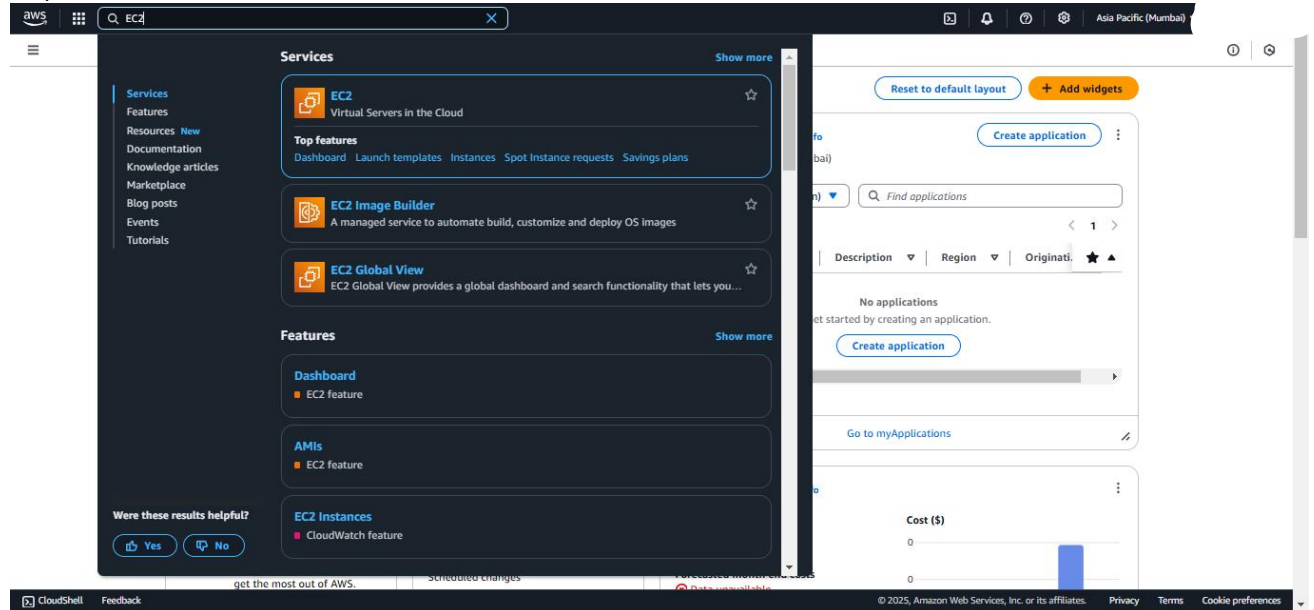


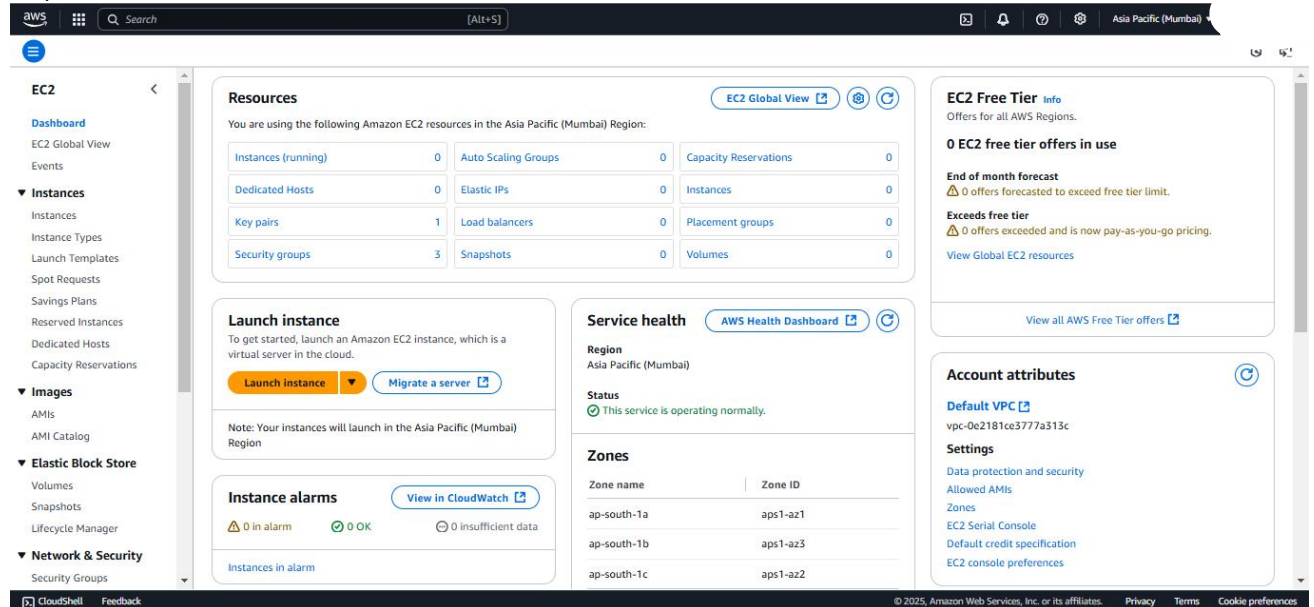
Assignment : 7

Hosting a website on EC2.

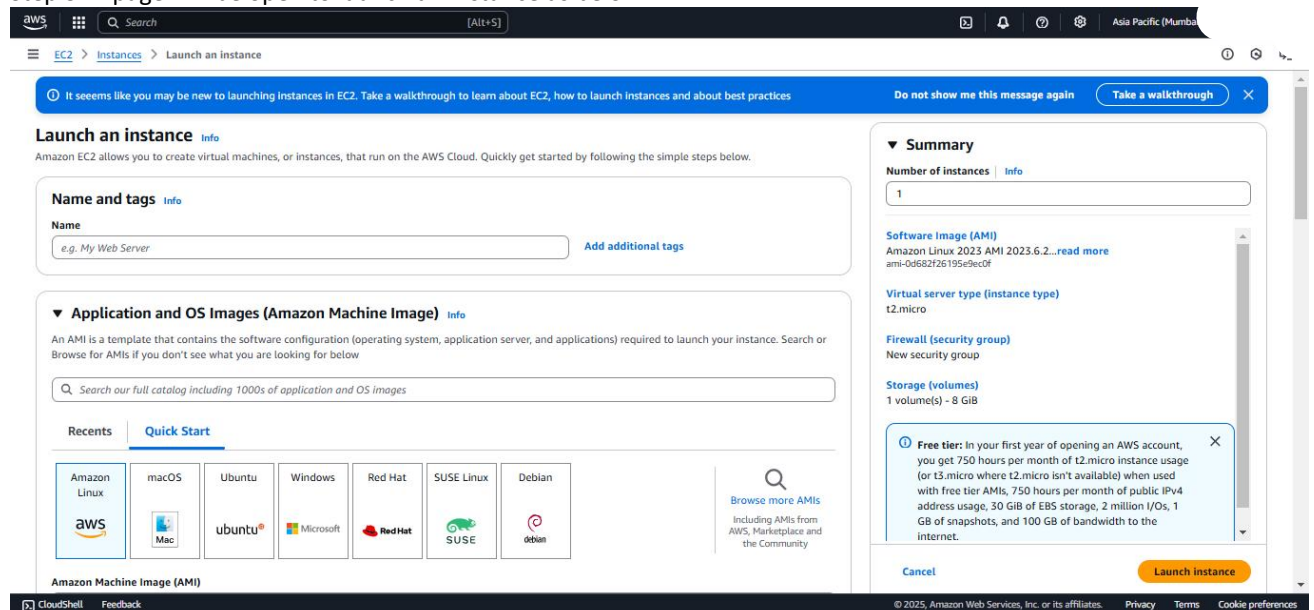
Step 1: Search for EC2 service in the search bar and click on it.



