# Configuring, Maintaining, and Accessing a Cloud VM-(AWS)

Aiswarya.D

## Contents

## 1. Overview

Task Description: Research and launch a free Virtual Machine within a cloud environment. The tutorial will include setting up the VM, getting its IP address, testing connectivity with ping and ssh into the VM. Extra work will include additional network tests and exploratory work done on the VM and proper shutdown of the VM when finished.

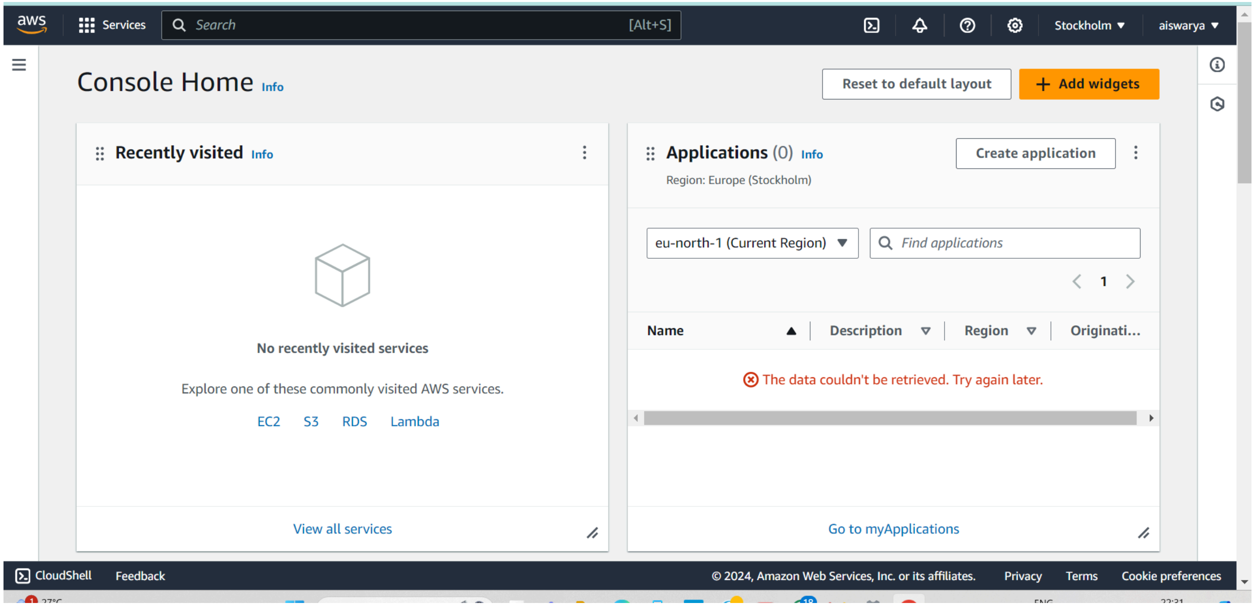
For the purpose of this report, I have taken AWS as my cloud provider.

## 2. Launch Process of EC2 Instance

### 2.1 Accessing the EC2 Console

I started with the initial setup process by accessing the EC2 Dashboard through the following actions:

1. Logged into the AWS Management Console and searched for EC2.



Region: Europe (Stockholm)

