


# Configuring Cisco Any Connect Secure Mobility Client

Manual: Configuring Cisco AnyConnect Client in Ubuntu

**Platform:**  Ubuntu

**Audience:**  NIC VPN Users

**Date:**  November 2025

---

---

# 1. Download & Extract

---

## 1.

### Go to user's home directory

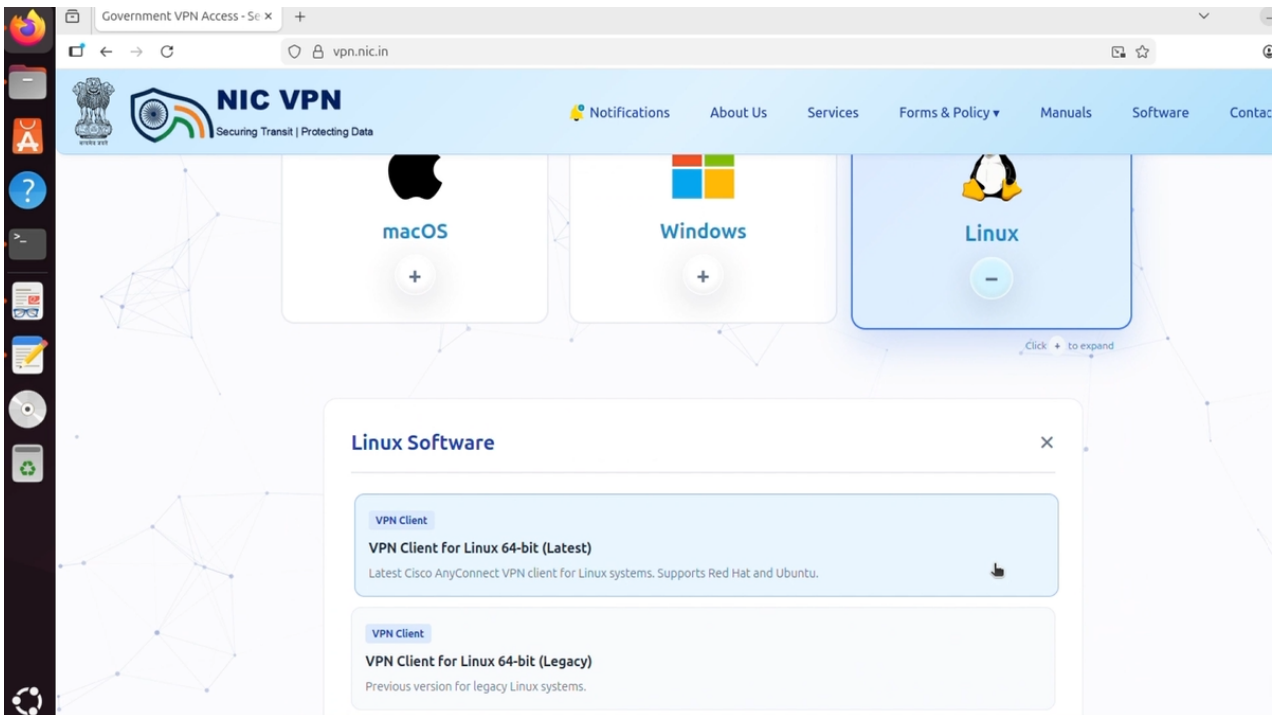
First, open a terminal and navigate to the **user's home directory**.

---

## 2.

### Download the Cisco AnyConnect Client

Download the **Cisco AnyConnect Client** from the website [vpn.nic.in](http://vpn.nic.in)



### 3.

#### Extract Folder and Install the VPN Client

Install the **VPN Client** by running the following commands.

