0.2           amazonlinux              183 k
 iptables-utils             x86_64             1.8.8-3.amzn2023.0.2           amazonlinux               43 k
 libnetfilter_conntrack     x86_64             1.0.8-2.amzn2023.0.2           amazonlinux               58 k
 libnfnetlink               x86_64             1.0.1-19.amzn2023.0.2          amazonlinux               30 k
 libnftnl                   x86_64             1.2.2-2.amzn2023.0.2           amazonlinux               84 k

Transaction Summary
================================================================================================================
Install  7 Packages

Total download size: 816 k
Installed size: 2.9 M
Downloading Packages:
(1/7): iptables-services-1.8.8-3.amzn2023.0.2.noarch.rpm                         495 kB/s |  18 kB     00:00
(2/7): iptables-libs-1.8.8-3.amzn2023.0.2.x86_64.rpm                             8.4 MB/s | 401 kB     00:00
(3/7): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm                              3.4 MB/s | 183 kB     00:00
(4/7): iptables-utils-1.8.8-3.amzn2023.0.2.x86_64.rpm                            1.5 MB/s |  43 kB     00:00
(5/7): libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64.rpm                    1.9 MB/s |  58 kB     00:00
(6/7): libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64.rpm                             1.0 MB/s |  30 kB     00:00
(7/7): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm                                  2.7 MB/s |  84 kB     00:00
----------------------------------------------------------------------------------------------------------------
Total                                                                            5.9 MB/s | 816 kB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                        1/1
  Installing       : libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64                                              1/7
  Installing       : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64                                     2/7
  Installing       : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64                                              3/7
  Installing       : libnftnl-1.2.2-2.amzn2023.0.2.x86_64                                                   4/7
  Installing       : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64                                               5/7
  Running scriptlet: iptables-nft-1.8.8-3.amzn2023.0.2.x86_64                                               5/7
  Installing       : iptables-utils-1.8.8-3.amzn2023.0.2.x86_64                                             6/7
  Installing       : iptables-services-1.8.8-3.amzn2023.0.2.noarch                                          7/7
  Running scriptlet: iptables-services-1.8.8-3.amzn2023.0.2.noarch                                          7/7
  Verifying        : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64                                              1/7
  Verifying        : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64                                               2/7
  Verifying        : iptables-services-1.8.8-3.amzn2023.0.2.noarch                                          3/7
  Verifying        : iptables-utils-1.8.8-3.amzn2023.0.2.x86_64                                             4/7
  Verifying        : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64                                     5/7
  Verifying        : libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64                                              6/7
  Verifying        : libnftnl-1.2.2-2.amzn2023.0.2.x86_64                                                   7/7

Installed:
  iptables-libs-1.8.8-3.amzn2023.0.2.x86_64                  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
  iptables-services-1.8.8-3.amzn2023.0.2.noarch              iptables-utils-1.8.8-3.amzn2023.0.2.x86_64
  libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64         libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64
  libnftnl-1.2.2-2.amzn2023.0.2.x86_64

Complete!
[ec2-user@ip-10-0-1-172 ~]$ iptables --version
iptables v1.8.8 (nf_tables)
[ec2-user@ip-10-0-1-172 ~]$ sudo systemctl start iptables
[ec2-user@ip-10-0-1-172 ~]$ sudo systemctl enable iptables
Created symlink /etc/systemd/system/multi-user.target.wants/iptables.service → /usr/lib/systemd/system/iptables.service.
[ec2-user@ip-10-0-1-172 ~]$ sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
[ec2-user@ip-10-0-1-172 ~]$ sudo iptables -t nat -L -n -v
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 MASQUERADE  all  --  *      eth0    0.0.0.0/0            0.0.0.0/0
```

```
[ec2-user@ip-10-0-1-172 ~]$ sudo service iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables: [  OK  ]
[ec2-user@ip-10-0-1-172 ~]$ cat /proc/sys/net/ipv4/ip_forward
1
[ec2-user@ip-10-0-1-172 ~]$ sudo iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
MASQUERADE  all  --  anywhere             anywhere
```

8. Update Private Routing Table
   0.0.0.0/0 > NAT Instance ID (Private subnet now has outbound internet without exposure)






```
┌──(kali㉿kali)-[~]
└─$ ping google.com
PING google.com (142.250.190.14) 56(84) bytes of data.
64 bytes from ord37s32-in-f14.1e100.net (142.250.190.14): icmp_seq=1 ttl=117 time=8.43 ms
64 bytes from ord37s32-in-f14.1e100.net (142.250.190.14): icmp_seq=2 ttl=117 time=8.43 ms
64 bytes from ord37s32-in-f14.1e100.net (142.250.190.14): icmp_seq=3 ttl=117 time=8.52 ms
64 bytes from ord37s32-in-f14.1e100.net (142.250.190.14): icmp_seq=4 ttl=117 time=8.48 ms
64 bytes from ord37s32-in-f14.1e100.net (142.250.190.14): icmp_seq=5 ttl=117 time=8.45 ms
64 bytes from ord37s32-in-f14.1e100.net (142.250.190.14): icmp_seq=6 ttl=117 time=8.44 ms
64 bytes from ord37s32-in-f14.1e100.net (142.250.190.14): icmp_seq=7 ttl=117 time=8.45 ms
64 bytes from ord37s32-in-f14.1e100.net (142.250.190.14): icmp_seq=8 ttl=117 time=8.45 ms
^C
--- google.com ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7011ms
rtt min/avg/max/mdev = 8.427/8.455/8.516/0.026 ms
```

Threat Model

Threats Considered

- Public subnet exposure
- SSH brute-force attempts
- NAT instance compromise
- Misconfigured route tables
- Lateral movement from public > private subnet
- Privilege escalation on EC2 instances
- Data exfiltration from private subnet

Mitigations

- SSH restricted top /32

- No public IPs in private subnet
- NAT instance hardened (IP forwarding, iptables)
- Strict SGs and NACLs
- CloudTrail + GuardDuty monitoring
- VPC Flow Logs for anomaly detection

Monitoring & Detection Plan

Logging

- VPC Flow Logs > CloudWatch
- CloudTrail > S3
- EC2 system logs

Detection

- GuardDuty for threat intelligence
- CloudWatch alarms for:
    - Unusual outbound traffic
    - SSH attempts
    - Route table changes
    - IAM changes

Visibility

- CloudWatch dashboards
- Flow log analysis

Cost-Control Strategy

- All EC2 instances free-tier eligible (t2.micro/t3.micro)
- NAT instance used instead of NAT Gateway
- No load balancers, RDS, or high-cost services
- Instances stopped when not in use
- No attack traffic to avoid egress charges
- CloudWatch logs retained minimally

This project demonstrates my ability to design secure cloud architectures, implement Linux-based NAT routing, enforce least-privilege network segmentation, and validate

connectivity in AWS. It reflects my interest in cloud security engineering and DevSecOps practices.