CLOUD SECURITY MANAGEMENT LAB FILE

Kamlesh Pareek

Sap ID – 500091593

Roll Number – R2142210389

Program – B.Tech. CSE CCVT B1(H)

Instructor – Dr. Ambika Aggarwal

INDEX

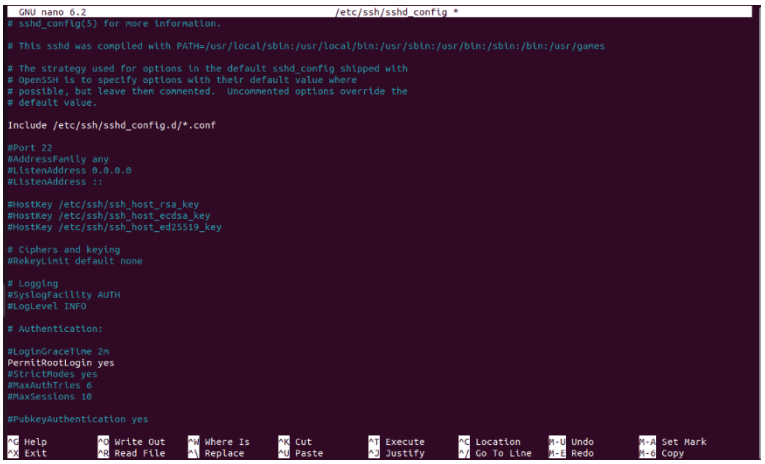
LAB – 1 SSH Key Based Authentication

1. Login to your guest OS and update it. Then install openssh-server(Web-Server) in your guest OS.
   sudo apt update
   sudo apt install -y openssh-server

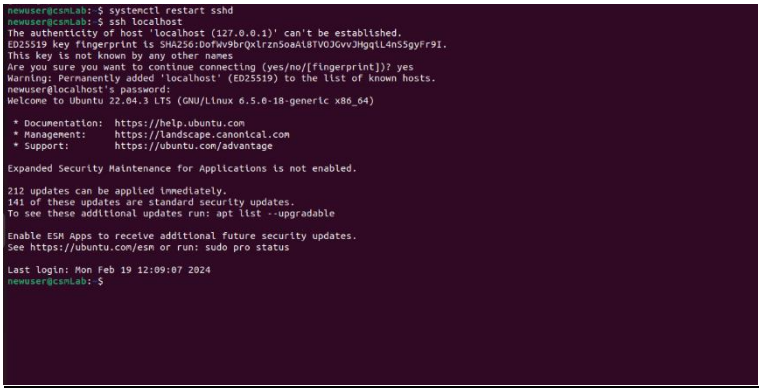
```
newuser@csmLab:~$ sudo apt update
Hit:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:4 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Fetched 229 kB in 3s (81.8 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
212 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