Step 2: Go to Dashboard section to check the status of the service, and then click on 'Launch Instance'.



Step 3: A page will be open to launch an instance as below.



Step 4: Name the instance and select 'Ubuntu' in the 'Application and OS image' section.

Step 4: Name the instance and select 'Ubuntu' in the 'Application and OS image' section.

Step 5: Scroll down to 'Key Pair' section, Create a new Key Pair as shown below; Then click on 'Create key pair'.

Step 5: Scroll down to 'Key Pair' section, Create a new Key Pair as shown below; Then click on 'Create key pair'.

Step 6: Scroll down to Network settings and check the 2 boxes to allow HTTPS & HTTP traffic from internet.

Step 6: Scroll down to Network settings and check the 2 boxes to allow HTTPS & HTTP traffic from internet.

Step 7: The instance is created successfully as shown below, and scroll down to click on 'View all instances'.

The screenshot shows the AWS Management Console after successfully launching an EC2 instance. A green banner at the top states: "Success Successfully initiated launch of instance (i-0c7fb8d929e0e25f9)". Below this is a "Launch log" section. The "Next Steps" section provides several options for managing the instance:

- Create billing and free tier usage alerts:** To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds. [Create billing alerts](#)
- Connect to your instance:** Once your instance is running, log into it from your local computer. [Connect to instance](#) [Learn more](#)
- Connect an RDS database:** Configure the connection between an EC2 instance and a database to allow traffic flow between them. [Connect an RDS database](#) [Create a new RDS database](#) [Learn more](#)
- Create EBS snapshot policy:** Create a policy that automates the creation, retention, and deletion of EBS snapshots. [Create EBS snapshot policy](#)
- Manage detailed monitoring:** Enable or disable detailed monitoring for the instance. If you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period.
- Create Load Balancer:** Create an application, network gateway or classic Elastic Load Balancer. [Create Load Balancer](#)
- Create AWS budget:** AWS Budgets allows you to create budgets, forecast spend, and take action on your costs and usage from a single location.
- Manage CloudWatch alarms:** Create or update Amazon CloudWatch alarms for the instance. [Manage CloudWatch alarms](#)

Step 8: In the Instances window, click on the instance id.

The screenshot shows the AWS Management Console's "Instances" page. The left sidebar contains navigation links for EC2, Images, Elastic Block Store, and Network & Security. The main content area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public
awsweb	i-0c7fb8d929e0e25f9	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	ec2-52-66-118-246.ap-...	52.66.

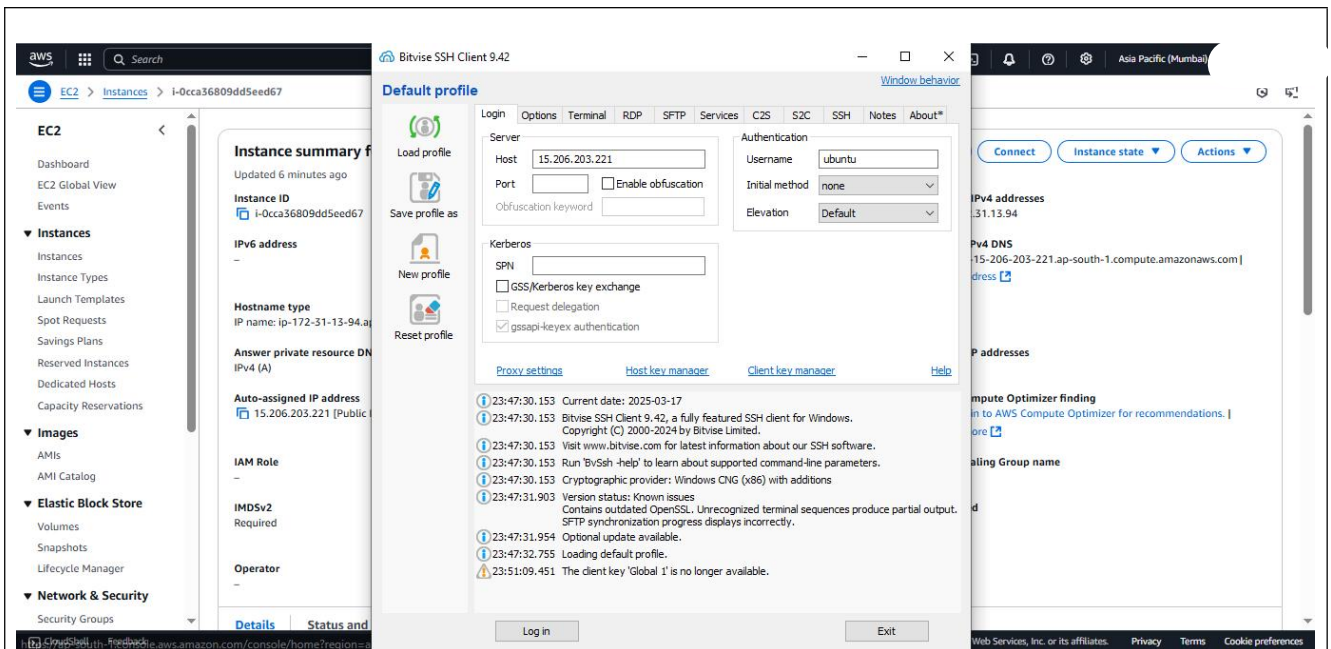
Below the table is a "Select an instance" section.

Step 9: Copy the IPV4 address shown in the Instance summary.