##### a) tar -xvf anyconnect-linux-64-5.1.8.122-k9.tar.gz

```
anyconnect-linux-64-5.1.8.122-k9.tar.gz anyconnect-linux-64-5.1.11-XJUK.1.8.122-k9.tar.gz.part 'certificate.pfx'
jeet@ubuntu: ~/Downloads$
jeet@ubuntu: ~/Downloads$ ls
anyconnect-linux-64-5.1.8.122-k9.tar.gz anyconnect-linux-64-5.1.11-XJUK.1.8.122-k9.tar.gz.part 'certificate.pfx'
jeet@ubuntu: ~/Downloads$ ls
anyconnect-linux-64-5.1.8.122-k9.tar.gz 'certificate.pfx'
jeet@ubuntu: ~/Downloads$
jeet@ubuntu: ~/Downloads$
jeet@ubuntu: ~/Downloads$
jeet@ubuntu: ~/Downloads$ tar -xvf anyconnect-linux-64-5.1.8.122-k9.tar.gz
```

##### b) cd cisco-secure-client-linux64-5.1.8.122/vpn/

```
jeet@ubuntu: ~/Downloads$
jeet@ubuntu: ~/Downloads$ ls
anyconnect-linux-64-5.1.8.122-k9.tar.gz 'certificate.pfx' cisco-secure-client-linux64-5.1.8.122
jeet@ubuntu: ~/Downloads$ cd cisco-secure-client-linux64-5.1.8.122/vpn/
```

##### c) ./vpn\_install.sh

```
cfom.so libacwebhelper.so libvpnipsec.so vpngdownloader-cli
cisco-secure-client.directory libboost_atomic.so license.txt vpn_install.sh
cisco-secure-client.menu libboost_chrono.so load_tun.sh vpnui
cisco_secure_client_uninstall.sh libboost_date_time.so manifesttool_vpn vpn_uninstall.sh
com.cisco.secureclient.gui.desktop libboost_filesystem.so OpenSource.html
DigiCertAssuredIDRootCA.pem libboost_regex.so resources
jeet@ubuntu: ~/Downloads/cisco-secure-client-linux64-5.1.8.122/vpn$ sudo ./vpn_install.sh
```

##### d) cd ../..

```
AnyconnectForMac.XSU libacwebhelper.so libvpnipsec.so vpngdownloader-cli
cfom.so libboost_atomic.so license.txt vpn_install.sh
cisco-secure-client.directory libboost_chrono.so load_tun.sh vpnui
cisco-secure-client.menu libboost_date_time.so manifesttool_vpn vpn_uninstall.sh
cisco_secure_client_uninstall.sh libboost_filesystem.so OpenSource.html
com.cisco.secureclient.gui.desktop libboost_regex.so resources
DigiCertAssuredIDRootCA.pem
jeet@ubuntu: ~/Downloads/cisco-secure-client-linux64-5.1.8.122/vpn$ cd ..
```

##### e) systemctl status vpnapgentd

```
jeet@ubuntu: ~/Downloads$
jeet@ubuntu: ~/Downloads$
jeet@ubuntu: ~/Downloads$ ls
anyconnect-linux-64-5.1.8.122-k9.tar.gz 'certificate.pfx' cisco-secure-client-linux64-5.1.8.122
jeet@ubuntu: ~/Downloads$
jeet@ubuntu: ~/Downloads$ systemctl status vpnapgentd.service
```

## 2. Tool Installation

---


### 1.

#### Create two folders

To store the **client** and **CA** certificates, create the following directories by running the following **commands**.

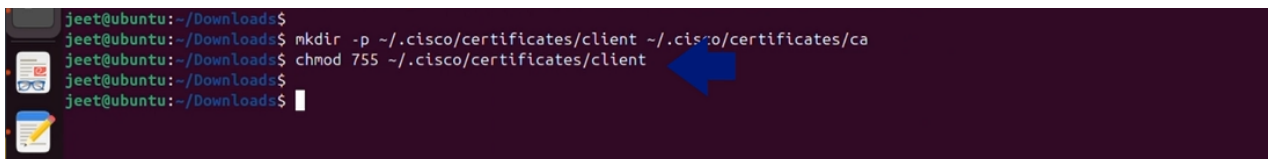
#### a) `mkdir -p ~/.cisco/certificates/client ~/.cisco/certificates/ca`

```
jeet@ubuntu: ~/Downloads$  
jeet@ubuntu: ~/Downloads$ ls  
jeet@ubuntu: ~/Downloads$ anyconnect-linux-64-5.1.8.122-k9.tar.gz certificate.pfx cisco-secure-client-linux64-5.1.8.122  
jeet@ubuntu: ~/Downloads$  
jeet@ubuntu: ~/Downloads$ mkdir -p ~/.cisco/certificates/client ~/.cisco/certificates/ca  
jeet@ubuntu: ~/Downloads$
```