```
newuser@csmLab:~$ sudo apt install -y openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-client openssh-sftp-server ssh-import-id
Suggested packages:
  keychain libpam-ssh monkeysphere ssh-askpass molly-guard
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
The following packages will be upgraded:
  openssh-client
1 upgraded, 4 newly installed, 0 to remove and 211 not upgraded.
Need to get 752 kB/1,658 kB of archives.
After this operation, 6,050 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-sftp-server amd64 1:8.9p1-3ubuntu0.6 [38.7 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-server amd64 1:8.9p1-3ubuntu0.6 [435 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ncurses-term all 6.3-2ubuntu0.1 [267 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 ssh-import-id all 5.11-0ubuntu1 [10.1 kB]
Fetched 752 kB in 3s (225 kB/s)
Preconfiguring packages ...
(Reading database ... 205319 files and directories currently installed.)
Preparing to unpack .../openssh-client_1%3a8.9p1-3ubuntu0.6_amd64.deb ...
Unpacking openssh-client (1:8.9p1-3ubuntu0.6) over (1:8.9p1-3ubuntu0.3) ...
Selecting previously unselected package openssh-sftp-server.
Preparing to unpack .../openssh-sftp-server_1%3a8.9p1-3ubuntu0.6_amd64.deb ...
Unpacking openssh-sftp-server (1:8.9p1-3ubuntu0.6) ...
Selecting previously unselected package openssh-server.
Preparing to unpack .../openssh-server_1%3a8.9p1-3ubuntu0.6_amd64.deb ...
Unpacking openssh-server (1:8.9p1-3ubuntu0.6) ...
Selecting previously unselected package ncurses-term.
Preparing to unpack .../ncurses-term_6.3-2ubuntu0.1_all.deb ...
Unpacking ncurses-term (6.3-2ubuntu0.1) ...
Selecting previously unselected package ssh-import-id.
```

2. Open /etc/ssh/sshd_config file using nano editor and uncomment PermitRootLogin and replace ‘prohibit-password’ with ‘yes’. Save and exit.
   nano /etc/ssh/sshd_config



3. Restart the server and check if server is working or not.
   systemctl restart sshd
   ssh localhost

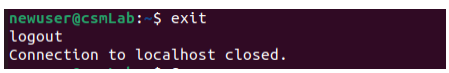


4. Generate ssh key for securely accessing the system shell.
   ssh-keygen

5. Now setup a passwordless SSH login for the user newuser on your local machine (localhost).
   ssh-copy-id newuser@localhost

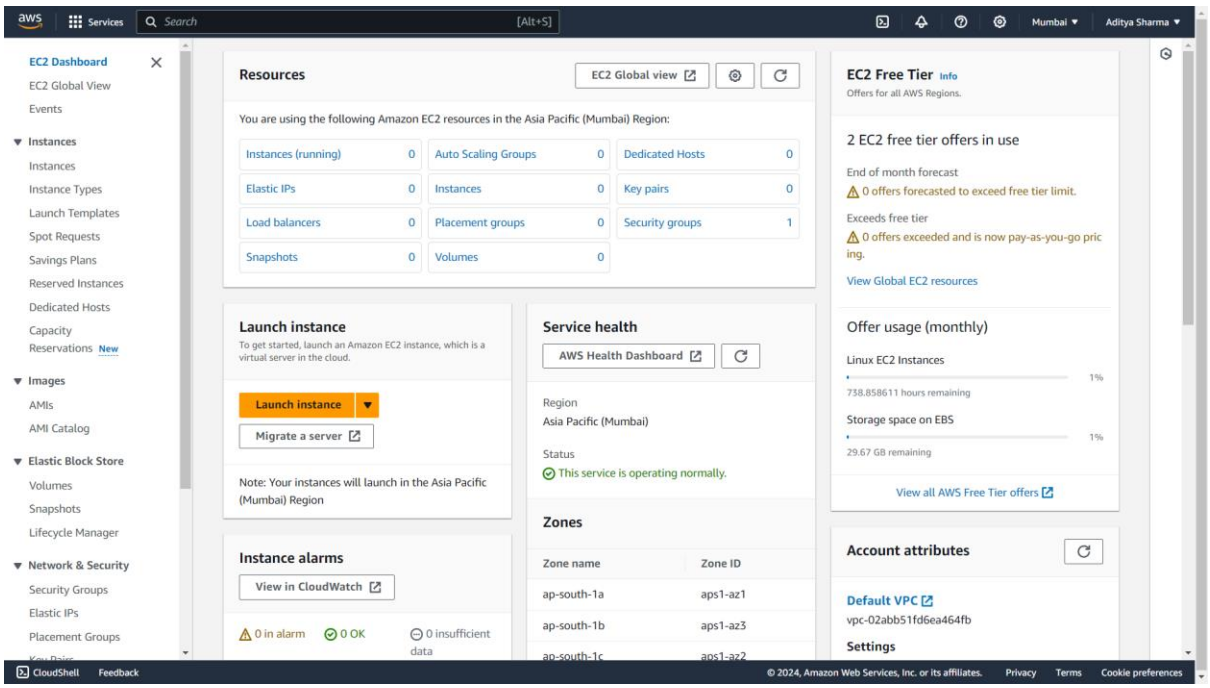
6. Now login to the newuser at localhost using ssh(secure shell)
   ssh newuser@localhost

7. exit

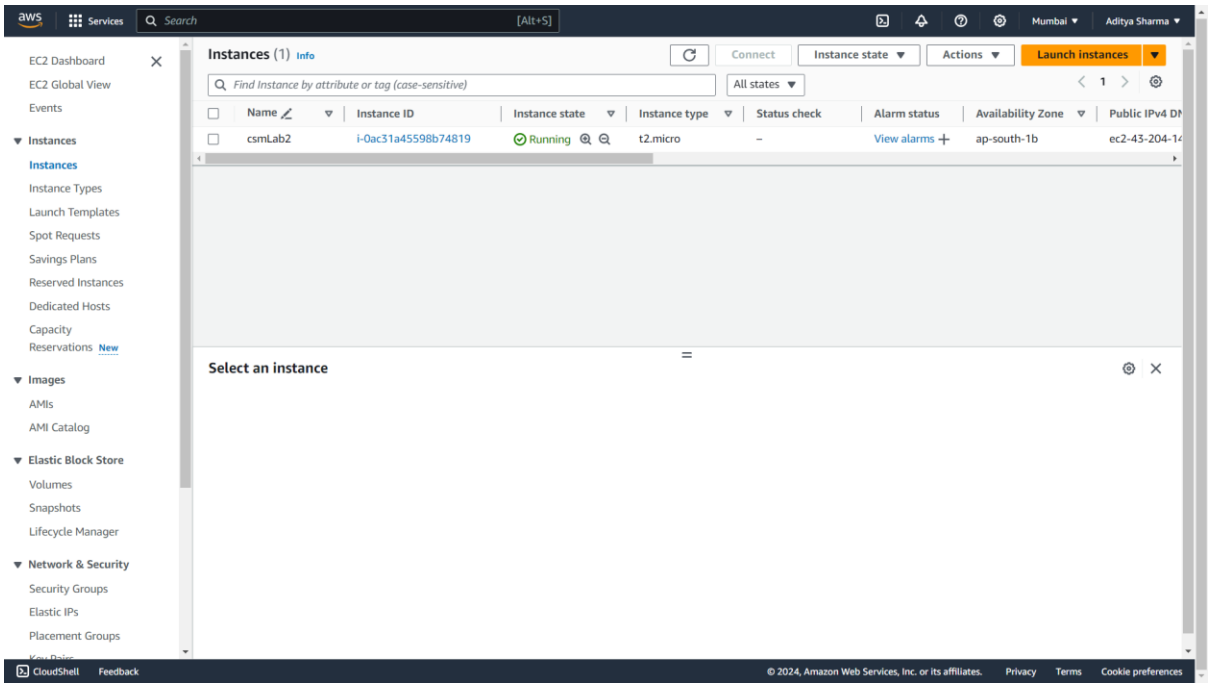


LAB – 2 Install a Webserver in VM

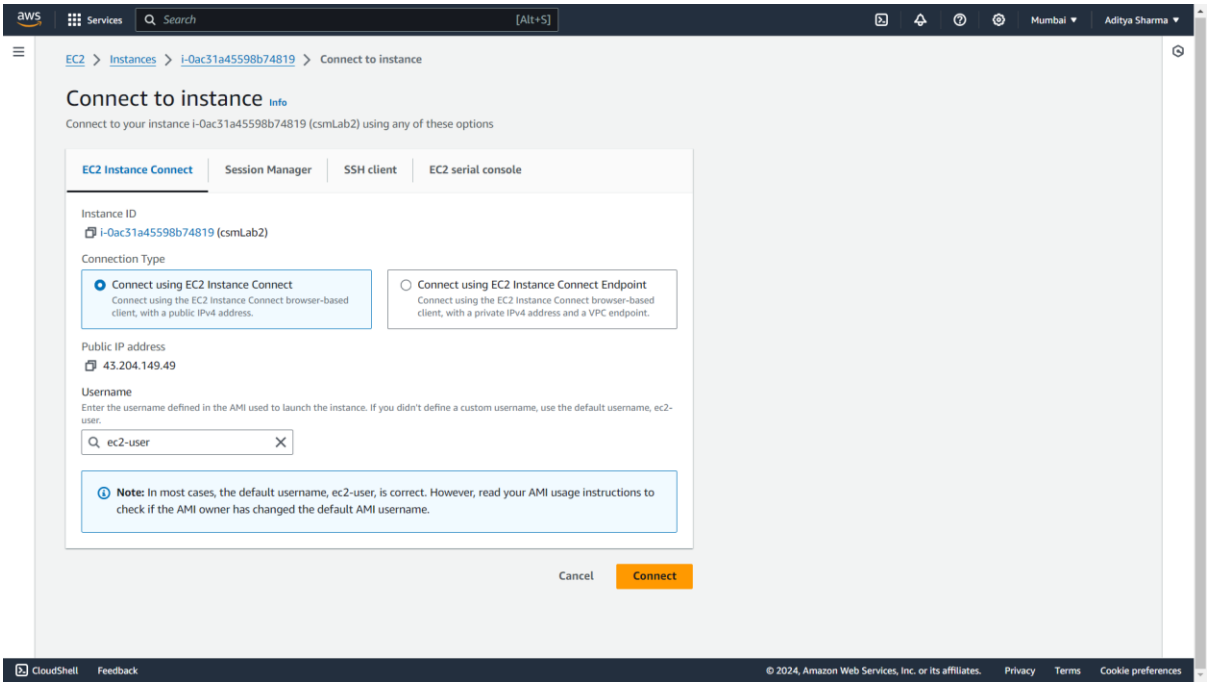
1. Login to your AWS management console. Go to EC2 service dashboard.



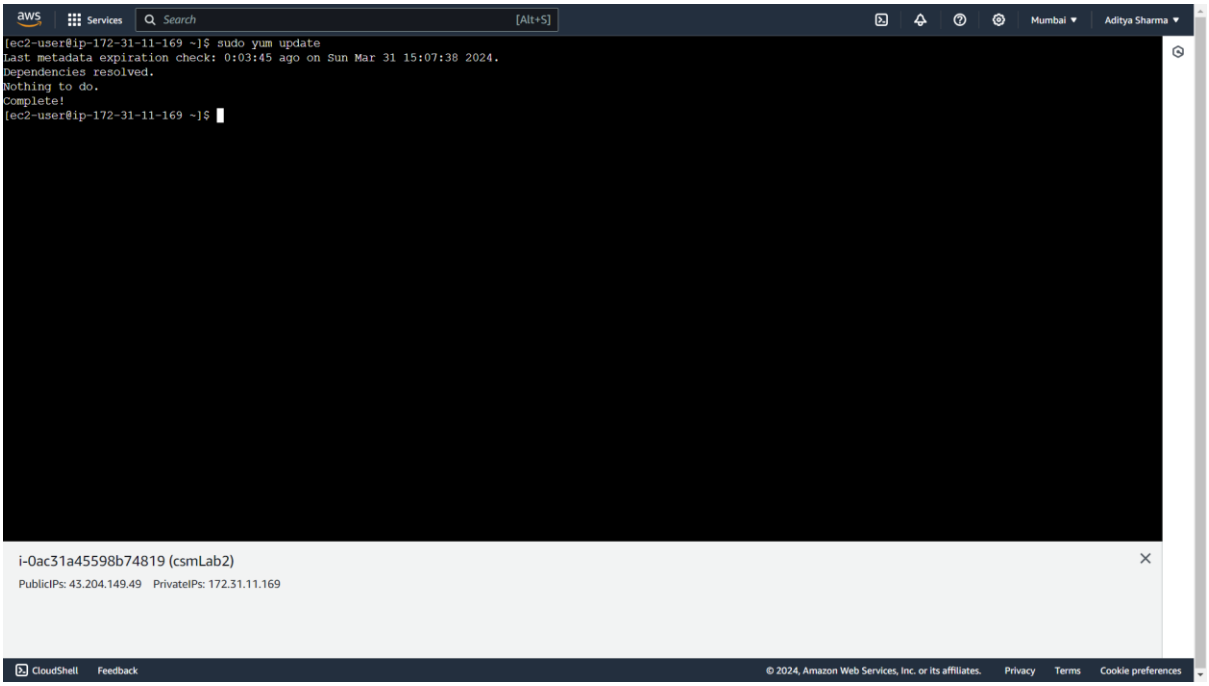
2. Create a new virtual machine (I am using AWS Linux image).



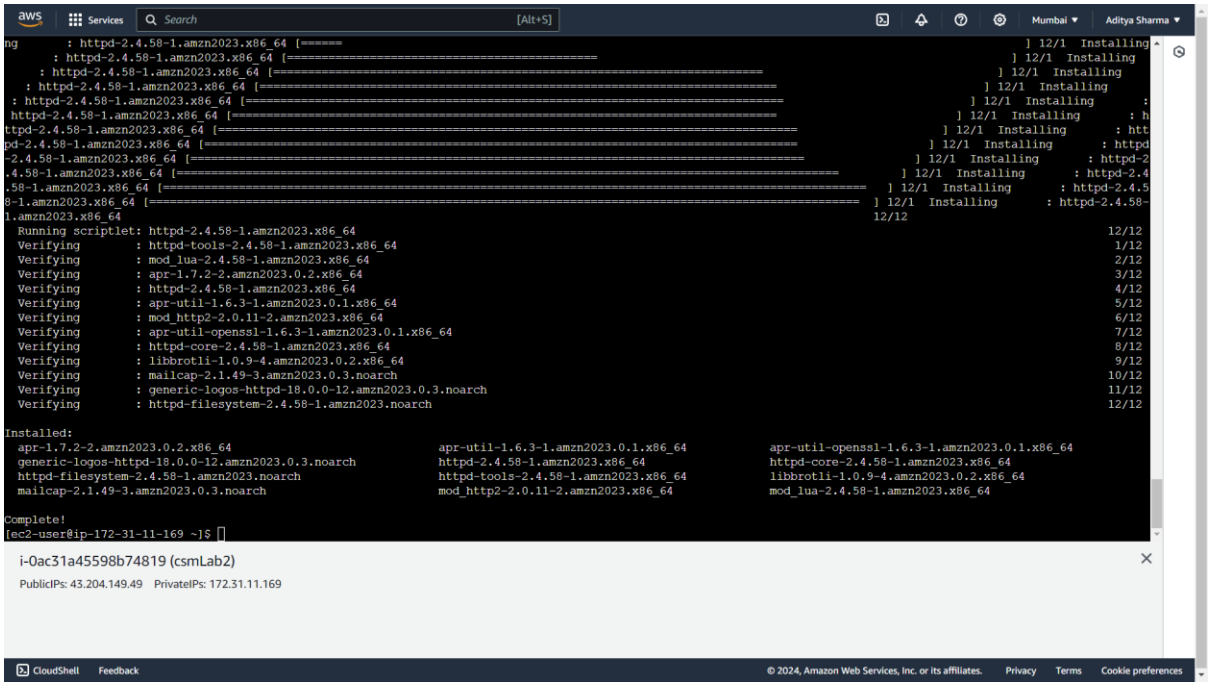
3. Connect to your VM.



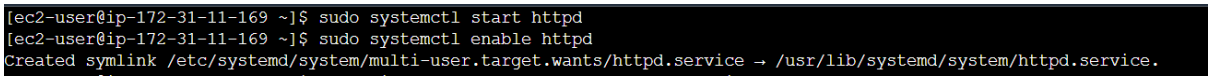
4. Update the instance using sudo yum update



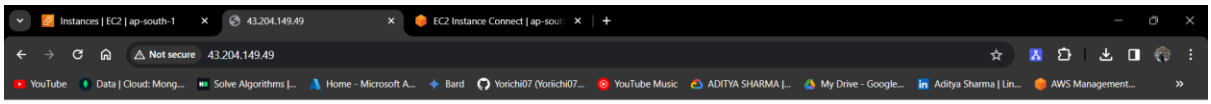