No recently visited services

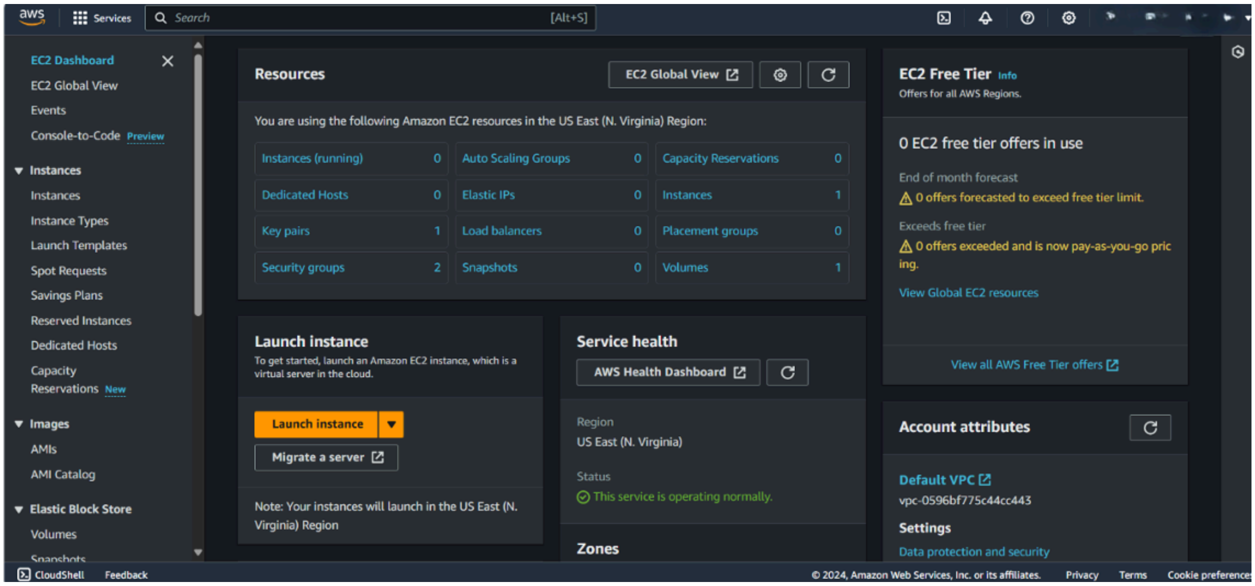
Explore one of these commonly visited AWS services.

The data couldn't be retrieved. Try again later.

© 2024, Amazon Web Services, Inc. or its affiliates.

Screenshot 1

2. Selecting EC2 from the results redirected me to the EC2 dashboard.



**Resources**

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

| | | | | | |
|---|---|---|---|---|---|
| Instances (running) | 0 | Auto Scaling Groups | 0 | Capacity Reservations | 0 |
| Dedicated Hosts | 0 | Elastic IPs | 0 | Instances | 1 |
| Key pairs | 1 | Load balancers | 0 | Placement groups | 0 |
| Security groups | 2 | Snapshots | 0 | Volumes | 1 |

**Launch instance**

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the US East (N. Virginia) Region

**Service health**

Region
US East (N. Virginia)

Status
This service is operating normally.

**Zones**

**EC2 Free Tier** Info

Offers for all AWS Regions.

0 EC2 free tier offers in use

End of month forecast
⚠ 0 offers forecasted to exceed free tier limit.

Exceeds free tier
⚠ 0 offers exceeded and is now pay-as-you-go pricing.

View Global EC2 resources

View all AWS Free Tier offers

**Account attributes**

Default VPC
vpc-0596bf775c44cc443

Settings
Data protection and security

© 2024, Amazon Web Services, Inc. or its affiliates.

