National
Supercomputing
Centre

Linux SSL VPN (Checkpoint) Client Guide

Version-05-05-2025

Disclaimer:-

Any links to third-party software available on this guide are provided “as is” without warranty of any kind, either expressed or implied and such software is to be used at your own risk.

The use of the third-party software links on this guide is done at your own discretion and risk and with agreement that you will be solely responsible for any damage to your computer system or loss of data that results from such activities.

You are solely responsible for adequate protection and backup of the data and equipment used in connection with any of the software linked to this guide, and we will not be liable for any damages that you may suffer connection with downloading, installing, using, modifying or distributing such software.

No advice or information, whether oral or written, obtained by you from us or from this guide shall create any warranty for the software.

Additionally, we make no warranty that:

- The third-party software will meet your requirements.
- The third-party software will be uninterrupted, timely, secure or error-free.
- The results from the use of the third-party software will be effective, accurate or reliable.
- The quality of the third-party software will meet your expectations.
- If errors or problems occur in connection with a download of the third-party software obtained from the links on this document, they will be corrected.

This guide covers checkpoint SSL VPN client setup for Linux OS. Recommended and tested OS in this document :

- Ubuntu 22.04 LTS
- Ubuntu 20.04 LTS

Although steps in this document are based on Ubuntu 20.04 LTS and Ubuntu 22.04 LTS, similar steps should be applicable for other Linux Distro.

There are two methods to setup SSL VPN client:-

1. Manual Installation.
2. Using ChrootVPN – a 3rd-party wrapper for Checkpoint R80 VPN client.

Requirements:

- Terminal
- Web browser
- Sudo privilege

First Method: Manual Installation

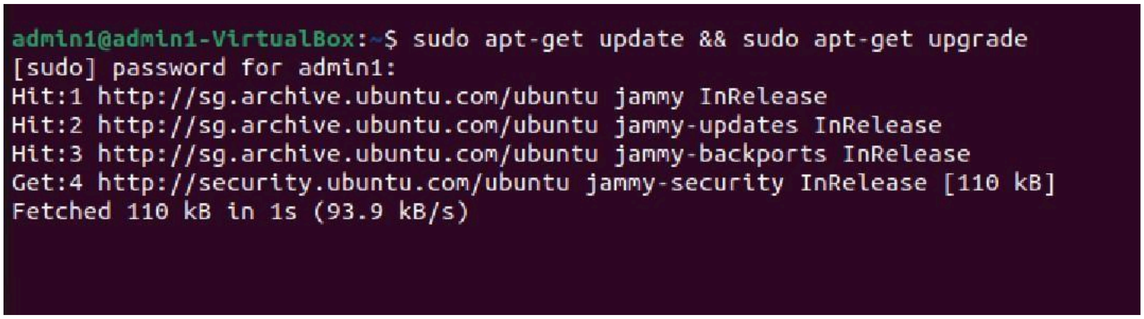
This method consists of few parts as below:

1. Install Prerequisite (Install & update ubuntu packages).
2. Install Firefox
3. Install Java
4. Install SSL network extender and its dependencies.
5. Post Installation.
6. Connect to ASPIRE2A+ VPN via Checkpoint Mobile Access Portal.

1. Install Prerequisite

a. Update the repo.

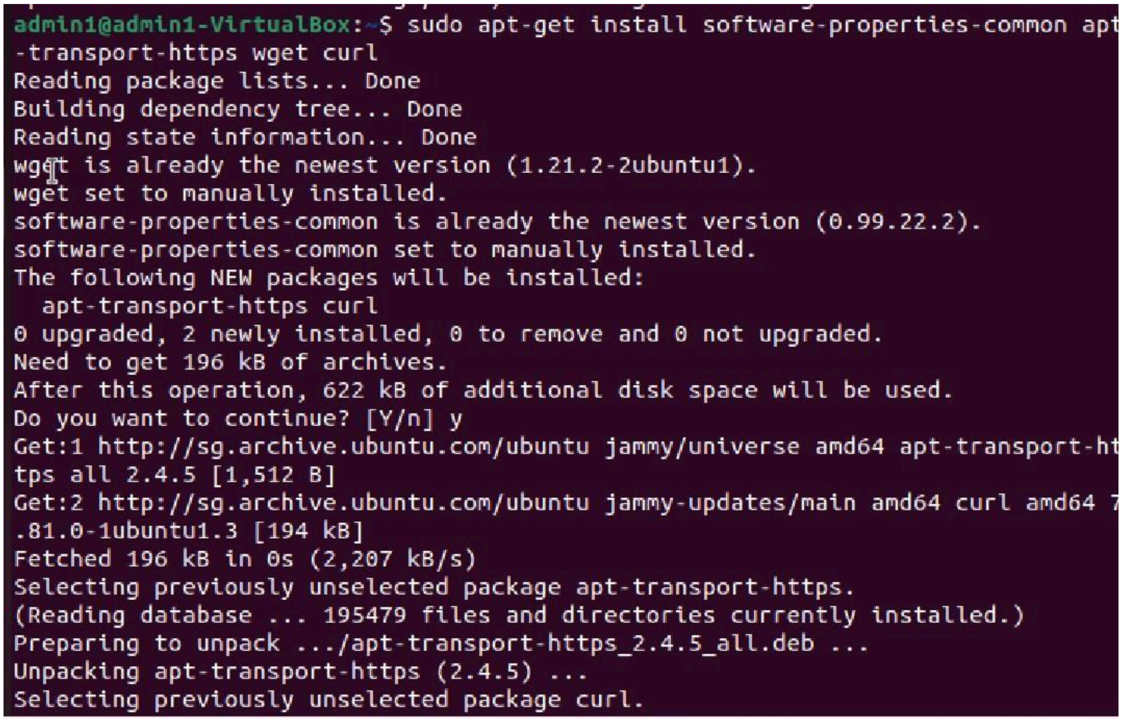
```
$ sudo apt-get update && sudo apt-get upgrade
```