5. Install the Apache web server using sudo yum install httpd -y



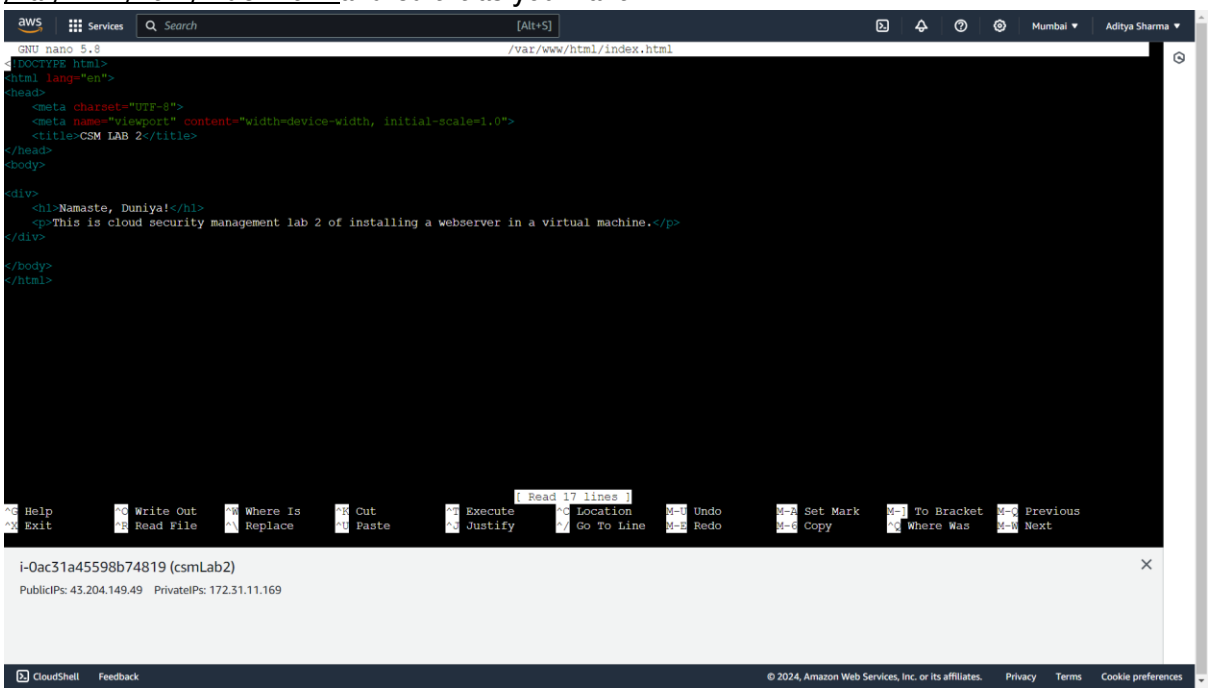
6. Start the apache webserver and configure it to start at system boot.



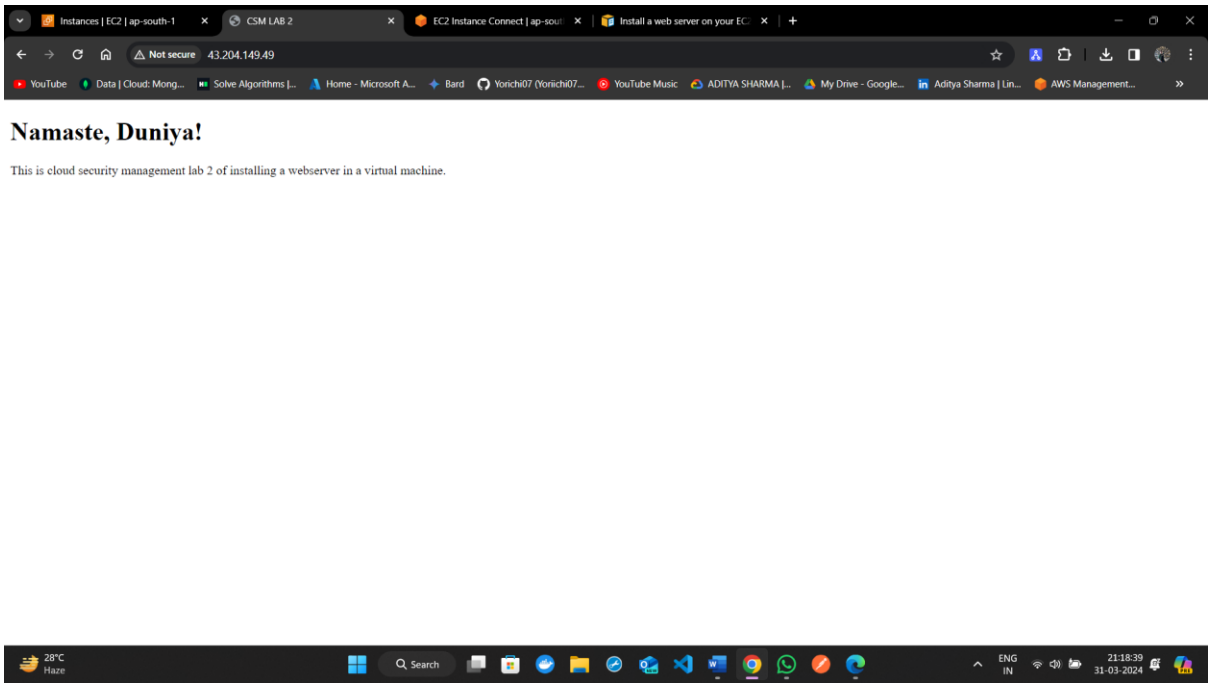
7. Connect to your webserver using public IP address of instance.



8. You can also edit the webpage. Open index.html using sudo nano /var/www/html/index.html and edit it as you want.



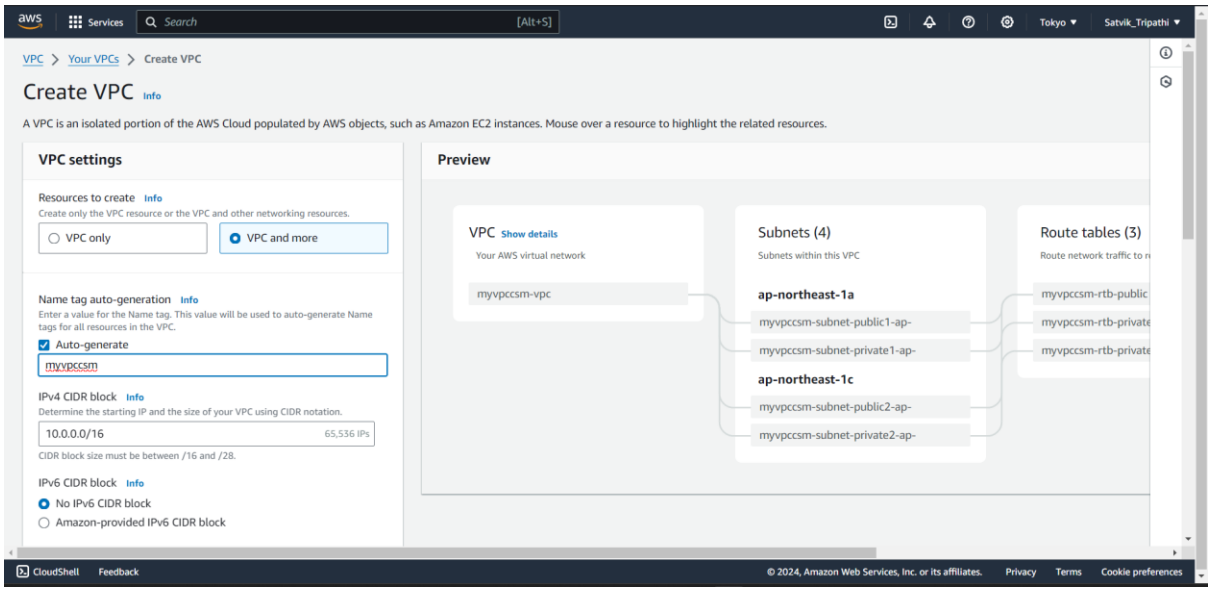
9. Verify the change in the website.



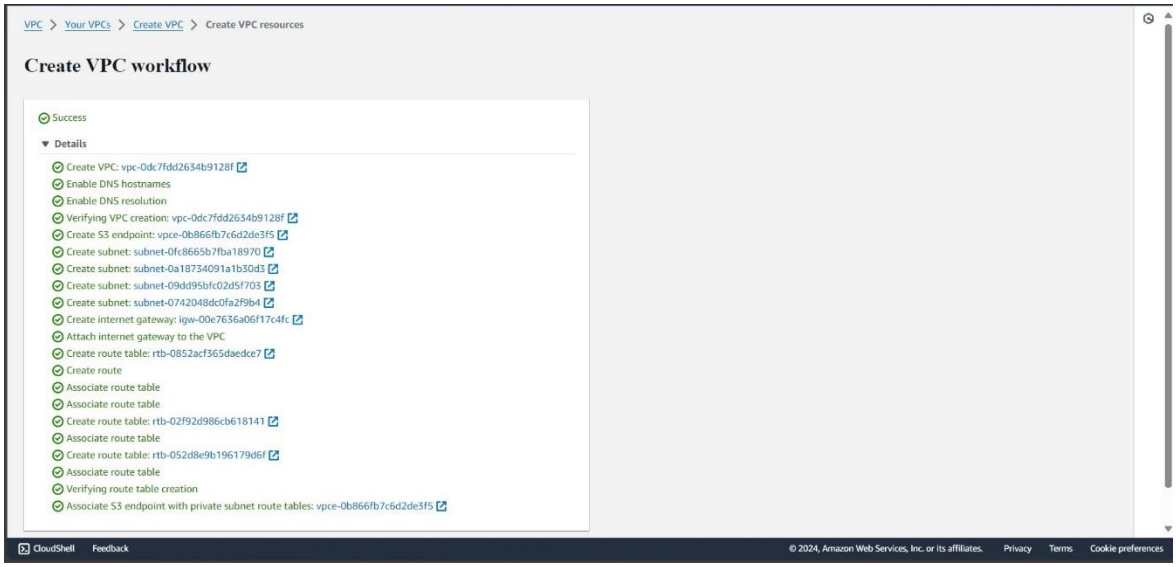
Experiment 3 - VPC

Aim: Create a VPC in AWS

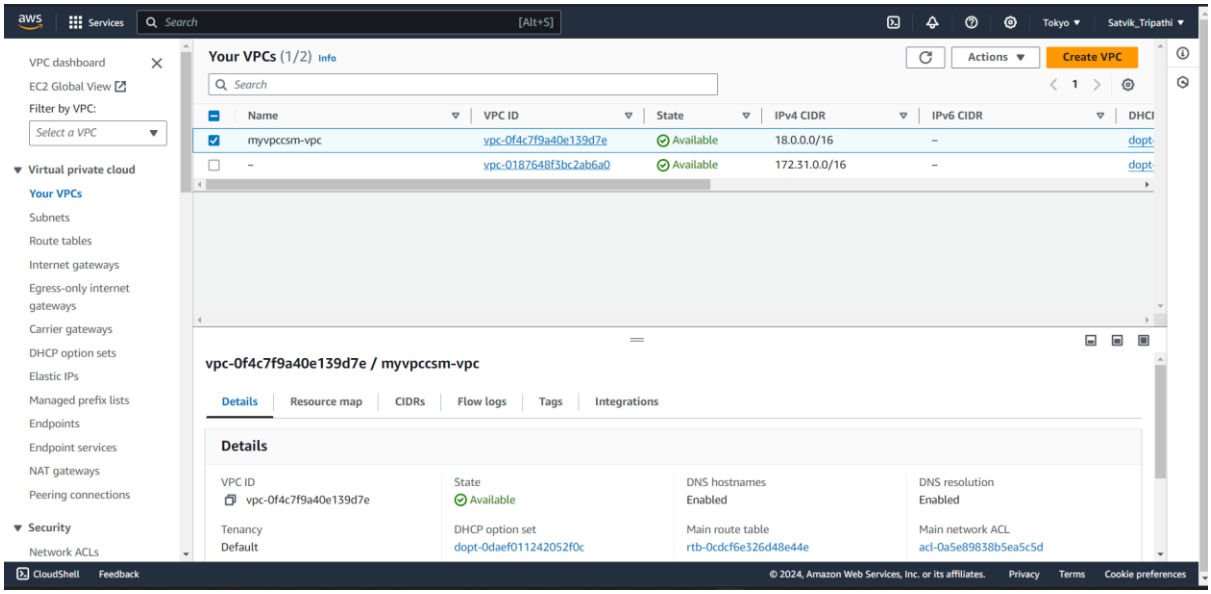
Step 1: Naming a new VPC that we are going to be using for this experiment.



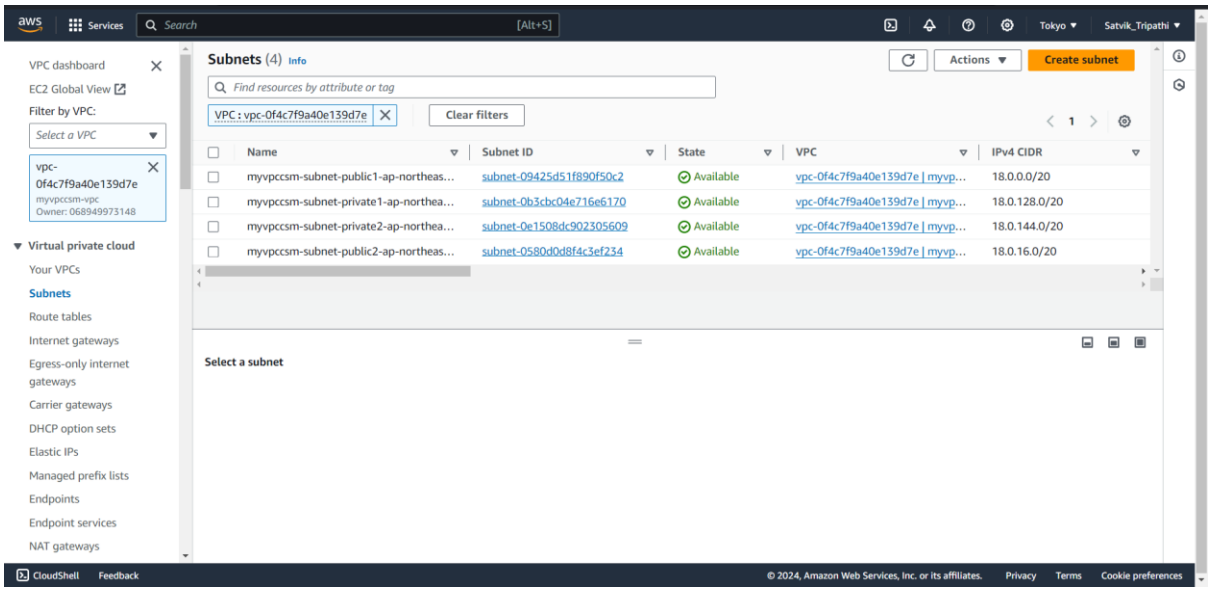
Step 2: VPC creation process initialized successfully



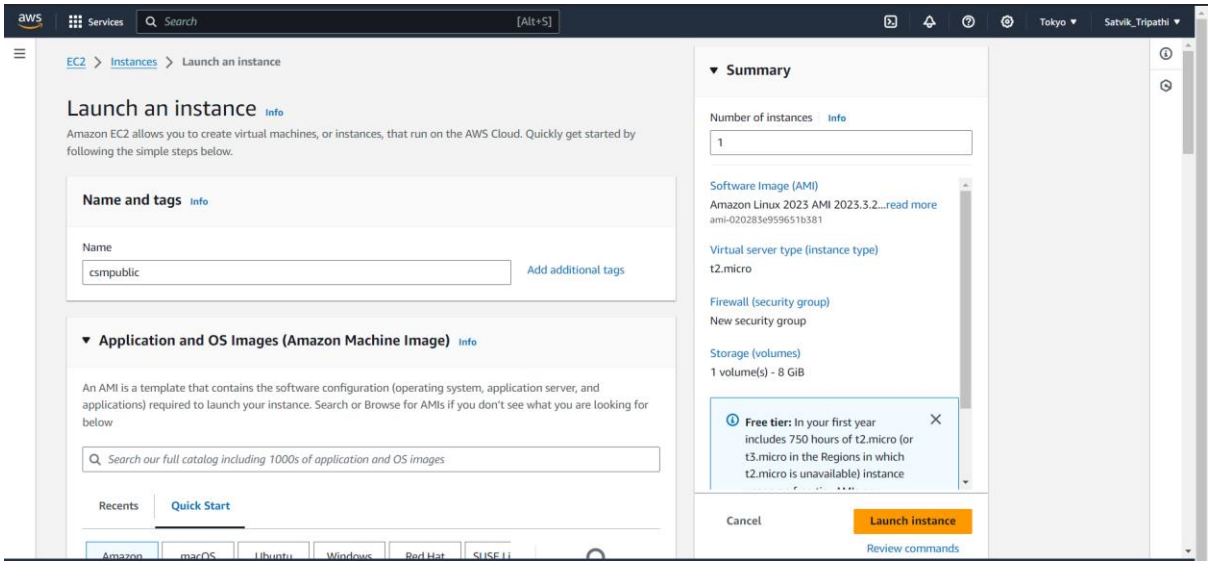
Step 3: Custom VPC created successfully and now we can use it.



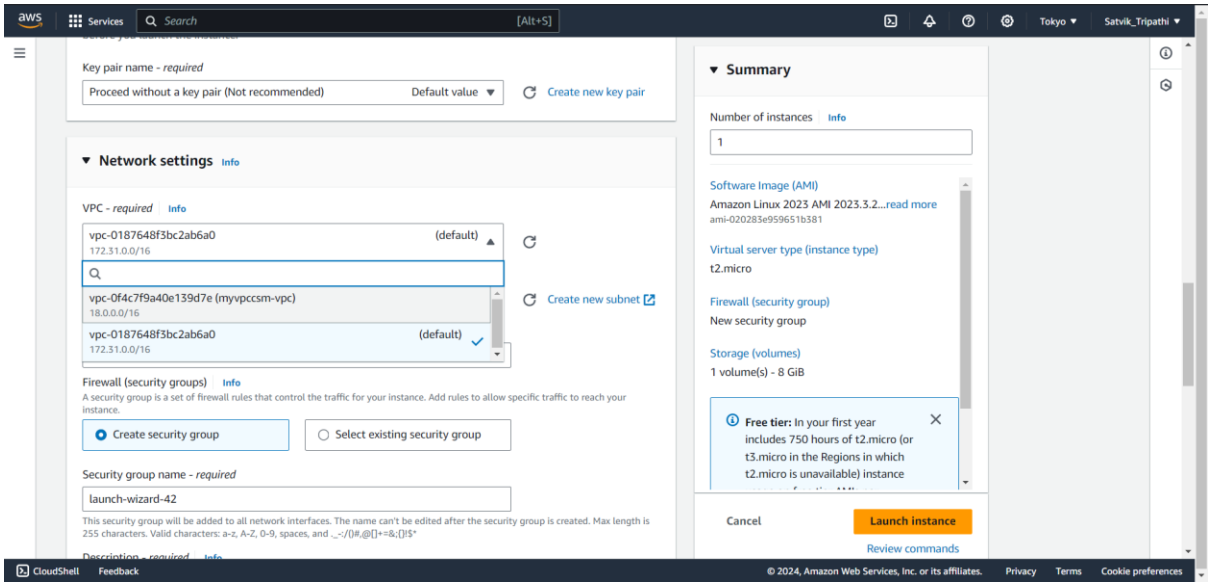
Step 4: Subnets of the created VPC are shown listed.

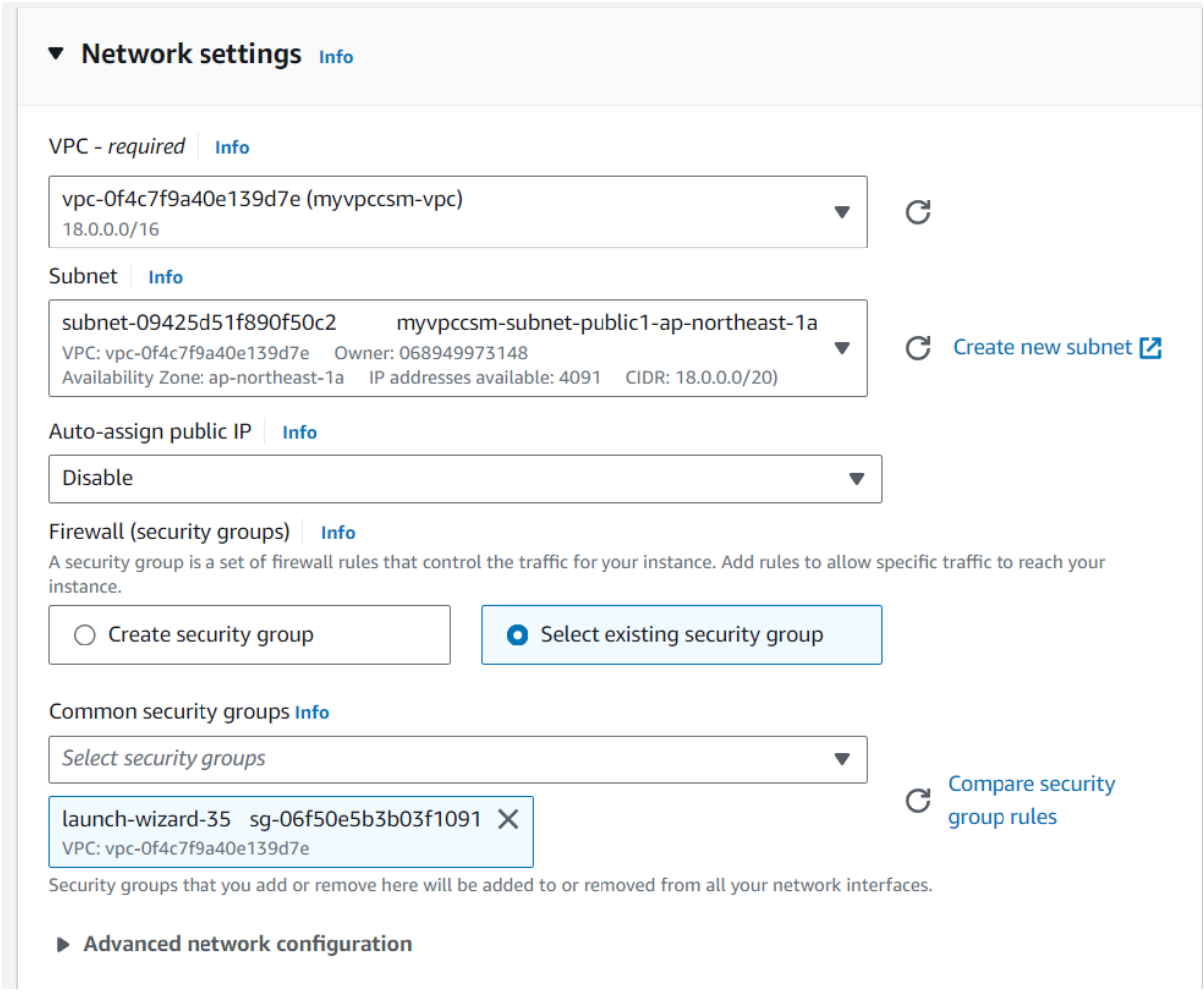


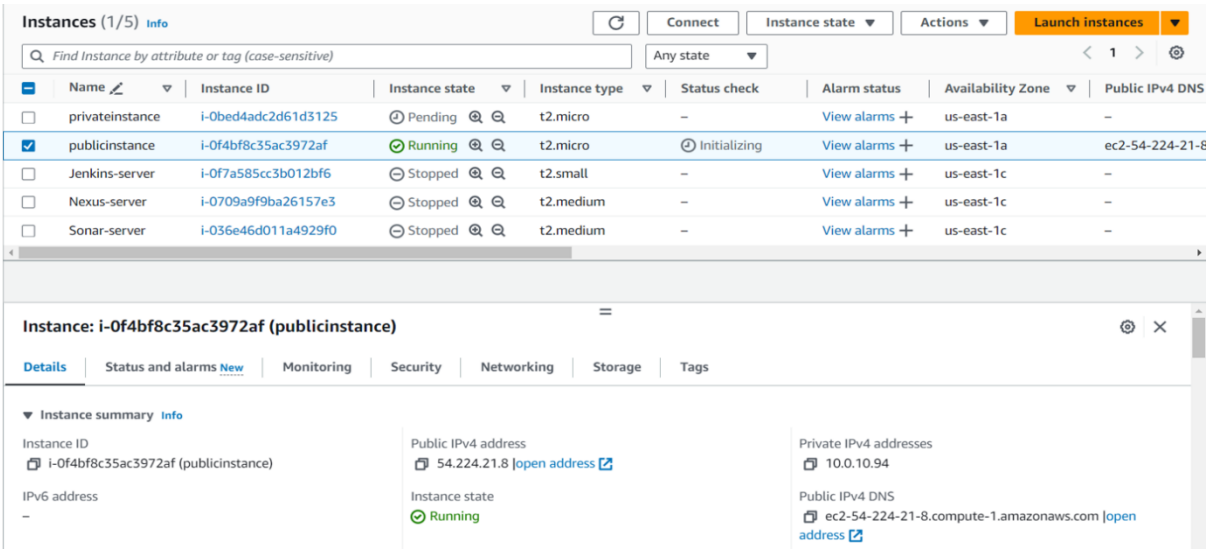
Step 5: Now we have to create a new instance that we will connect using public subnet

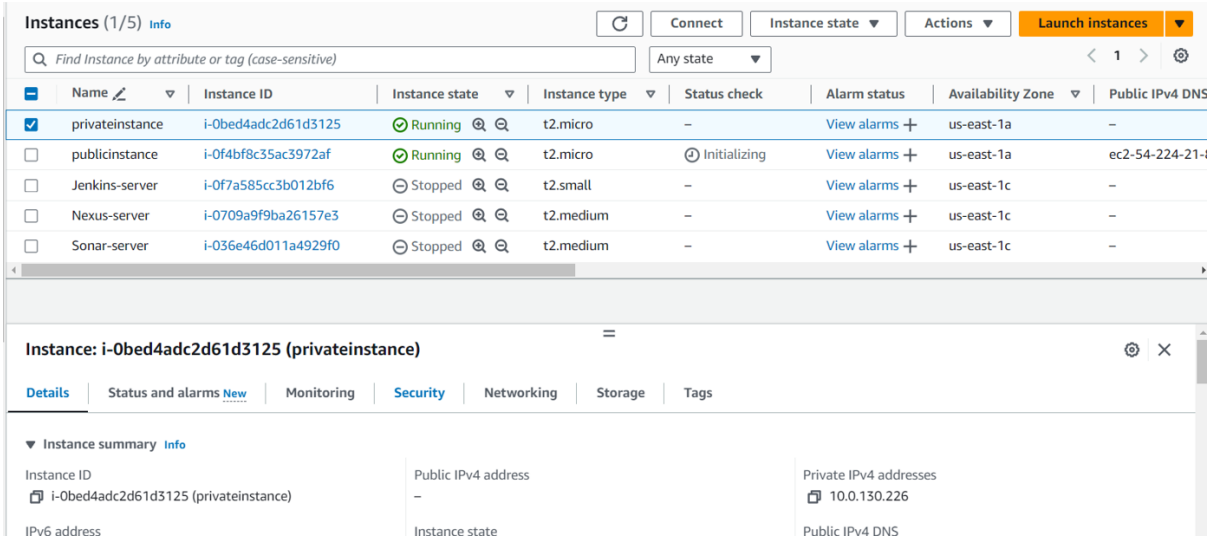


Step 6: Selecting key pair, the VPC we created and a public subnet before launching the instance.




Network settings Info
VPC - required Info
vpc-0f4c7f9a40e139d7e (myvpccsm-vpc)
18.0.0.0/16
Subnet Info
subnet-09425d51f890f50c2 myvpccsm-subnet-public1-ap-northeast-1a
VPC: vpc-0f4c7f9a40e139d7e Owner: 068949973148
Availability Zone: ap-northeast-1a IP addresses available: 4091 CIDR: 18.0.0.0/20)
Create new subnet
Auto-assign public IP Info
Disable
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
launch-wizard-35 sg-06f50e5b3b03f1091
VPC: vpc-0f4c7f9a40e139d7e
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


Instances (1/5) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
Any state
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
privateinstance i-0bed4adc2d61d3125 Pending t2.micro – View alarms us-east-1a –
publicinstance i-0f4bf8c35ac3972af Running t2.micro Initializing View alarms us-east-1a ec2-54-224-21-8
Jenkins-server i-0f7a585cc3b012bf6 Stopped t2.small – View alarms us-east-1c –
Nexus-server i-0709a9f9ba26157e3 Stopped t2.medium – View alarms us-east-1c –
Sonar-server i-036e46d011a4929f0 Stopped t2.medium – View alarms us-east-1c –
Instance: i-0f4bf8c35ac3972af (publicinstance)
Details
Status and alarms New
Monitoring
Security
Networking
Storage
Tags
Instance summary Info
Instance ID
i-0f4bf8c35ac3972af (publicinstance)
Public IPv4 address
54.224.21.8 |open address
Private IPv4 addresses
10.0.10.94
IPv6 address
–
Instance state
Running
Public IPv4 DNS
ec2-54-224-21-8.compute-1.amazonaws.com |open address



Step 7: Download your keys

Step 8: Open git bash and ssh into the public instance using ssh command