A terminal window screenshot showing the execution of the command `mkdir -p ~/.cisco/certificates/client ~/.cisco/certificates/ca`. The terminal output shows the command being entered and executed successfully. A blue arrow points to the command line.

#### b) `chmod 755 ~/.cisco/certificates/client`

```
jeet@ubuntu: ~/Downloads$  
jeet@ubuntu: ~/Downloads$ mkdir -p ~/.cisco/certificates/client ~/.cisco/certificates/ca  
jeet@ubuntu: ~/Downloads$ chmod 755 ~/.cisco/certificates/client  
jeet@ubuntu: ~/Downloads$  
jeet@ubuntu: ~/Downloads$
```

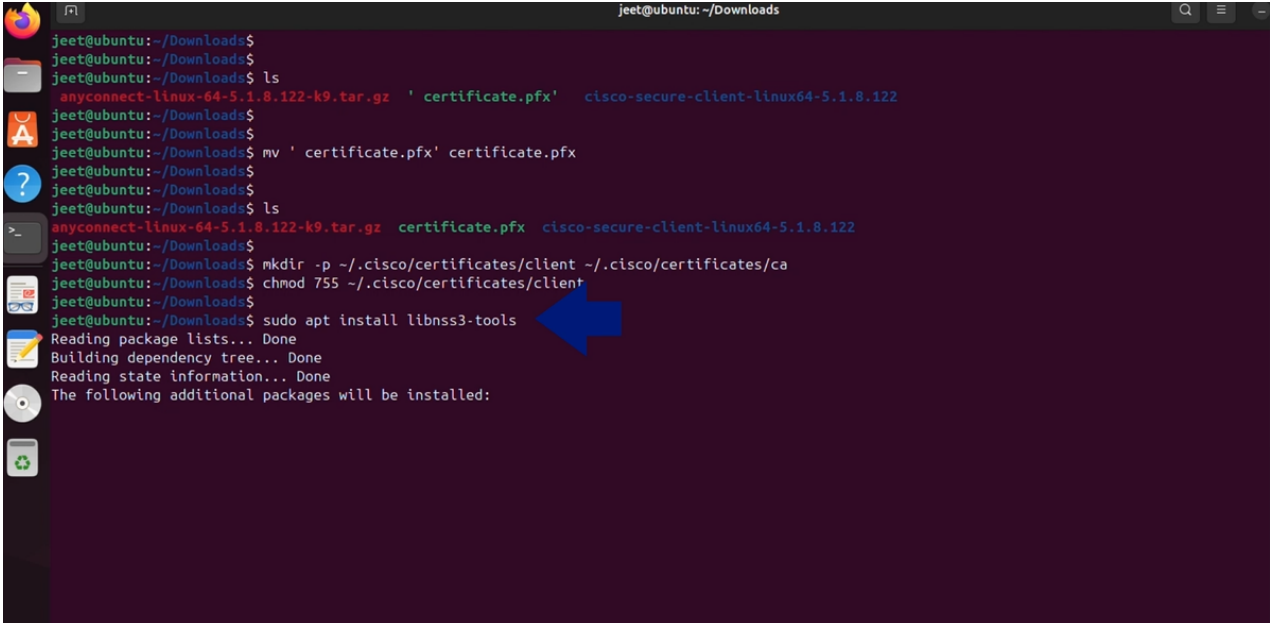
A terminal window screenshot showing the execution of the command `chmod 755 ~/.cisco/certificates/client`. The terminal output shows the command being entered and executed successfully. A blue arrow points to the command line.