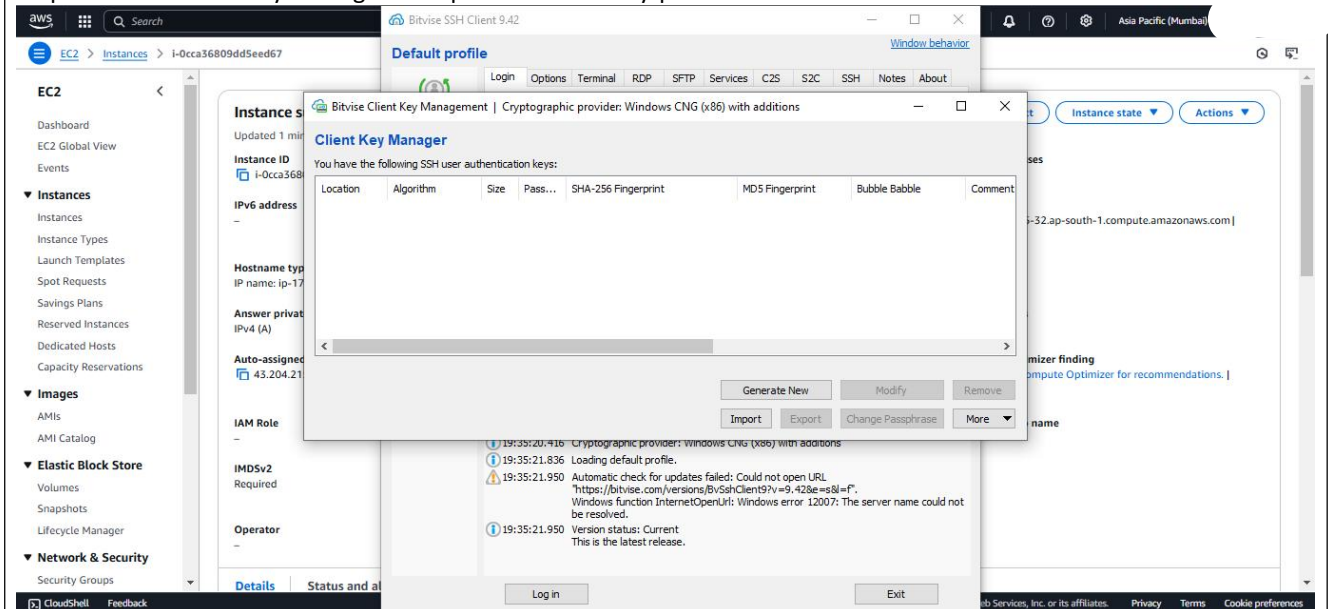
The screenshot shows the "Instance summary" page for the instance i-0cca36809dd5eed67 (awsweb). The summary is organized into several sections:

- Instance ID:** i-0cca36809dd5eed67
- IPv6 address:** -
- Hostname type:** IP name: ip-172-31-13-94.ap-south-1.compute.internal
- Answer private resource DNS name:** IPv4 (A)
- Auto-assigned IP address:** 15.206.203.221 [Public IP]
- IAM Role:** -
- IMDSv2:** Required
- Operator:** -
- Public IPv4 address:** 15.206.203.221 [open address](#)
- Instance state:** Running
- Private IP DNS name (IPv4 only):** ip-172-31-13-94.ap-south-1.compute.internal
- Instance type:** t2.micro
- VPC ID:** vpc-0e2181ce3777a313c [open address](#)
- Subnet ID:** subnet-0e06b1c09b402835f [open address](#)
- Instance ARN:** arn:aws:ec2:ap-south-1:586794457897:instance/i-0cca36809dd5eed67
- Private IPv4 addresses:** 172.31.13.94
- Public IPv4 DNS:** ec2-15-206-203-221.ap-south-1.compute.amazonaws.com | [open address](#)
- Elastic IP addresses:** -
- AWS Compute Optimizer finding:** Opt-in to AWS Compute Optimizer for recommendations. | [Learn more](#)
- Auto Scaling Group name:** -
- Managed:** false

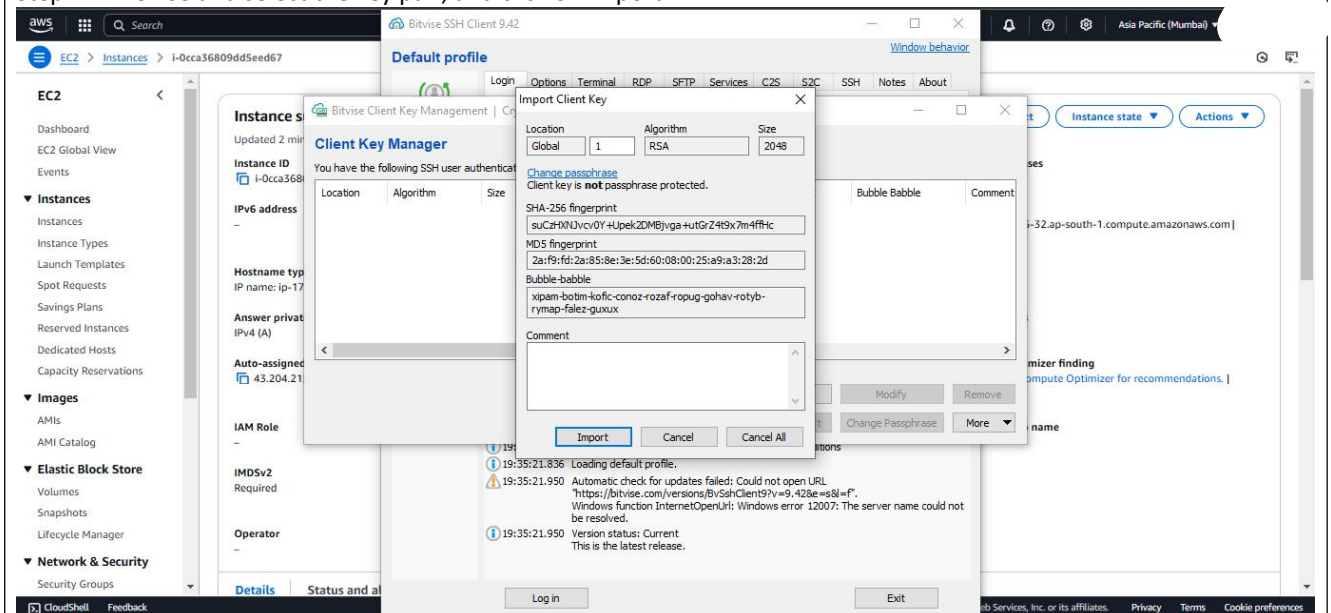
Step 10: Go to "https://bitwise.com/ssh-client-download" and download the "Bitwise SSH Client" and install it in the system.