```
$ ssh -i "public-instance.pem" ubuntu@ec2-54-224-21-8.compute-1.amazonaws.com
The authenticity of host 'ec2-54-224-21-8.compute-1.amazonaws.com (54.224.21.8)' can't be established.
ED25519 key fingerprint is SHA256:jg2J9IsyAdi+61v1ZTTFVFgD7UPgB5GNU3fZeWwktwM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-224-21-8.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Feb 23 12:14:53 UTC 2024

  System load:  0.09033203125     Processes:             104
  Usage of /:   20.6% of 7.57GB   Users logged in:       0
  Memory usage: 21%               IPv4 address for eth0: 10.0.10.94
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

ubuntu@ip-10-0-10-94:~$
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
```

Step 9: Now we also try to ssh into the private instance but in this case we will fail because our private instance is in a private subnet which does not have Internet access because of which to connect to the private instance we must first connect to the public instance i.e.. we set up the Bastion host

```
$ ssh -i "private-instance.pem" ubuntu@10.0.130.226
ssh: connect to host 10.0.130.226 port 22: Connection timed out
```

Step 10: Run the ssh-agent using the command

eval $(ssh-agent)

Also add the key to ssh agent using the command

ssh-add <key-path>

Step 11: Now we go forward and edit the config file of ssh on our host system

nano ~/.ssh/config

Add the following file

Host bastion

HostName bastion-host-public-ip

User ec2-user

IdentityFile /path/to/key1.pem

ForwardAgent yes

Host private-instance

HostName private-instance-private-ip

User ec2-user

IdentityFile /path/to/key2.pem

ProxyJump bastion

Step 12: Now ssh to the bastion host first and then logout

ssh bastion

```
$ ssh bastion
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Feb 23 12:33:01 UTC 2024

  System load:  0.0               Processes:             99
  Usage of /:   20.8% of 7.57GB   Users logged in:       0
  Memory usage: 21%               IPv4 address for eth0: 10.0.10.94
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Fri Feb 23 12:29:03 2024 from 49.43.160.87
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-10-94:~$
```

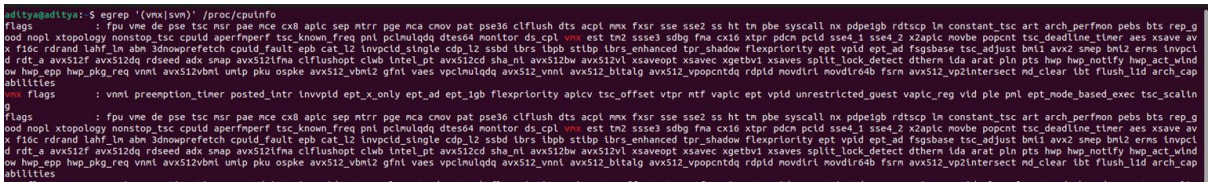
S-12) We now know that the public instance is accessible so now we SSH into the private instance

ssh private-instance

```
ubuntu@ip-10-0-130-226:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc fq_codel state UP group default qlen 1000
    link/ether 0e:24:c9:77:9d:d7 brd ff:ff:ff:ff:ff:ff
    inet 10.0.130.226/20 metric 100 brd 10.0.143.255 scope global dynamic eth0
       valid_lft 2320sec preferred_lft 2320sec
    inet6 fe80::c24:c9ff:fe77:9dd7/64 scope link
       valid_lft forever preferred_lft forever
ubuntu@ip-10-0-130-226:~$
```

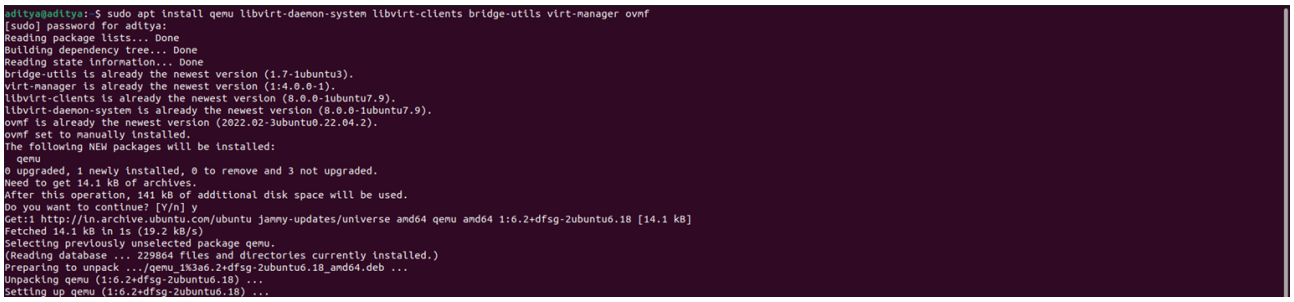
Experiment 4 – Installation and Configuration of Virtualisation using KVM

Step 1: Check if your hardware supports hardware virtualization. Use lscpu in your terminal if vmx gets colored then virtualization is supported.



Step 2: Update and upgrade your system using sudo apt-get update && sudo apt-get upgrade -y

Step 3: Now install necessary packages (virt-manager, qemu libvirt-daemon-system, libvirt clients, bridge-utils, ovmf) using sudo apt install qemu libvirt-daemon-system libvirt-clients bridge-utils virt-manager ovmf .After running its recommended to reboot your system.



Step 4: Now check the status of libvertd using sudo systemctl status libvertd. “libvirtd” is the daemon for the libvirt virtualization management library. It is responsible for managing virtual machines (VMs) and other virtualization tasks on the host system.

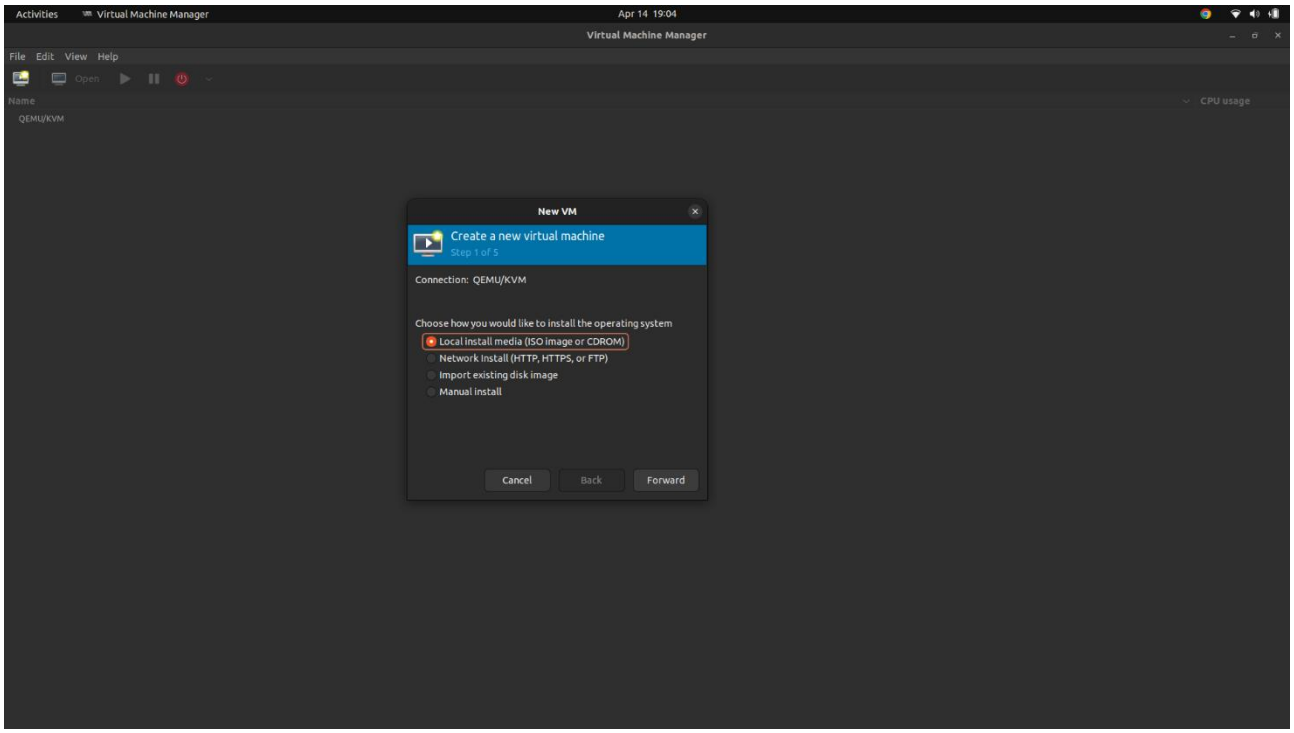
```
aditya@aditya:~$ sudo systemctl status libvirtd
● libvirtd.service - Virtualization daemon
     Loaded: loaded (/lib/systemd/system/libvirtd.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2024-04-14 18:31:20 IST; 7min ago
TriggeredBy: ● libvirtd-ro.socket
             ● libvirtd.socket
             ● libvirtd-admin.socket
       Docs: man:libvirtd(8)
             https://libvirt.org
   Main PID: 6909 (libvirtd)
      Tasks: 21 (limit: 32768)
     Memory: 19.4M
        CPU: 589ms
     CGroup: /system.slice/libvirtd.service
             ├─6909 /usr/sbin/libvirtd
             ├─8053 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefile-ro --dhcp-script=/usr/lib/libvirt/libvirt_leaseshelper
             └─8054 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefile-ro --dhcp-script=/usr/lib/libvirt/libvirt_leaseshelper

Apr 14 18:31:20 aditya systemd[1]: Started Virtualization daemon.
Apr 14 18:31:21 aditya dnsmasq[8053]: started, version 2.90 cachesize 150
Apr 14 18:31:21 aditya dnsmasq[8053]: compile time options: IPv6 GNU-getopt DBus no-UBus i18n IDN2 DHCP DHCPv6 no-Lua TFTP conntrack ipset no-nftset auth cryptohash DNSSEC loop-detect inotify dumpfile
Apr 14 18:31:21 aditya dnsmasq-dhcp[8053]: DHCP, IP range 192.168.122.2 -- 192.168.122.254, lease time 1h
Apr 14 18:31:21 aditya dnsmasq-dhcp[8053]: DHCP, sockets bound exclusively to interface virbr0
Apr 14 18:31:21 aditya dnsmasq[8053]: reading /etc/resolv.conf
Apr 14 18:31:21 aditya dnsmasq[8053]: using nameserver 127.0.0.53#53
Apr 14 18:31:21 aditya dnsmasq[8053]: read /etc/hosts - 9 names
Apr 14 18:31:21 aditya dnsmasq[8053]: read /var/lib/libvirt/dnsmasq/default.addnhosts - 0 names
Apr 14 18:31:21 aditya dnsmasq-dhcp[8053]: read /var/lib/libvirt/dnsmasq/default.hostsfile
```

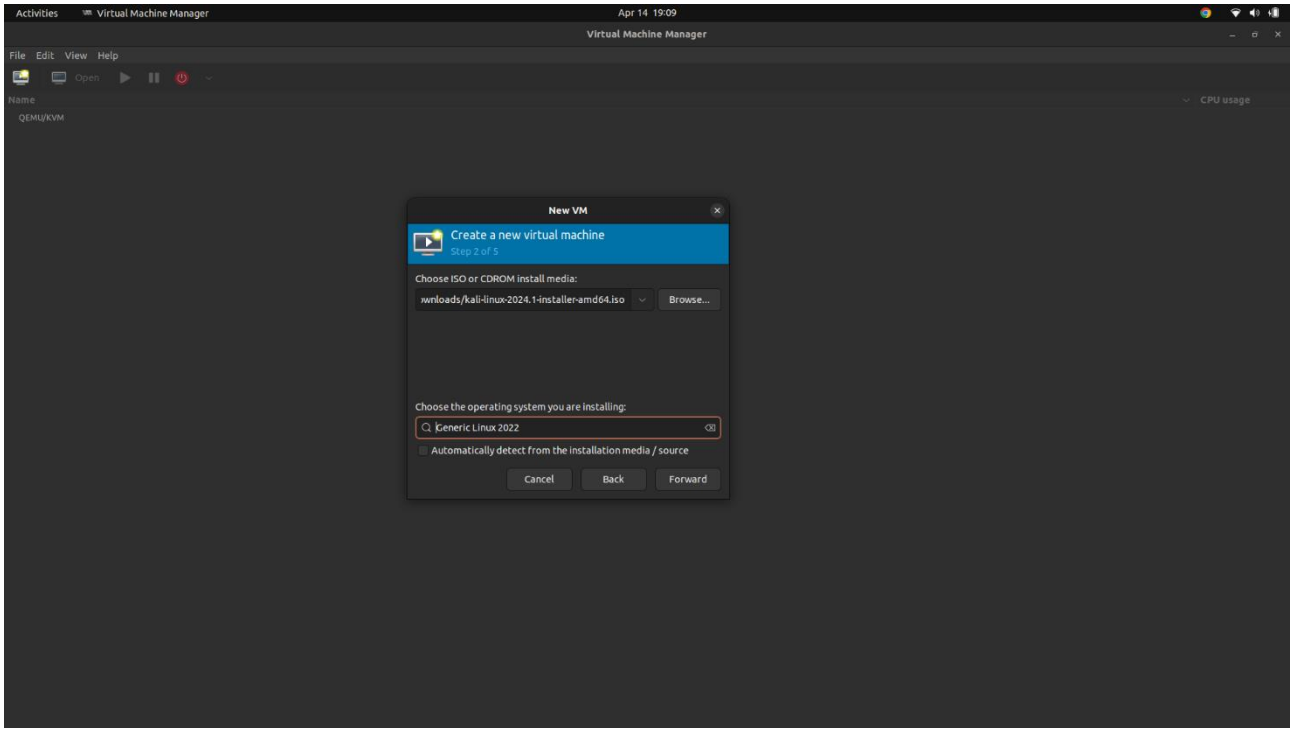
Step 5: Run group command and notice a new group “libvert” is added. Add current user to libvert group to allow them to handle virtual machines using sudo useradd g $USER libvirt and sudo useradd g $USER libvirt-kvm