## 2.

### Install required certificate tools

Install the required **certificate tools** by running the following **commands**.

a) `sudo apt install libnss3-tools`

A terminal window screenshot showing a series of commands and their outputs. The user is in the directory ~/Downloads. The commands and outputs are: 1. `ls` showing files: `anyconnect-linux-64-5.1.8.122-k9.tar.gz`, `certificate.pfx`, and `cisco-secure-client-linux64-5.1.8.122`. 2. `mv 'certificate.pfx' certificate.pfx`. 3. `ls` showing the same files. 4. `mkdir -p ~/.cisco/certificates/client ~/.cisco/certificates/ca`. 5. `chmod 755 ~/.cisco/certificates/client`. 6. `sudo apt install libnss3-tools`. The output for the last command shows: `Reading package lists... Done`, `Building dependency tree... Done`, `Reading state information... Done`, and `The following additional packages will be installed:`. A blue arrow points to the `sudo apt install libnss3-tools` command.

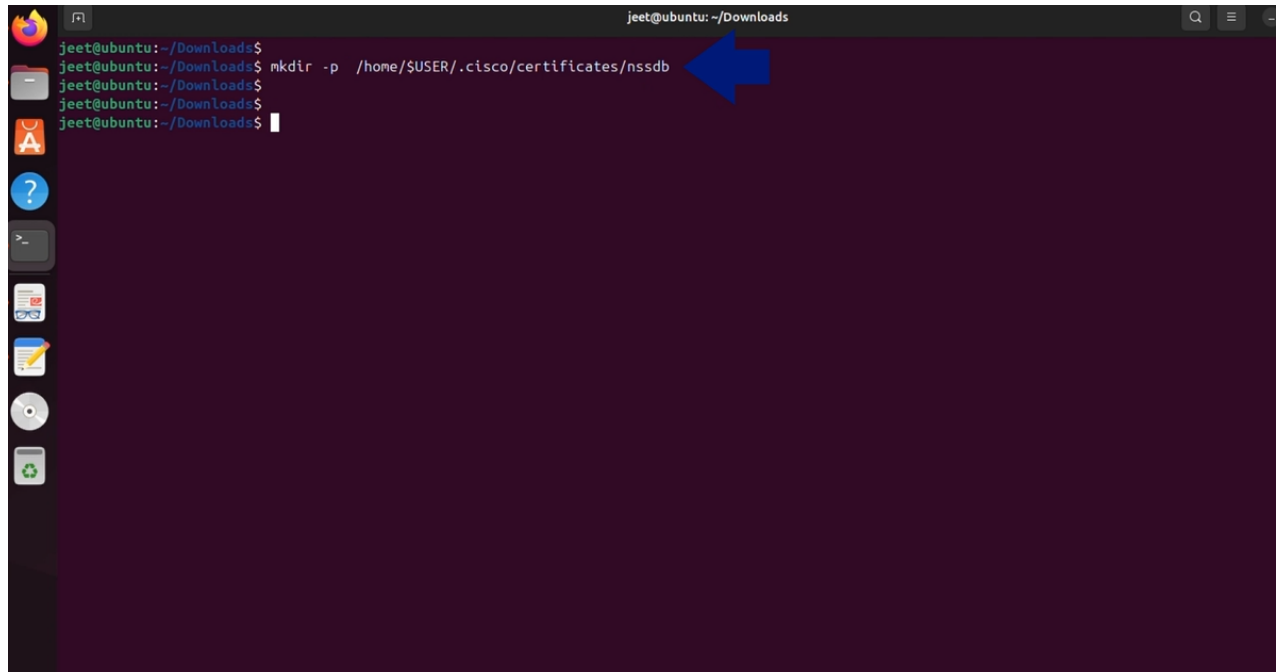
```
jeet@ubuntu: ~/Downloads
jeet@ubuntu:~/Downloads$ 
jeet@ubuntu:~/Downloads$ 
jeet@ubuntu:~/Downloads$ ls
anyconnect-linux-64-5.1.8.122-k9.tar.gz  'certificate.pfx'  cisco-secure-client-linux64-5.1.8.122
jeet@ubuntu:~/Downloads$ 
jeet@ubuntu:~/Downloads$ mv 'certificate.pfx' certificate.pfx
jeet@ubuntu:~/Downloads$ 
jeet@ubuntu:~/Downloads$ 
jeet@ubuntu:~/Downloads$ ls
anyconnect-linux-64-5.1.8.122-k9.tar.gz  certificate.pfx  cisco-secure-client-linux64-5.1.8.122
jeet@ubuntu:~/Downloads$ 
jeet@ubuntu:~/Downloads$ mkdir -p ~/.cisco/certificates/client ~/.cisco/certificates/ca
jeet@ubuntu:~/Downloads$ chmod 755 ~/.cisco/certificates/client
jeet@ubuntu:~/Downloads$ 
jeet@ubuntu:~/Downloads$ sudo apt install libnss3-tools
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
```

