

**20CSE522J-NETWORK SECURITY**

**LAB REPORT**

*Submitted by*

**Anil Kumar Sah [Reg.No:RA2212033010001]**

*Under the Guidance of*

**Dr.Vinoth Kumar S**

(Associate Professor, Department of Information Technology)

*In partial fulfillment of the Requirements for the Degree*

**MASTER OF TECHNOLOGY**



**DEPARTMENT OF FACULTY OF ENGINEERING AND TECHNOLOGY**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY KATTANKULATHUR –603203**

**December- 2023**

## **BONAFIDE CERTIFICATE**

Certified that this subject code “**20CSE522J-Network Security**” is the bonafide work of “**Anil Kumar Sah [Reg No: RA2212033010001]**”, who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

**Dr.Vinoth Kumar S.**

Associate Professor

**SUBJECT INCHARGE.**

Department of Networking and  
Communications

Signature of Internal Examiner

**Dr.Annapurani Panaiyappan K.**

Professor

**HEAD OF THE DEPARTMENT**

Department of Networking and  
Communications

Signature of External Examiner

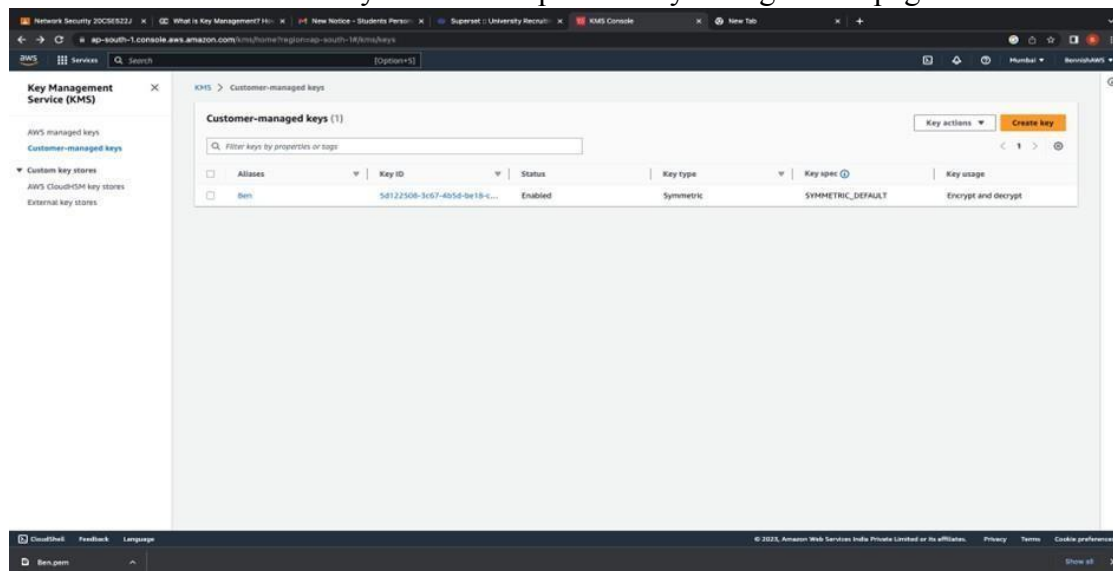
## TABLE OF CONTENT

EX.NO	DATE	TITLE	PAGE NO.	MARKS
1	31/07/2023	Implementation of different key management	4	
2	05/08/2023	To Create a VPN (Virtual private Network) over WAN	8	
3	14/08/2023	Eavesdropping Attacks and Its Prevention Using SSH	18	
4	22/08/2023	Kismet	22	
5	31/08/2023	Security Group Policies Management	28	
6	14/09/2023	Implementing NAT	31	
7	22/09/2023	Scapy	36	
8	05/10/2023	Cowpatty	39	

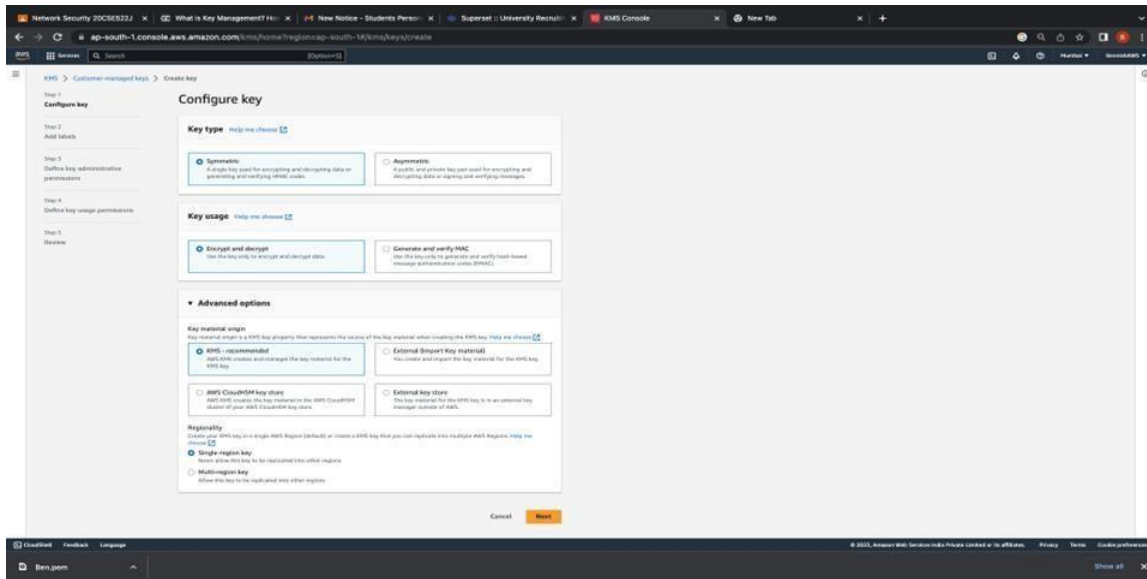
EX NO: 1	Implementation of different key management
DATE: 31/07/2023	

**AIM:** Implementing diverse key management strategies within the AWS network security framework, enhancing data protection, access controls, and encryption mechanisms for improved overall security posture.

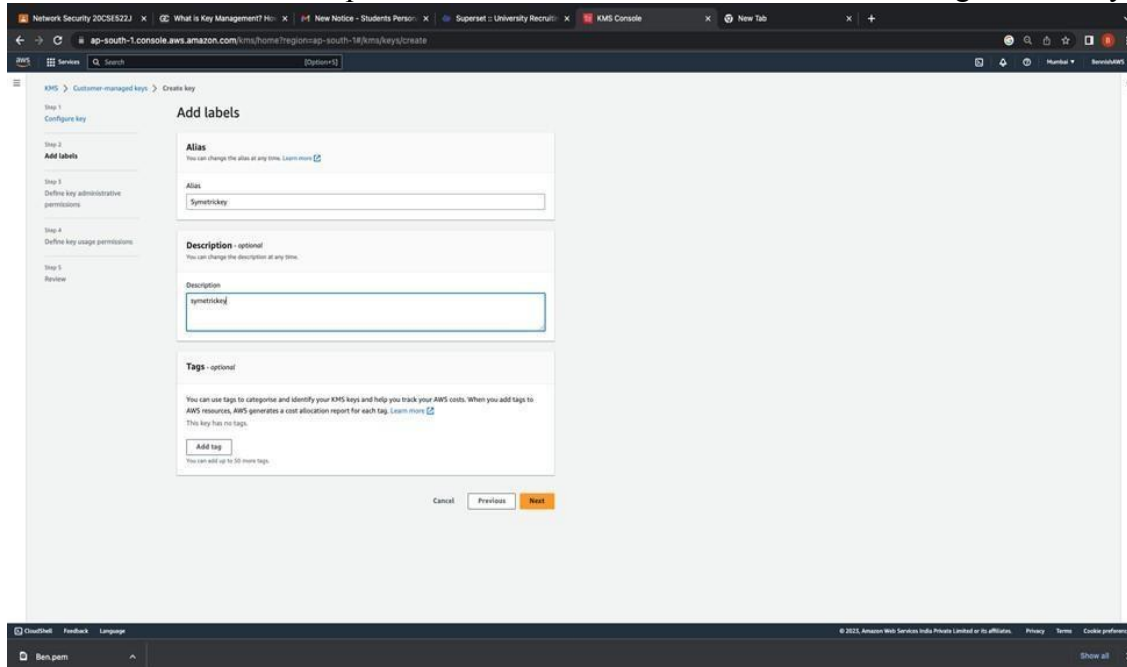
1. Sign in to the AWS management console and select the Key Management Service .
2. Select the Create key and it will open the key configuration page .



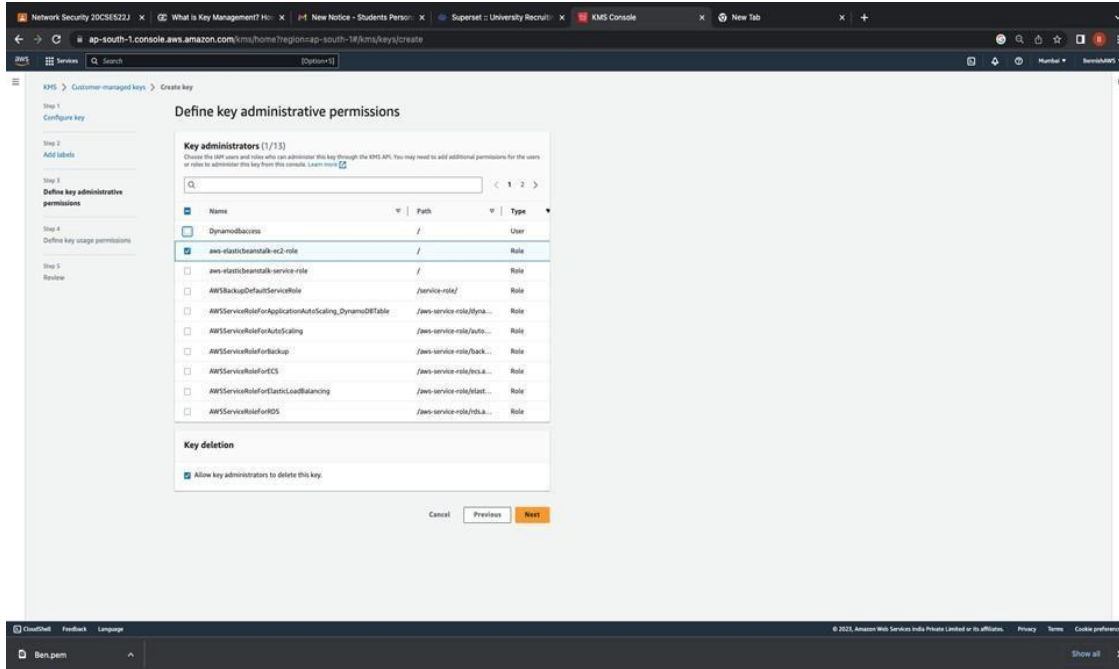
3. For symmetric key encryption ,select the key type as symmetric and key usage as encrypt and decrypt.



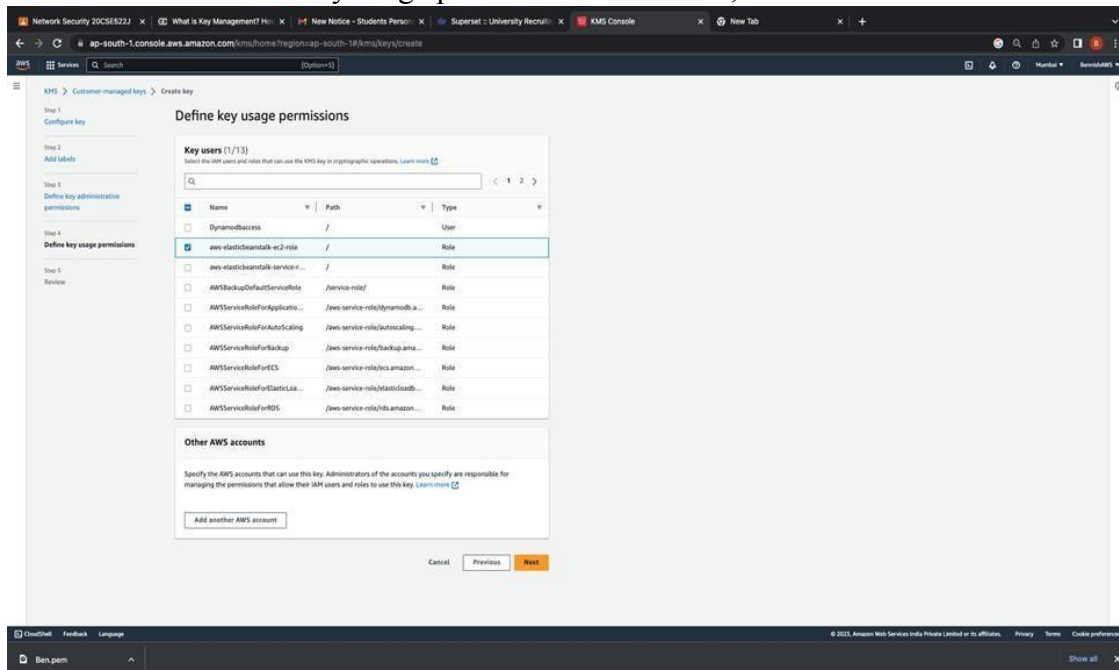
4. Let the advanced options be default and select next for the Labeling for the Key.