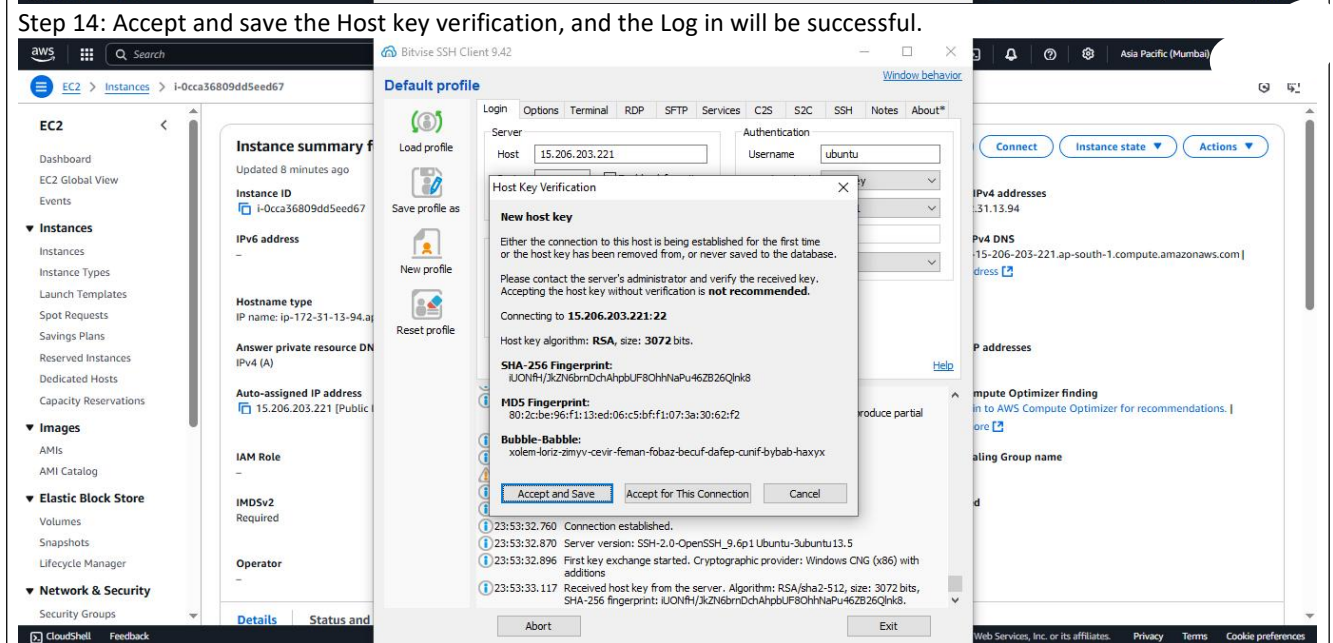
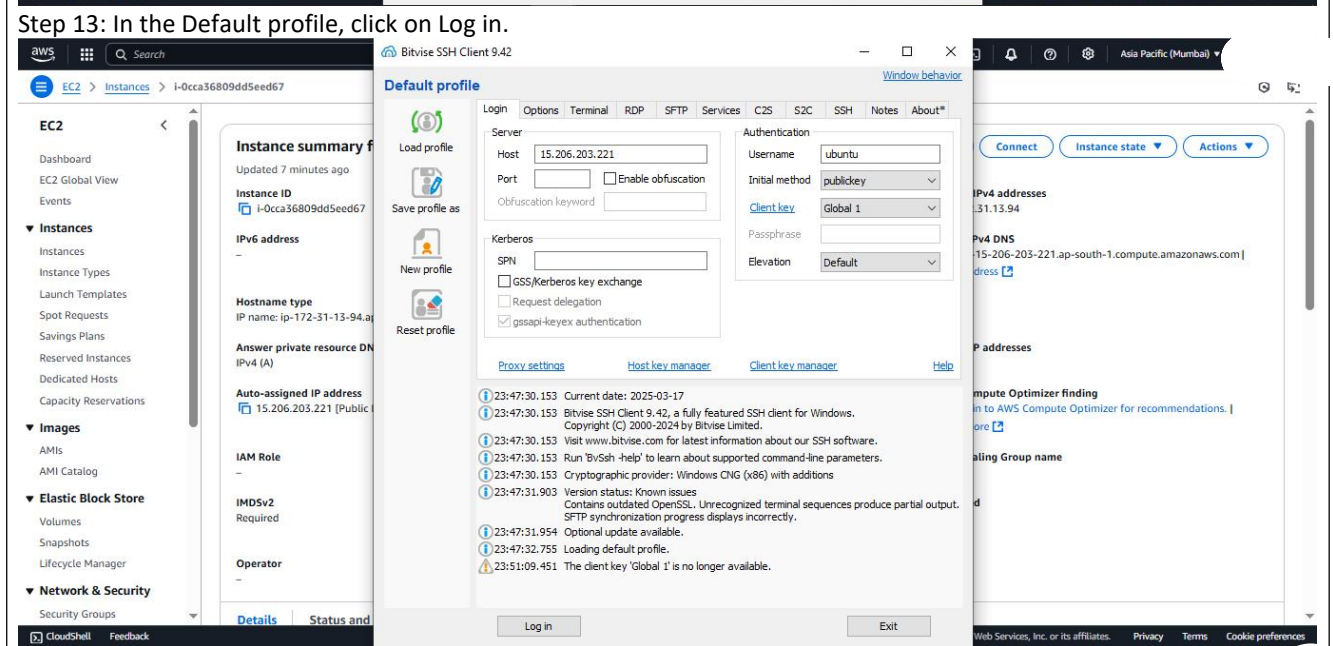
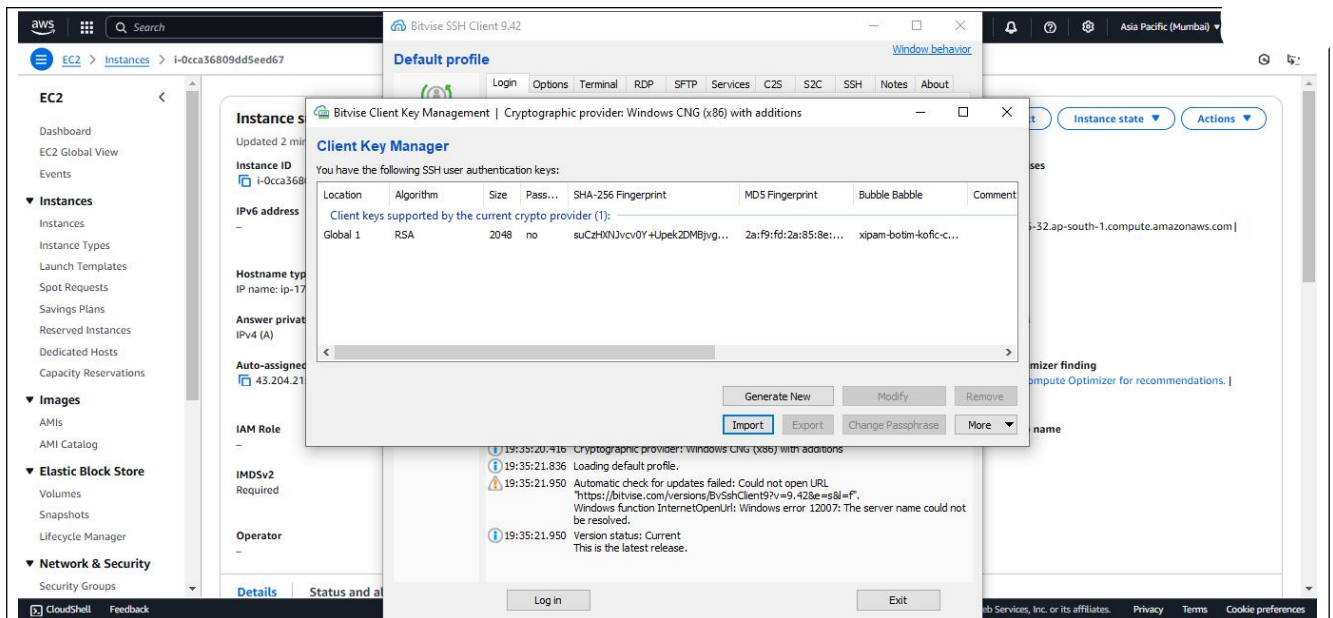