```
aditya@aditya:~$ sudo useradd -g $USER libvirt
sudo useradd -g $USER libvirt-KVM
aditya@aditya:~$
```

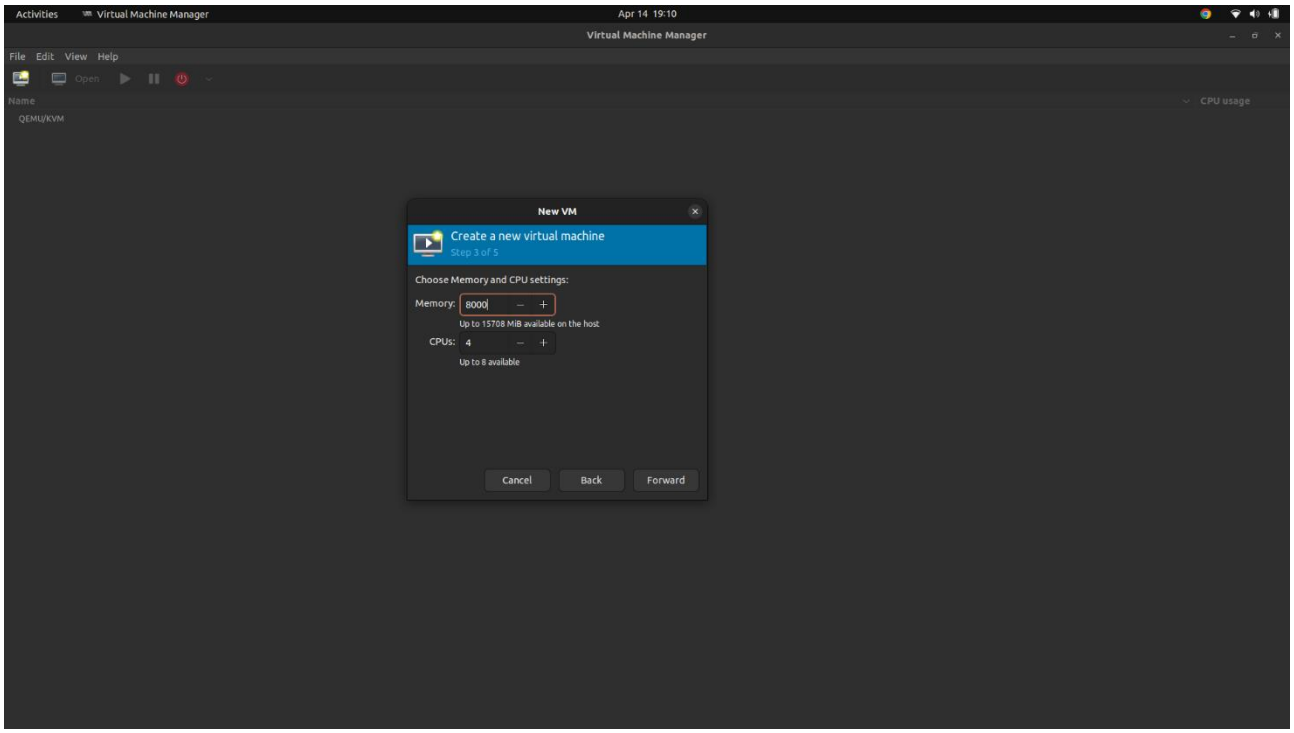
Step 6: Now open Virtual Machine Manager app which is a GUI interface that allows use to run and manage virtual machines. Click on file and select New Virtual Machine. Then select Local install media.



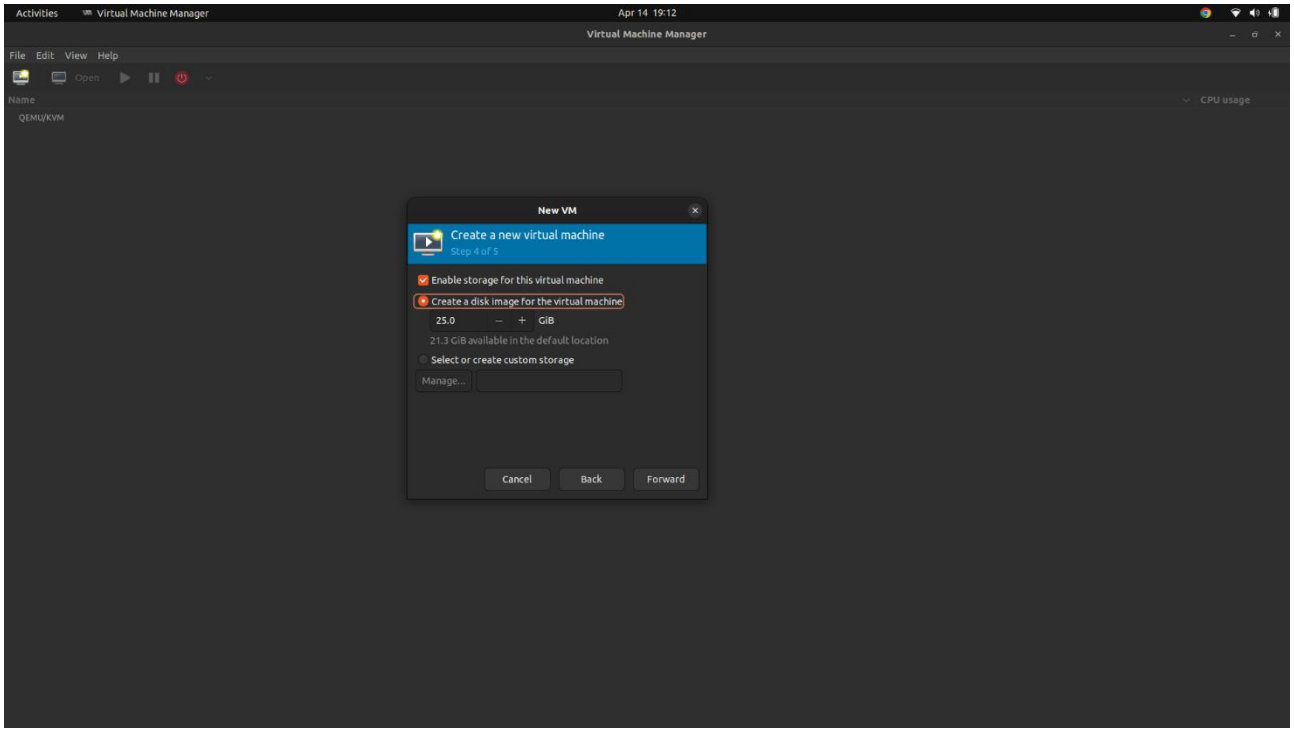
Step 7: Browse your iso file and choose OS. I have downloaded iso file of kali linux.



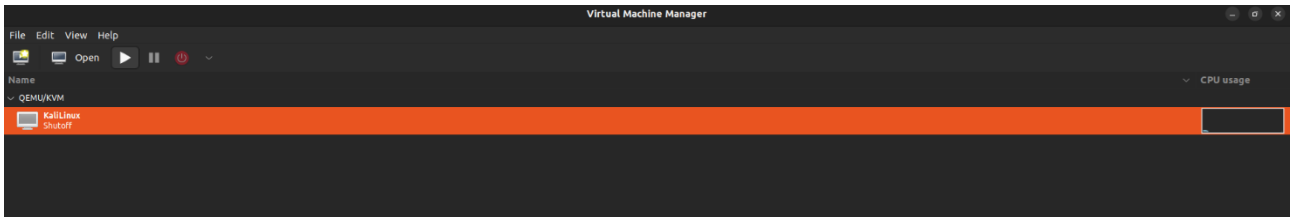
Step 8: Assign memory and cores to the virtual machine.



Step 9: Then allocate disk space to the virtual machine and click forward and then finish.



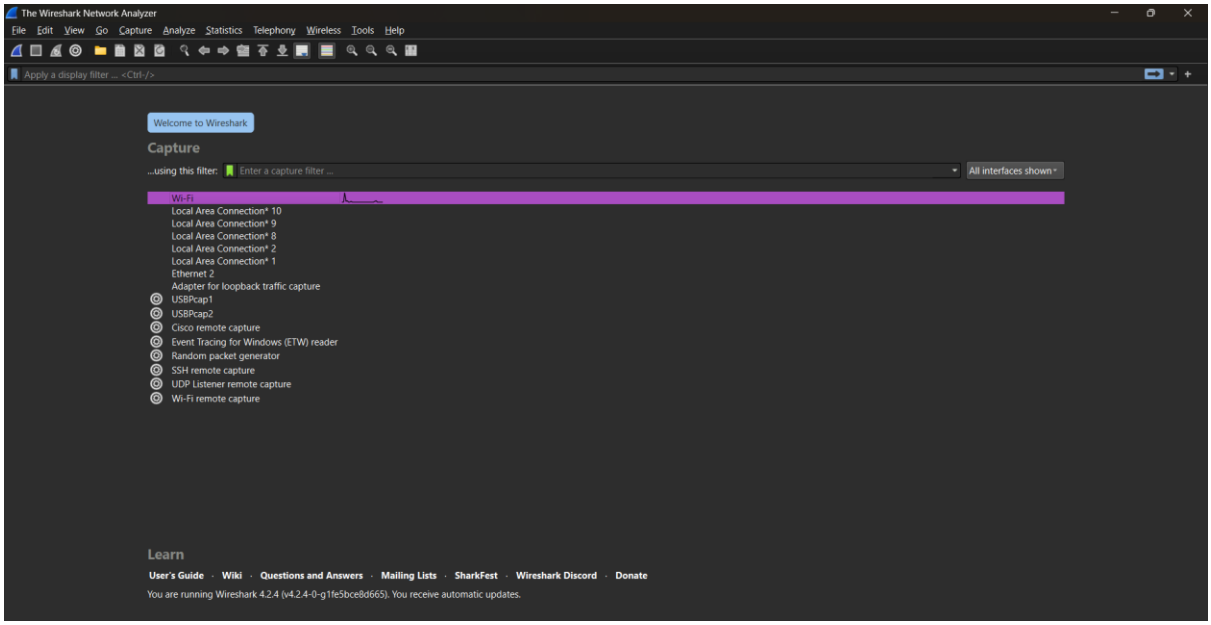
Step 10: Complete the installation of the virtual machine and start it. You can also run virtual machine using virsh start KaliLinux