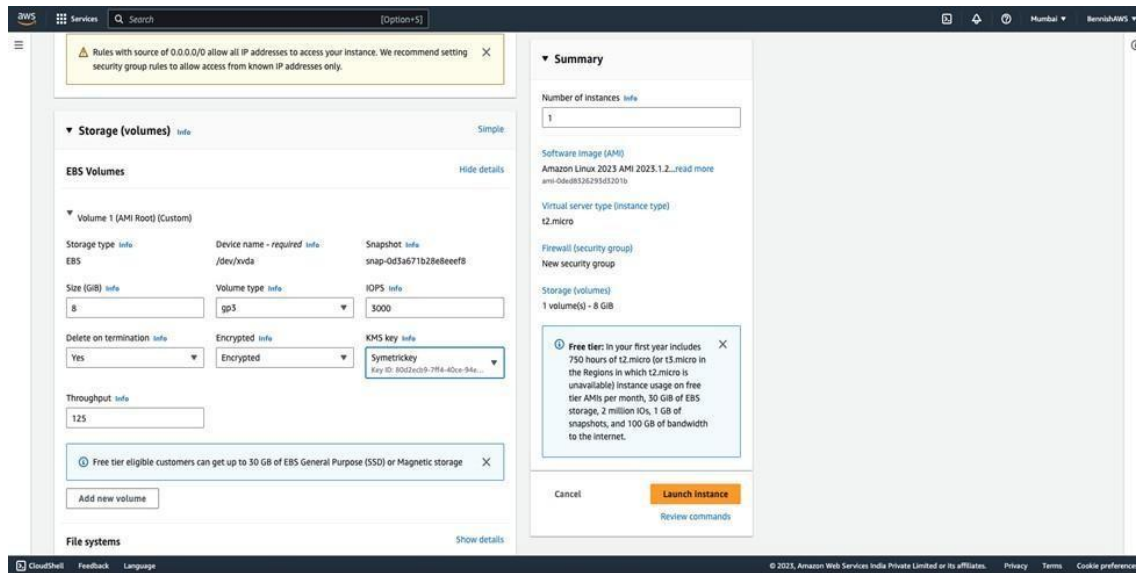
5. Select next for the defining the administrative permission and select the ec2 role administrative permission for the key.



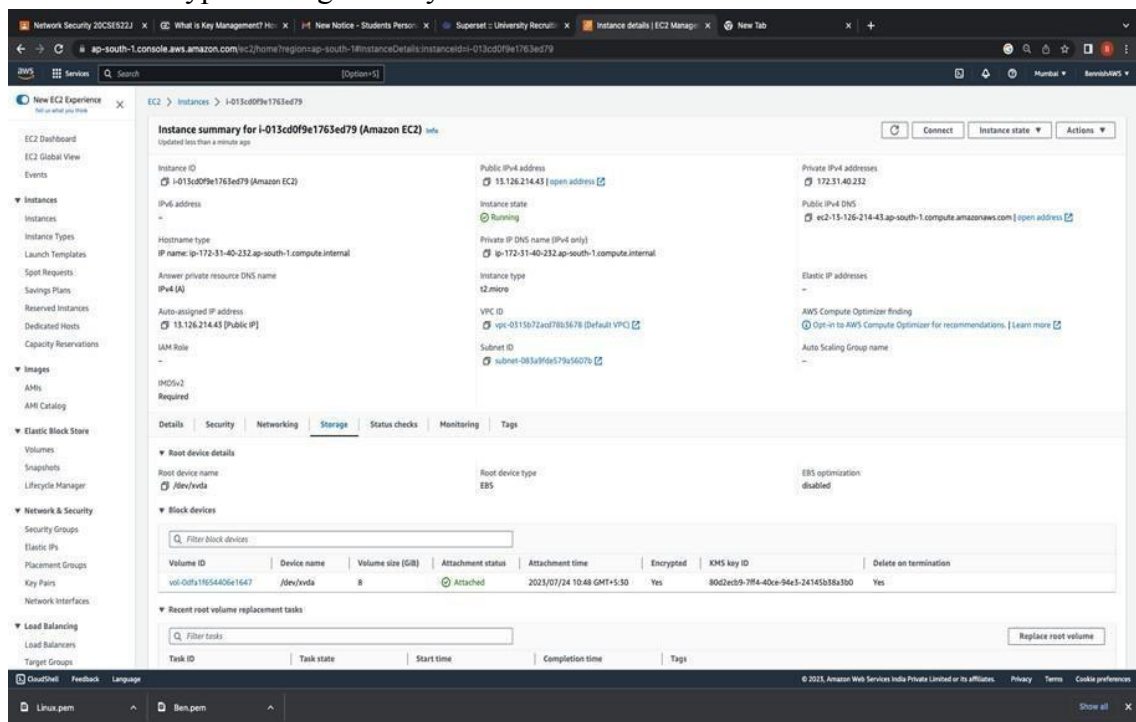
6. Next select the key usage permission as ec2 role , then review and finish.



7. Create a EC2 instance ,and in configuration storage select in encrypted and select the KMS key which we created and launch the instance.



8. In the created EC2 instance ,in the storage details we can notice that the storage is encrypted using the Key which we created.



**Result:** Implementation of different key management is successfully done.

EX NO:2

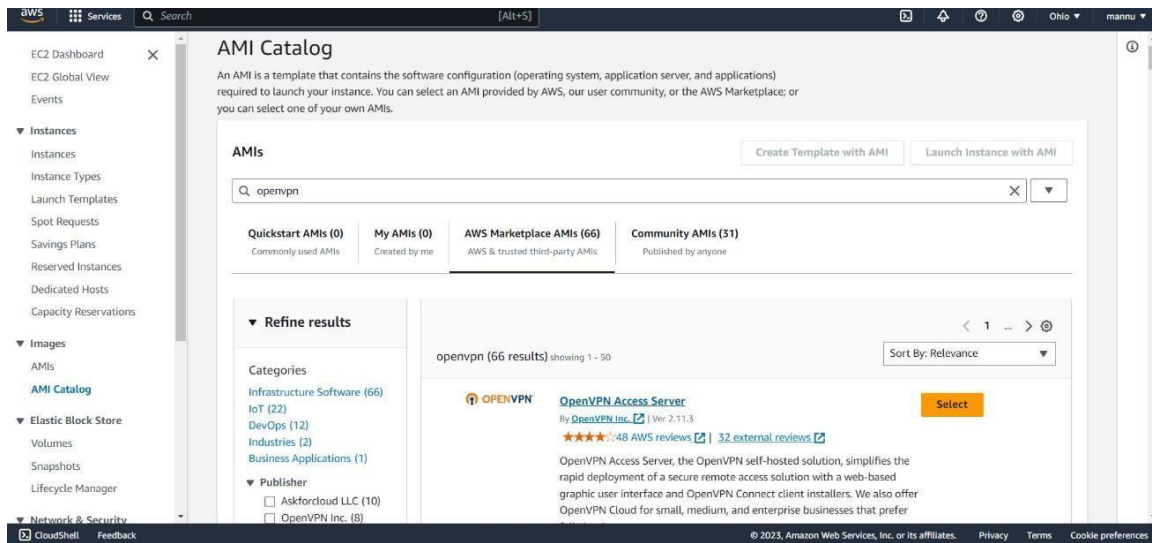
DATE:05/08/2023

## Create a VPN (Virtual private Network) over WAN

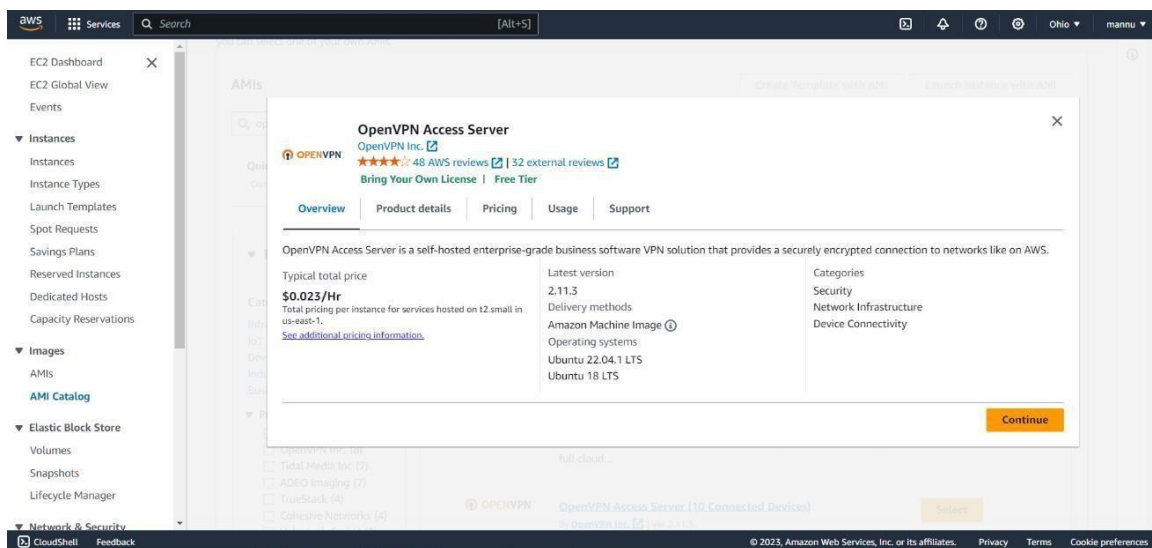
**AIM:** Create a Virtual Private Network over WAN.