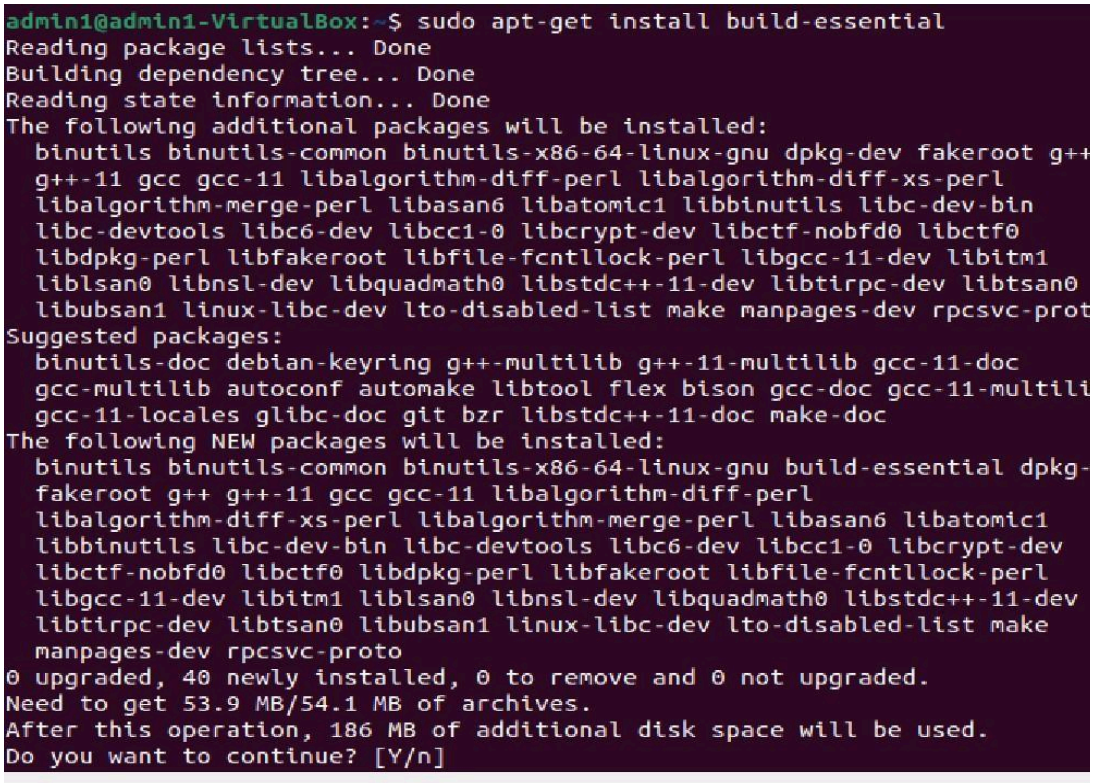
```
admin1@admin1-VirtualBox:~$ sudo apt-get update && sudo apt-get upgrade
[sudo] password for admin1:
Hit:1 http://sg.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://sg.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://sg.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Fetched 110 kB in 1s (93.9 kB/s)
```

b. Install development package

```
$ sudo apt-get install software-properties-common apt-transport-https
wget curl build-essential
```



```
admin1@admin1-VirtualBox:~$ sudo apt-get install software-properties-common apt
-transport-https wget curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
wget is already the newest version (1.21.2-2ubuntu1).
wget set to manually installed.
software-properties-common is already the newest version (0.99.22.2).
software-properties-common set to manually installed.
The following NEW packages will be installed:
  apt-transport-https curl
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 196 kB of archives.
After this operation, 622 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://sg.archive.ubuntu.com/ubuntu jammy/universe amd64 apt-transport-ht
tps all 2.4.5 [1,512 B]
Get:2 http://sg.archive.ubuntu.com/ubuntu jammy-updates/main amd64 curl amd64 7
.81.0-1ubuntu1.3 [194 kB]
Fetched 196 kB in 0s (2,207 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 195479 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.4.5_all.deb ...
Unpacking apt-transport-https (2.4.5) ...
Selecting previously unselected package curl.
```



```
admin1@admin1-VirtualBox:~$ sudo apt-get install build-essential
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu dpkg-dev fakeroot g++
  g++-11 gcc gcc-11 libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan6 libatomic1 libbinutils libc-dev-bin
  libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0
  libdpkg-perl libfakeroot libfile-fcntllock-perl libgcc-11-dev libitm1
  liblsan0 libnsl-dev libquadmath0 libstdc++-11-dev libtirpc-dev libtsan0
  libubsan1 linux-libc-dev lto-disabled-list make manpages-dev rpcsvc-prot
Suggested packages:
  binutils-doc debian-keyring g++-multilib g++-11-multilib gcc-11-doc
  gcc-multilib autoconf automake libtool flex bison gcc-doc gcc-11-multili
  gcc-11-locales glibc-doc git bzr libstdc++-11-doc make-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-
  fakeroot g++ g++-11 gcc gcc-11 libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6 libatomic1
  libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev
  libctf-nobfd0 libctf0 libdpkg-perl libfakeroot libfile-fcntllock-perl
  libgcc-11-dev libitm1 liblsan0 libnsl-dev libquadmath0 libstdc++-11-dev
  libtirpc-dev libtsan0 libubsan1 linux-libc-dev lto-disabled-list make
  manpages-dev rpcsvc-proto
0 upgraded, 40 newly installed, 0 to remove and 0 not upgraded.
Need to get 53.9 MB/54.1 MB of archives.
After this operation, 186 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

2. Install Firefox

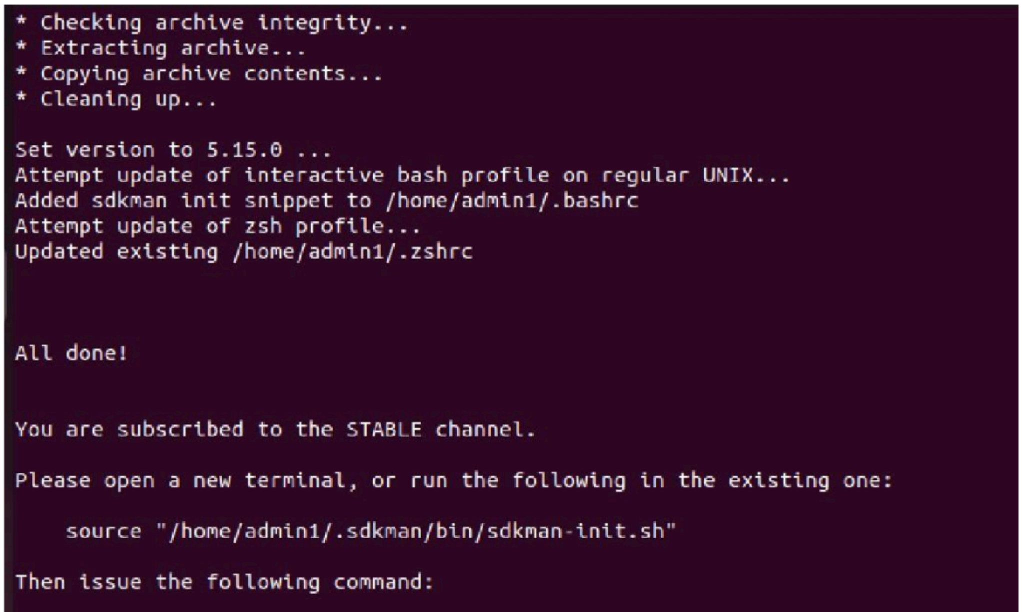
By default, most Linux OS come with firefox pre-installed, if firefox is not installed in your OS, you may install it using the command below:

```
$ sudo apt-get install firefox
```

3. Install Java

a. Install SDKMAN! as a Java repository.

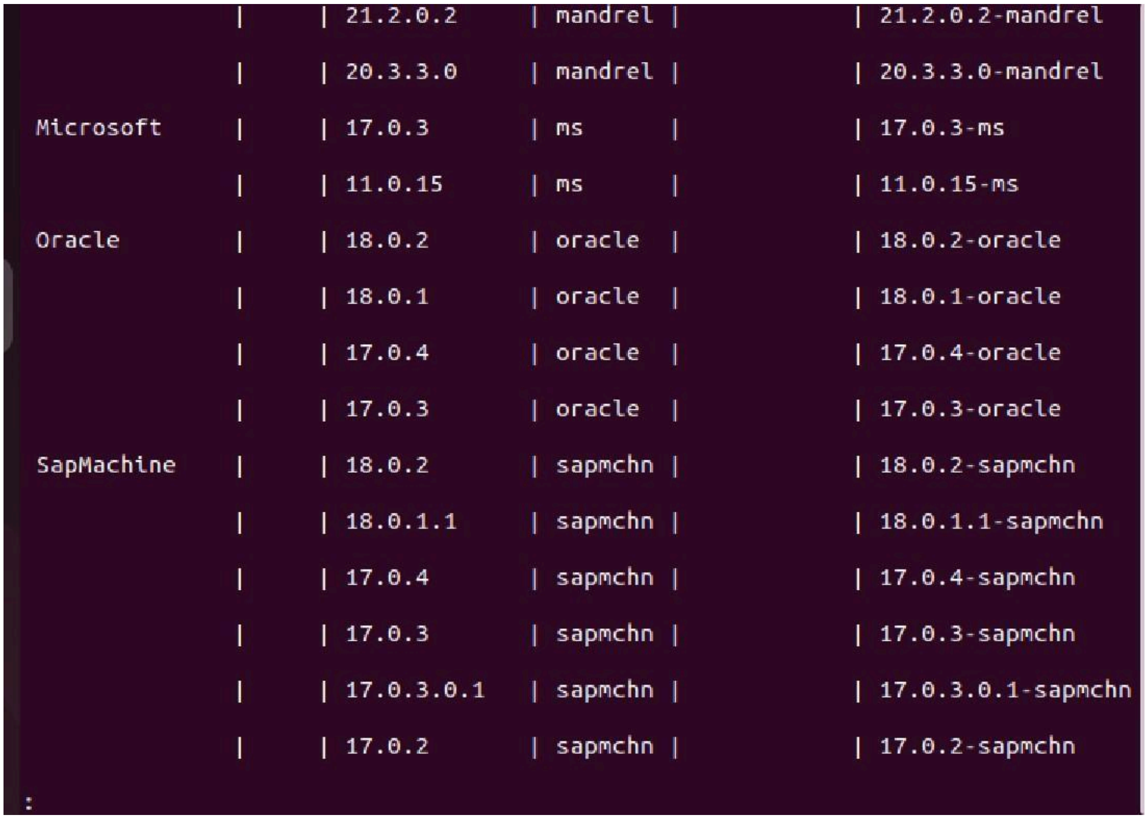
```
$ cd ~
$ curl -s "https://get.sdkman.io" | bash
$ source "$HOME/.sdkman/bin/sdkman-init.sh"
```



```
* Checking archive integrity...
* Extracting archive...
* Copying archive contents...
* Cleaning up...

Set version to 5.15.0 ...
Attempt update of interactive bash profile on regular UNIX...
Added sdkman init snippet to /home/admin1/.bashrc
Attempt update of zsh profile...
Updated existing /home/admin1/.zshrc



All done!


You are subscribed to the STABLE channel.

Please open a new terminal, or run the following in the existing one:

    source "/home/admin1/.sdkman/bin/sdkman-init.sh"

Then issue the following command:
```

b. List the available Java versions and select the Java oracle version.

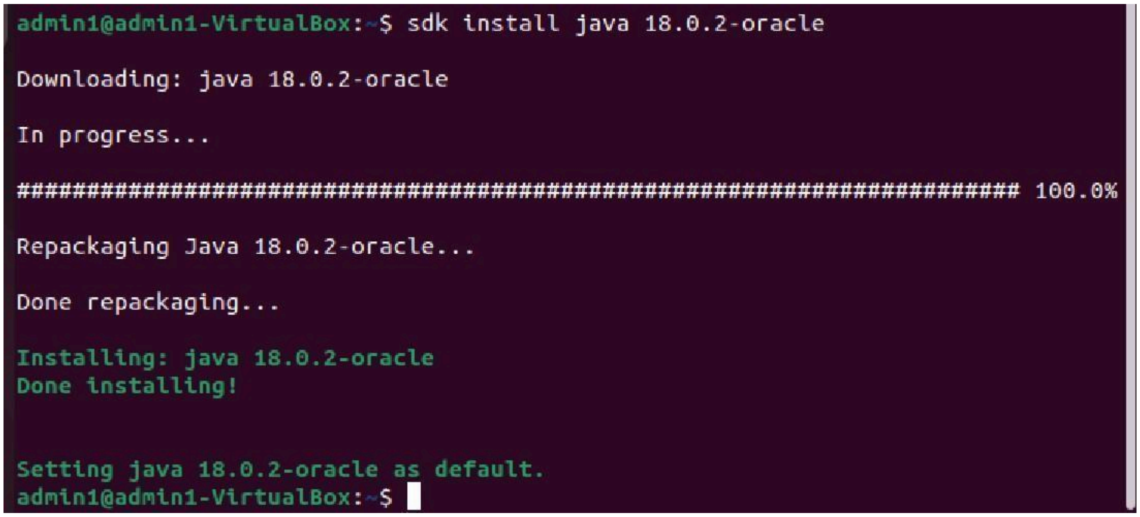
```
$ sdk list java
```