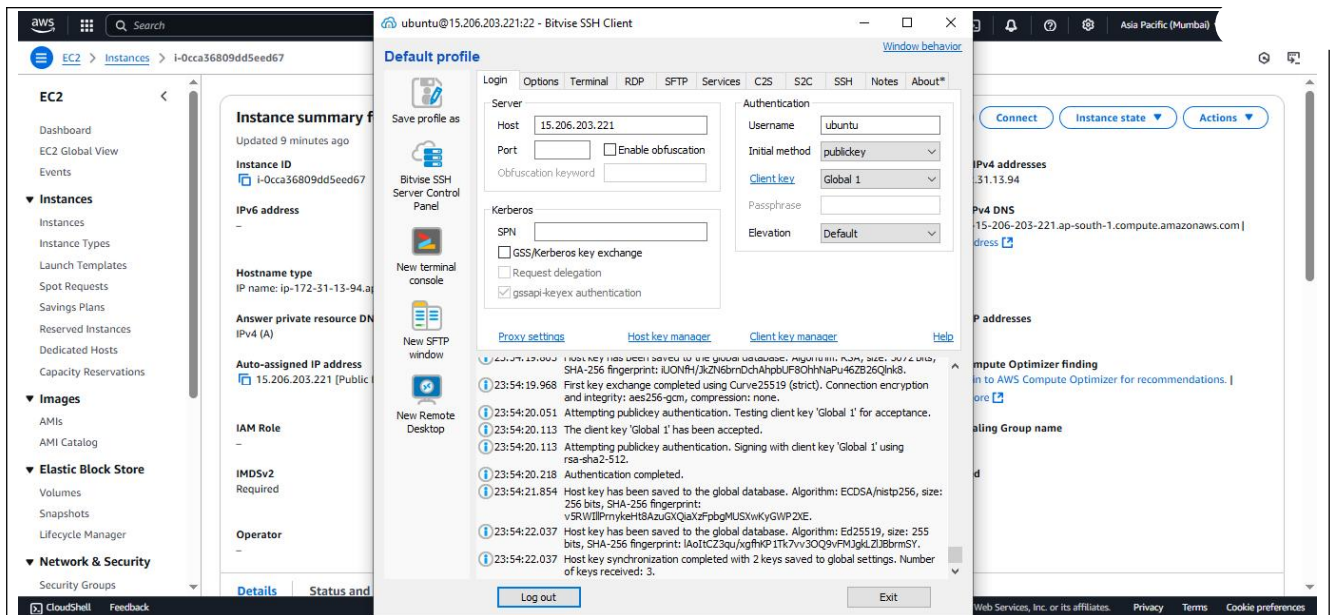
Step 11: Go to Client Key Manager to import the saved key-pair.