### PROCEDURE:

1. First login to AWS account and search AMI, and after that select openvpn access server free version

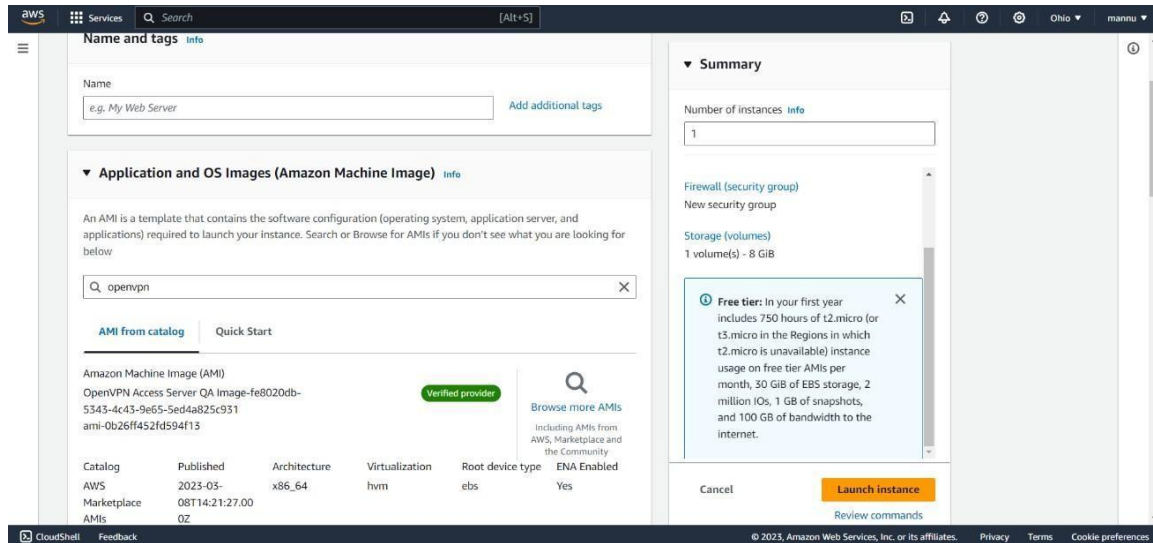


2. Selected openvpn server will be continue.

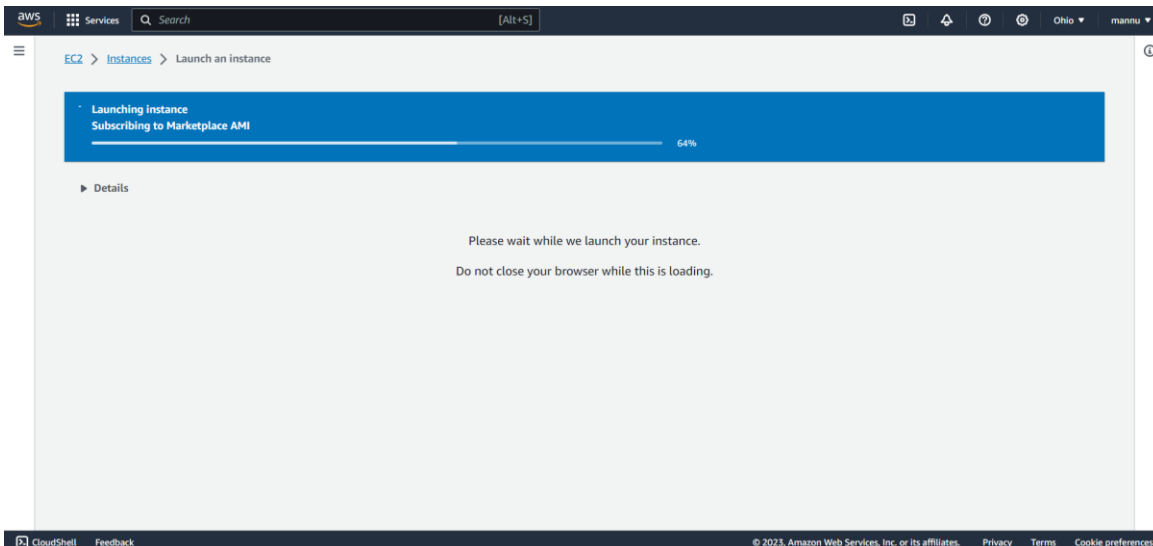


3. After selecting create an instance and launch it.

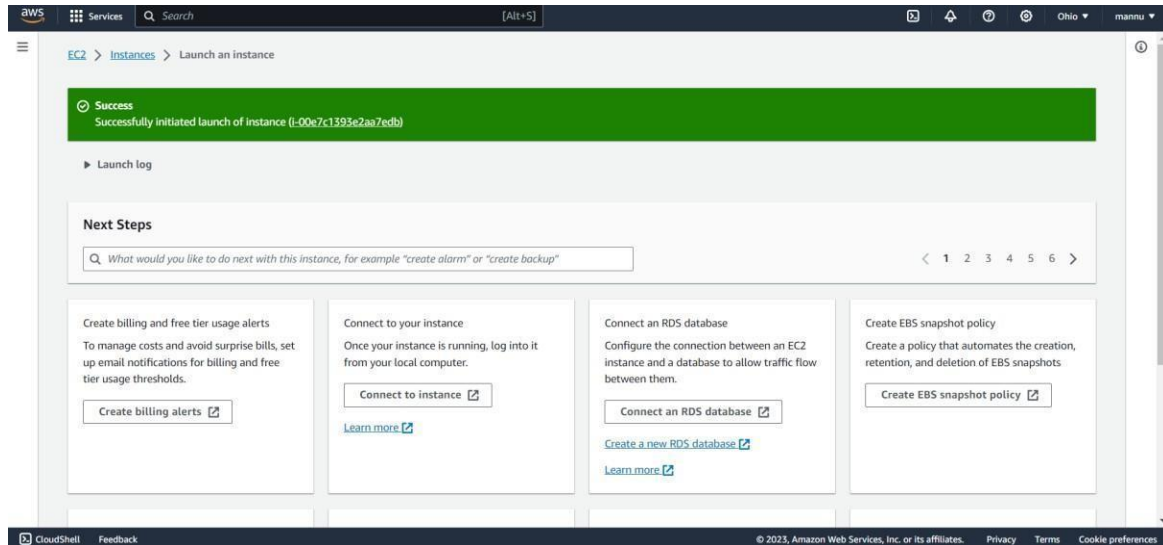




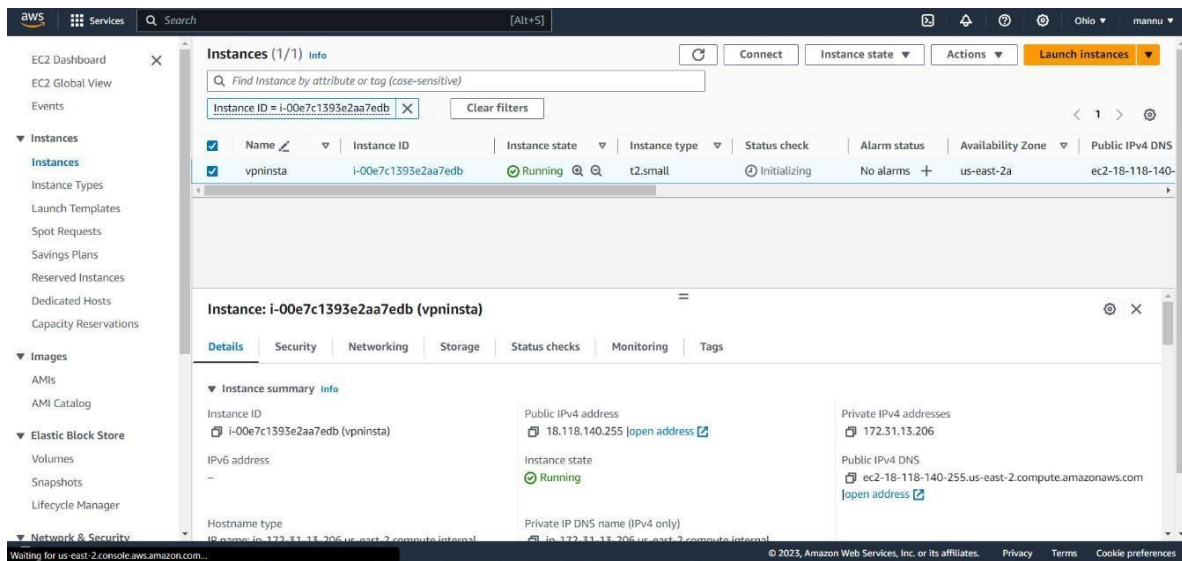
#### 4. Launching Instance.



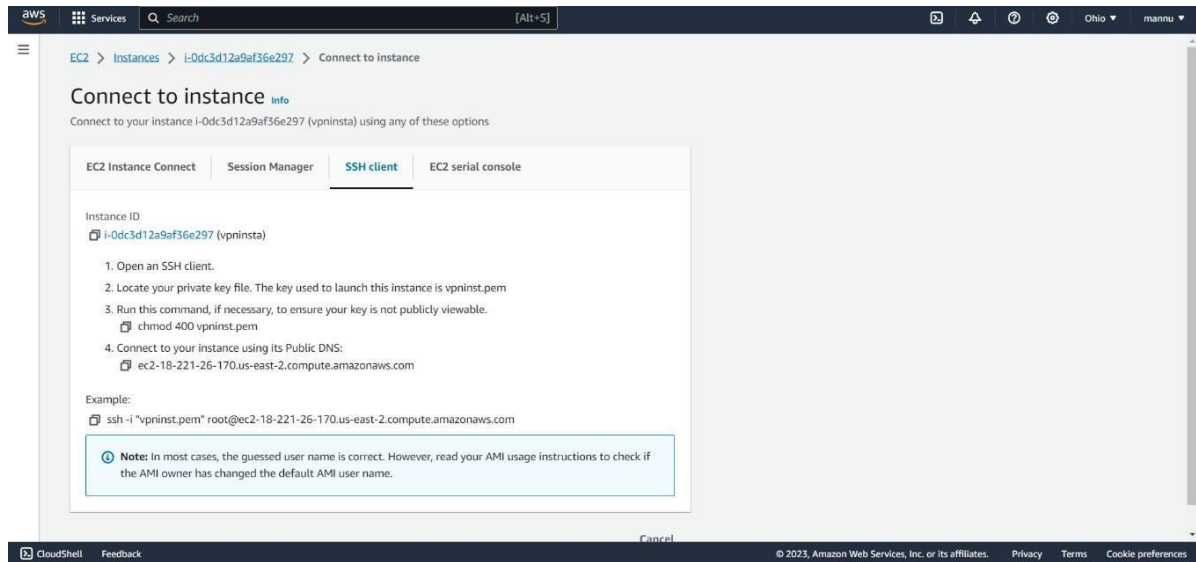
#### 5. Successfully launched



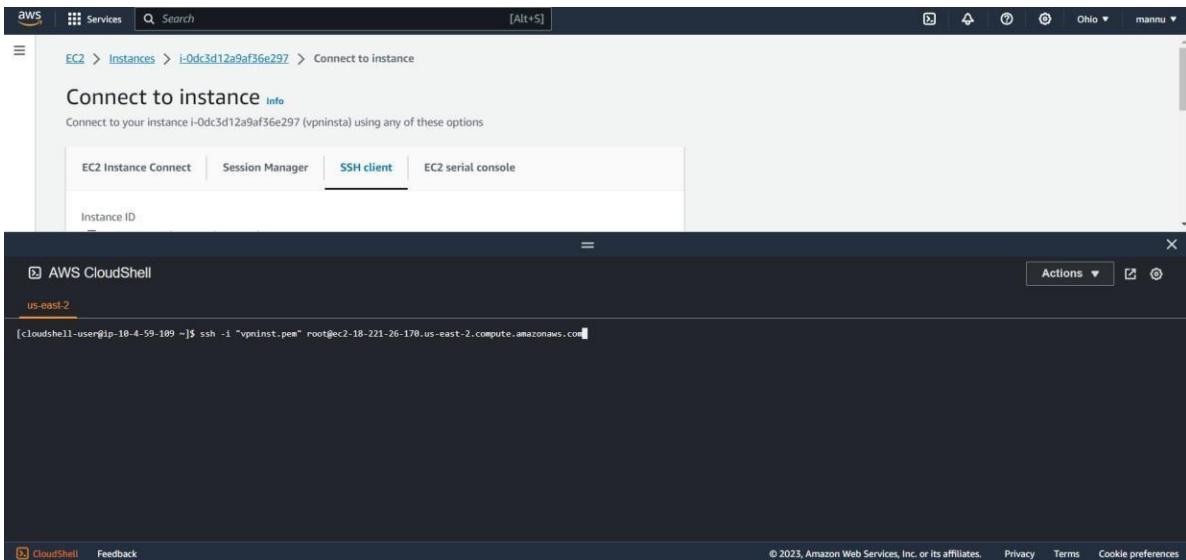
6, Copy public ipV4 address.



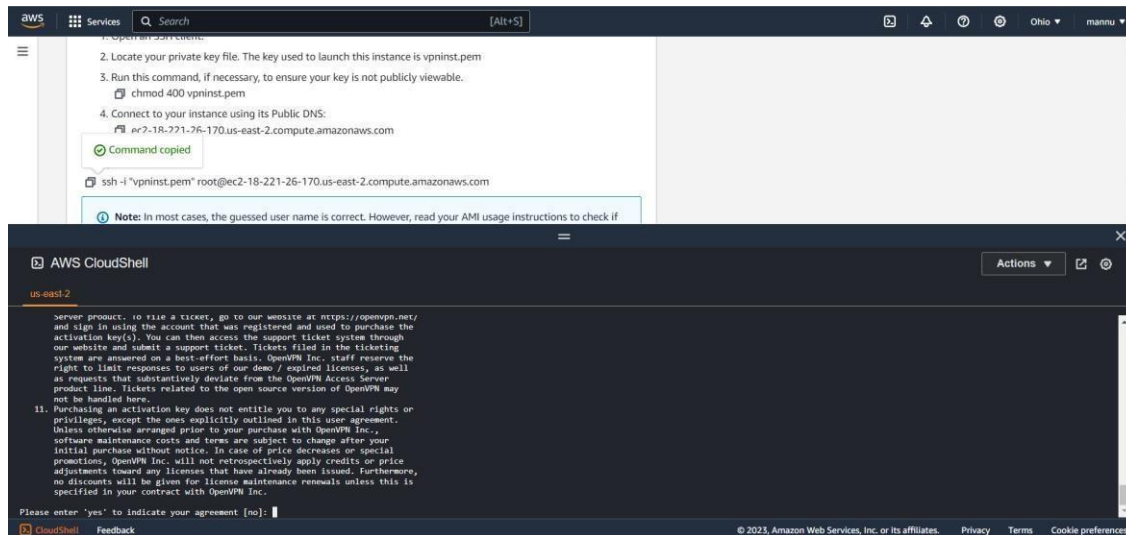
7. Open the link and connect to the instance.



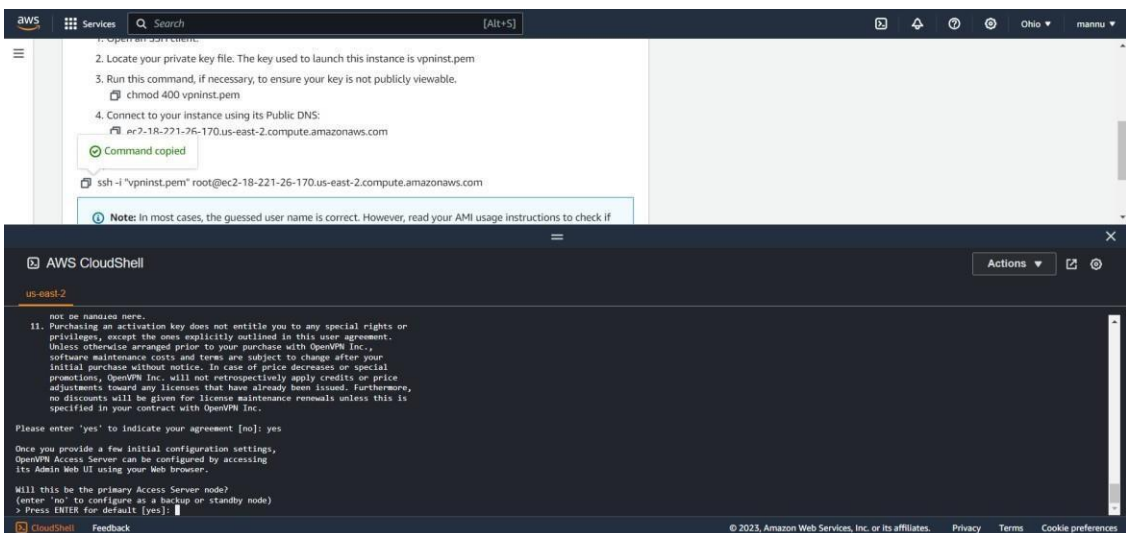
## 1. Enter the commands



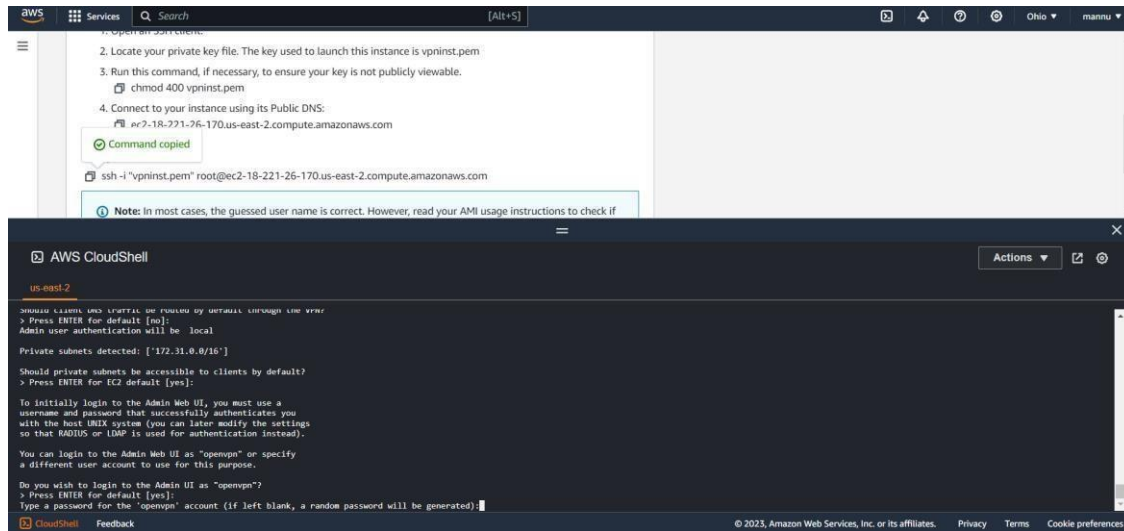
9. After that allow that command.



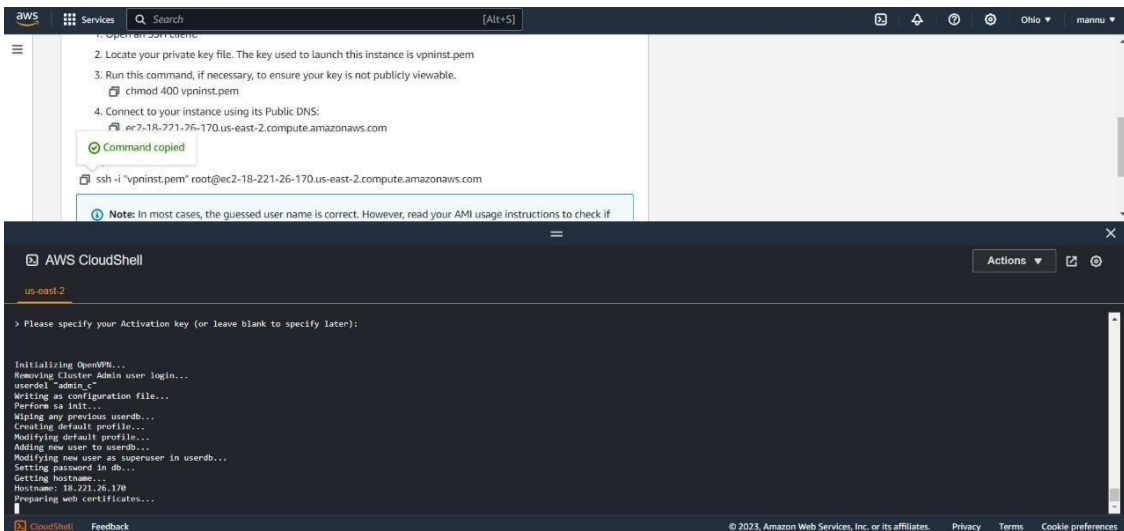
## 10. Follow all pseudo commands.