```
               |     | 21.2.0.2     | mandrel |            | 21.2.0.2-mandrel

               |     | 20.3.3.0     | mandrel |            | 20.3.3.0-mandrel

 Microsoft     |     | 17.0.3       | ms      |            | 17.0.3-ms

               |     | 11.0.15      | ms      |            | 11.0.15-ms

 Oracle        |     | 18.0.2       | oracle  |            | 18.0.2-oracle

               |     | 18.0.1       | oracle  |            | 18.0.1-oracle

               |     | 17.0.4       | oracle  |            | 17.0.4-oracle

               |     | 17.0.3       | oracle  |            | 17.0.3-oracle

 SapMachine    |     | 18.0.2       | sapmchn |            | 18.0.2-sapmchn

               |     | 18.0.1.1     | sapmchn |            | 18.0.1.1-sapmchn

               |     | 17.0.4       | sapmchn |            | 17.0.4-sapmchn

               |     | 17.0.3       | sapmchn |            | 17.0.3-sapmchn

               |     | 17.0.3.0.1   | sapmchn |            | 17.0.3.0.1-sapmchn

               |     | 17.0.2       | sapmchn |            | 17.0.2-sapmchn

:
```

```
PRESS Q TO QUIT
```

```
$ sdk install java 18.0.2-oracle
```

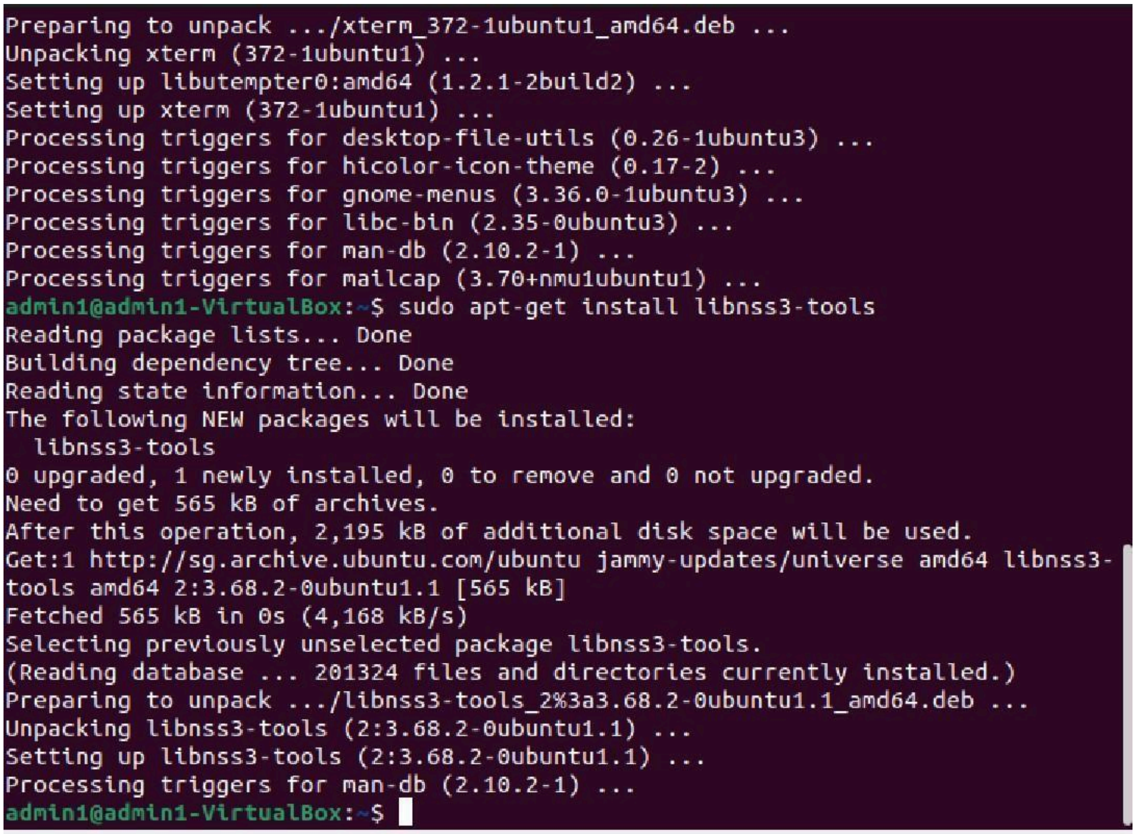


```
admin1@admin1-VirtualBox:~$ sdk install java 18.0.2-oracle

Downloading: java 18.0.2-oracle

In progress...

######################################################################## 100.0%

Repackaging Java 18.0.2-oracle...

Done repackaging...

Installing: java 18.0.2-oracle
Done installing!