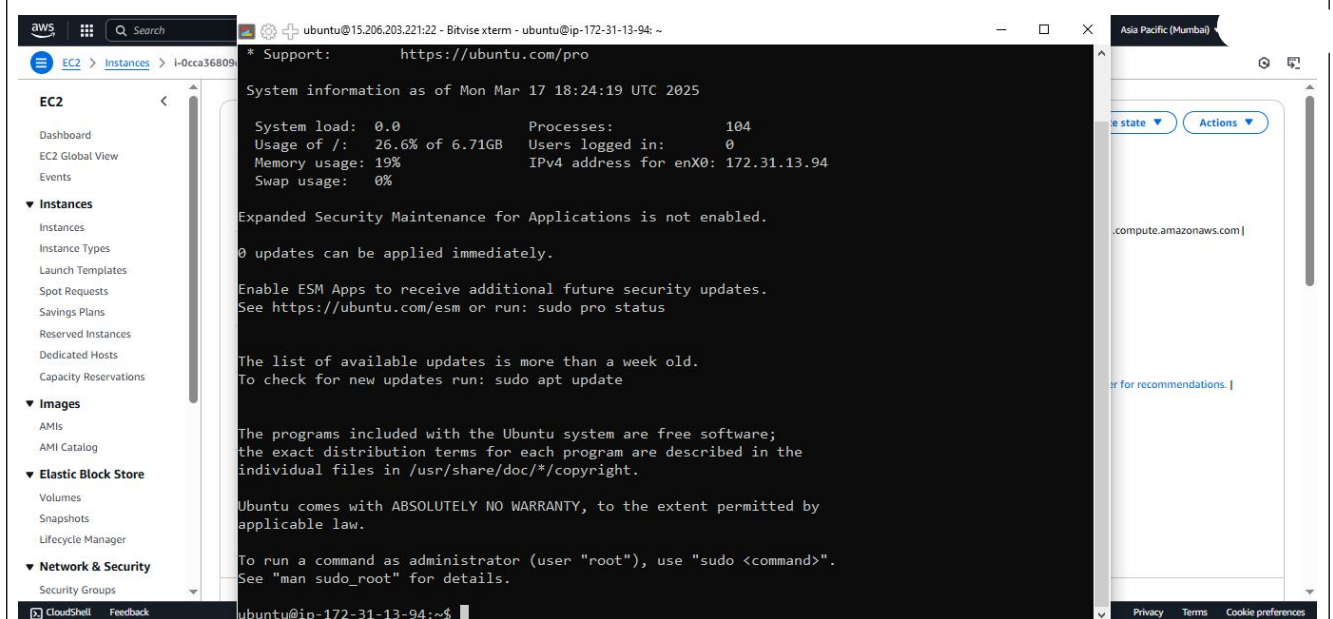
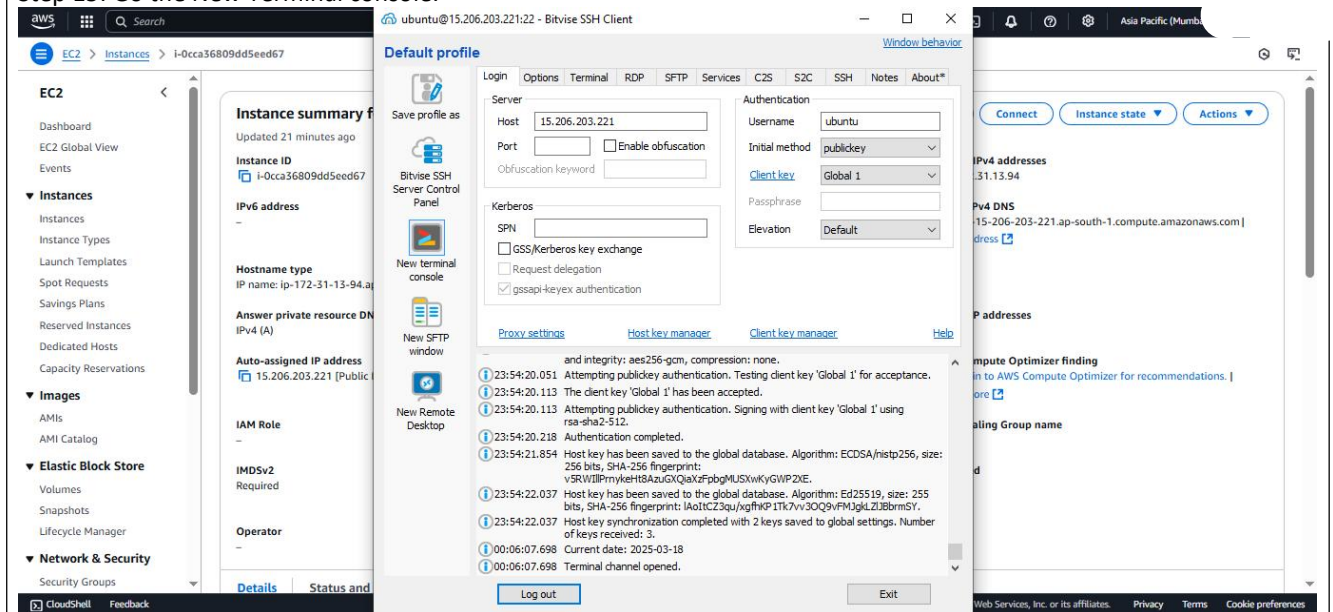
Step 12: Browse and select the key pair, and click on Import.







Step 15: Go the New Terminal console.



Step 16: Run "sudo apt-get update".

The screenshot shows the AWS CloudShell interface with a terminal window. The terminal output for the command `sudo apt-get update` is as follows:

```
ubuntu@ip-172-31-13-94:~$ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [921 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [209 kB]
Get:15 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]
Get:16 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [13.4 kB]
Get:17 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1040 kB]
Get:18 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [262 kB]
Get:19 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [364 kB]
Get:20 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [25.8 kB]
Get:21 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [759 kB]
```

Step 17: Run "sudo apt-get upgrade".

The screenshot shows the AWS CloudShell interface with a terminal window. The terminal output for the command `sudo apt-get upgrade` is as follows:

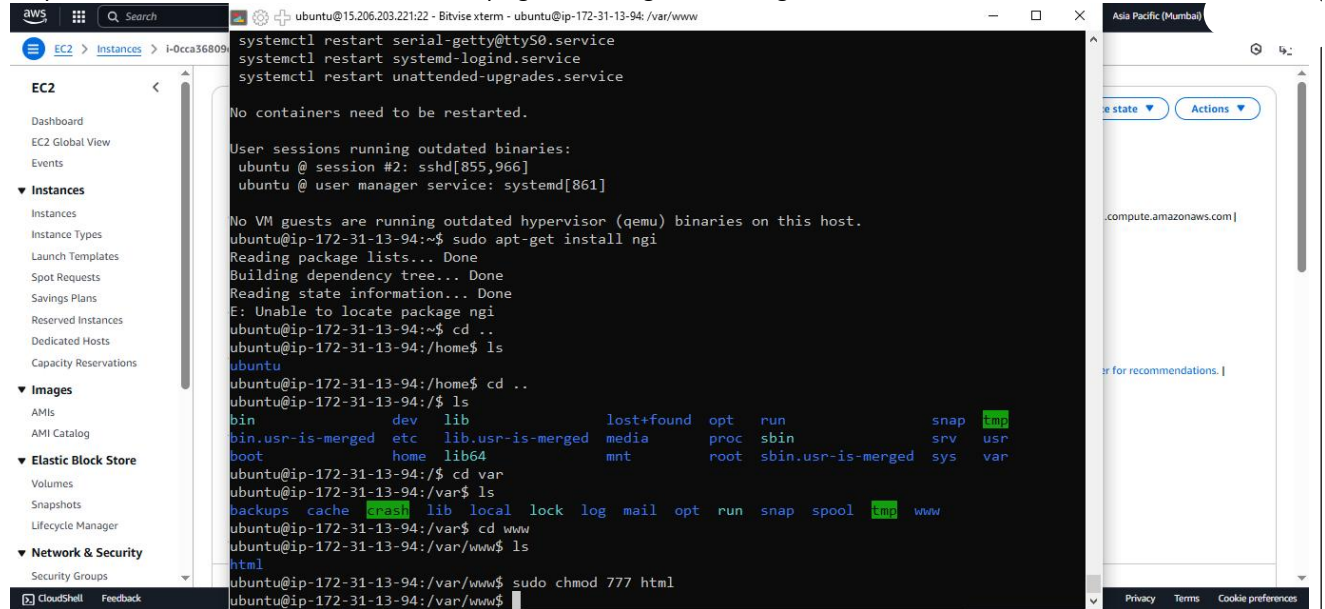
```
ubuntu@ip-172-31-13-94:~$ sudo apt-get upgrade
Get:53 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Get:54 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [448 B]
Fetched 32.8 MB in 22s (1497 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-13-94:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following upgrades have been deferred due to phasing:
  landscape-common libnss-systemd libpam-systemd libplymouth5 libsystemd-shared libsystemd0
  libudev1 plymouth plymouth-theme-ubuntu-text systemd systemd-dev systemd-resolved systemd-sysv
  udev
The following packages have been kept back:
  linux-aws linux-headers-aws linux-image-aws
The following packages will be upgraded:
  apport apport-core-dump-handler base-files bind9-dnswriter bind9-host bind9-libs bpftrace
  bsdxattrutils bsdutils cloud-init cryptsetup cryptsetup-bin cryptsetup-initramfs dmccventd
  dmsetup dracut-install eject fdisk fwupd gir1.2-packagekit-glib-1.0 initramfs-tools
  initramfs-tools-bin initramfs-tools-core intel-microcode kmod krb5-locales libaio1t64 libattr1
  libblkid1 libbsd0 libc-bin libc6 libcap2 libcap2-bin libcryptsetup12 libdevmapper-event1.02.1
  libdevmapper1.02.1 libdrm-common libdrm2 libdw1t64 libelf1t64 libfdisk1 libfwupd2 libgmp10
  libgnutls30t64 libgpg-error-1.0 libgpg-error0 libgssapi-krb5-2 libidn2-0 libk5crypto3 libkmod2
  libkrb5-3 libkrb5support0 libldap-common libldap2 liblvm2cmd2.03 libmd0 libmount1 libmpfr6
  libnghttp2-14 libnl-3-200 libnl-genl-3-200 libnl-route-3-200 libnvmelt64 libopeniscsiur
  libpackagekit-glib2-18 libpam-cap libpcre2-8-0 libperl5.38t64 libpolkit-agent-1-0
  libpolkit-gobject-1-0 libpython3.12-minimal libpython3.12-stdlib libpython3.12t64 libselinux1
  libsmartcols1 libsqlite3-0 libssl3t64 libtasn1-6 libunistring5 libunwind8 libuuid1 libxml2
  linux-tools-common locales lvm2 motd-news-config mount open-iscsi openssh-client openssh-server
  openssl-sftp-server openssl packagekit packagekit-tools perl perl-base perl-modules-5.38 polkitd
  pollinate python-apt-common python3-apport python3-apt python3-distupgrad python3-jinja2
  python3-problem-report python3.12 python3.12-minimal rsync snapd sosreport tzdata tzdata-legacy
```