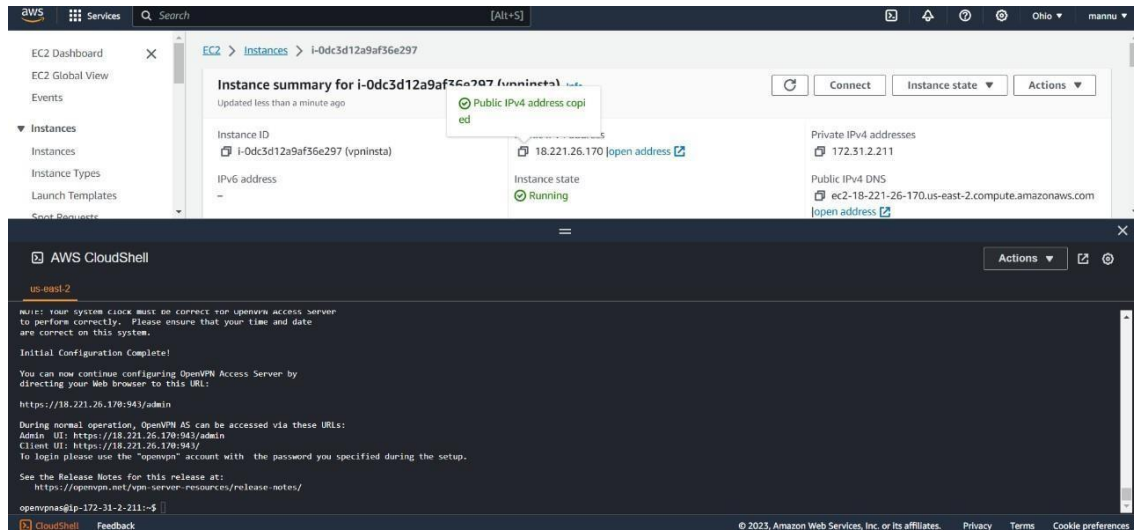
## 11. Set password.



12. Status showing that the password is successfully updated.



13. Now again copy the ipv4 address.



14. Copy the link and enter the new webpage.

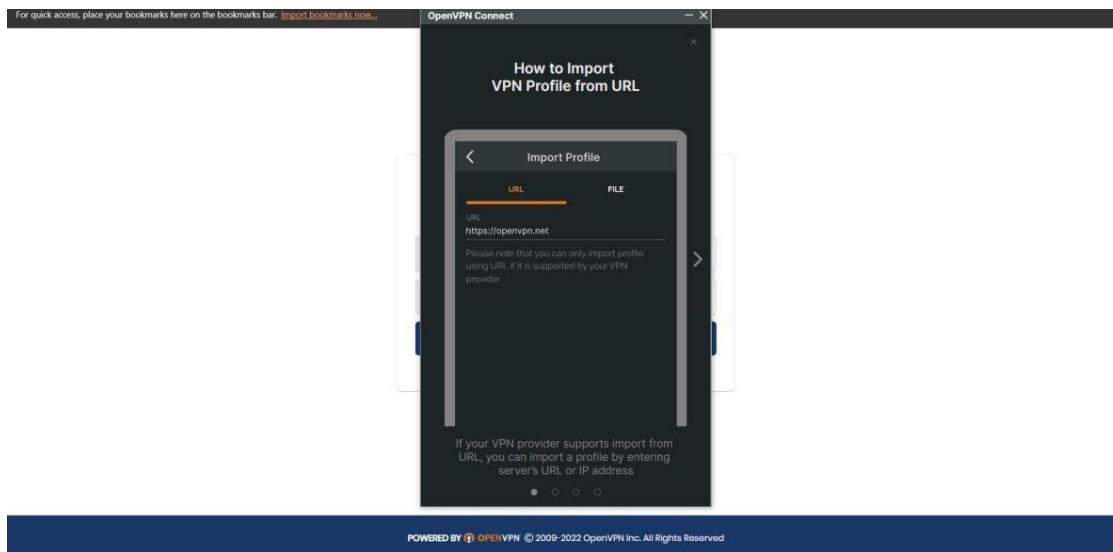
The screenshot shows the OpenVPN Access Server User Login page. The page features the OpenVPN Access Server logo at the top. Below the logo, there is a 'User Login' section with two input fields: 'Username' and 'Password'. A 'Sign In' button is located below the password field.


15. Run the software downloaded in the drive.



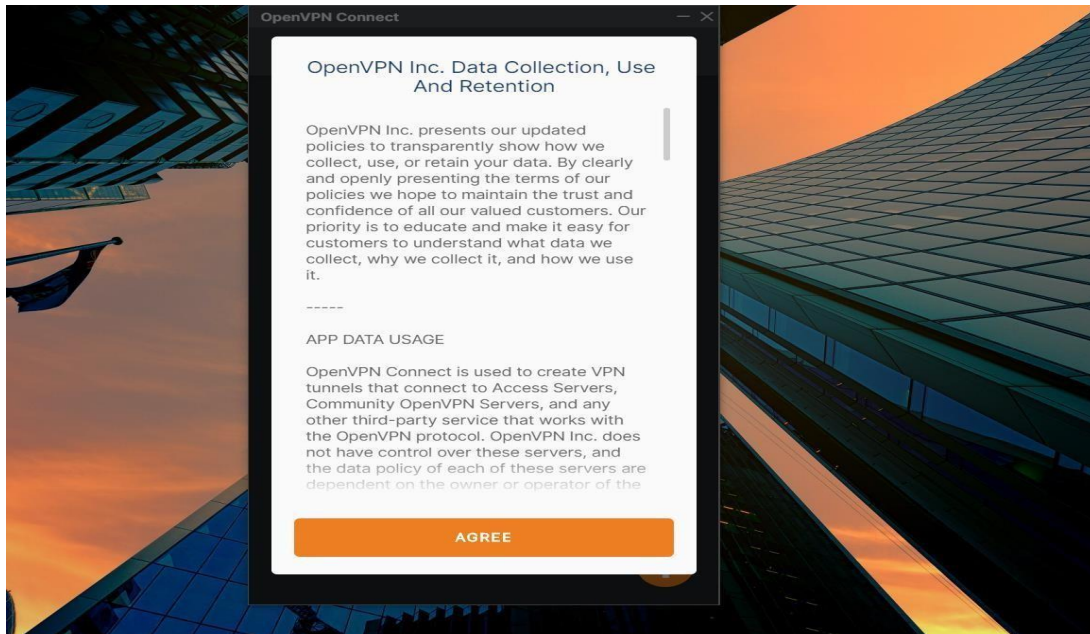
POWERED BY  OPENVPN © 2009-2022 OpenVPN Inc. All Rights Reserved