### 3.

#### Create a directory

To store the **NSS** database, create the following directory by running the following **command**.

```
a) mkdir -p /home/$USER/.cisco/certificates/nssdb
```



A terminal window screenshot showing the execution of the command to create the directory. The terminal title is "jeet@ubuntu: ~/Downloads". The command "mkdir -p /home/\$USER/.cisco/certificates/nssdb" is entered and executed. A blue arrow points to the command line. The terminal output shows the command being executed and the prompt returning.

```
jeet@ubuntu: ~/Downloads$  
jeet@ubuntu: ~/Downloads$ mkdir -p /home/$USER/.cisco/certificates/nssdb  
jeet@ubuntu: ~/Downloads$  
jeet@ubuntu: ~/Downloads$  
jeet@ubuntu: ~/Downloads$
```

### 3. Import Certificate (DSC)

---

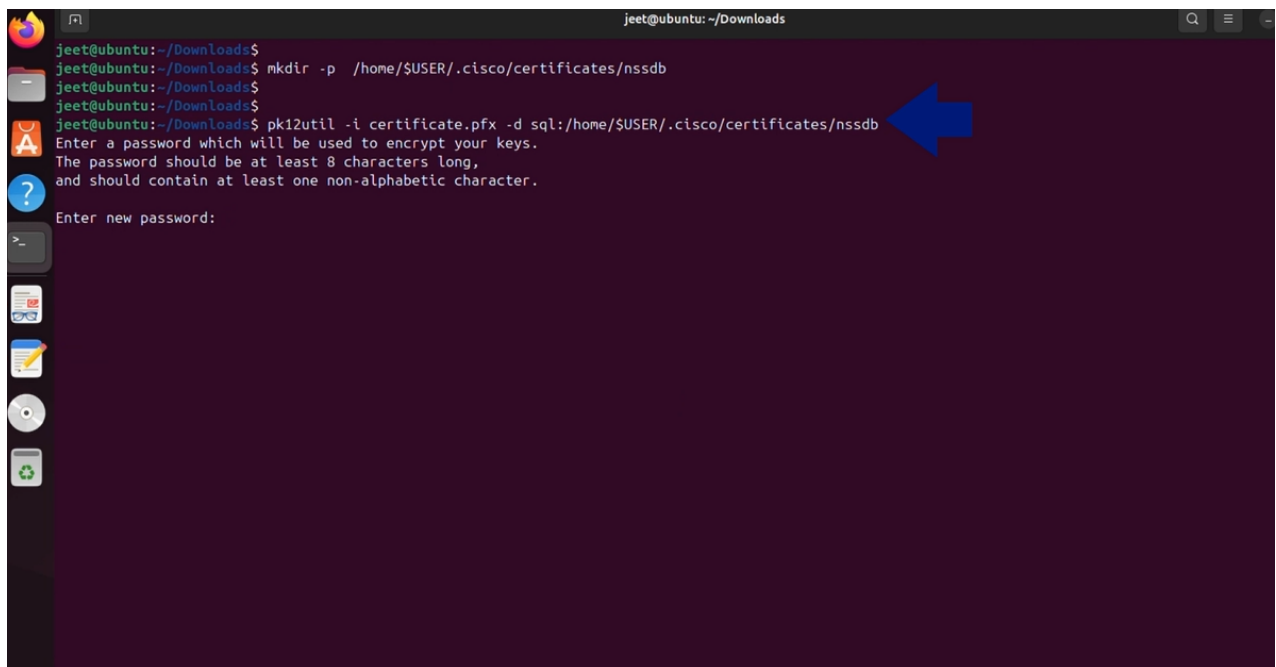
#### 1.