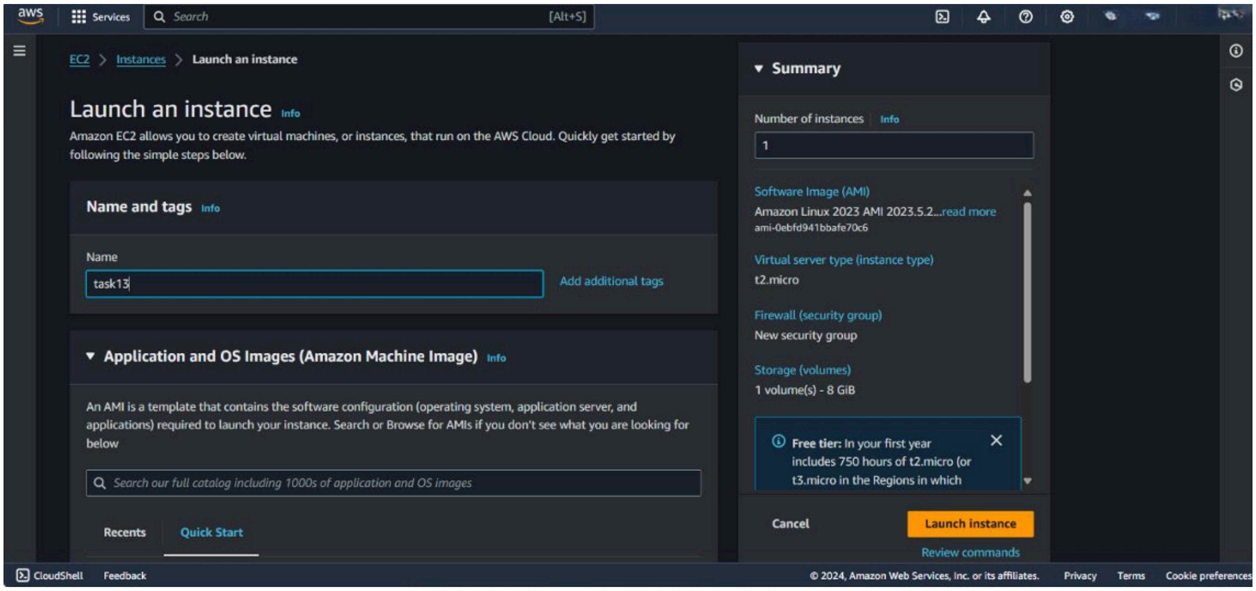
Screenshot 2

## 2.2 Launching an EC2 Instance

To begin creating an EC2 instance

1. I clicked the Launch Instance button on the EC2 dashboard.

2. This took me to the Launch an Instance page where I could configure the settings for the VM.



EC2 > Instances > Launch an instance

**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
task13

**▼ Application and OS Images (Amazon Machine Image)** Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

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

**▼ Summary**

Number of instances Info
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...read more
ami-0ebfd941bbafe70c6

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which

© 2024, Amazon Web Services, Inc. or its affiliates.

Screenshot 3

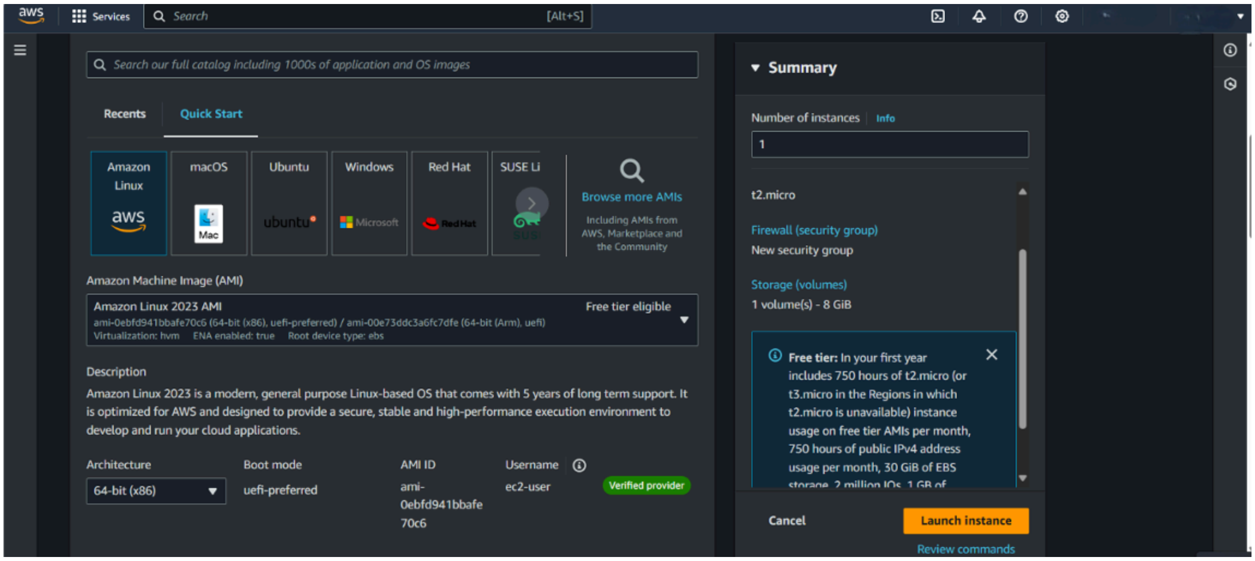
## 2.3 Selecting the Amazon Machine Image (AMI)

The Operating system to be used will depend on the Amazon Machine Image -

1. Explored the available AMIs; chose Amazon Linux, which is free-tier eligible.

2. Other choices are macOS,Ubuntu,or Windows Server, but I

This is the chosen Amazon Linux for this configuration.