Step 18: Then run command "sudo apt-get install nginx".

The screenshot shows the AWS CloudShell interface with a terminal window. The terminal output for the command `sudo apt-get install nginx` is as follows:

```
ubuntu@ip-172-31-13-94:~$ sudo apt-get install nginx
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-13-94:~$ sudo apt-get install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  nginx-common
Suggested packages:
  fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  nginx nginx-common
0 upgraded, 2 newly installed, 0 to remove and 17 not upgraded.
Need to get 552 kB of archives.
After this operation, 1596 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx-common all 1.24
  .0-2ubuntu7.1 [31.2 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx amd64 1.24.0-2u
  buntu7.1 [521 kB]
Fetched 552 kB in 0s (23.3 MB/s)
Preconfiguring packages ...
Selecting previously unselected package nginx-common.
(Reading database ... 70568 files and directories currently installed.)
Preparing to unpack .../nginx-common_1.24.0-2ubuntu7.1_all.deb ...
Unpacking nginx-common (1.24.0-2ubuntu7.1) ...
Selecting previously unselected package nginx.
Preparing to unpack .../nginx_1.24.0-2ubuntu7.1_amd64.deb ...
Unpacking nginx (1.24.0-2ubuntu7.1) ...
Setting up nginx (1.24.0-2ubuntu7.1) ...
Setting up nginx-common (1.24.0-2ubuntu7.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service -> /usr/lib/systemd/system/
  nginx.service.
```