```
aditya@aditya:~$ virsh start KaliLinux
Domain 'KaliLinux' started
```

Experiment 5 – Pentesting-Wireshark Tool

Wireshark is an essential tool for pentesting thick clients and most things in a Windows environment.

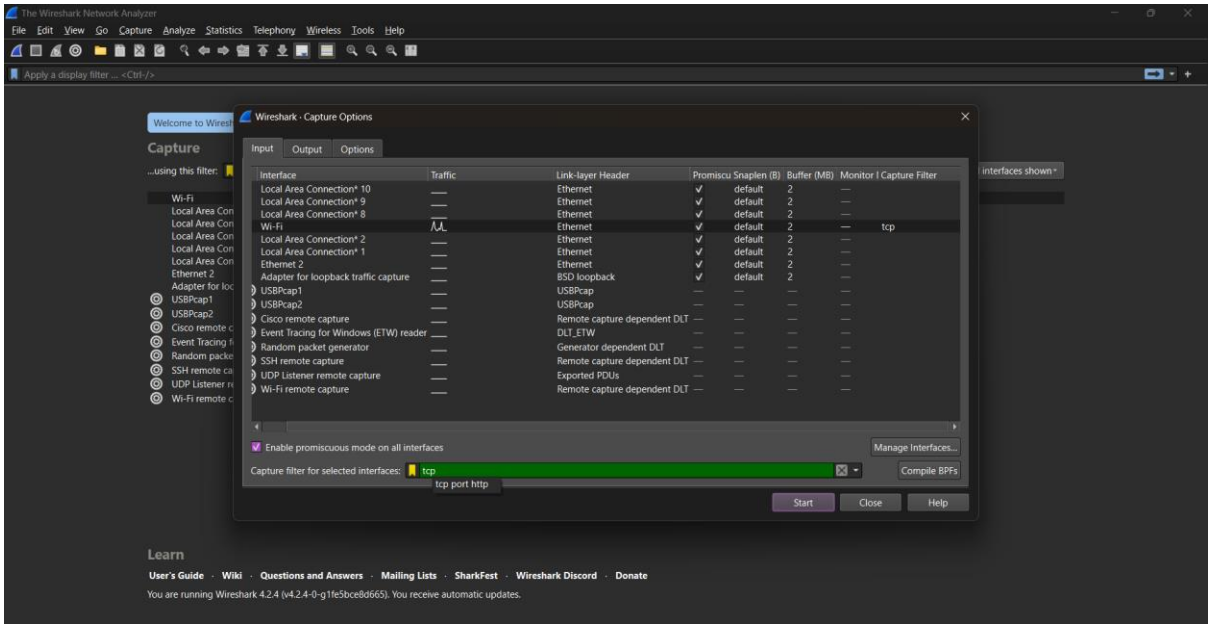


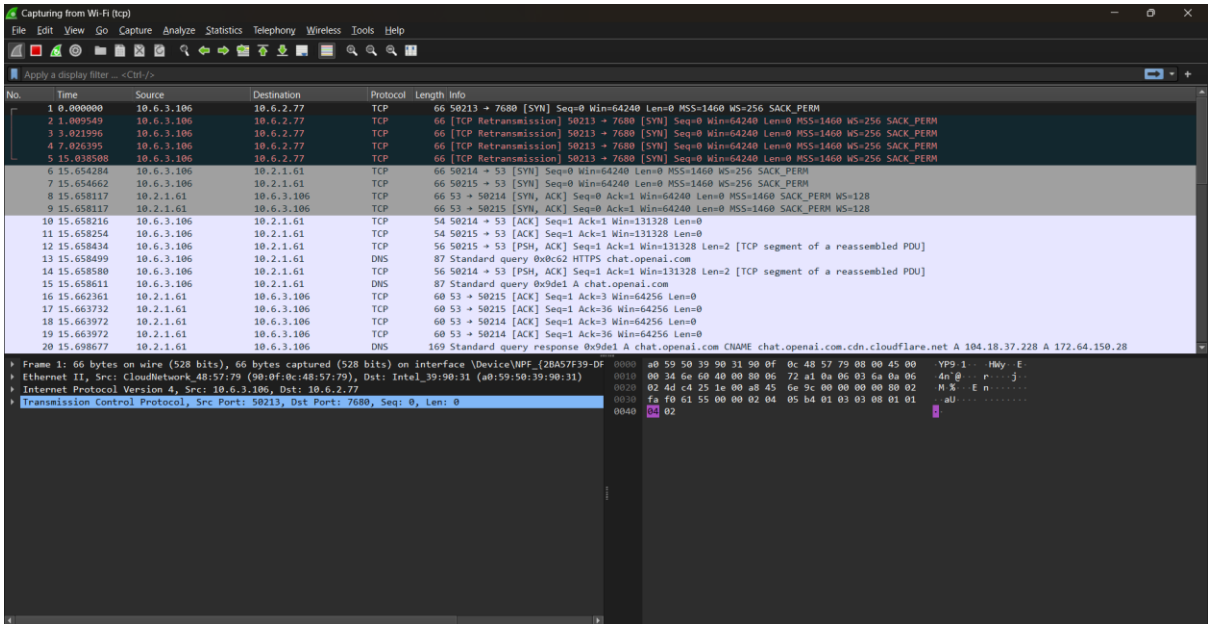
Capture vs Display Filters

Capture filters – completely ignore traffic set by the filter. Display filters – filters existing captured traffic, opening the filter in a new window.

- **Capture Filters:** Determine which packets are captured during a session. Applied before packets are stored. Use BPF (Berkeley Packet Filter) syntax. Help reduce captured data by filtering out unwanted packets. Limited in protocol-specific filtering.

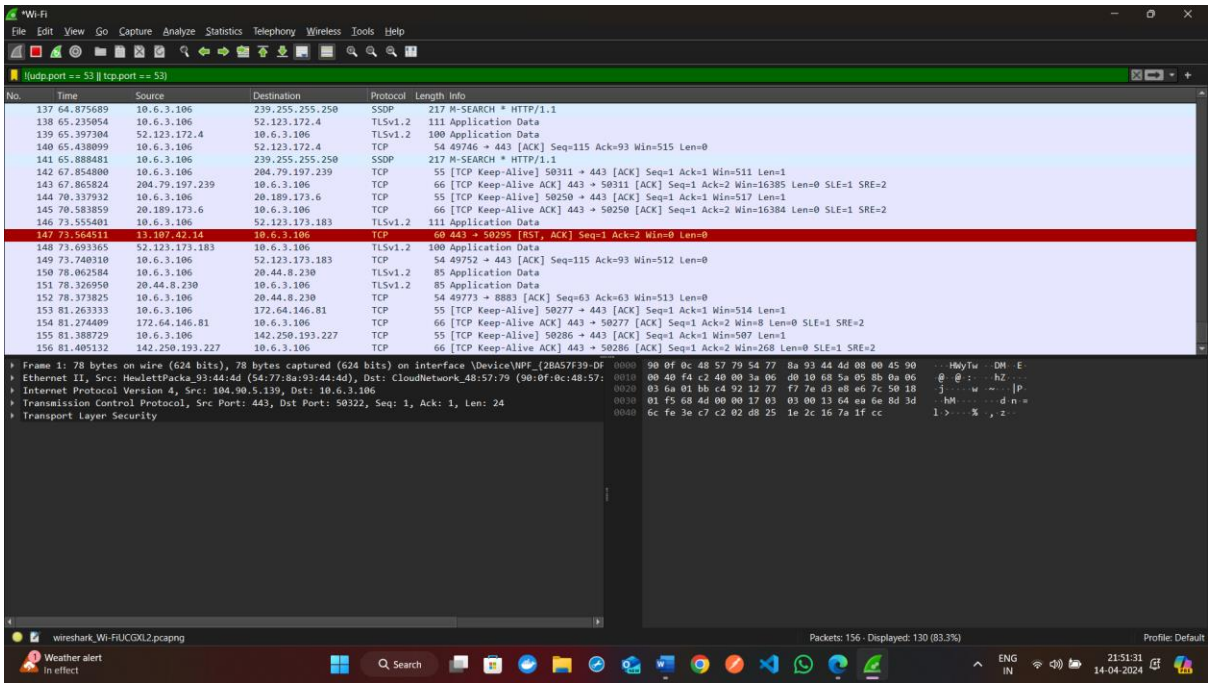
  I have applied TCP(Transmission Control Protocol) capture filtering below.





- **Display Filters:** Filter and analyse captured packets after they've been stored. Applied during analysis. Use Wireshark's filter toolbar or menu options. Offer more flexibility, including protocol-specific filtering. Allow for detailed analysis of captured traffic.

  Applied non-DNS port display filter

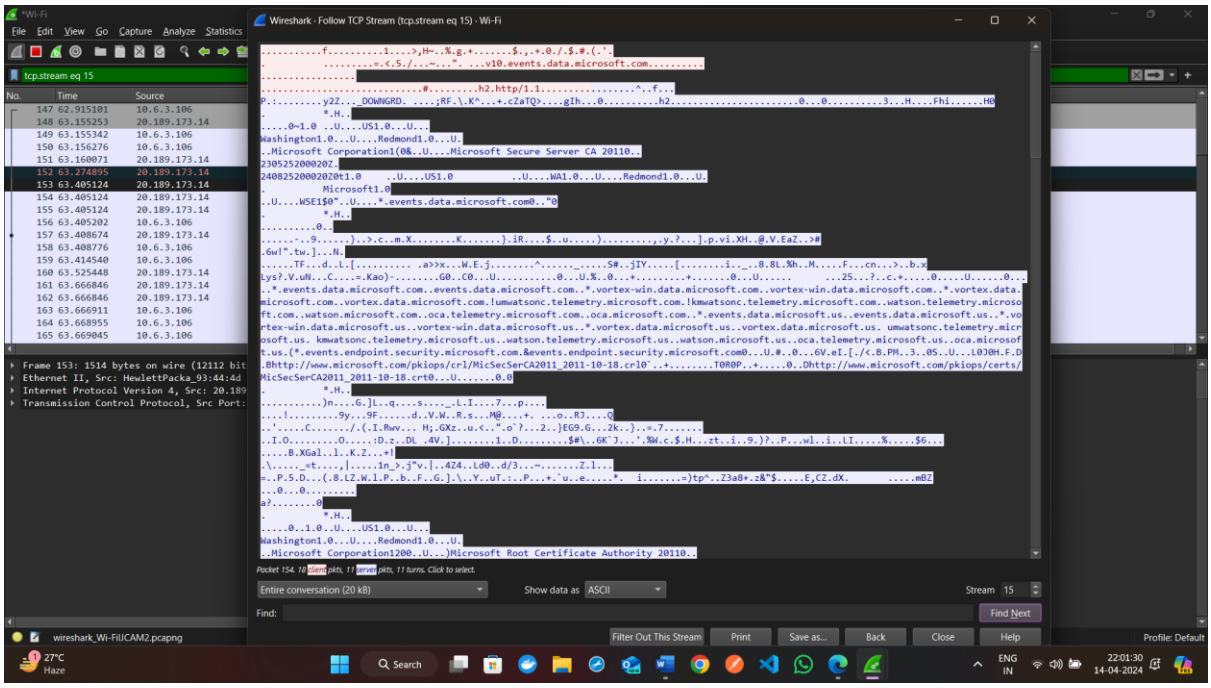


FOLLOW Traffic streams

The traffic you're interested in will often be spread out over a number of inbound and outbound packets. This can be frustrating when trying to view sensitive HTTP request/response pairs and most application-level data in general.

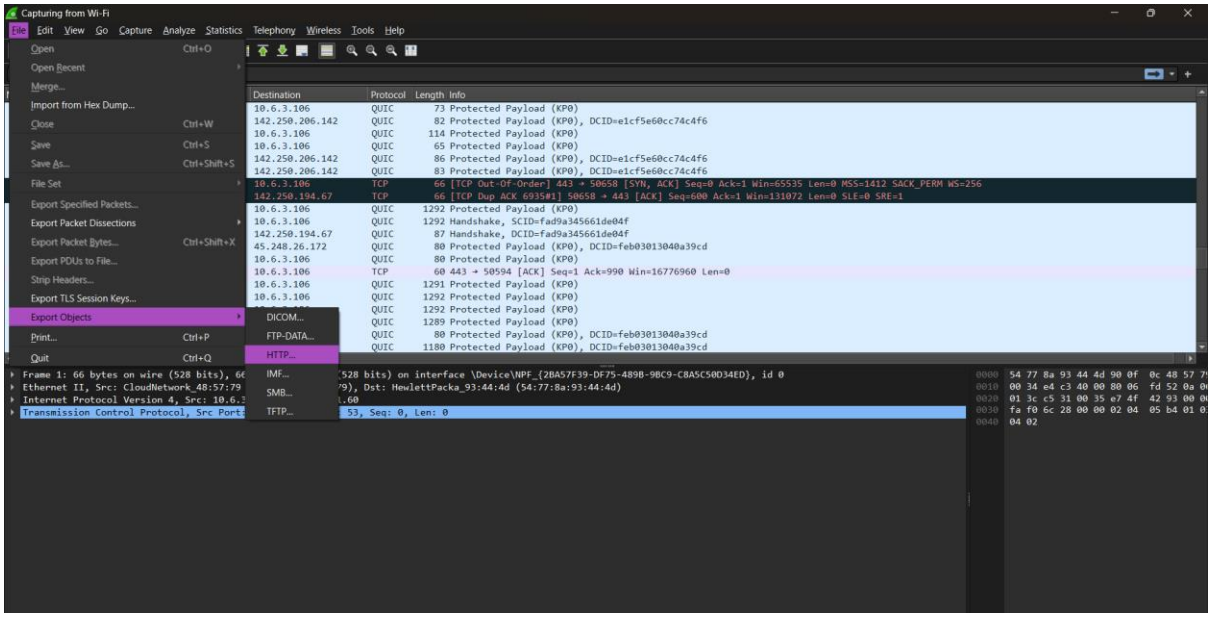
Fortunately, Wireshark allows you to select a packet and view the entire TCP stream it belongs to.

Inbound and outbound traffic will be highlighted in red and blue to show the application layer communication without packet headers.



Exporting Objects

Often during a pentest you may be looking to grab sensitive information from plain text streams. Wireshark has an “Export objects” function that combines protocol dissectors with content extractors to dump objects contained in streams. This can often reveal Jpegs from video streams, PDFs from HTTP downloads, and so on.

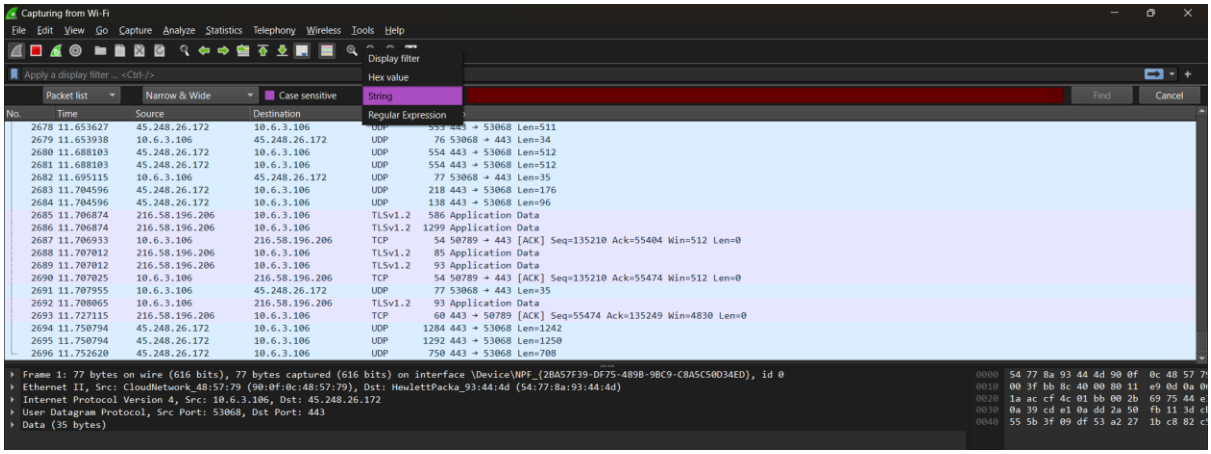


Remember -.pcap files are universal

Always remember that pcap files are not proprietary to Wireshark. As a pentester you surely will find it often more convenient to use tcpdump as a collector and use Wireshark on a different system to analyze the traffic.

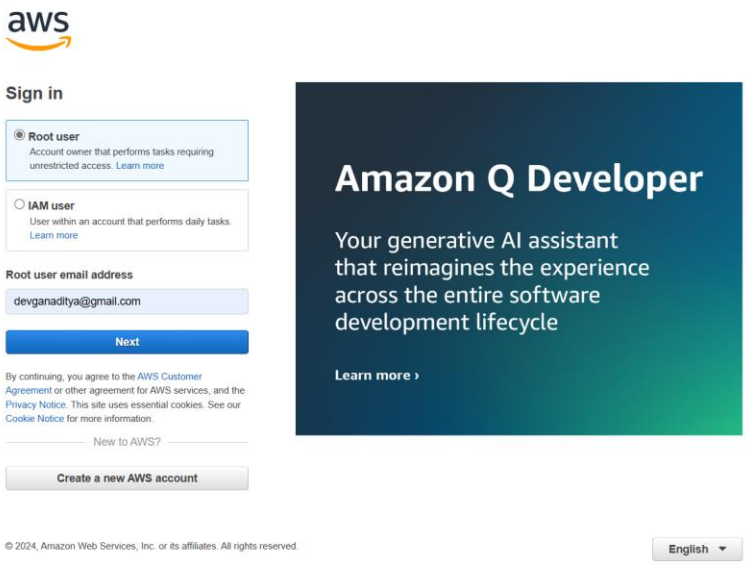
Searching for Strings

Searching for strings is not entirely trivial. Hitting Ctrl+F will bring up the search bar, however you must select string from the dropdown to search packet payloads for ascii strings.

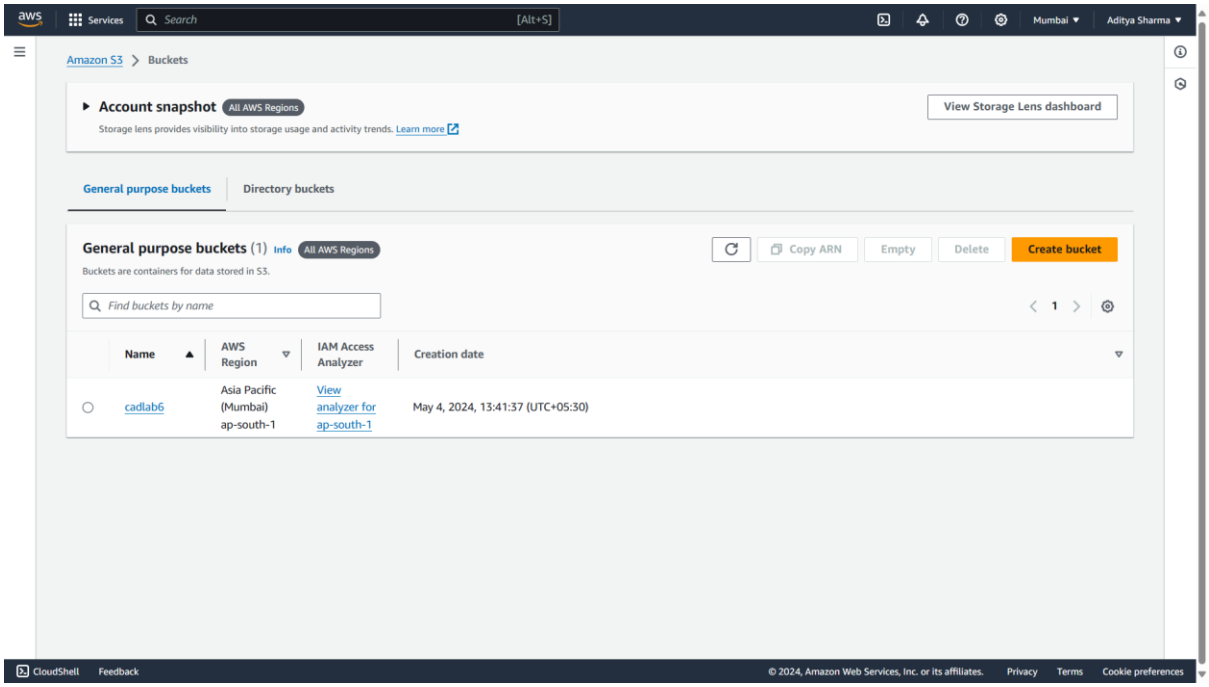


Experiment 6 – Implement and Evaluate AWS S3 as Storage as a Service

Step 1: Go to https://console.aws.amazon.com/ and click on "Create a New AWS Account" if you don't have one already. Login to the AWS Management Console.

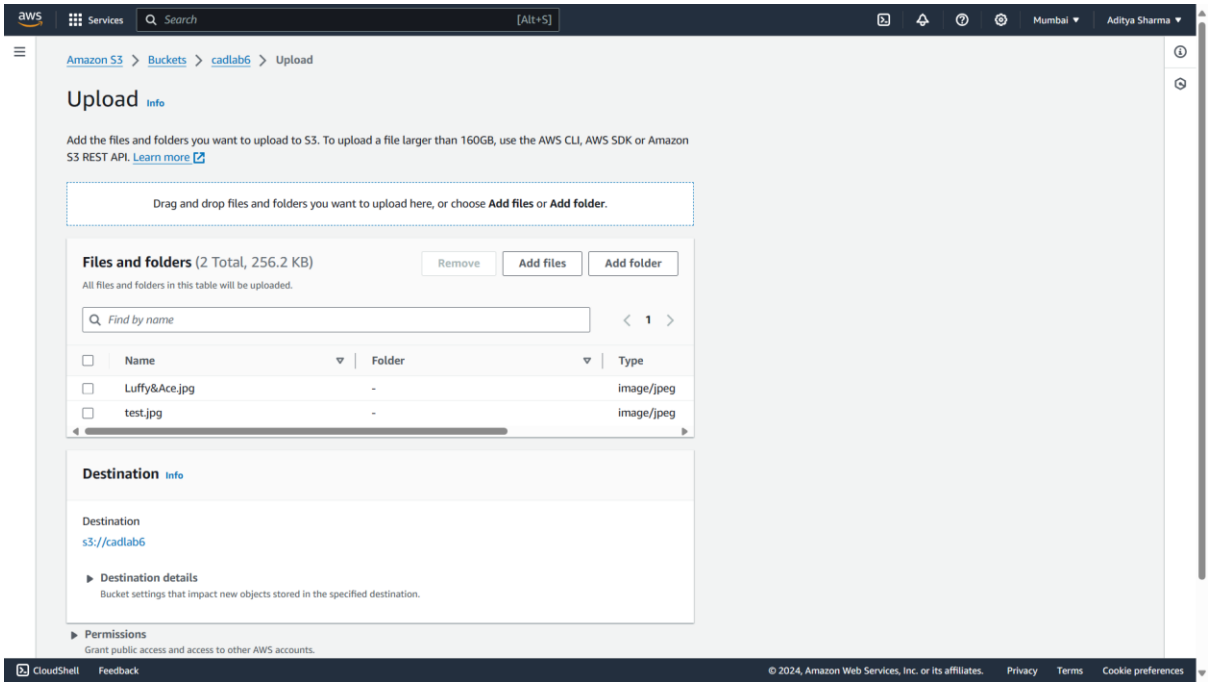


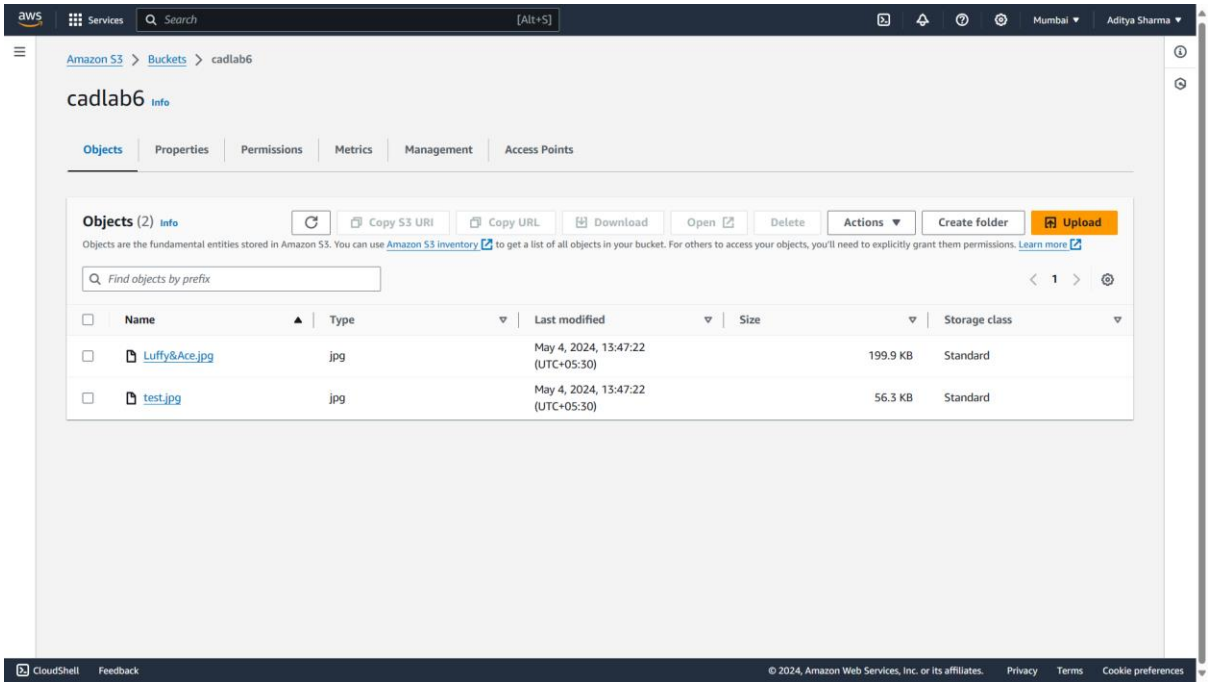
Step 2: Search for "S3" in the services search bar and open S3 service. Click on "Create Bucket". Choose a unique bucket name following AWS naming conventions. Select a region where you want your data stored. Click on "Create" to create your S3 bucket.



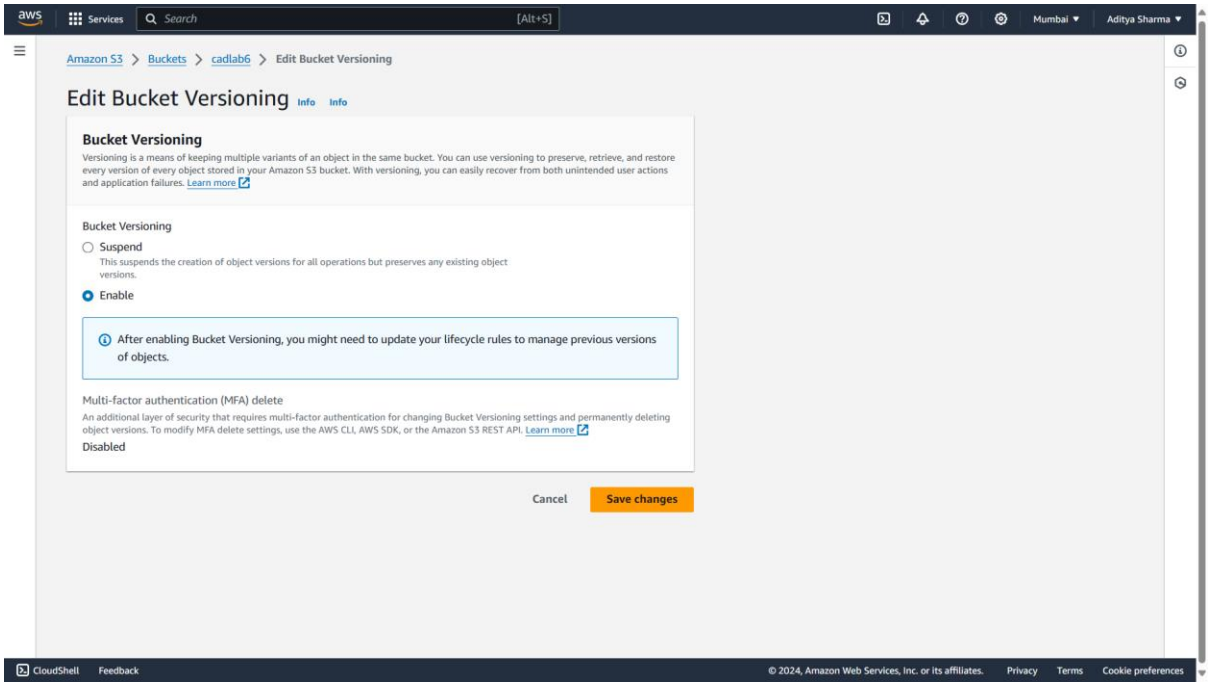
Step 3: To upload objects in S3 bucket, there are multiple ways: By using AWS Management Console, by using AWS CLI, and by using SDKs. I will be doing using AWS Management Console.

Click on your newly created bucket. Click on “Upload” button. Select the files you want to upload from your local machine and click "Upload."

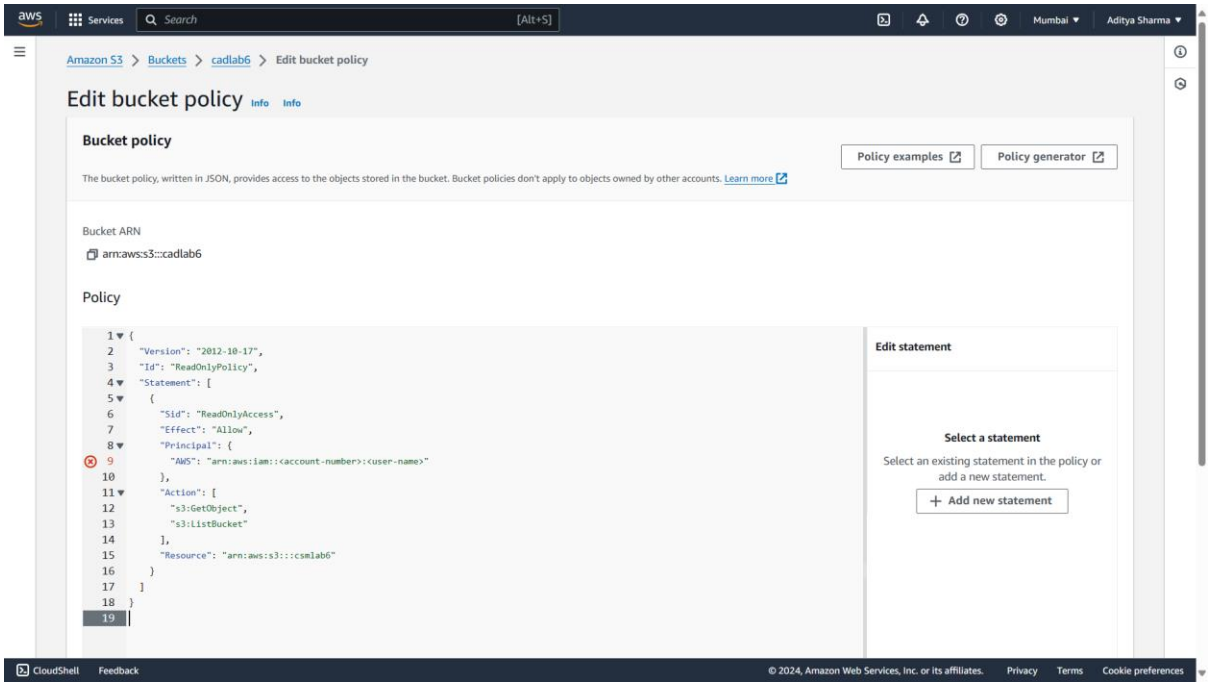




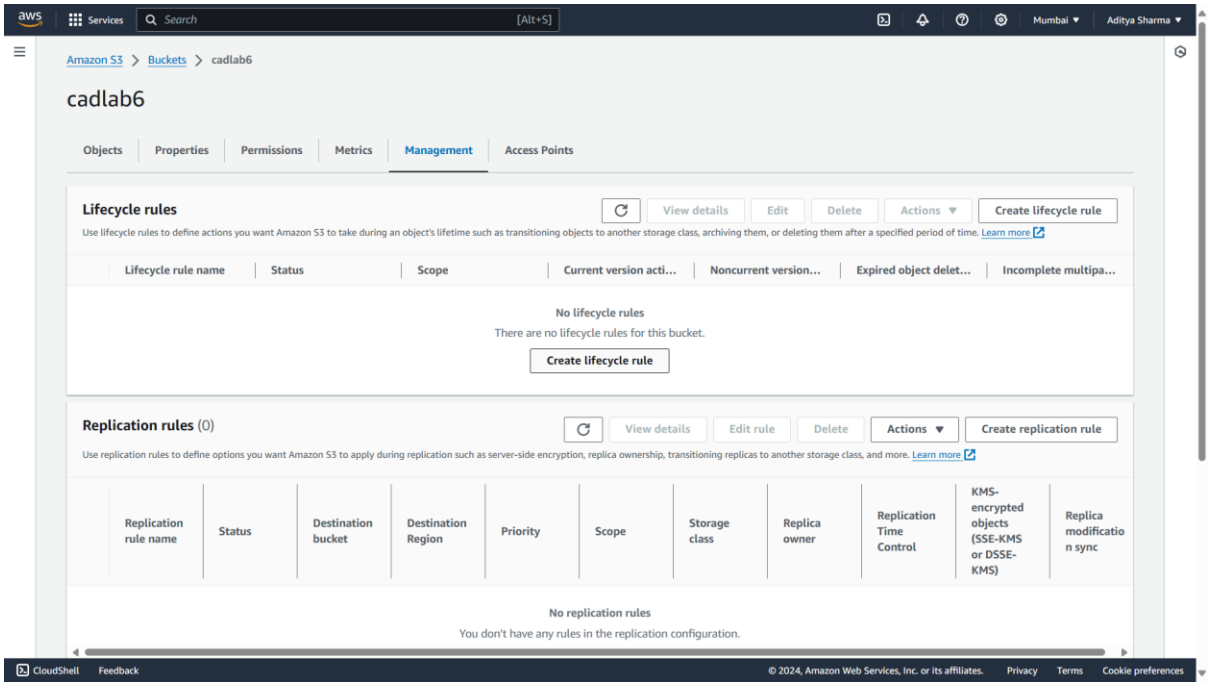
Step 4: Click on “Properties” tab and under Bucket Versioning, click on Edit. Click on Enable and Save Changes.Versioning keeps track of all object versions uploaded, allowing you to recover previous versions if needed.



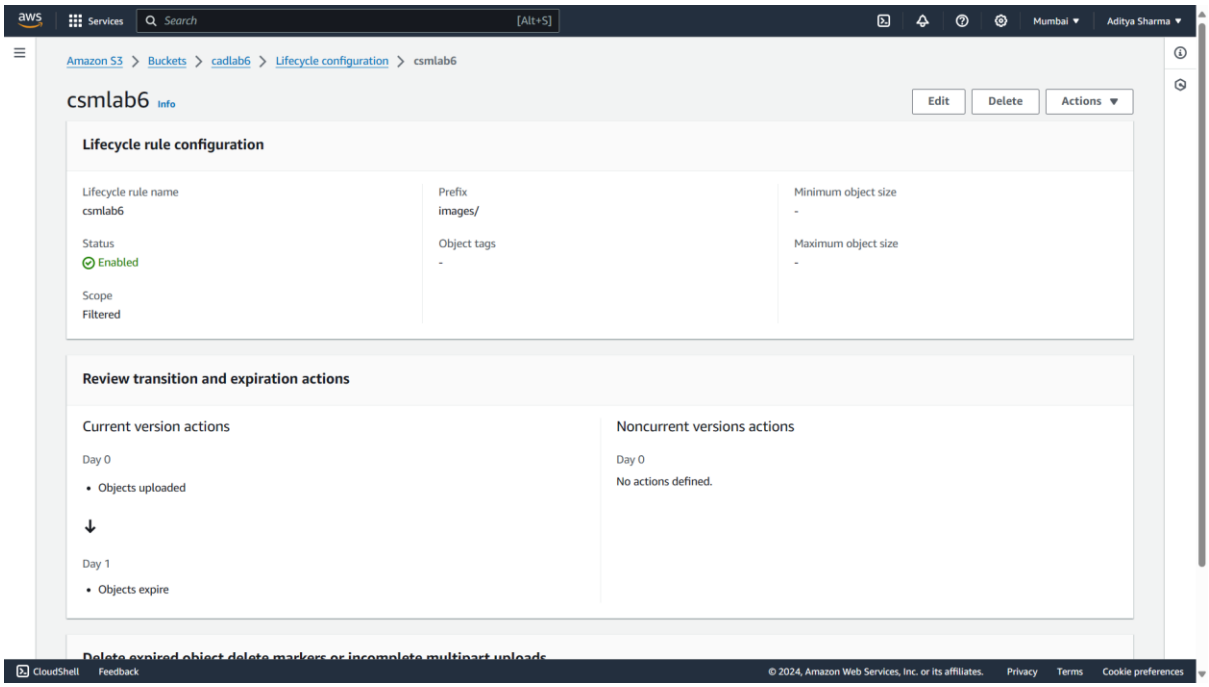
Step 5: For implementing Access Control, use Bucket Policies. Click on “Permissions” tab and under “Bucket Policy” click on Edit and below is an example of giving read only access to an IAM user.



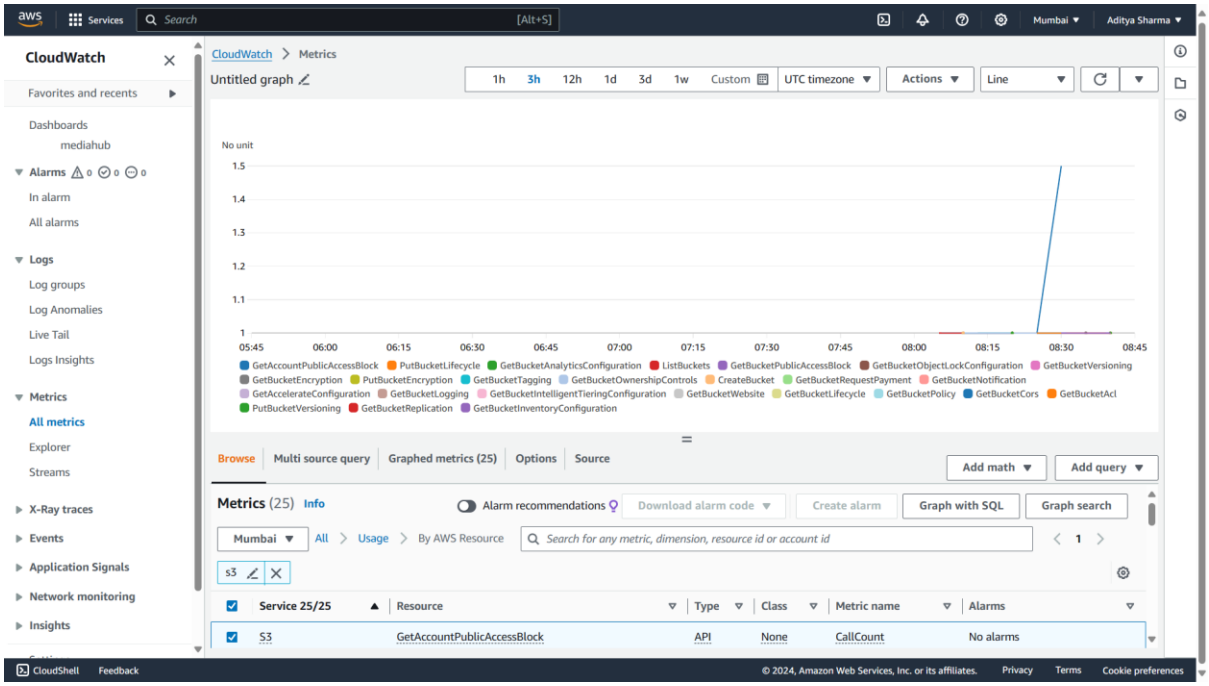