16. After installation you can see this new tab.

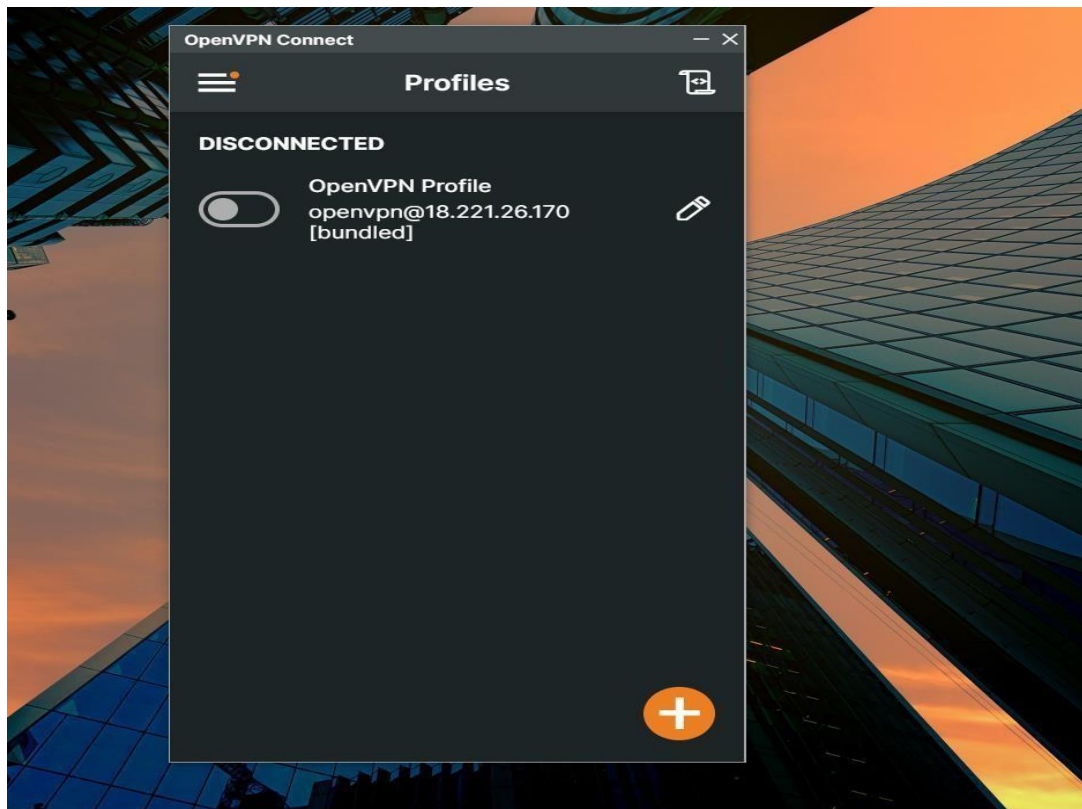


POWERED BY  OPENVPN © 2009-2022 OpenVPN Inc. All Rights Reserved



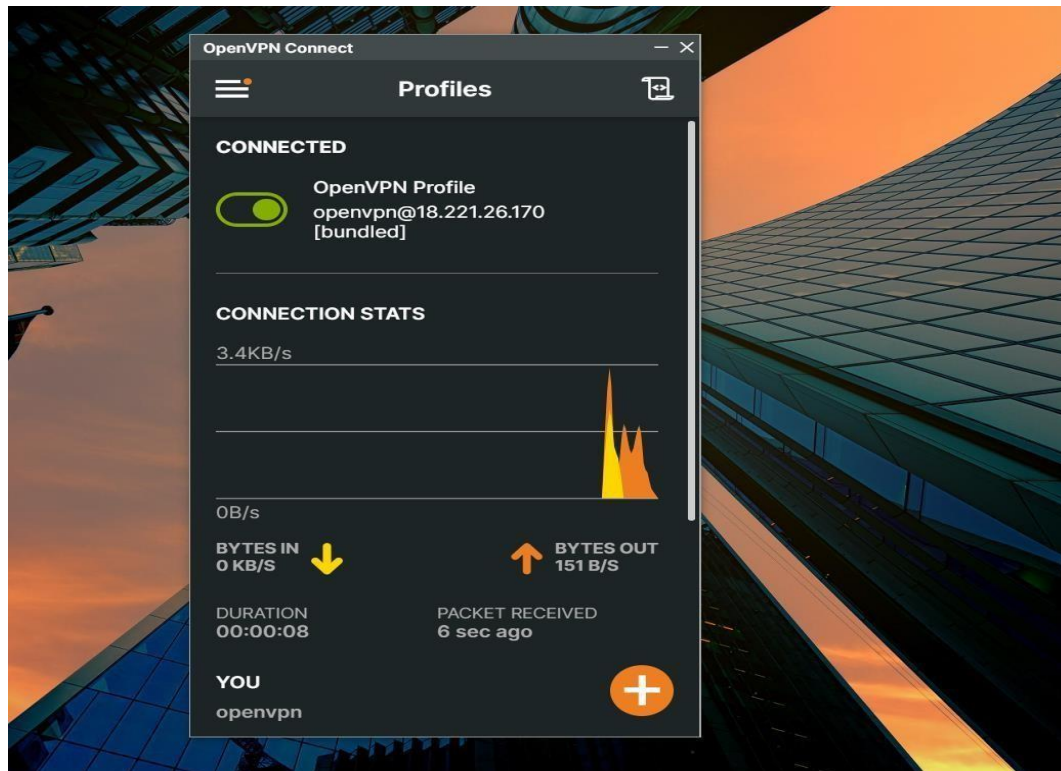


17. Connect it.



18. After connection the required result is obtained.





**RESULT:** The program is successfully completed and the output is verified.

EX NO:3

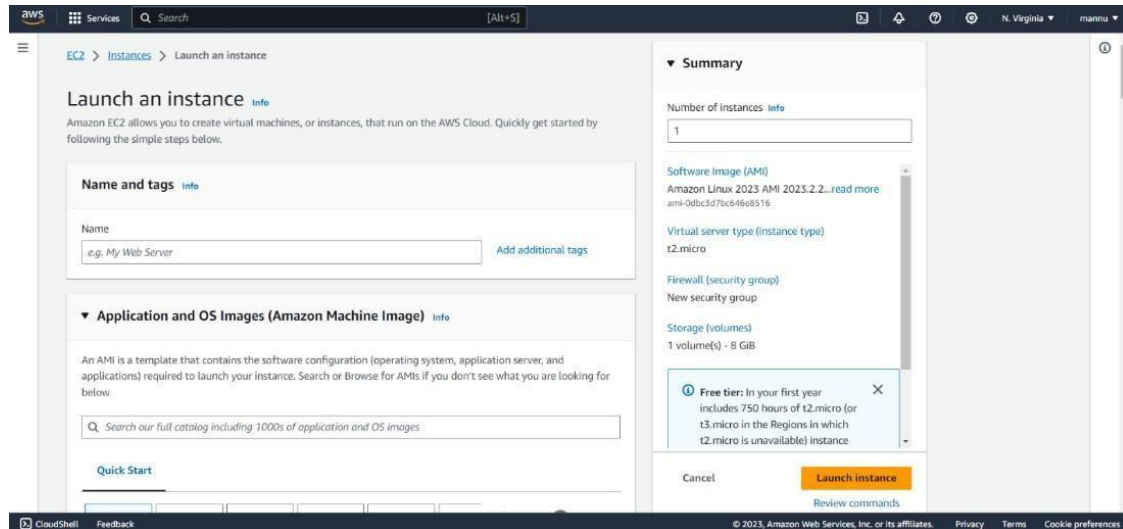
DATE:14/08/2023

## Eavesdropping Attacks and Prevention using SSH submission.

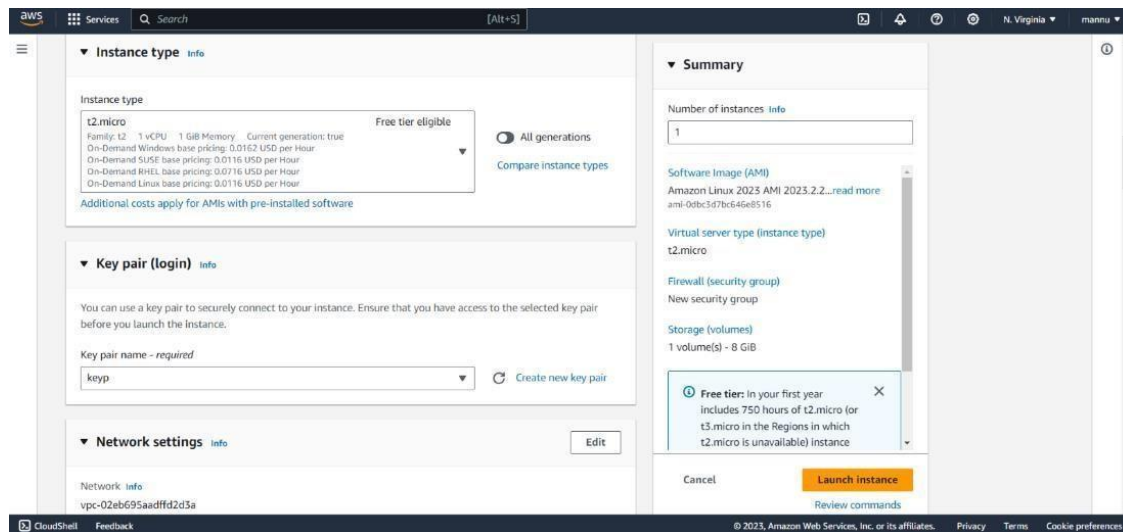
**Aim:** Eavesdropping Attacks and Prevention using SSH submission.

### Procedure:

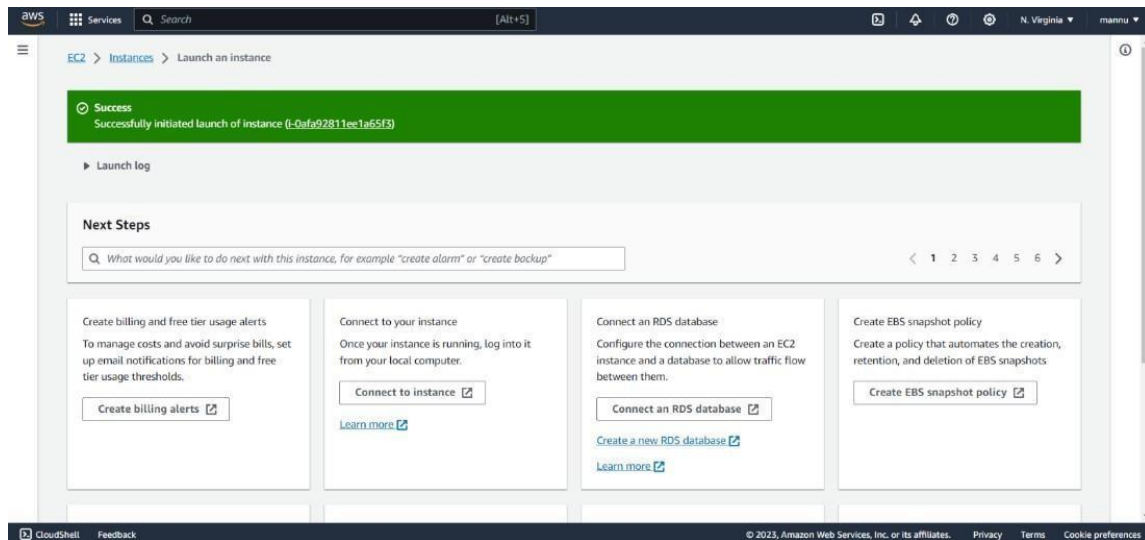
#### 1.Login to Aws account and create an instance



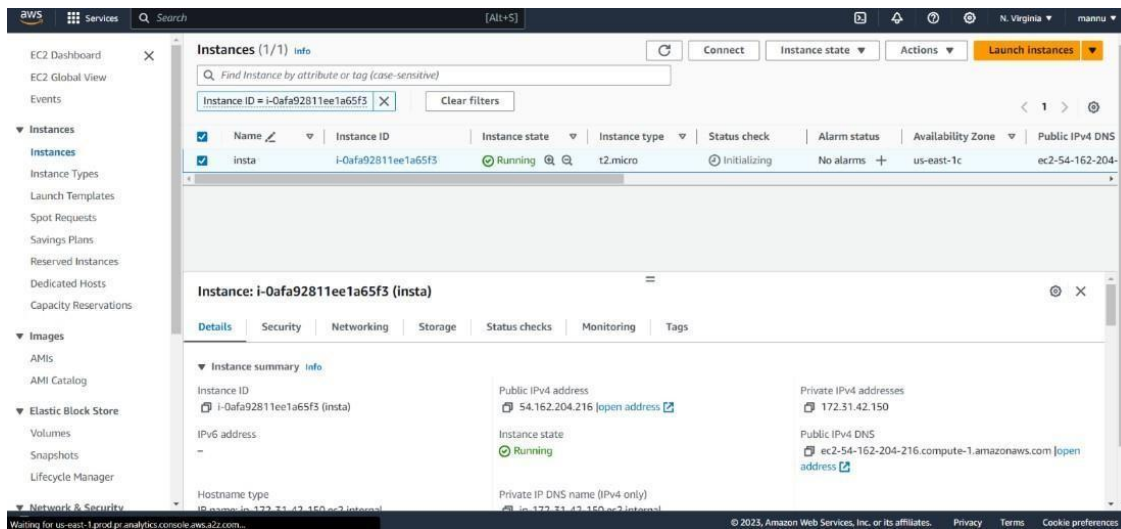
#### 2.Launch Instance.



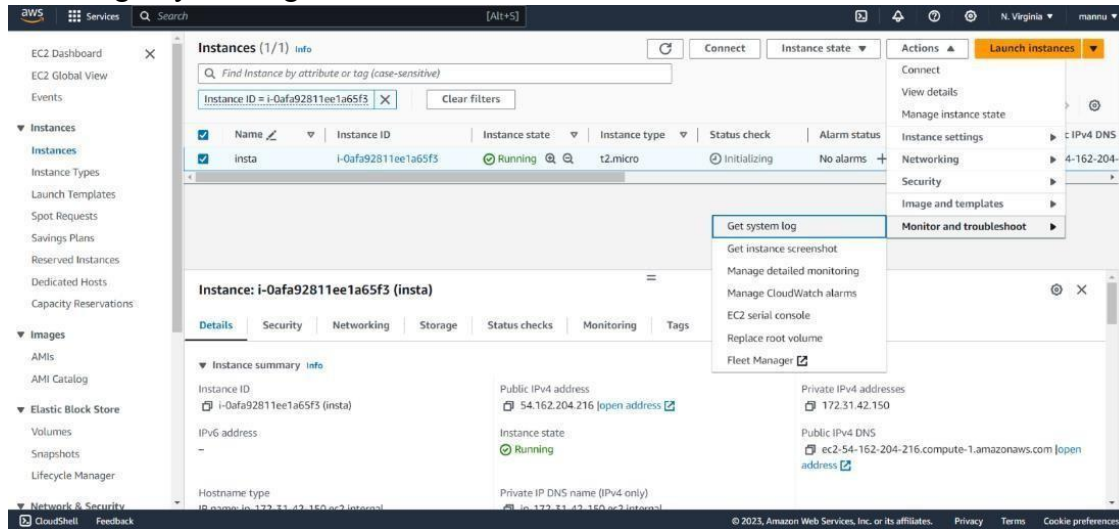
### 3. Instance Launched.



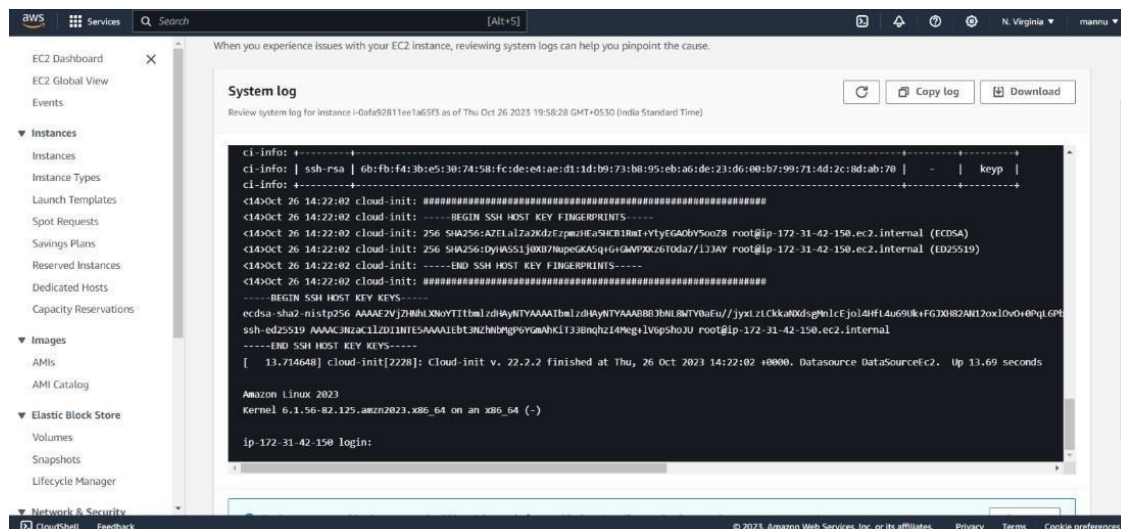
### 4. Select newly created instance.



5. Select get system log.



6. After selecting this cmd is open and running some code leave sometime.



7. After running copy ssh second line and paste in cmd.



EX NO: 4	KISMET
DATE: 22/08/2023	

**AIM:** To install and see the working of Kismet to find and monitor nearby WIFI.

## **PROCEDURE:**

### **Step 1- Install Kismet**

To install Kismet on Kali Linux, we'll first clone the git repository with the command below.

**git clone <https://www.kismetwireless.net/git/kismet.git>**

Depending on which OS you're using, Kismet may not need any dependencies. But to ensure Kismet runs correctly, we should install Kismet's slightly lengthy list of dependencies. These are needed because Kismet deals with detecting, decoding, logging, and sorting lots of wireless data while controlling a wireless card, which requires several libraries to be installed. You can do this by running the following in a terminal window.

**sudo apt-get install build-essential git libmicrohttpd-dev zlib1g-dev libnl-3-dev libnl-genl-3-dev libcap-dev libpcap-dev libncurses5-dev libnm-dev libdw-dev libsqlite3-dev**

Next, navigate to the Kismet directory we created using cd, and configure the installation.

**cd kismet**

**./configure**

This will configure the installation for your particular OS distribution. When that process is complete, create the installation with:

## **make**

When this is complete, we'll run the resulting file to complete the installation with the `suidinstall` option. This is important because Kismet is directly taking in signals and writing data to your computer. It is a terrible idea to do this as a root user because if any of that data is malicious, it could be executed as root.

When unprivileged users need to accomplish tasks that require privileges, like controlling the wireless network adapter, Linux lets us give privileges to programs instead of users so we don't have to make everyone, including malware, root.

To mitigate [giving root access], Kismet uses separate processes to control the network interfaces and capture packets. These capture programs are much smaller than Kismet itself and do minimal (or no) processing on the contents of the packets they receive

Run the following to complete the SUID installation

## **sudo make suidinstall**

After Kismet is installed, add yourself to the Kismet group to be able to capture packets as a non-root user. Be sure to replace "YourUsername" with your actual username.

## **sudo usermod -a -G kismet YourUsername**

## **Step 2- Put Your Wireless Card in Monitor Mode**

Attach your wireless network card to your computer, and if needed, attach it to the virtual machine using the "USB" settings. To find your card, you can use the `ip` or `ifconfig` commands. Your card should be named something like "wlan1" or "wlan0."

Once you have the name of your card, you can put the card in monitor mode by running the command below.

## **sudo airmon-ng start YourCardName**

This will put `YourCardName` (be sure to replace with your actual card's name) in monitor mode. Your card will be renamed to add a "mon" at the end of the name of the

card. So, if it was named "wlan0" before, it will now be named "wlan0mon". This change lets us immediately identify that a card is in wireless monitor mode.

We will use this new name for the card to launch Kismet.

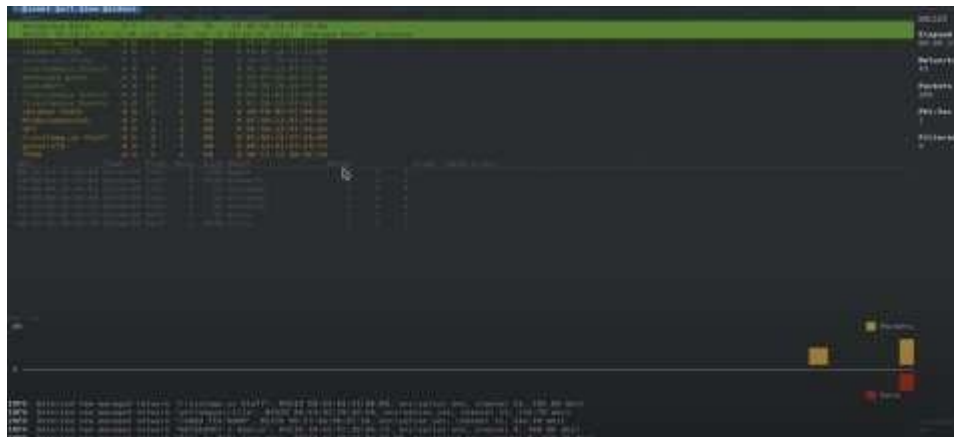
### **Step 3-Launch Kismet**

Starting Kismet is simple. To start as a non-root user, you can simply type the following.