Amazon Machine Image (AMI)

Amazon Linux 2023 AMI  Free tier eligible
ami-0ebfd941bbafe70c6 (64-bit (x86), uefi-preferred) / ami-00e73ddc3a6fc7dfe (64-bit (Arm), uefi)
Virtualization: hvm  ENA enabled: true  Root device type: ebs

Description

Amazon Linux 2023 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.

| Architecture | Boot mode | AMI ID | Username |
|---|---|---|---|
| 64-bit (x86) | uefi-preferred | ami-0ebfd941bbafe70c6 | ec2-user |

Summary

Number of instances: 1

t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of

Screenshot 4

## 2.4 Choosing an Instance Type

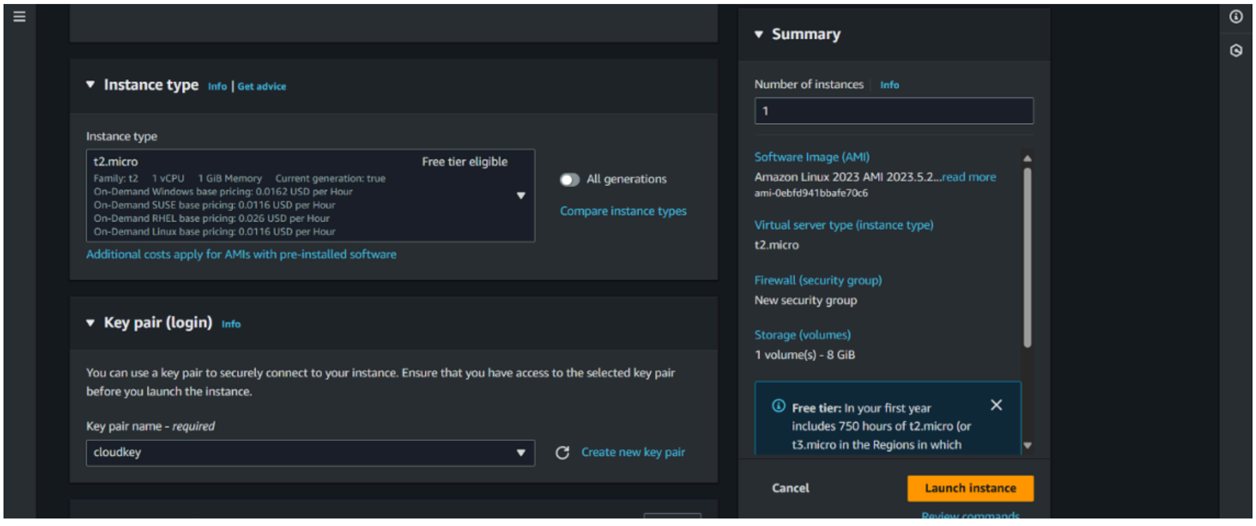
Instance types define the resource configuration (CPU, memory, etc.):

1. I selected t2 micro instance type, which is suitable for small workload and within

Under the AWS free-tier offering.

2. Selected key pair named as cloudkey [a key pair consists of a public key and a

The private key is used safely to access your EC2 instances.



**Instance type** Info | Get advice

Instance type

t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

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*
cloudkey

**Summary**

Number of instances: 1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...read more
ami-0ebfd941bbafe70c6

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which

Screenshot 5

## 2.5 Configuring Instance Details

I configure other settings for the instance.

1. Here, I used the default VPC for networking, and most of the options defaulted to unchanged.
2. That was enough as I did not require advanced networking settings.