Setting java 18.0.2-oracle as default.
admin1@admin1-VirtualBox:~$
```

4. Install SSL Network Extender and its dependencies

a. Install xterm and libnss3-tools

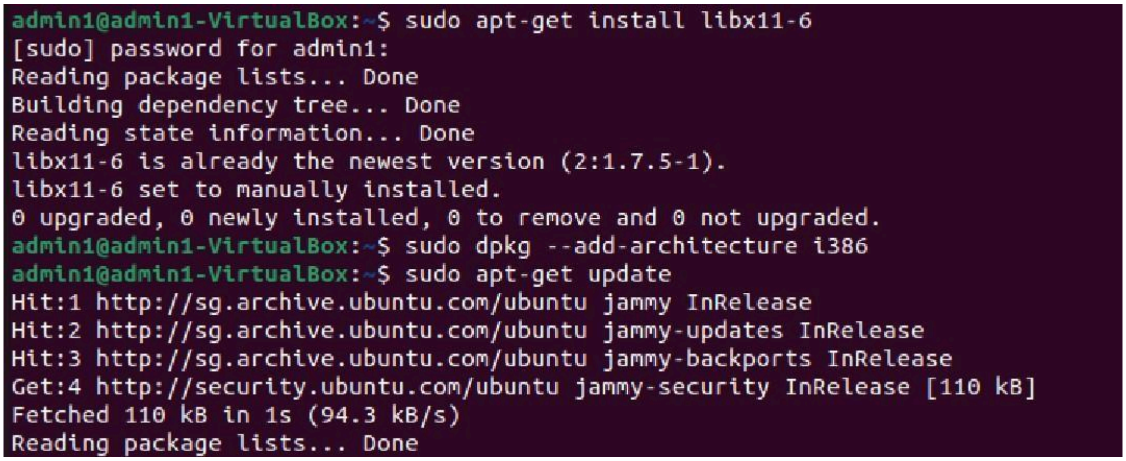
```
$ sudo apt-get install xterm libnss3-tools
```



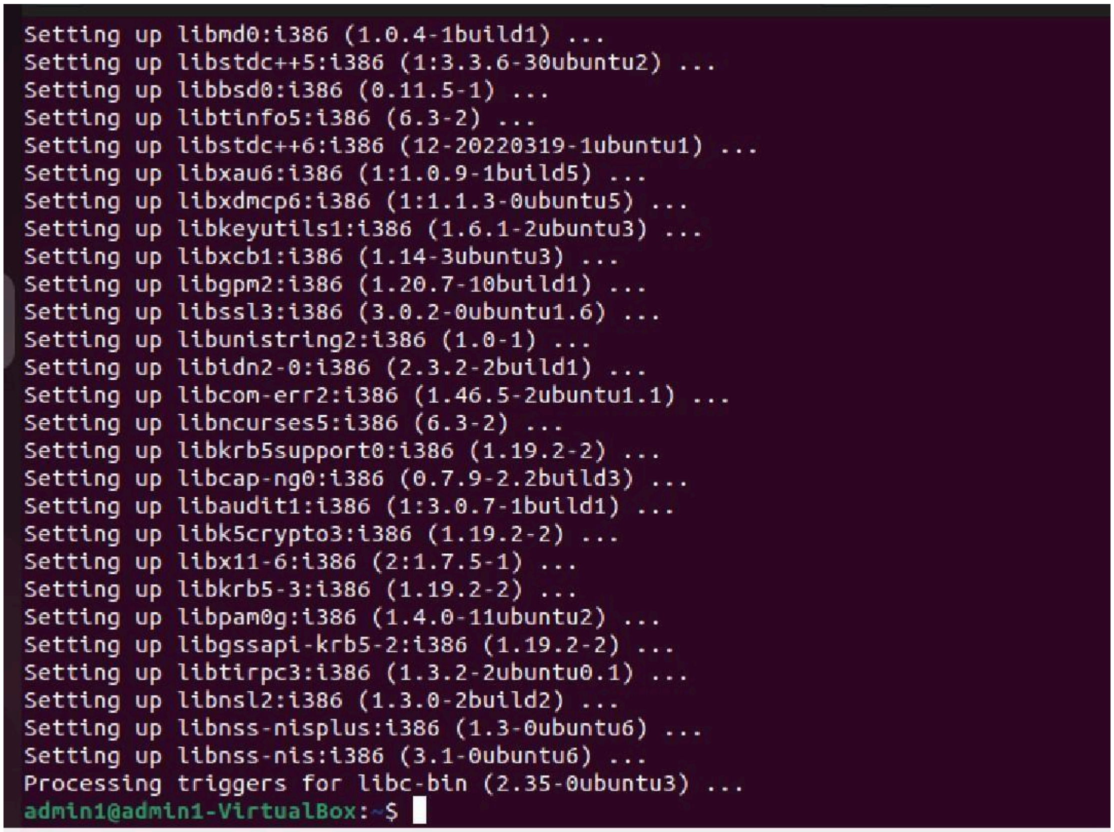
```
Preparing to unpack .../xterm_372-1ubuntu1_amd64.deb ...
Unpacking xterm (372-1ubuntu1) ...
Setting up libutempter0:amd64 (1.2.1-2build2) ...
Setting up xterm (372-1ubuntu1) ...
Processing triggers for desktop-file-utils (0.26-1ubuntu3) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu3) ...
Processing triggers for libc-bin (2.35-0ubuntu3) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for mailcap (3.70+nmu1ubuntu1) ...
admin1@admin1-VirtualBox:~$ sudo apt-get install libnss3-tools
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  libnss3-tools
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 565 kB of archives.
After this operation, 2,195 kB of additional disk space will be used.
Get:1 http://sg.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 libnss3-
tools amd64 2:3.68.2-0ubuntu1.1 [565 kB]
Fetched 565 kB in 0s (4,168 kB/s)
Selecting previously unselected package libnss3-tools.
(Reading database ... 201324 files and directories currently installed.)
Preparing to unpack .../libnss3-tools_2%3a3.68.2-0ubuntu1.1_amd64.deb ...
Unpacking libnss3-tools (2:3.68.2-0ubuntu1.1) ...
Setting up libnss3-tools (2:3.68.2-0ubuntu1.1) ...
Processing triggers for man-db (2.10.2-1) ...
admin1@admin1-VirtualBox:~$
```

b. Install 32-bit compatible libraries.

```
$ sudo apt-get install libx11-6
$ sudo dpkg --add-architecture i386
$ sudo apt-get update $ sudo apt-get install libx11-6:i386
libc6:i386 libncurses5:i386 libstdc++6:i386 libstdc++5:i386
libpam0g:i386
```



```
admin1@admin1-VirtualBox:~$ sudo apt-get install libx11-6
[sudo] password for admin1:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libx11-6 is already the newest version (2:1.7.5-1).
libx11-6 set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
admin1@admin1-VirtualBox:~$ sudo dpkg --add-architecture i386
admin1@admin1-VirtualBox:~$ sudo apt-get update
Hit:1 http://sg.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://sg.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://sg.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Fetched 110 kB in 1s (94.3 kB/s)
Reading package lists... Done
```



```
Setting up libmd0:i386 (1.0.4-1build1) ...
Setting up libstdc++5:i386 (1:3.3.6-30ubuntu2) ...
Setting up libbsd0:i386 (0.11.5-1) ...
Setting up libtinfo5:i386 (6.3-2) ...
Setting up libstdc++6:i386 (12-20220319-1ubuntu1) ...
Setting up libxau6:i386 (1:1.0.9-1build5) ...
Setting up libxdmcp6:i386 (1:1.1.3-0ubuntu5) ...
Setting up libkeyutils1:i386 (1.6.1-2ubuntu3) ...
Setting up libxcb1:i386 (1.14-3ubuntu3) ...
Setting up libgpm2:i386 (1.20.7-10build1) ...
Setting up libssl3:i386 (3.0.2-0ubuntu1.6) ...
Setting up libunistring2:i386 (1.0-1) ...
Setting up libidn2-0:i386 (2.3.2-2build1) ...
Setting up libcom-err2:i386 (1.46.5-2ubuntu1.1) ...
Setting up libncurses5:i386 (6.3-2) ...
Setting up libkrb5support0:i386 (1.19.2-2) ...
Setting up libcap-ng0:i386 (0.7.9-2.2build3) ...
Setting up libaudit1:i386 (1:3.0.7-1build1) ...
Setting up libk5crypto3:i386 (1.19.2-2) ...
Setting up libx11-6:i386 (2:1.7.5-1) ...
Setting up libkrb5-3:i386 (1.19.2-2) ...
Setting up libpam0g:i386 (1.4.0-11ubuntu2) ...
Setting up libgssapi-krb5-2:i386 (1.19.2-2) ...
Setting up libtirpc3:i386 (1.3.2-2ubuntu0.1) ...
Setting up libnsl2:i386 (1.3.0-2build2) ...
Setting up libnss-nisplus:i386 (1.3-0ubuntu6) ...
Setting up libnss-nis:i386 (3.1-0ubuntu6) ...
Processing triggers for libc-bin (2.35-0ubuntu3) ...
admin1@admin1-VirtualBox:~$
```

c. Download and install SSL network extender and cshell. This step requires sudo privilege.

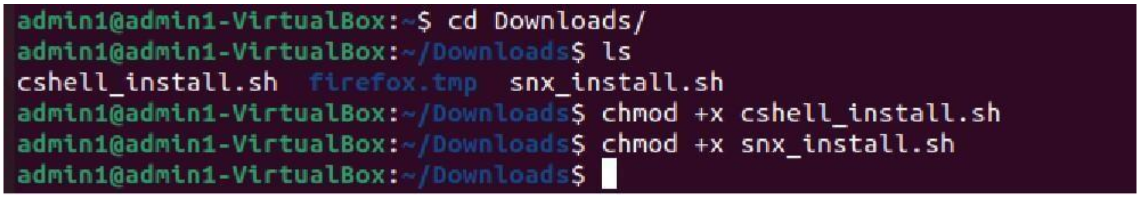
```
$ wget --no-check-certificate
```

https://vpn1.nscc.sg/sslvpn/SNX/INSTALL/snx_install.sh