**kismet -c YourCardNameMon**

### **Step 4-Persistent Network Surveillance**

Once we start Kismet, we should see a list of all the Wi-Fi devices we can detect nearby. The number of devices detected will vary depending on if you're scanning 2.4 GHz, 5 GHz, or both. If you have the ability to add an antenna to your wireless network adapter, a higher gain (or directional) antenna can extend your range and the number of devices detected.



Upon highlighting a network, the first thing we'll notice is the list of wireless clients appears in the main window. These are clients that are associated with the network.





To learn more about a specific network's clients, you can, after highlighting the client window, we can see more information about each client in real time. If you have a network that you want to monitor persistently, it's a good idea to note

the channel number. Since Kismet is exploring all channels by hopping through them, you will miss all transmissions on one channel while Kismet is scanning another. This packet fragmentation can cause you to lose data, so once you identify the network you wish to watch, you should switch from "scanning" to persistently monitoring one channel. This will allow you to capture all activity on the channel

To do this, click on "Kismet" in the top-left corner, and then click on "Config Channel"

In the configuration window, select "Lock," and then enter the number of the channel you want to monitor.

### **Step5-Watch for Patterns & Explore Around You**

Human behavior will have an effect on the wireless signals around you, and Kismet can let you watch these normally perceivable changes in the wireless environment. It doesn't matter that these networks are encrypted because the relationships between them and plaintext portions of packets are more than enough. By watching the type of traffic flowing across networks, we can take a step beyond simply seeing what is around us and instead begin to learn how these networks are used and by whom.

In particular, Kismet has an "Alerts" section in the menu under "Windows" that will warn you of any suspicious wireless behavior. This can detect things like networks switching channels, deauth packets, networks spoofing other networks, and APs that are rapidly switching names. Our writers accidentally turned on

a Hak5 Wi-Fi\_Pineapple while monitoring with Kismet and nearly had a panic attack when a torrent of incredibly serious sounding alert messages started cascading down our screen detecting what was obviously targeted Wi-Fi hacking.

**STEP 6-Hiding Your Activity from Cheap & Easy Wireless Surveillance** Earlier, I mentioned that Wi-Fi can be detected nearly a mile away using a directional Wi-Fi antenna. These signals are so strong that they are a backup for GPS navigation for the military via NAVSOP (Navigation via Signals of Opportunity). If the military can fly planes by the light of your Wi-Fi network, maybe it's time to consider if you need it turned to the very highest setting. which it almost definitely is right now., in order to just get Wi-Fi in your house or business. Most people have logged into their router exactly once and never change any of the settings beyond the required ones. While the instructions\_are different \_for \_each brand of router, nearly every brand will have a power setting. You can turn this down. Way down. Manufacturers jack it all the way up by default so that you don't complain about the signal strength. If you don't have trouble with your Wi-Fi range, reduce it so it only covers the area you need. Anything you want kept secret should be hard-wired, plain and simple. If you can't block the signals from going out of your house and being picked up by a sensitive antenna, don't put those signals out in the first place. If you have to, you can use Kismet to test the range of when someone can pick up data from your network.

### **Hiding Your Devices from the Kismet List**

For client devices, including smartphones, turn off the Wi-Fi setting whenever you don't need it. Your Wi-Fi card can be used to track you anywhere, not just at home or work. This is true even while you are not connected to Wi-Fi. Devices that rely on Wi-Fi to function you can't do much about. Smartphone manufacturers try to randomize the MAC address that your phone advertises while walking around, but this goes out the window as soon as the phone tries to associate with a network it thinks it knows. This is super easy to do to a crowd of people, which means it doesn't stand up to a real attack. Don't believe me? If you change your phone's

mobile hotspot to "Google Starbucks," nearly every smartphone nearby will connect to you and reveal its true MAC address, allowing you to track it.

Country	City	Lat	Long	Alt	Pop	Area	Time	Code	Notes
China	Guangzhou	23.12	113.27	44	12,400,000	7,434	UTC+8	94	Guangdong
China	Shanghai	31.22	121.47	4	22,000,000	6,340	UTC+8	95	Shanghai
China	Beijing	39.90	116.40	43	19,000,000	16,411	UTC+8	96	Beijing
China	Chengdu	30.57	104.06	1,600	14,000,000	17,000	UTC+8	97	Sichuan
China	Wuhan	30.59	114.30	39	8,000,000	8,396	UTC+8	98	Hubei
China	Xi'an	34.26	108.95	2,400	7,000,000	10,108	UTC+8	99	Shaanxi
China	Harbin	45.75	126.63	150	9,000,000	75,000	UTC+8	100	Heilongjiang
China	Qingdao	36.06	120.37	10	7,000,000	10,654	UTC+8	101	Shandong
China	Nanjing	32.06	118.78	15	8,000,000	6,594	UTC+8	102	Jiangsu
China	Hangzhou	30.27	120.15	5	8,000,000	16,883	UTC+8	103	Zhejiang
China	Shenzhen	22.54	114.05	1	12,000,000	3,277	UTC+8	104	Guangdong
China	Guangzhou	23.12	113.27	44	12,400,000	7,434	UTC+8	105	Guangdong
China	Shanghai	31.22	121.47	4	22,000,000	6,340	UTC+8	106	Shanghai
China	Beijing	39.90	116.40	43	19,000,000	16,411	UTC+8	107	Beijing
China	Chengdu	30.57	104.06	1,600	14,000,000	17,000	UTC+8	108	Sichuan
China	Wuhan	30.59	114.30	39	8,000,000	8,396	UTC+8	109	Hubei
China	Xi'an	34.26	108.95	2,400	7,000,000	10,108	UTC+8	110	Shaanxi
China	Harbin	45.75	126.63	150	9,000,000	75,000	UTC+8	111	Heilongjiang
China	Qingdao	36.06	120.37	10	7,000,000	10,654	UTC+8	112	Shandong
China	Nanjing	32.06	118.78	15	8,000,000	6,594	UTC+8	113	Jiangsu
China	Hangzhou	30.27	120.15	5	8,000,000	16,883	UTC+8	114	Zhejiang
China	Shenzhen	22.54	114.05	1	12,000,000	3,277	UTC+8	115	Guangdong
China	Guangzhou	23.12	113.27	44	12,400,000	7,434	UTC+8	116	Guangdong
China	Shanghai	31.22	121.47	4	22,000,000	6,340	UTC+8	117	Shanghai
China	Beijing	39.90	116.40	43	19,000,000	16,411	UTC+8	118	Beijing
China	Chengdu	30.57	104.06	1,600	14,000,000	17,000	UTC+8	119	Sichuan
China	Wuhan	30.59	114.30	39	8,000,000	8,396	UTC+8	120	Hubei
China	Xi'an	34.26	108.95	2,400	7,000,000	10,108	UTC+8	121	Shaanxi
China	Harbin	45.75	126.63	150	9,000,000	75,000	UTC+8	122	Heilongjiang
China	Qingdao	36.06	120.37	10	7,000,000	10,654	UTC+8	123	Shandong
China	Nanjing	32.06	118.78	15	8,000,000	6,594	UTC+8	124	Jiangsu
China	Hangzhou	30.27	120.15	5	8,000,000	16,883	UTC+8	125	Zhejiang
China	Shenzhen	22.54	114.05	1	12,000,000	3,277	UTC+8	126	Guangdong
China	Guangzhou	23.12	113.27	44	12,400,000	7,434	UTC+8	127	Guangdong
China	Shanghai	31.22	121.47	4	22,000,000	6,340	UTC+8	128	Shanghai
China	Beijing	39.90	116.40	43	19,000,000	16,411	UTC+8	129	Beijing
China	Chengdu	30.57	104.06	1,600	14,000,000	17,000	UTC+8	130	Sichuan
China	Wuhan	30.59	114.30	39	8,000,000	8,396	UTC+8	131	Hubei
China	Xi'an	34.26	108.95	2,400	7,000,000	10,108	UTC+8	132	Shaanxi
China	Harbin	45.75	126.63	150	9,000,000	75,000	UTC+8	133	Heilongjiang
China	Qingdao	36.06	120.37	10	7,000,000	10,654	UTC+8	134	Shandong
China	Nanjing	32.06	118.78	15	8,000,000	6,594	UTC+8	135	Jiangsu
China	Hangzhou	30.27	120.15	5	8,000,000	16,883	UTC+8	136	Zhejiang
China	Shenzhen	22.54	114.05	1	12,000,000	3,277	UTC+8	137	Guangdong
China	Guangzhou	23.12	113.27	44	12,400,000	7,434	UTC+8	138	Guangdong
China	Shanghai	31.22	121.47	4	22,000,000	6,340	UTC+8	139	Shanghai
China	Beijing	39.90	116.40	43	19,000,000	16,411	UTC+8	140	Beijing
China	Chengdu	30.57	104.06	1,600	14,000,000	17,000	UTC+8	141	Sichuan
China	Wuhan	30.59	114.30	39	8,000,000	8,396	UTC+8	142	Hubei
China	Xi'an	34.26	108.95	2,400	7,000,000	10,108	UTC+8	143	Shaanxi
China	Harbin	45.75	126.63	150	9,000,000	75,000	UTC+8	144	Heilongjiang
China	Qingdao	36.06	120.37	10	7,000,000	10,654	UTC+8	145	Shandong
China	Nanjing	32.06	118.78	15	8,000,000	6,594	UTC+8	146	Jiangsu
China	Hangzhou	30.27	120.15	5	8,000,000	16,883	UTC+8	147	Zhejiang
China	Shenzhen	22.54	114.05	1	12,000,000	3,277	UTC+8	148	Guangdong
China	Guangzhou	23.12	113.27	44	12,400,000	7,434	UTC+8	149	Guangdong
China	Shanghai	31.22	121.47	4	22,000,000	6,340	UTC+8	150	Shanghai

**RESULTS:** Thus the kismet is installed and worked successfully.

EX NO:5	SECURITY GROUP POLICIES MANAGEMENT
DATE:31/08/2023	

**AIM:** To do the security group policy management in our system.

## **PROCEDURE:**

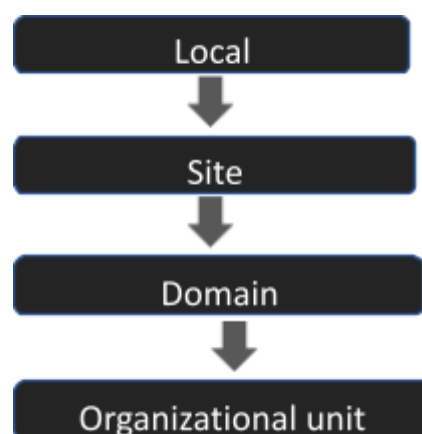
### **Implementing Group Policy Objects**

Group policies are typically linked to multiple objects in the Active Directory, including sites, domains, or organizational units. However, they cannot be directly linked to individual users or security groups.

When multiple GPOs (Group Policy Objects) are linked to organizational units, they are processed following a hierarchy. The GPO linked to the highest-level OU is processed first, followed by its child OUs, and so on, until it reaches the specific computer or user object. The GPO associated with the last processed OU takes precedence and overrides any previous settings.

In terms of processing order, the GPOs are processed based on the hierarchy and object type, with sites being processed first, then domains, and finally organizational units. This means that the GPOs linked to sites have the lowest precedence, followed by domain-linked GPOs, and finally, organizational unit-linked GPOs, which have the highest precedence in the processing order. This order is essential to understand as it determines which GPO settings apply and which ones take precedence when there are conflicting policies.

The order of processing for Group Policy Objects is from the broadest level, such as the site, down to the most specific level, like the organizational unit, with the last processed GPO having the highest priority and overriding any conflicting settings.



Group Policy is processed in order

**NOTE:** To view the precedence of Group Policies on any site, domain, or Organizational Unit (OU), go to the "Group Policy inheritance" tab. The first object that applies last takes precedence over all others.

## **Implementing Group Policy on an Organizational Unit**

First, ensure that you have the Group Policy Management Console (GPMC) installed in the administrative tools. If not, follow these steps:

1. Go to Administrative Tools > Group Policy Management.
2. Right-click on Group Policy Objects and select "New."
3. Name your policy.
4. Click "Link an Existing GPO" to link a GPO to an Organizational Unit.
5. Select the Group Policy Object you want to link.

Note: You can edit the GPO by right-clicking and selecting "Edit."

## **Configuring Group Policy Settings**

A Group Policy Object (GPO) is a collection of Group Policy settings. Normally, a user can edit GPOs that are locally installed on each computer, yet an administrator can create non-local GPOs.

**To disable Windows Messenger from running, follow these steps:**

1. Create a GPO as mentioned above.
2. Right-click and edit the Group Policy settings.
3. Your new GPO will appear to the right. Right-click on the GPO and select "Edit."
4. Navigate to User Configuration > Administrative Templates > Windows Components > Windows Messenger.
5. Two Group Policy settings will appear. Double-click on "Do not allow Windows Messenger to be run."
6. There are three options: Not configured, enabled, and disabled. Not configured means that the GPO takes precedence and allows users to use Windows Messenger. Enabled means that you are not allowing Windows Messenger to run, and disabled allows users to run Windows Messenger. (In case you forget what each means, you can press the "Explain" tab, and a clear description will show.)

Note: After configuring the GPO, you can still link it to an OU, Domain, or site by right-clicking on one of the objects and choosing "Link an existing GPO."

## **Determining Applied GPOs**

To determine what effective Group Policies are applied to a user or computer, use the Group Policy Results Wizard. Open the Group Policy Management, then:

1. Right-click on "Group Policy Results" and select "Group Policy Results Wizard."
2. Select a user or computer.
3. A report will be displayed under "Group Policy Results."
4. Click "Show the report" to identify the section of interest.