## 2.6 Setting Up Security Group Rules

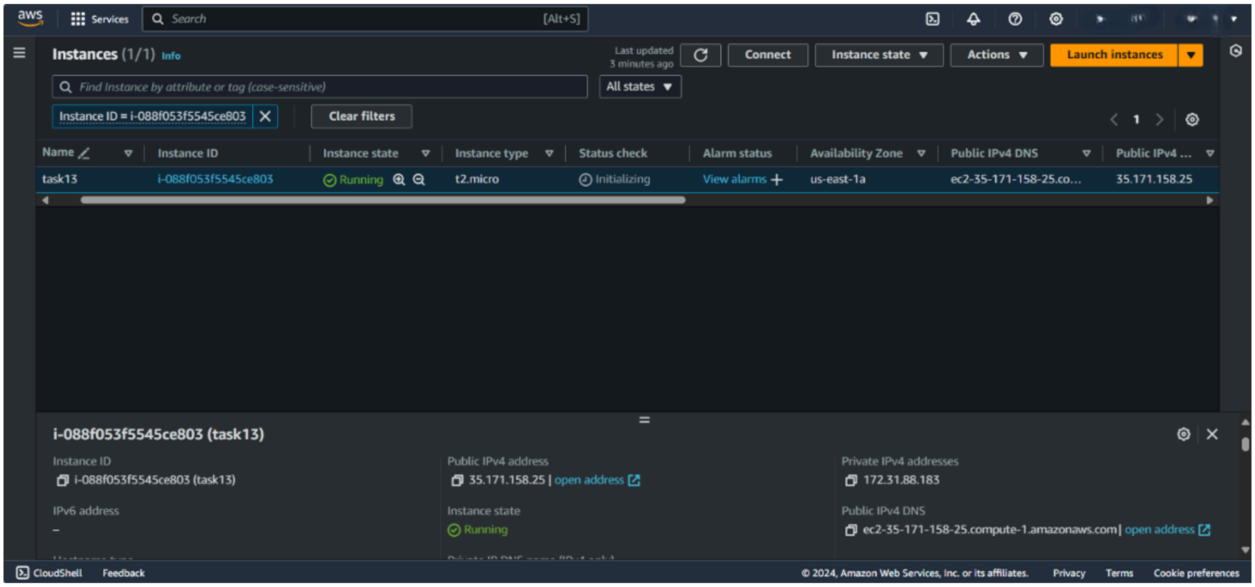
The Security Group is a virtual firewall that controls traffic to and from the instance:

1. Prior to starting, I had SSH (port 22) pre-approved for this instance so that it could be accessed remotely for
Linux.
2. I subsequently changed the inbound rules to allow specific traffic. For instance, I added to allow ICMP as it would be in order to test pinging the server. (Refer screenshot 8)

## 2.7 Final Review and Launch

1. Confirm that the configuration settings are correct.

2. Verify Setup After confirming that the setup was valid, I selected the Launch Instance option.



Instance ID = i-088f053f5545ce803

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 DNS | Public IPv4 ... |
|---|---|---|---|---|---|---|---|---|
| task13 | i-088f053f5545ce803 | Running | t2.micro | Initializing | View alarms + | us-east-1a | ec2-35-171-158-25.co... | 35.171.158.25 |

i-088f053f5545ce803 (task13)

Instance ID: i-088f053f5545ce803 (task13)

Public IPv4 address: 35.171.158.25 | open address

Private IPv4 addresses: 172.31.88.183

IPv6 address: –

Instance state: Running

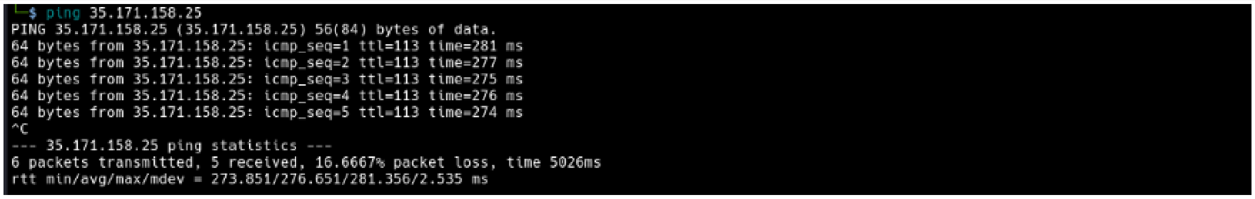
Public IPv4 DNS: ec2-35-171-158-25.compute-1.amazonaws.com| open address