Step 6: For Lifecycle Policies, click on “Management” tab and under Lifecycle Rules click on create lifecycle rule.



I have created a rule to expire objects after 1 day.



Step 7: You can evaluate the performance of S3 bucket using CloudWatch.



Step 8:

Security Features:

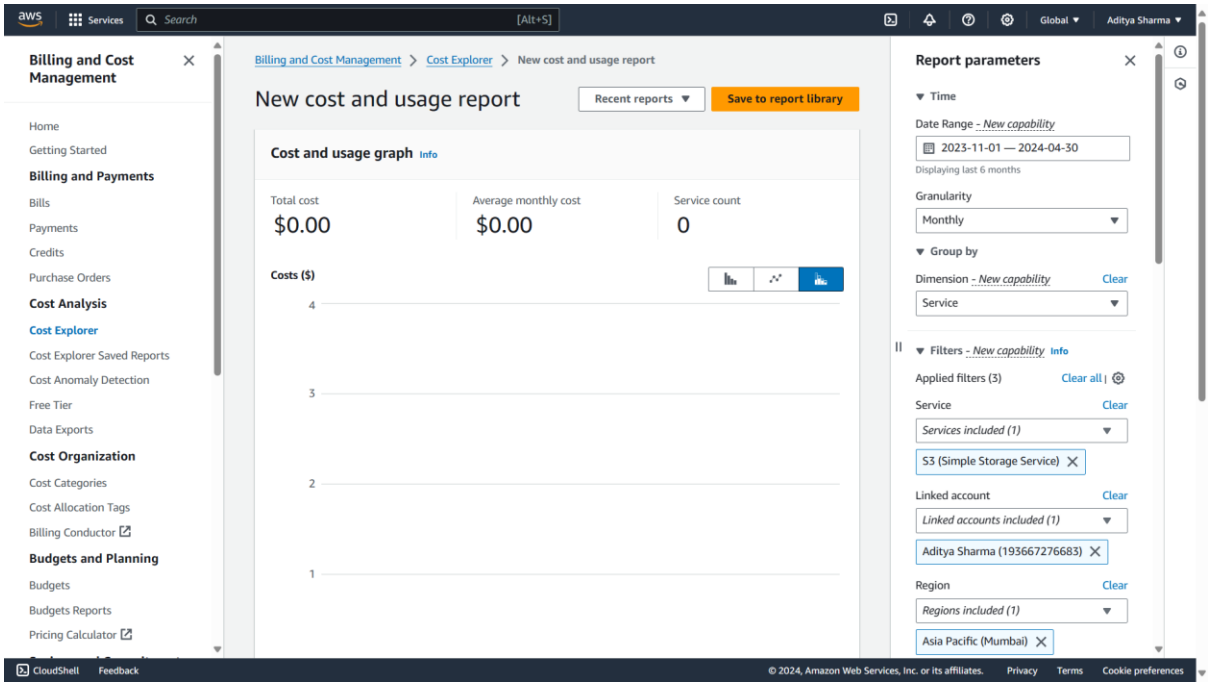
- **Encryption:** S3 encrypts data at rest (using AES-256) and in transit (using SSL/TLS). You can choose between server-side encryption (SSE) options like SSE-S3 and SSE-KMS for additional control over encryption keys.
- **IAM Policies:** Control access to your S3 bucket and objects using granular IAM policies that define who can access the data and what actions they can perform.
- **Bucket Logging:** Enable S3 bucket logging to track access requests and changes made to your S3 bucket. This helps with auditing and security analysis.

Compliance Considerations:

- Identify any specific security or compliance requirements for your data storage. S3 offers various features to support compliance with regulations like HIPAA, PCI DSS, and GDPR.
- Utilize features like encryption and access control to meet compliance requirements.

Step 9: Cost Exploration and Management can be done using Billing and Cost Management console of AWS. Go to cost explorer under cost analysis.

Select S3 service, linked account, region etc.

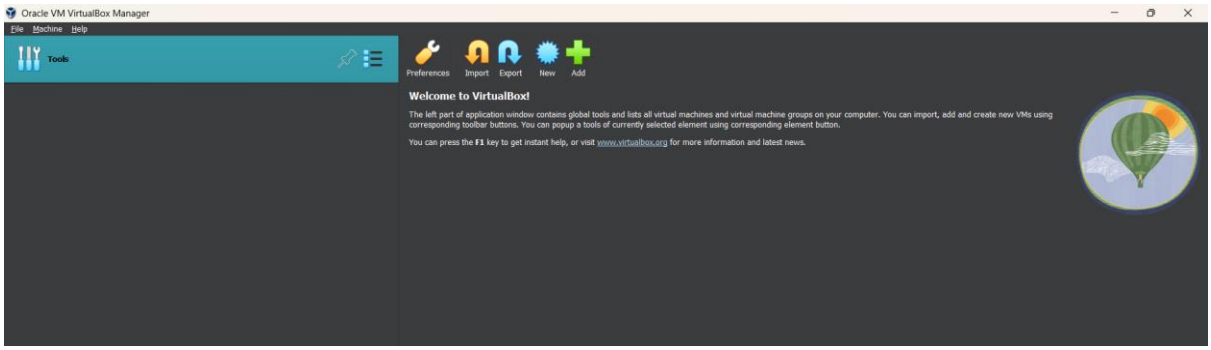

Billing and Cost Management
Home
Getting Started
Billing and Payments
Bills
Payments
Credits
Purchase Orders
Cost Analysis
Cost Explorer
Cost Explorer Saved Reports
Cost Anomaly Detection
Free Tier
Data Exports
Cost Organization
Cost Categories
Cost Allocation Tags
Billing Conductor
Budgets and Planning
Budgets
Budgets Reports
Pricing Calculator
Billing and Cost Management > Cost Explorer > New cost and usage report
New cost and usage report
Recent reports
Save to report library
Cost and usage graph Info
Total cost
$0.00
Average monthly cost
$0.00
Service count
0
Costs ($)
4
3
2
1
Report parameters
Time
Date Range - New capability
2023-11-01 — 2024-04-30
Displaying last 6 months
Granularity
Monthly
Group by
Dimension - New capability
Clear
Service
Filters - New capability Info
Applied filters (3)
Clear all
Service
Clear
Services included (1)
S3 (Simple Storage Service)
Linked account
Clear
Linked accounts included (1)
Aditya Sharma (193667276683)
Region
Clear
Regions included (1)
Asia Pacific (Mumbai)
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Experiment 7 – Implement Paravirtualization using Oracle Virtual Box

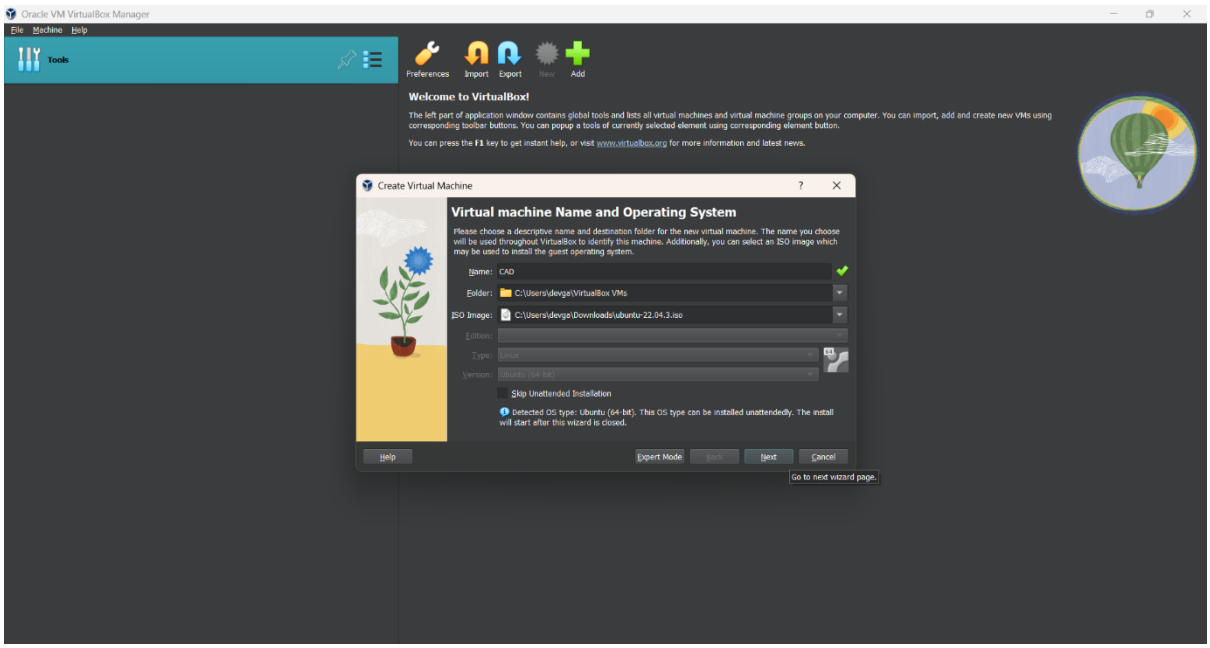
Paravirtualization is a type of virtualization where software instructions from the guest operating system running inside a virtual machine can use “hypercalls” that communicate directly with the hypervisor. This provides an interface very similar to software running natively on the host hardware.

The main benefits of paravirtualization are where instructions are not compatible with full virtualization or where more immediate access to underlying hardware is required for performance reasons. For timing-critical functions, paravirtualization can provide the speed of native code alongside some of the benefits of virtualization, such as sharing hardware between multiple operating systems.

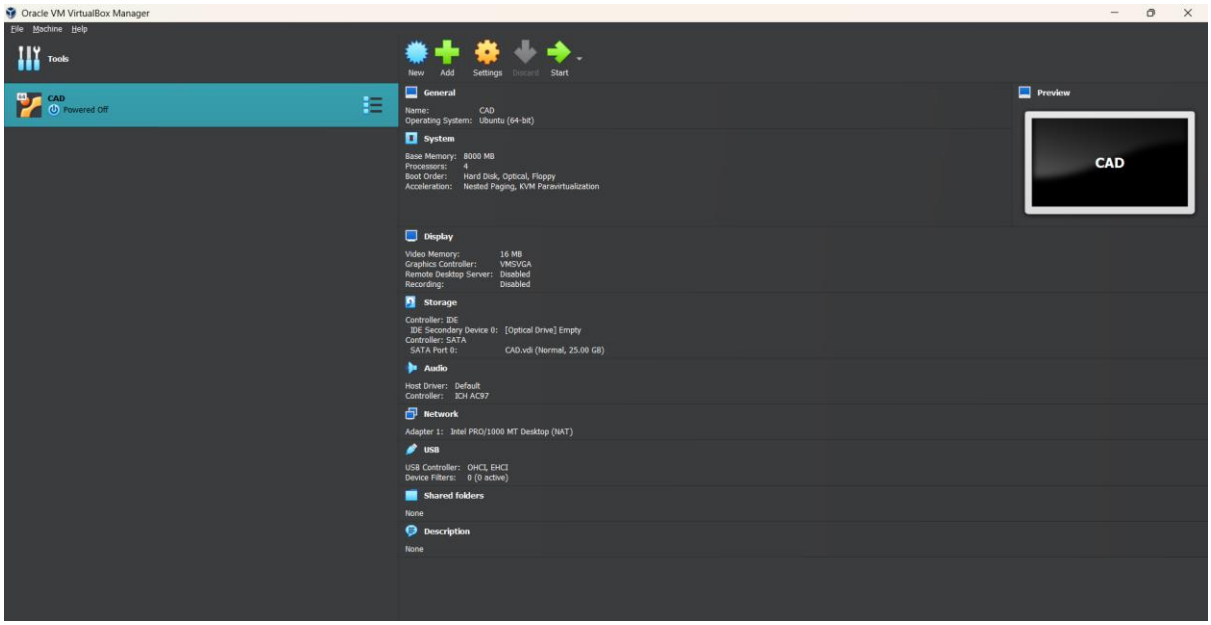
Step 1: Launch the Oracle VM VirtualBox application on your host machine.

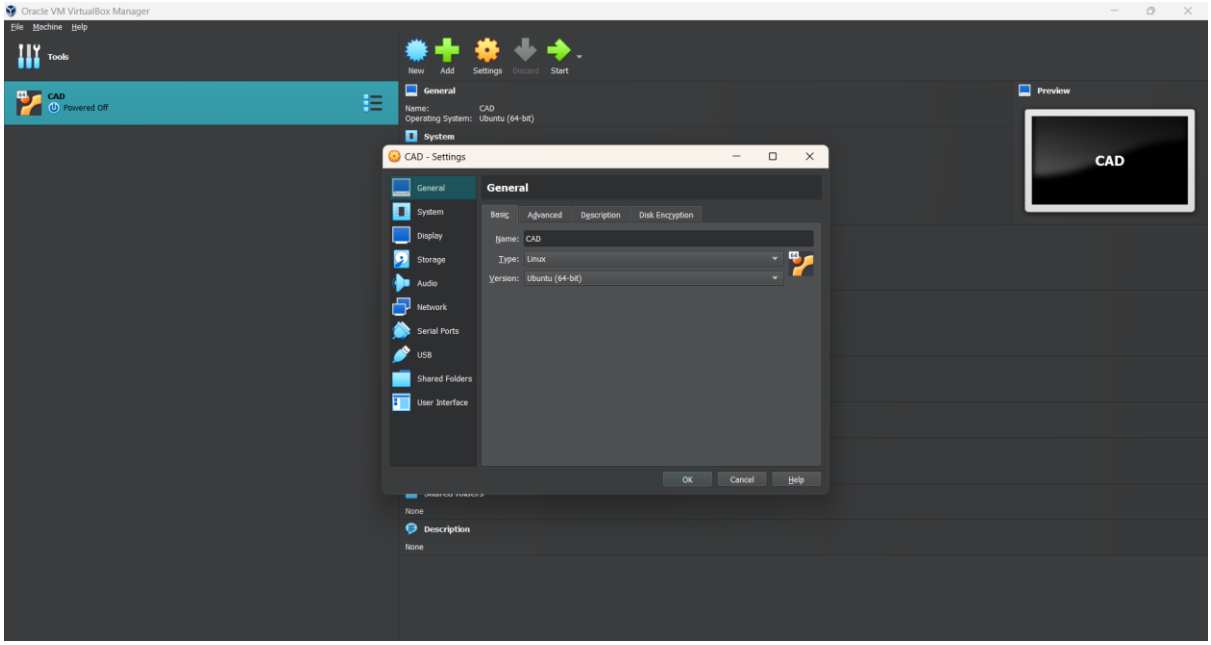


Step 2: In VirtualBox Manager window, click on the new to create a virtual machine.



Step 3: I am using Ubuntu VM. Right click on the virtual machine and choose "Settings" from the context menu.



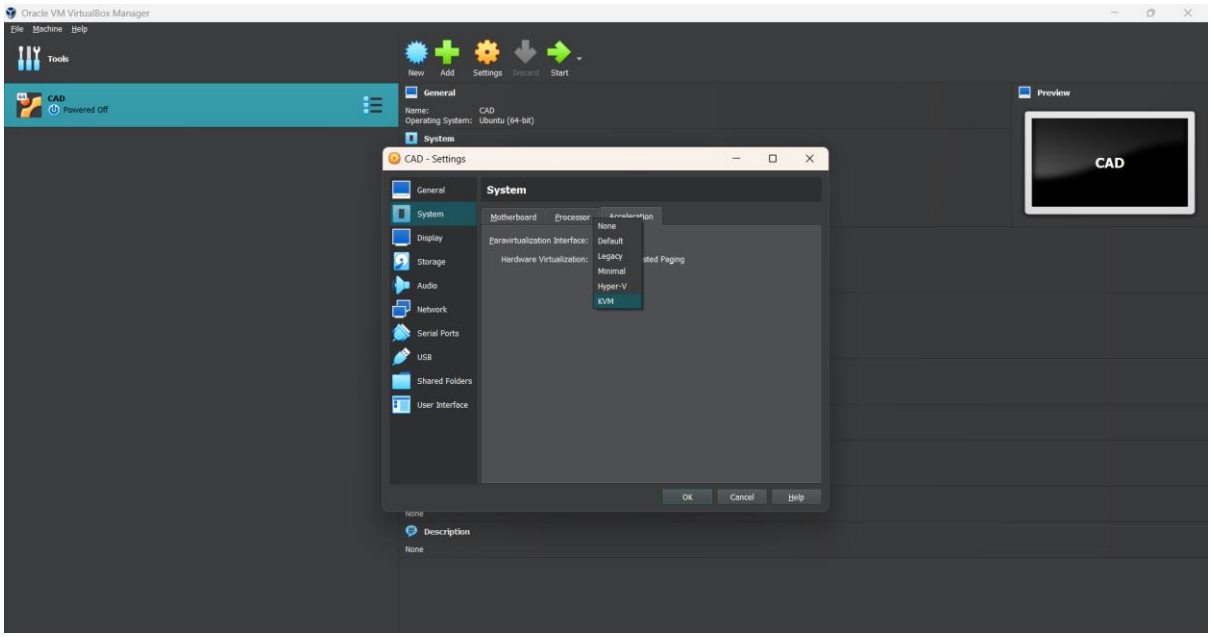


Step 4: In the Settings window, locate the "System" section on the left panel and click on it. Locate the "Acceleration" tab. Here, look for the setting labelled "Paravirtualization Interface".

For Ubuntu – KVM

For Windows – HyperV

For Mac – Minimal



Step 5: Click Ok and Start your virtual machine. With paravirtualization, Ubuntu will now benefit from the performance improvements offered by the paravirtualization (Reduced overhead, Faster Communication, Improved I/O performance, Enhanced Graphics and Potential Resource Efficiency).

Step 6: Install sysbench to observe change in performance.