## **Security Policies**

Some of the other popular security options you can set in Group Policy are:

### **1. Password Policy**

Go to Group Policy Object Editor > Computer Configuration > Windows Settings > Security Settings > Account Policies > Password Policy. There are six options:

- a. Enforce password history (keep a history of all passwords assigned and don't allow repetition).
- b. Maximum password age (Allows a maximum amount of time a password can be kept before changed. After it expires, the user has to assign a new password).
- c. Minimum password age (Determines the minimum amount of time a user can keep his/her password).
- d. Password must meet complex requirements (lower case, upper case, numbers, and special characters).
- e. Store passwords using reversible encryption (stores passwords in the equivalent of plaintext).

### **2.Account Lockout Policy**

Go to Computer Configuration > Windows Settings > Security Settings > Account Policies > Account Lockout Policy.

- a. Account lockout duration (A defined number of times in which the user is allowed to enter the correct password).
- b. Account lockout threshold (A defined number of times in which the user can enter an incorrect password).
- c. Reset account lockout counter after (resets the counter for lockouts after a predetermined time).

### **3.User Rights Assignment**

Go to Computer Configuration > Windows Settings > Security Settings > Local Policies > User Rights Assignment. You can choose either to deny or grant a right for users.

### **4.Security Options**

Go to Computer Configuration > Windows Settings > Security Settings > Local Policies > Security Options. You can choose to grant or deny a number of security options to certain users or groups.

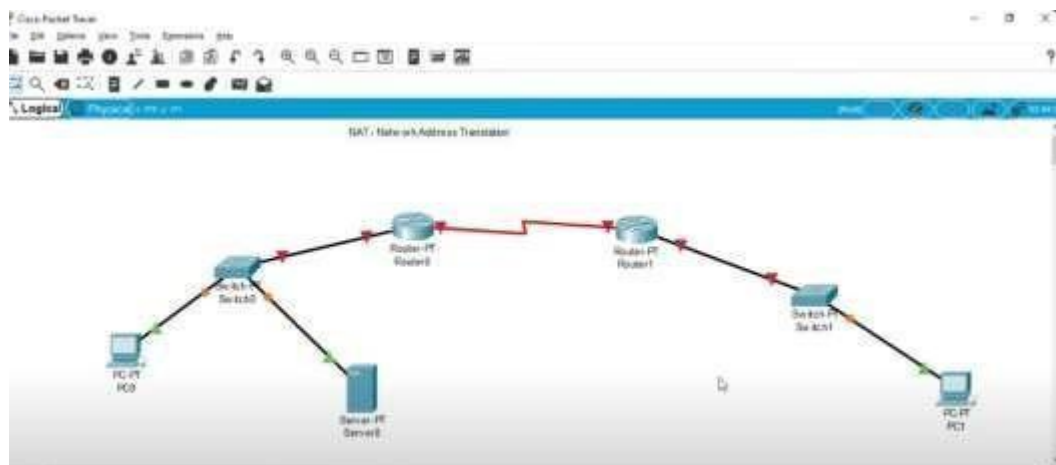
**RESULT:** The implementation of security group management policy

EX NO: 6	IMPLEMENTING NAT
DATE: 14/09/2023	

**AIM:** To implement NAT using Cisco Packet Tracer.

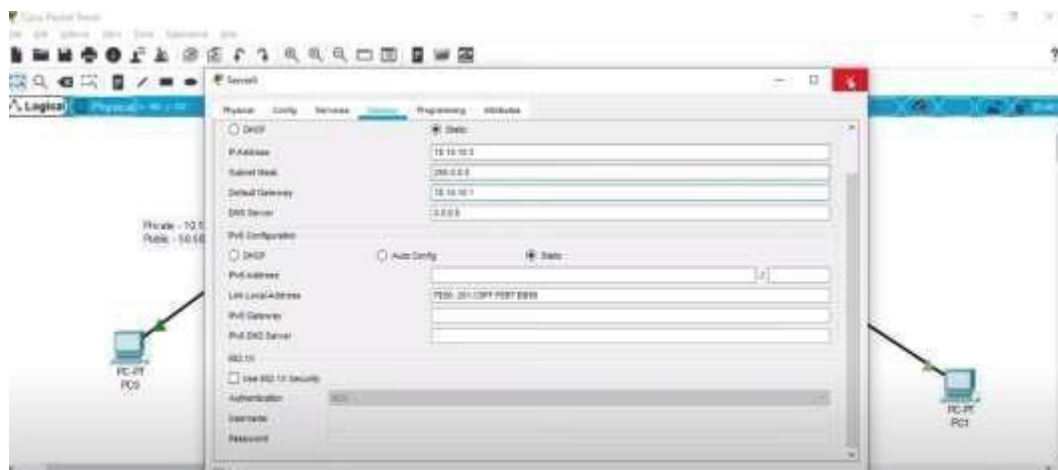
## PROCEDURE:

**Step 1:** Select routers, switches, PC and servers and connect with zigzag wire.

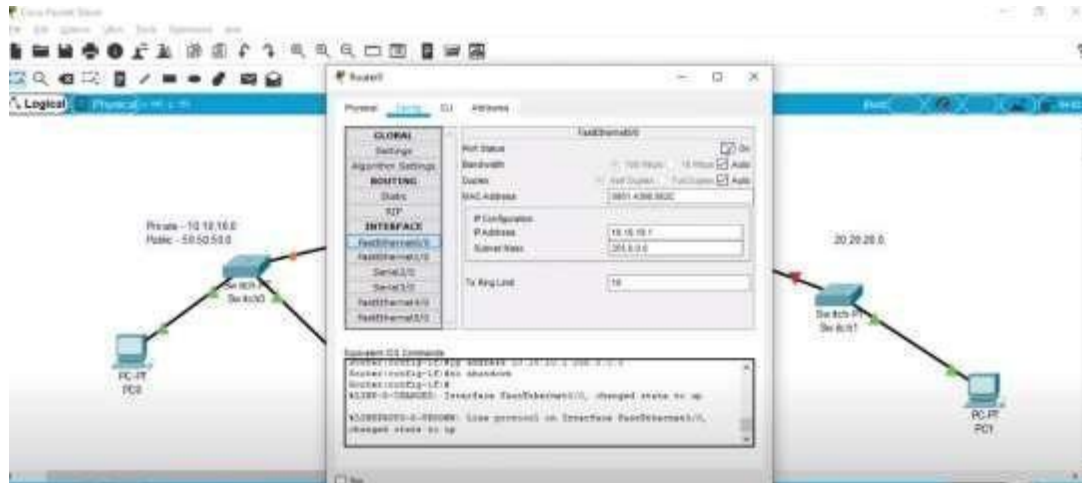


**Step 2:** Make three different network with different public and private address. **Step**

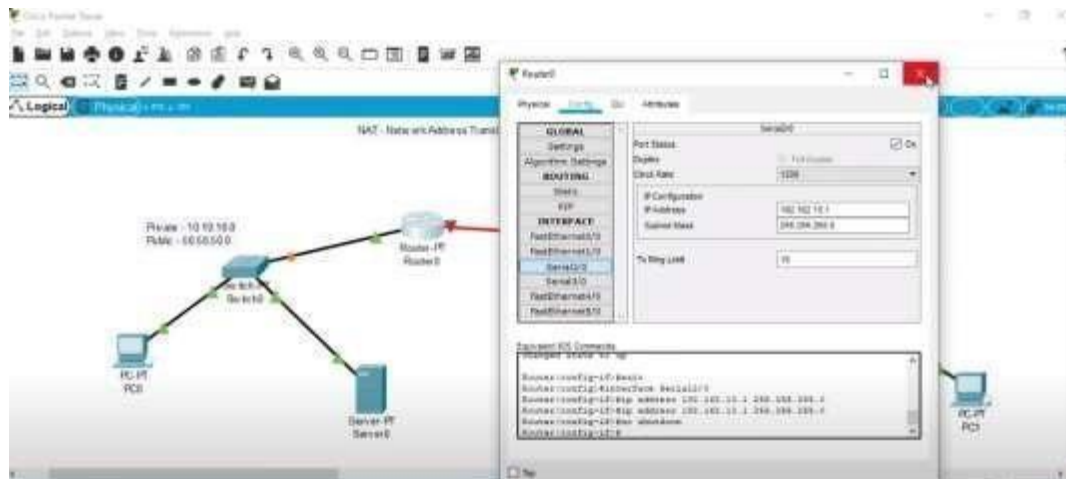
**3:** Now click on pe 0 and click on desktop, go to ip configuration and enter ip Address, subnet mask and default gateway.



**Step 4:** Do same way as step 3 for server and PC1.



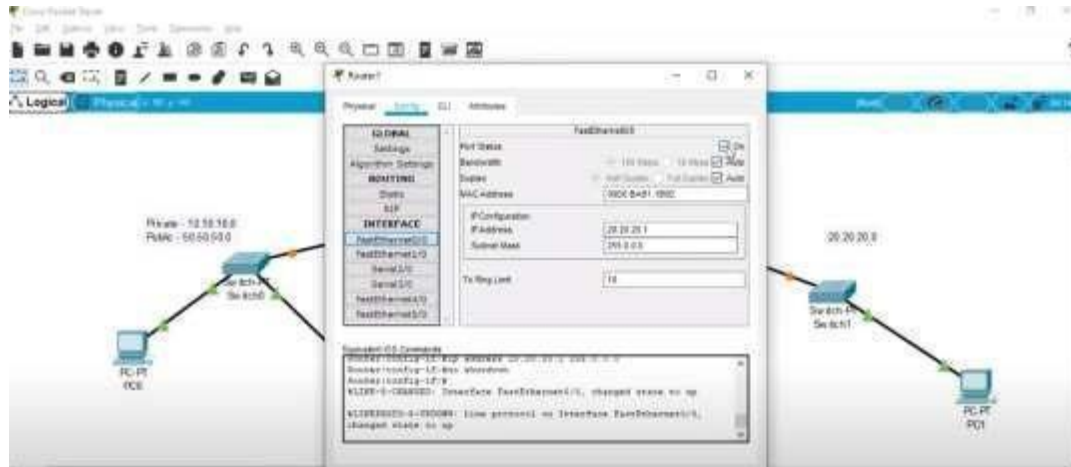
**Step 5:** Now click on router 0 and go to config and click Fast Ethernet 0/0, enter ip Address and subnet mask and check on in top right corner for network 1 i.e. 10.10.10.0.



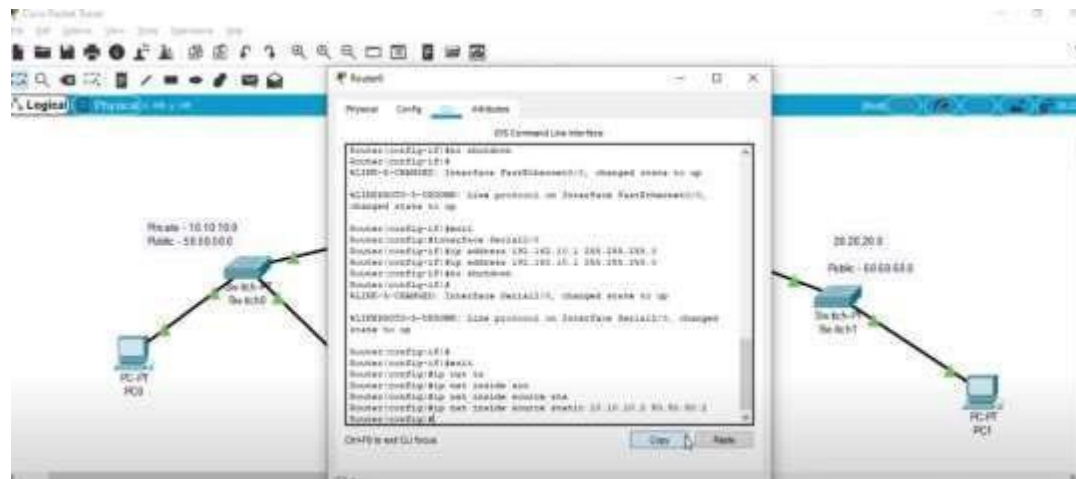
**Step 6:** Again in config router, go to serial 2/0, now enter the ip address and subnet Mask of network 2 i.e. 192.162.10.0 and click on ON check box in topbar.

**Step 7:** Do same way as step5 and step 6 for router 1 also for both network I.e 20.20.20.0 and 192.162.10.0.



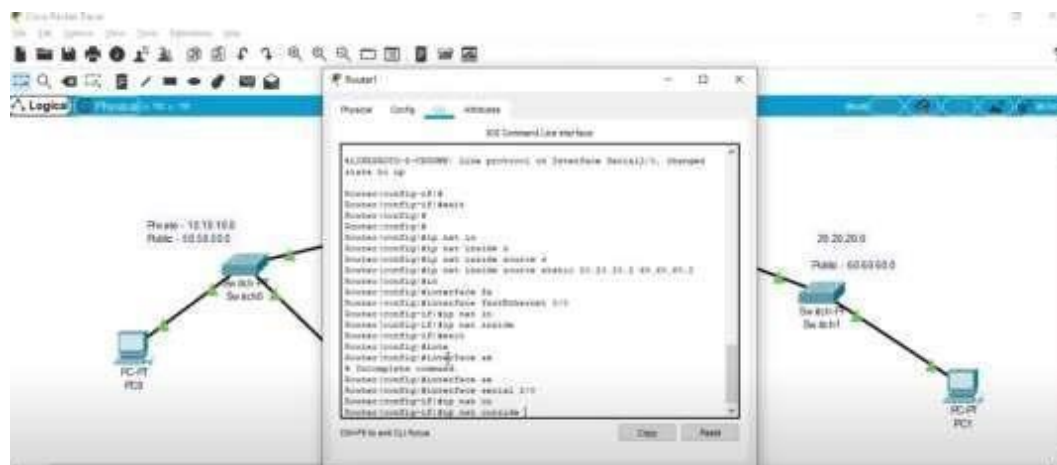


**Step 8:** Click on router 0 and go to CLI ,enter the following commands.



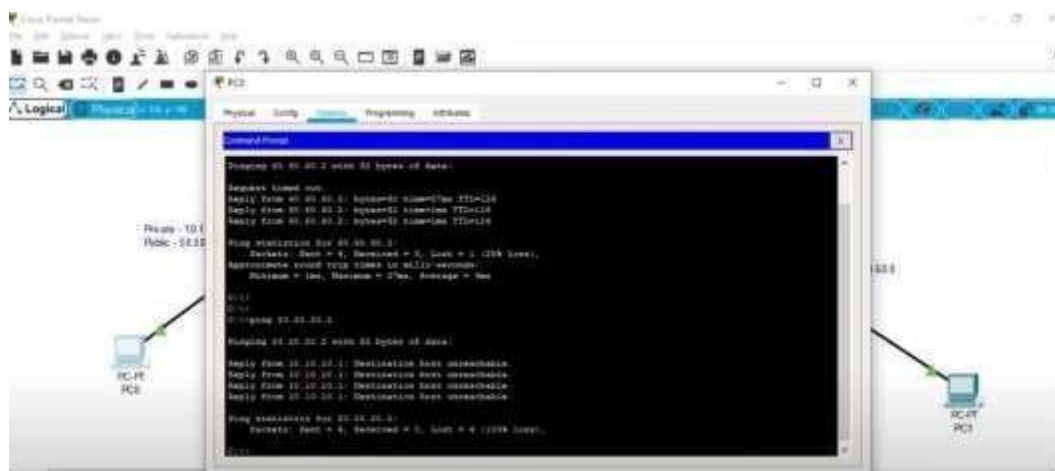
**Step 9:** Give static router information for router O and router I as

**Step 10:** To see the ip route of router 1 ,write the following command and you Will able to see the connection of router directly or via.