#### Import Certificate to NSS Database

Import the **certificate** to the **NSS database** by running the following command.

```
a) pk12util -i certificate.pfx -d sql:/home/$USER/.cisco/certificates/nssdb
```

**Note:** The first password is of your choice for the certificate store, you can enter the password of your choice. Whereas the second password is the private key share with you on your registered contact number.



```
jeet@ubuntu: ~/Downloads
jeet@ubuntu:~/Downloads$ mkdir -p /home/$USER/.cisco/certificates/nssdb
jeet@ubuntu:~/Downloads$
jeet@ubuntu:~/Downloads$ pk12util -i certificate.pfx -d sql:/home/$USER/.cisco/certificates/nssdb
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.
Enter new password:
```



### 3.

#### Extract and Secure PEM Certificate

Extract the **PEM certificate** and move it to the **client certificate directory** by running the following commands.

##### a) `openssl pkcs12 -in certificate.pfx -out certificate_name.pem -nodes -clcerts -legacy`

```
le-3462cc5c-8790-4130-a65e-1adbbcd96b47          Trust Attributes
vpncsa - in                                         SSL,S/MIME,JAR/XPI
jeet@ubuntu:~/Downloads$ openssl pkcs12 -in certificate.pfx -out certificate_name.pem -nodes -clcerts -legacy
```

##### b) `mv certificate_name.pem ~/.cisco/certificates/client/`

```
jeet@ubuntu:~/Downloads$ openssl pkcs12 -in certificate.pfx -out certificate_name.pem -nodes -clcerts -legacy
Enter Import Password:
jeet@ubuntu:~/Downloads$ mv certificate_name.pem ~/.cisco/certificates/client/
```

##### c) `chmod 700 ~/.cisco/certificates/client`

```
jeet@ubuntu:~/Downloads$ mv certificate_name.pem ~/.cisco/certificates/client/
jeet@ubuntu:~/Downloads$ chmod 700 ~/.cisco/certificates/client
```

##### d) `chmod 600 ~/.cisco/certificates/client/certificate_name.pem`

```
jeet@ubuntu:~/Downloads$ mv certificate_name.pem ~/.cisco/certificates/client/
jeet@ubuntu:~/Downloads$ chmod 700 ~/.cisco/certificates/client
jeet@ubuntu:~/Downloads$ chmod 600 ~/.cisco/certificates/client/certificate_name.pem
```