```
aditya@CAD:~$ sudo apt install sysbench
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libaio1 libluajit-5.1-2 libluajit-5.1-common libmysqlclient21 libpq5 mysql-common
The following NEW packages will be installed:
  libaio1 libluajit-5.1-2 libluajit-5.1-common libmysqlclient21 libpq5 mysql-common sysbench
0 upgraded, 7 newly installed, 0 to remove and 252 not upgraded.
Need to get 1,862 kB of archives.
After this operation, 8,511 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libaio1 amd64 0.3.112-13build1 [7,176 B]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy/universe amd64 libluajit-5.1-common all 2.1.0~beta3+dfsg-6 [44.3 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy/universe amd64 libluajit-5.1-2 amd64 2.1.0~beta3+dfsg-6 [238 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 mysql-common all 5.8+1.0.8 [7,212 B]
Get:5 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libmysqlclient21 amd64 8.0.36-0ubuntu0.22.04.1 [1,302 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpq5 amd64 14.11-0ubuntu0.22.04.1 [144 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu jammy/universe amd64 sysbench amd64 1.0.20+ds-2 [120 kB]
Fetched 1,862 kB in 6s (323 kB/s)
Selecting previously unselected package libaio1:amd64.
(Reading database ... 205322 files and directories currently installed.)
Preparing to unpack .../0-libaio1_0.3.112-13build1_amd64.deb ...
Unpacking libaio1:amd64 (0.3.112-13build1) ...
```

Run following CPU and Memory tests before paravirtualization and after paravirtualization to compare results.

CPU - sysbench --test=cpu --cpu-max-prime=20000 run

Memory - sysbench --test=memory --memory-oper=read --memory-block-size=1024 run

```
aditya@CAD:~$ sysbench --test=cpu --cpu-max-prime=20000 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:   727.53

General statistics:
    total time:                          10.0010s
    total number of events:              7278

Latency (ms):
         min:                                    1.04
         avg:                                    1.37
         max:                                   22.79
         95th percentile:                        1.73
         sum:                                 9983.21

Threads fairness:
    events (avg/stddev):           7278.0000/0.00
    execution time (avg/stddev):   9.9832/0.00
```

```
aditya@CAD:~$ sysbench --test=memory --memory-oper=read --memory-block-size=1024 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: read
  scope: global

Initializing worker threads...

Threads started!

Total operations: 34064685 (3404941.20 per second)

33266.29 MiB transferred (3325.14 MiB/sec)


General statistics:
    total time:                          10.0007s
    total number of events:              34064685

Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                   19.14
         95th percentile:                        0.00
         sum:                                 3228.16

Threads fairness:
    events (avg/stddev):           34064685.0000/0.00
    execution time (avg/stddev):   3.2282/0.00
```