```
$ wget --no-check-certificate
```

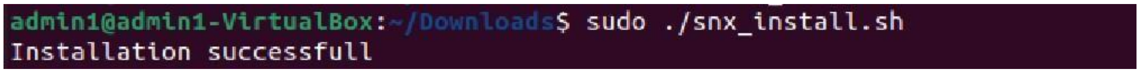
https://vpn1.nscc.sg/sslvpn/SNX/INSTALL/cshell_install.sh

```
$ chmod +x snx_install.sh
$ chmod +x cshell_install.sh
```



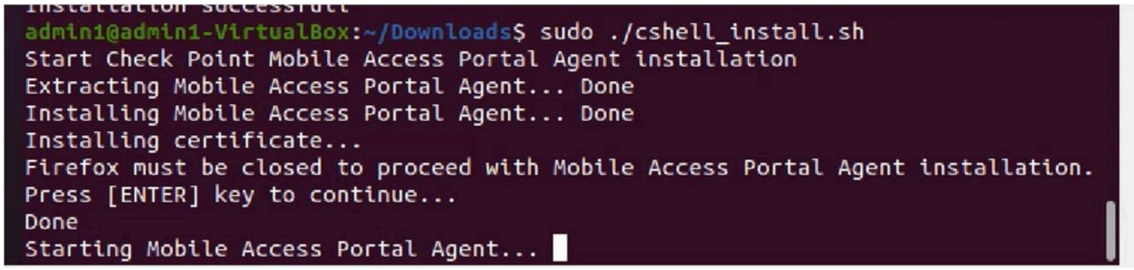
```
admin1@admin1-VirtualBox:~$ cd Downloads/
admin1@admin1-VirtualBox:~/Downloads$ ls
cshell_install.sh  firefox.tmp  snx_install.sh
admin1@admin1-VirtualBox:~/Downloads$ chmod +x cshell_install.sh
admin1@admin1-VirtualBox:~/Downloads$ chmod +x snx_install.sh
admin1@admin1-VirtualBox:~/Downloads$
```

```
$ sudo ./snx_install.sh
```



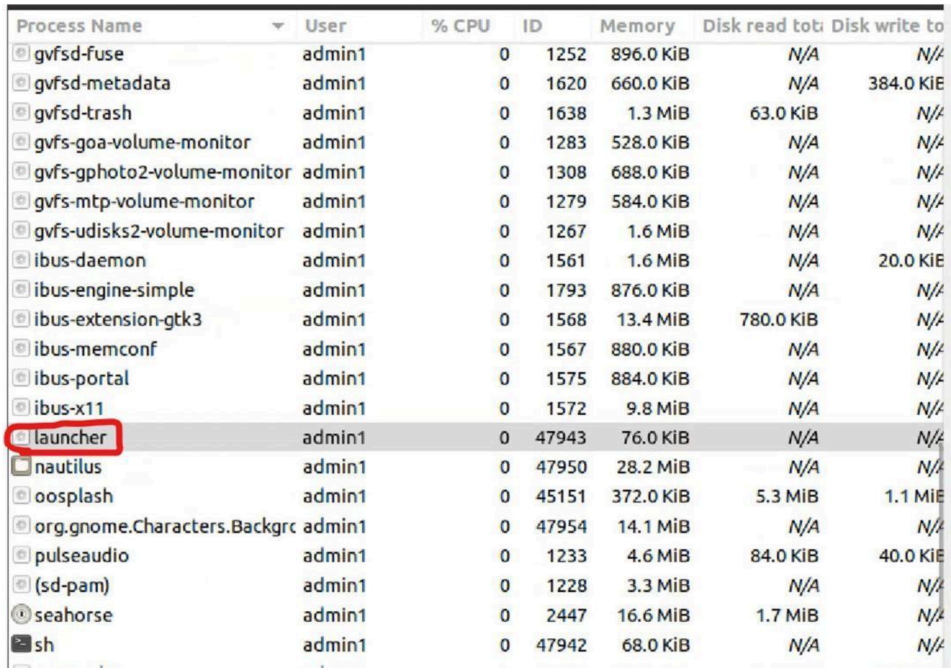
```
admin1@admin1-VirtualBox:~/Downloads$ sudo ./snx_install.sh
Installation successfull
```