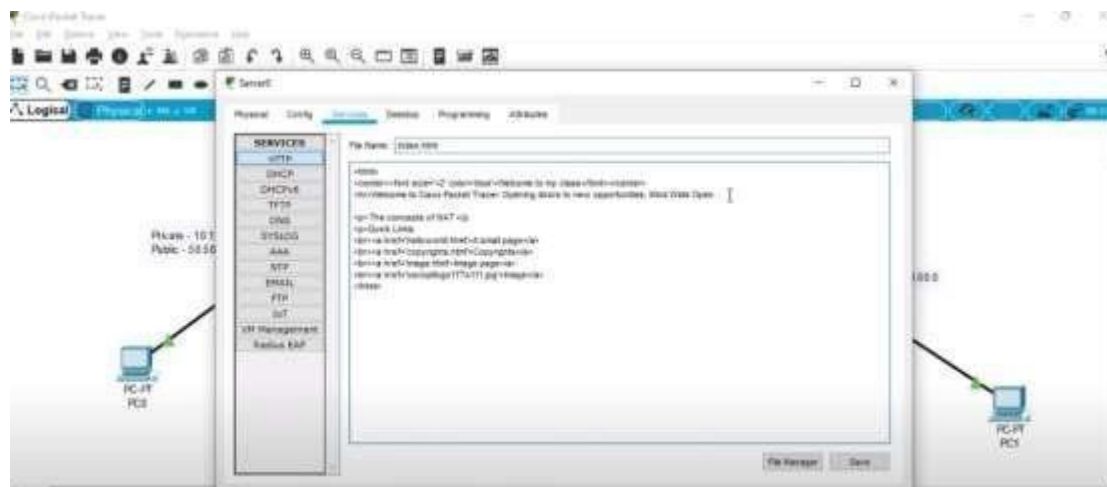
**Step 11:** Now if we pass the packet from pc 0 to pc 1 it will fail .So for that clickOn pc 0 and go to command prompt and enter the command ping 60.60.60.2 and

You will see that its pinged but if we ping private address i.e 20.20.20.2,it ll fail And show destination is unreachable. In that way private ip address will be hidden And only public IP address is accessible



**Step 12:** Click on server and then go to service and click on http and edit index file

And save it.



**Step 13:** Click on pe 1 and go to web browser and enter the public ip address of Network 1 as 50.50.50.3 and click on Go. After this you will see the index page of Network 1. But if we enter the private ip address of network 10.10.10.3, it will not

show address. In such any way since no one can private enter network the private address, they can only access public address.



**RESULT:** Implemented NAT successfully.

EX NO: 7	SCAPY
DATE: 22/09/2023	

**Aim:** To install and see the working of Scapy.

### **Procedure:**

#### **Installation of scapy module:**

As scapy module is not included in Python3 library by default, we have to add it into our Python library using pip. Execute this command in your Linux terminal to get the scapy module for Python3.

```
pip3 install scapy-python3
```

#### **Some important functions for creating Network scanner**

**ARP()**: This function defined in scapy module which allows us to create ARP packets (request or response). By default, if we are calling it, it will create an ARP request packet for us.

```
import scapy.all as scapy
```

```
request=scapy.ARP()
```

**Summary()**: This method provide us the status of the packet that we have created. It does not provide the detailed information about the packet, it just gives us the basic idea like what is the type of packet, what is the destination of the packet etc. For example if we want to create an ARP packet using ARP() method which is present in the scapy module and want to see the summary of the packet then we can do this by creating the object of ARP class

```
import scapy.all as scapy
```

```
)
```

```
request=scapy.ARP()
```

```
(print(request.summary()))
```

```
root@kali:~/Desktop# python3 networkscanner.py ARP
who has 0.0.0.0 says 192.168.215.144
root@kali:~/Desktop#
```

```
import scapy.all as scapy
request=scapy.ARP()
print(request.show())
```

```
PS D:\program\python> C:\Users\AJay_Fewer\code\stations\on-python\python-2021-1-24\223\223.py
one line python debugger launched: "223" -- "d:/program/python/scapy.py"
begin session:
finished sending 1 packets.
.
.
Received 1 packets, got 1 answers, remaining 0 packets
b4:ch:57:af:35:29 unknown 192.168.43.1
<adding: TCP:0 UDP:0 ICMP:0 Other:1>, <unanswered: TCP:0 UDP:0 ICMP:0 Other:0>
aaa| ARP |aaa
  htype      = 0x1
  ptype      = IPv4
  hlen       = None
  plen       = None
  op         = who-has
  hwsrc      = 48:68:ba:1f:12:dd
  psrc       = 192.168.43.136
  haddr      = 00:00:00:00:00:00
  paddr      = 192.168.43.1
```

In our example we will create an ARP packet and the with the help of `ls()` function, we will see what are the available fields for this packet.

1. Create an ARP packet using ARP() method.

2. Set the network range using variable.
3. Create an Ethernet packet using Ether() method.
4. Set the destination to broadcast using variable hwdst.
5. Combine ARP request packet and Ethernet frame using /".
6. Send this to your network and capture the response from different devices.
7. Print the IP and MAC address from the response packets.

**Below is the Python implementation**

```
import scapy.all as scapy
request = scapy.ARP(
request.pdst = 'x'
broadcast=scapy.Ether()
broadcast.dst = 'ff:ff:ff:ff:ff:ff'
request_broadcast = broadcast / request
clients=scapy.srp(request_broadcast, timeout=1)[0]
for element in clients:
print(element[1].psrc + "
"+element[1].hwsrc)
```

```
Received 1439 packets, got 48 answers,  
172.17.0.1      10:f0:05:41:f0:dc  
172.17.8.1      00:15:17:b4:b1:dd  
172.17.223.0    04:b1:67:ca:8f:67  
172.17.16.1     00:15:17:b4:b1:dd  
172.17.24.1     00:15:17:b4:b1:dd  
172.17.32.1     00:15:17:b4:b1:dd  
172.17.40.1     00:15:17:b4:b1:dd  
172.17.48.1     00:15:17:b4:b1:dd  
172.17.56.1     00:15:17:b4:b1:dd  
172.17.64.1     00:15:17:b4:b1:dd  
172.17.72.1     00:15:17:b4:b1:dd  
172.17.208.1    00:15:17:b4:b1:dd  
172.17.210.1    26:fa:6f:71:5f:c7  
172.17.224.1    00:15:17:b4:b1:dd  
172.17.240.1    00:15:17:b4:b1:dd  
172.17.222.1    34:f6:4b:a3:63:8f  
172.17.223.2    20:16:b9:8d:05:c7  
172.17.216.3    38:e6:0a:de:6c:d4  
172.17.221.3    20:34:fb:72:53:ed
```

**RESULT:** Implemented Scapy successfully.

EX NO: 8	COWPATTY
DATE: 05/10/2023	

**Aim:** To find Cowpatty

### Procedure:

Cowpatty is one of the hundreds of pieces of software that are included in the BackTrack suite of software. For some reason, it was not placed in the `/pentest/wireless` directory, but instead was left the `/usr/local/bin` directory, so let's navigate there.

- `cd /usr/local/bin`

Because cowpatty is in the `/usr/local/bin` directory and this directory should be in your PATH, we should be able to run it from any directory in BackTrack.

Find the Cowpatty Help Screen

To get a brief rundown of the cowpatty simply type:

### Cowpatty

```
root@kali:~# cowpatty --help
cowpatty 4.6 - WPA-PSK dictionary attack. <jwright@hasborg.com>
cowpatty: invalid option -- '-'

Usage: cowpatty [options]

    -f      Dictionary file
    -d      Hash file (genpmk)
    -r      Packet capture file
    -s      Network SSID (enclose in quotes if SSID includes spaces)
    -c      Check for valid 4-way frames, does not crack
    -h      Print this help information and exit
    -v      Print verbose information (more -v for more verbosity)
    -V      Print program version and exit

root@kali:~#
```

BackTrack will provide you a brief help screen. Take a note that cowpatty requires all of the following.



- a word list
- a file where the password hash has been captured
- the SSID of the target AP

Place the Wireless Adapter in Monitor Mode

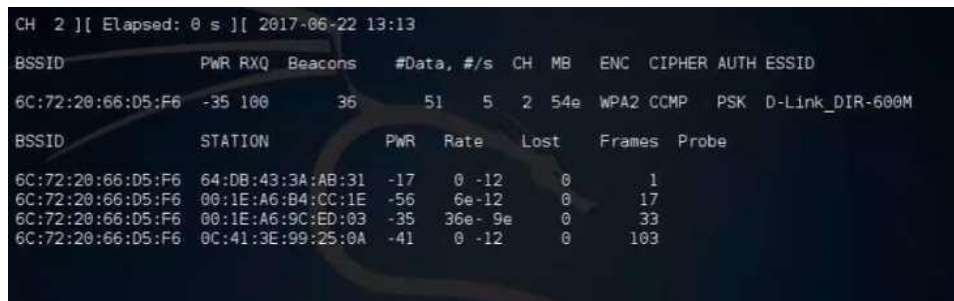
Just as in cracking with aircrack-ng, we need to put the wireless adapter into monitor mode.

- **airmon-ng start wlan0**
- **airodump-ng --bssid 00:25:9C:97:4F:48 -c 9 -w cowpatty mon0**

This will start a dump on the selected AP (**00:25:9C:97:4F:48**), on the selected channel (**-c 9**) and save the the hash in a file named **cowcrack**.

Capture the Handshake

Now when someone connects to the AP, we'll capture the hash and airdump-ng will show us it has been captured in the upper right-hand corner.



```
CH  2  ][ Elapsed: 0 s ][ 2017-06-22 13:13
```

BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
6C:72:20:66:D5:F6	-35	100	36	51 5	2	54e	WPA2	CCMP	PSK	D-Link_DIR-600M

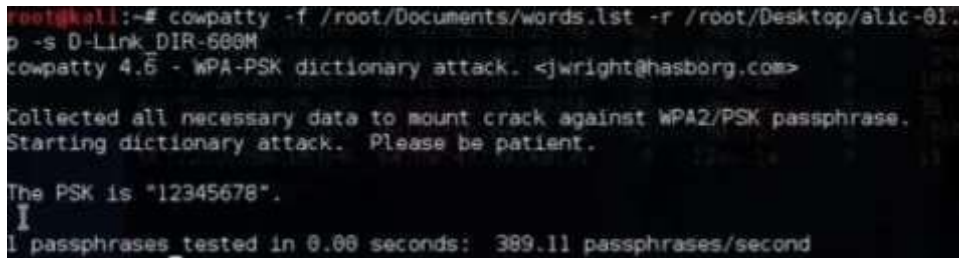
BSSID	STATION	PWR	Rate	Lost	Frames	Probe
6C:72:20:66:D5:F6	64:DB:43:3A:AB:31	-17	0 -12	0	1	
6C:72:20:66:D5:F6	00:1E:A6:B4:CC:1E	-56	6e-12	0	17	
6C:72:20:66:D5:F6	00:1E:A6:9C:ED:03	-35	36e- 9e	0	33	
6C:72:20:66:D5:F6	0C:41:3E:99:25:0A	-41	0 -12	0	103	

Run the cowpatty.

Now that we have the hash of the password, we can use it with cowpatty and our

wordlist to crack the hash.

- **cowpatty -f /pentest/passwords/wordlists/darkc0de.lst -r /root/cowcrack- 01.cap -s Mandela2**



```
root@kali:~# cowpatty -f /root/Documents/words.lst -r /root/Desktop/alic-01.cap -s D-Link DIR-600M
cowpatty 4.5 - WPA-PSK dictionary attack. <jwright@hasborg.com>
Collected all necessary data to mount crack against WPA2/PSK passphrase.
Starting dictionary attack. Please be patient.
The PSK is "12345678".
1 passphrases tested in 0.00 seconds: 389.11 passphrases/second
```

As you can see in the screenshot above, cowpatty is generating a hash of every word on our wordlist with the SSID as a seed and comparing it to the captured hash. When the hashes match, it displays the password of the A

## Make Your Own Hash

Although running cowpatty can be rather simple, it can also be very slow. The password hash is hashed with SHA1 with a seed of the SSID. This means that the same password on different SSIDs will generate different hashes. This prevents us from simply using a rainbow table against all APs. Cowpatty must take the password list you provide and compute the hash with the SSID for each word.

This is very CPU intensive and slow.

Cowpatty now supports using a pre-computed hash file rather than a plain-text word file, making the cracking of the WPA2-PSK password 1000x faster! Pre-computed hash files are available from the Church of WiFi, and these pre-computed hash files are generated using 172,000 dictionary file and the 1,000 most popular SSIDs. As useful as this is, if your SSID is not in that 1,000 Hash list really doesn't help us.

In that case, we need to generate our own hashes for our target SSID. We can do this by using an application called **genpmk**. We can generate our hash file for the "darkcode" wordlist for the SSID "Mandela2" by typing:

**genpmk -f /pentest/passwords/wordlists/darkc0de.lst -d hashes sS**

## **Mandela2**

### **Using Our Hash**

Once we have generated our hashes for the particular SSIDs, we can then crack the password with cowpatty by typing:

- **cowpatty -d hashfile -r dumpfile -s ssid**

**RESULT:** Thus, the CowPatty is implemented successfully.