Step 19: Then we run the command “sudo apt-get install nginx” and navigate to the html file.



```
aws ubuntu@15.206.203.221:22 - Bitvise xterm - ubuntu@ip-172-31-13-94: /var/www
systemctl restart serial-getty@ttyS0.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service

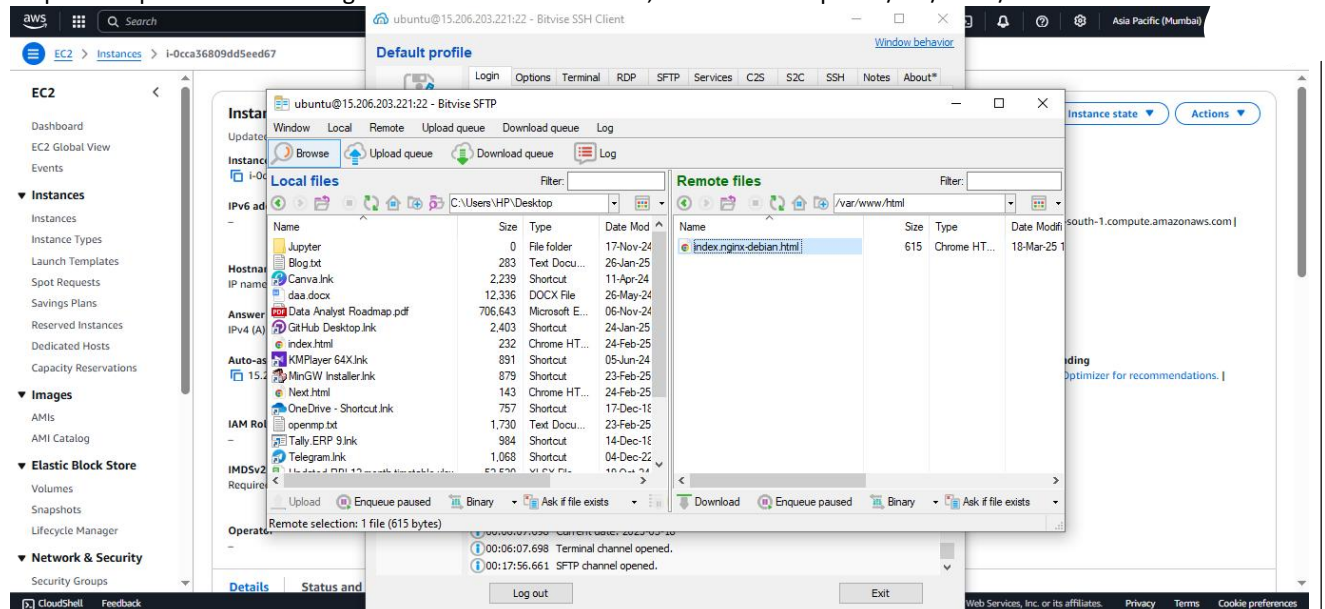
No containers need to be restarted.

User sessions running outdated binaries:
ubuntu @ session #2: sshd[855,966]
ubuntu @ user manager service: systemd[861]

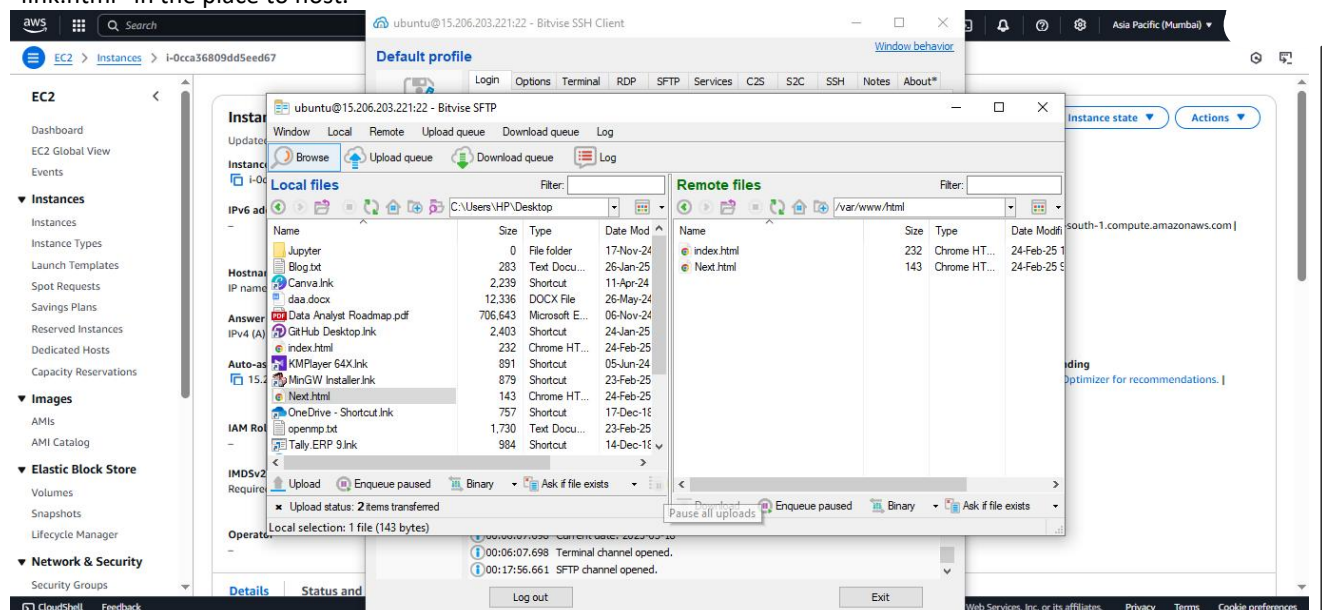
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-13-94:~$ sudo apt-get install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package nginx
ubuntu@ip-172-31-13-94:~$ cd ..
ubuntu@ip-172-31-13-94:/home$ ls
ubuntu
ubuntu@ip-172-31-13-94:/home$ cd ..
ubuntu@ip-172-31-13-94:/ $ ls
bin          dev          lib          lost+found  opt         run         snap        tmp
bin.usr-is-merged  etc         lib.usr-is-merged  media      proc       sbin       srv         usr
boot         home        lib64        mnt        root       sbin.usr-is-merged  sys         var

ubuntu@ip-172-31-13-94:/ $ cd var
ubuntu@ip-172-31-13-94:/var$ ls
backups  cache  crash  lib  local  lock  log  mail  opt  run  snap  spool  tmp  www
ubuntu@ip-172-31-13-94:/var$ cd www
ubuntu@ip-172-31-13-94:/var/www$ ls
html
ubuntu@ip-172-31-13-94:/var/www$ sudo chmod 777 html
ubuntu@ip-172-31-13-94:/var/www$
```

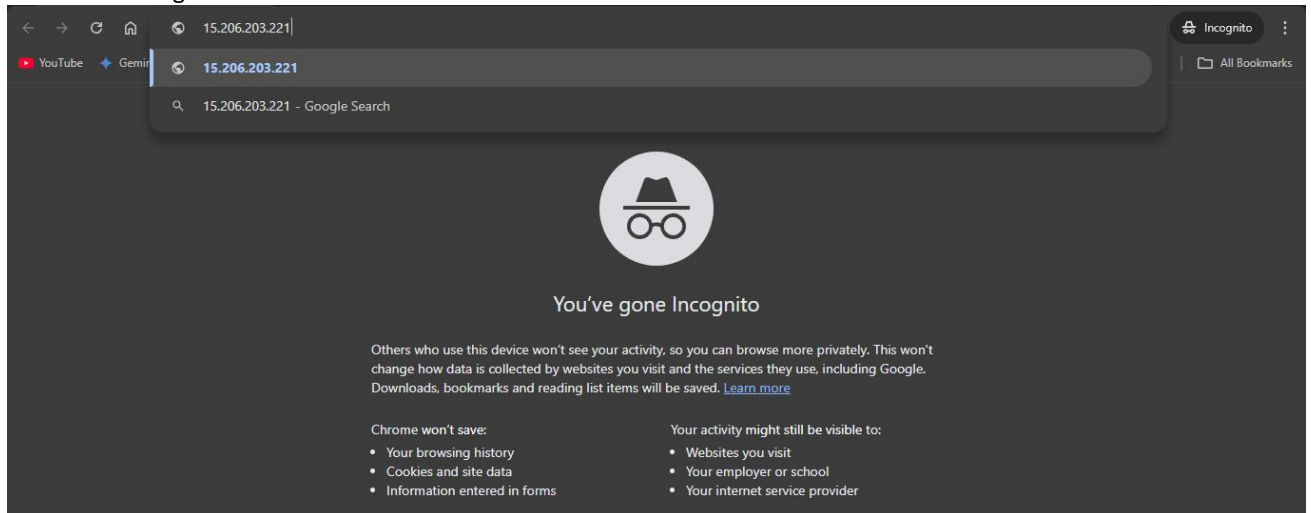
Step 20: Step 20: Then we navigate to “New SFTP window”, to folder with path “/var/www/html”.



Step 21: We delete the default html file “index.nginx-debian.html”. And we put our html files “index.html” and “link.html” in the place to host.



Step 24: Now we check whether the host is successful by copying the link in Incognito Mode. And checking whether the sites functioning.



Welcome to Index Page

This is a simple HTML page.

[Go to Link Page](#)

Page 1: Index.html

Welcome to Link Page

Here are some useful links:

- [Back to Index Page](#)
- [Visit YouTube](#)

Page 2: Link.html

Result: We successfully hosted a site on EC2.