**Note:** Enter your private key after running first command.

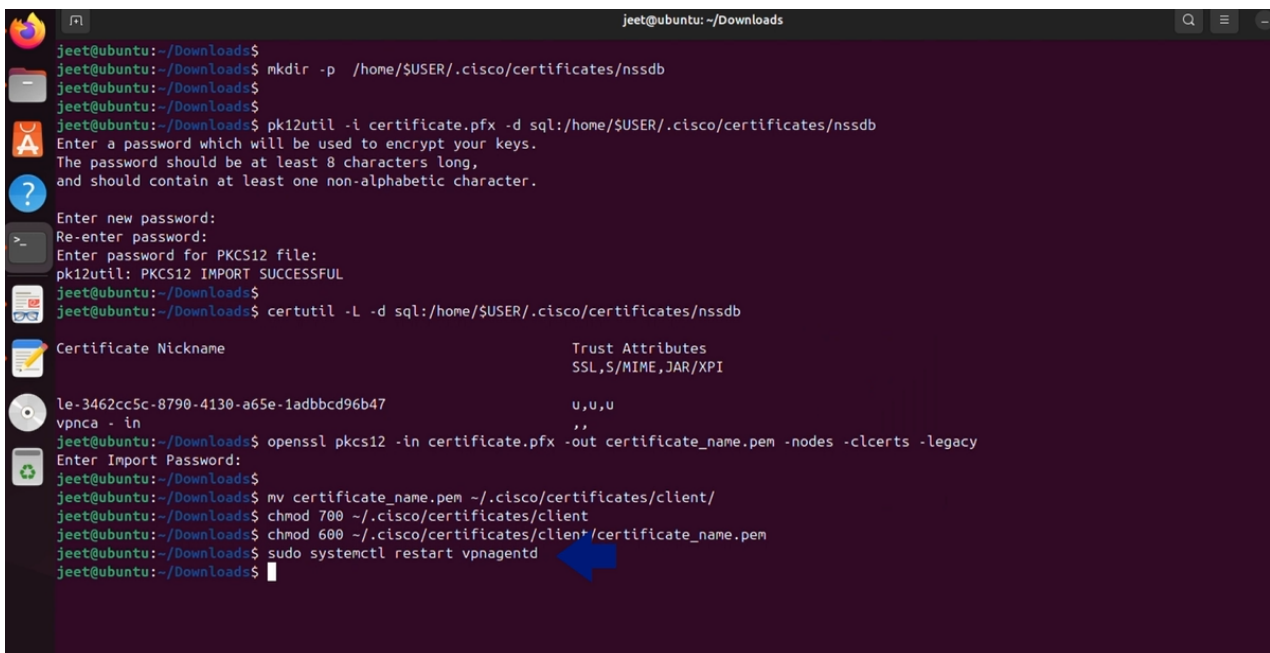
## 4. Connect to VPN

### 1.

#### Restart the certificate

Restart the **Certificate** by running the following **command**.

a) `sudo systemctl restart vpnapgentd`



```
jeet@ubuntu: ~/Downloads
jeet@ubuntu:~/Downloads$ mkdir -p /home/$USER/.cisco/certificates/nssdb
jeet@ubuntu:~/Downloads$
jeet@ubuntu:~/Downloads$ pk12util -i certificate.pfx -d sql:/home/$USER/.cisco/certificates/nssdb
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.
Enter new password:
Re-enter password:
Enter password for PKCS12 file:
pk12util: PKCS12 IMPORT SUCCESSFUL
jeet@ubuntu:~/Downloads$
jeet@ubuntu:~/Downloads$ certutil -L -d sql:/home/$USER/.cisco/certificates/nssdb
Certificate Nickname                               Trust Attributes
SSL,S/MIME,JAR/XPI

le-3462cc5c-8790-4130-a65e-1adbbcd96b47           u,u,u
vpncsa - in                                       ''
jeet@ubuntu:~/Downloads$ openssl pkcs12 -in certificate.pfx -out certificate_name.pem -nodes -clcerts -legacy
Enter Import Password:
jeet@ubuntu:~/Downloads$
jeet@ubuntu:~/Downloads$ mv certificate_name.pem ~/.cisco/certificates/client/
jeet@ubuntu:~/Downloads$ chmod 700 ~/.cisco/certificates/client
jeet@ubuntu:~/Downloads$ chmod 600 ~/.cisco/certificates/client*/certificate_name.pem
jeet@ubuntu:~/Downloads$ sudo systemctl restart vpnapgentd
jeet@ubuntu:~/Downloads$
```

## 2.

### Connect to Cisco Secure Client

Connect to the **Cisco Secure Client**. Enter **sconnect.nic.in** as the server address and enter your **username** and **password**.