Screenshot 6

## 3. Connectivity Testing and Instance Access

### 3.1 Verifying that connectivity from public IP exists by using the ping command

Once the instance was live, I checked public IP connectivity:



```
└─$ ping 35.171.158.25
PING 35.171.158.25 (35.171.158.25) 56(84) bytes of data.
64 bytes from 35.171.158.25: icmp_seq=1 ttl=113 time=281 ms
64 bytes from 35.171.158.25: icmp_seq=2 ttl=113 time=277 ms
64 bytes from 35.171.158.25: icmp_seq=3 ttl=113 time=275 ms
64 bytes from 35.171.158.25: icmp_seq=4 ttl=113 time=276 ms
64 bytes from 35.171.158.25: icmp_seq=5 ttl=113 time=274 ms
^C
--- 35.171.158.25 ping statistics ---
6 packets transmitted, 5 received, 16.6667% packet loss, time 5026ms
rtt min/avg/max/mdev = 273.851/276.651/281.356/2.535 ms
```

Screenshot 7

### 3.2 SSH to the EC2 INSTANCE

To connect to this virtual machine from a remote system, I applied the SSH command as given below:

1. I used the generated security key [cloudkey] from the setup of an instance to securelyLog in to the instance from my local machine.

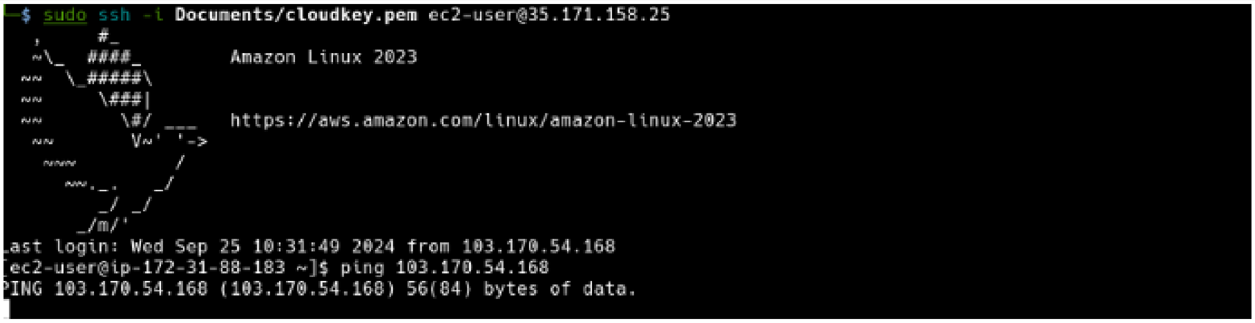
2. I ran the following command on my Kali Linux:
Connection:

sudo ssh -i Documents/cloudkey.pem ec2-user@52.91.114.191

## 4. Testing connectivity to the Public IP from VM in the cloud

### 4.1 Ping Test from the EC2 Instance

I attempted to verify public IP connectivity by pinging my public IP from within the EC2
instance itself:
1. I could not ping the public IP since my ISP had restrictions on blocking ICMP (ping)
packets.
2. This is a common limitation where Internet Service Providers avoid ICMP traffic  for some security or network performance reasons.



```
└─$ sudo ssh -i Documents/cloudkey.pem ec2-user@35.171.158.25
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Wed Sep 25 10:31:49 2024 from 103.170.54.168
[ec2-user@ip-172-31-88-183 ~]$ ping 103.170.54.168
PING 103.170.54.168 (103.170.54.168) 56(84) bytes of data.
```

Screenshot 8

**5. Terminating the VM**

I returned to the AWS EC2 dashboard to shut down the VM so that it wouldn't continue to run without a charge or resource.

## RESULTS SUMMARY

1. I was able to start a free-tier VM on AWS and retrieved the public IP.
2. After changing the security settings, I was then able to ping the VM using my Kali Linux machine.
3. I logged in to the VM using SSH, ran ip a, captured the ,, screenshot of it: ,
4. Tried pinging public IP from inside the cloud VM; the request was blocked because of ISP restriction .
5. Having done all those tasks, I shutdown and terminated my VM according to the requirement