```
$ sudo ./cshell_install.sh
```



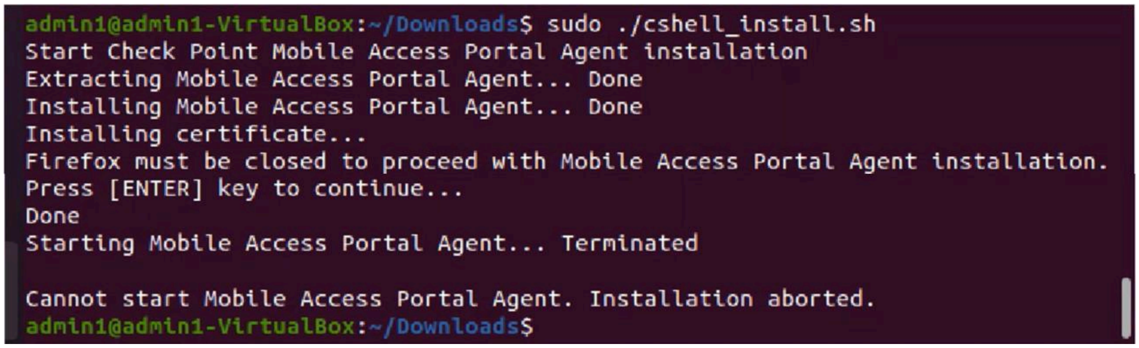
```
Installation successful
admin1@admin1-VirtualBox:~/Downloads$ sudo ./cshell_install.sh
Start Check Point Mobile Access Portal Agent installation
Extracting Mobile Access Portal Agent... Done
Installing Mobile Access Portal Agent... Done
Installing certificate...
Firefox must be closed to proceed with Mobile Access Portal Agent installation.
Press [ENTER] key to continue...
Done
Starting Mobile Access Portal Agent...
```

d. It is expected that it will be stuck at “**Starting Mobile Access Portal Agent.**” Please do not use CTRL+C. Open the **“System Monitor**”, find the service name ”**launcher**” and kill it.



Process Name	User	% CPU	ID	Memory	Disk read tot	Disk write to
gvfsd-fuse	admin1	0	1252	896.0 KiB	N/A	N/A
gvfsd-metadata	admin1	0	1620	660.0 KiB	N/A	384.0 KiB
gvfsd-trash	admin1	0	1638	1.3 MiB	63.0 KiB	N/A
gvfs-goa-volume-monitor	admin1	0	1283	528.0 KiB	N/A	N/A
gvfs-gphoto2-volume-monitor	admin1	0	1308	688.0 KiB	N/A	N/A
gvfs-mtp-volume-monitor	admin1	0	1279	584.0 KiB	N/A	N/A
gvfs-udisks2-volume-monitor	admin1	0	1267	1.6 MiB	N/A	N/A
ibus-daemon	admin1	0	1561	1.6 MiB	N/A	20.0 KiB
ibus-engine-simple	admin1	0	1793	876.0 KiB	N/A	N/A
ibus-extension-gtk3	admin1	0	1568	13.4 MiB	780.0 KiB	N/A
ibus-memconf	admin1	0	1567	880.0 KiB	N/A	N/A
ibus-portal	admin1	0	1575	884.0 KiB	N/A	N/A
ibus-x11	admin1	0	1572	9.8 MiB	N/A	N/A
launcher	admin1	0	47943	76.0 KiB	N/A	N/A
nautilus	admin1	0	47950	28.2 MiB	N/A	N/A
oosplash	admin1	0	45151	372.0 KiB	5.3 MiB	1.1 MiB
org.gnome.Characters.Backgrc	admin1	0	47954	14.1 MiB	N/A	N/A
pulseaudio	admin1	0	1233	4.6 MiB	84.0 KiB	40.0 KiB
(sd-pam)	admin1	0	1228	3.3 MiB	N/A	N/A
seahorse	admin1	0	2447	16.6 MiB	1.7 MiB	N/A
sh	admin1	0	47942	68.0 KiB	N/A	N/A

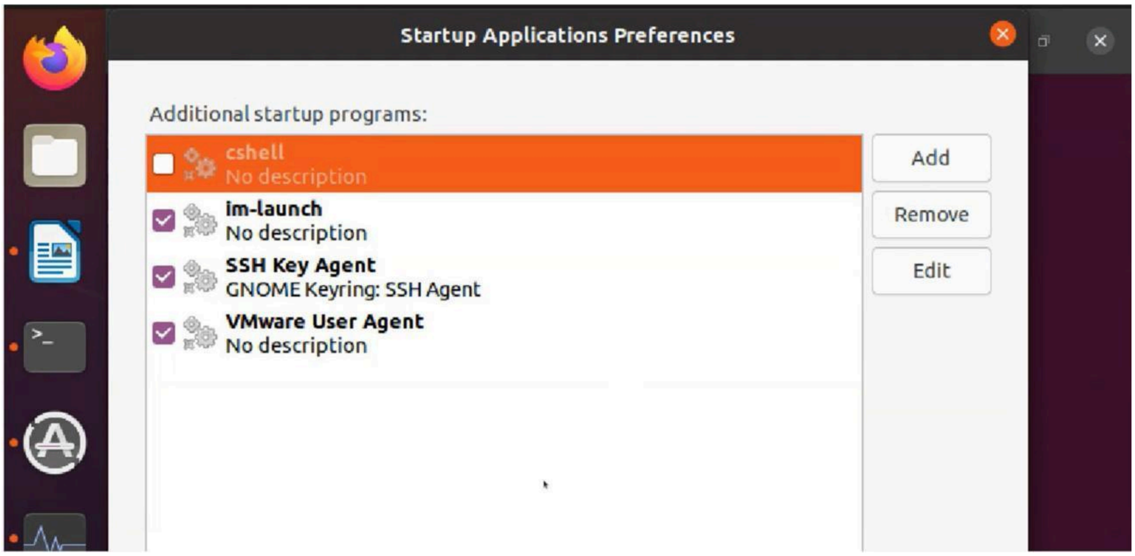
e. Back to the terminal, the **Mobile Access Portal Agent** is terminated as shown below.



```
admin1@admin1-VirtualBox:~/Downloads$ sudo ./cshell_install.sh
Start Check Point Mobile Access Portal Agent installation
Extracting Mobile Access Portal Agent... Done
Installing Mobile Access Portal Agent... Done
Installing certificate...
Firefox must be closed to proceed with Mobile Access Portal Agent installation.
Press [ENTER] key to continue...
Done
Starting Mobile Access Portal Agent... Terminated

Cannot start Mobile Access Portal Agent. Installation aborted.
admin1@admin1-VirtualBox:~/Downloads$
```

5. Post Installation

Open “**Startup Application Preferences**", uncheck the cshell.



Additional startup programs:

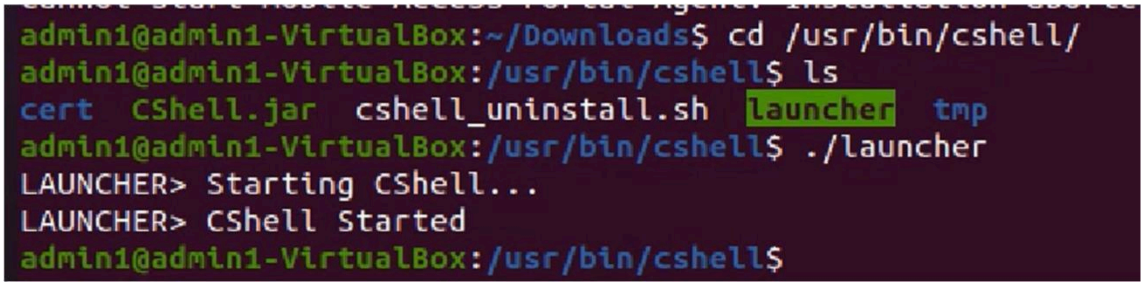
□ cshell
No description

☑ im-launch
No description

☑ SSH Key Agent
GNOME Keyring: SSH Agent

☑ VMware User Agent
No description

Go back to Terminal, navigate to /usr/bin/cshell, run the script name “launcher” , and you should get the result as below.

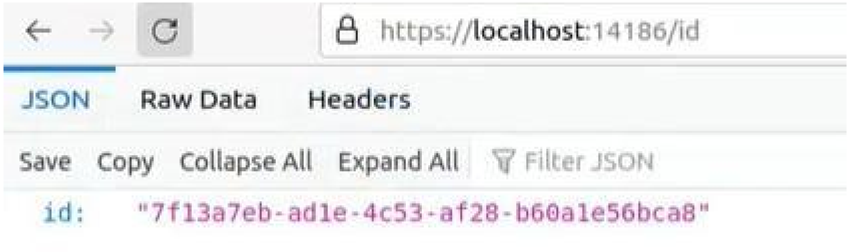


```
admin1@admin1-VirtualBox:~/Downloads$ cd /usr/bin/cshell/
admin1@admin1-VirtualBox:/usr/bin/cshell$ ls
cert  CShell.jar  cshell_uninstall.sh  launcher  tmp
admin1@admin1-VirtualBox:/usr/bin/cshell$ ./launcher
LAUNCHER> Starting CShell...
LAUNCHER> CShell Started
admin1@admin1-VirtualBox:/usr/bin/cshell$
```

To verify cshell process:

```
$ ps -ef | grep -i cshell
cshell     3502      1721 1 21:06 ?       00:00:02 java -jar /usr/bin/cshell/CShell.jar
/tmp/cshell.fifo
3574    2333  0 21:09 pts/0    00:00:00 grep --color=auto -i cshell
```

In some cases, the self-signed SSL certificate is not recognized by the cshell. In this case, please open the web browser and go to `https://localhost:14186/id` to accept the certificate.



https://localhost:14186/id

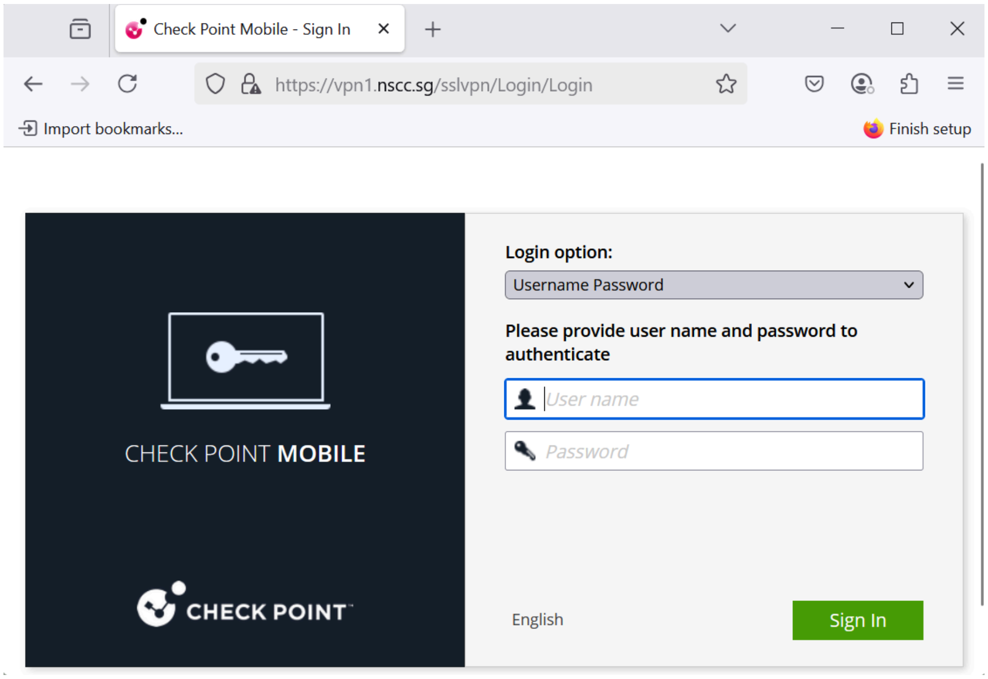
id: "7f13a7eb-ad1e-4c53-af28-b60a1e56bca8"

6. Connect to ASPIRE2A+ VPN via Checkpoint Mobile Access Portal.

NOTE: Please make sure the cshell is running before proceeding below instructions. Refer to step 4 to check and verify.

a. Open the browser and go to the ASPIRE2A+ SSLVPN page.

b. https://vpn1.nscc.sg/sslvpn

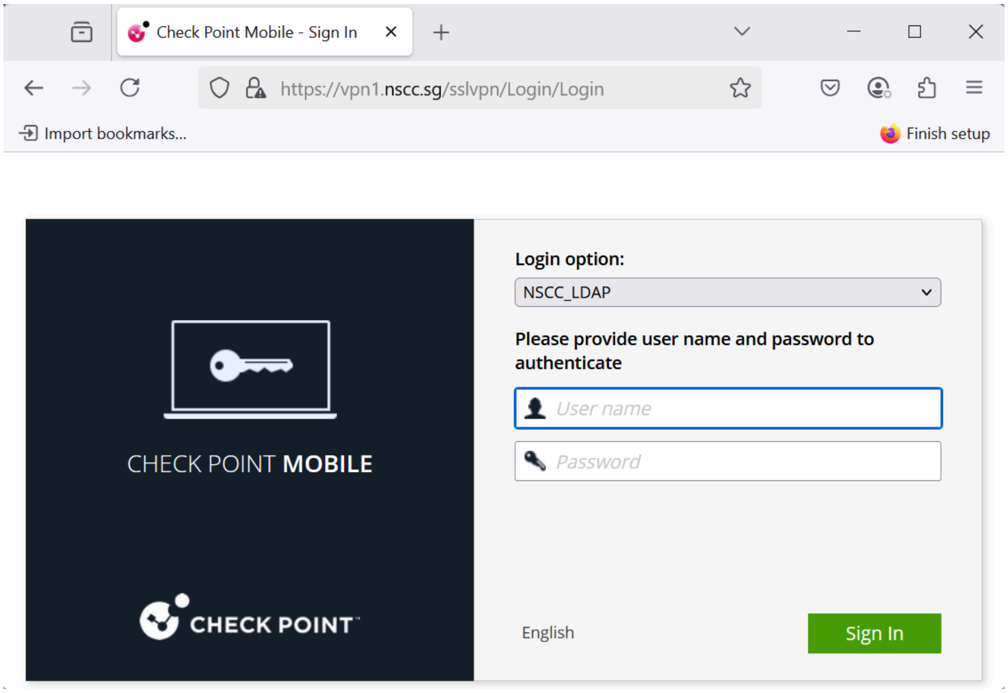


https://vpn1.nscc.sg/sslvpn/Login/Login

Login option: Username Password

Please provide user name and password to authenticate

c. Set the login option as "**NSCC_LDAP**" and login with your ASPIRE2A credentials. Approve the DUO 2FA notification in your mobile device.

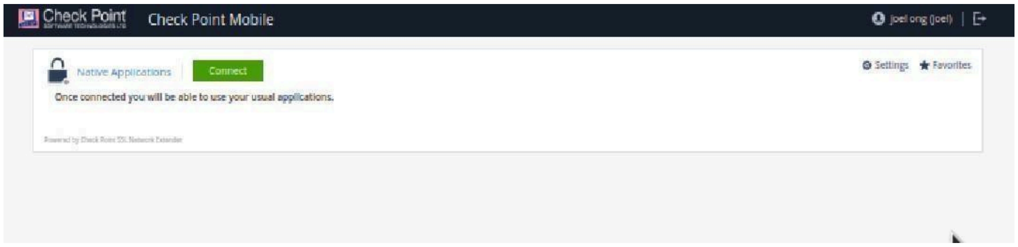


https://vpn1.nscc.sg/sslvpn/Login/Login

Login option: NSCC_LDAP

Please provide user name and password to authenticate

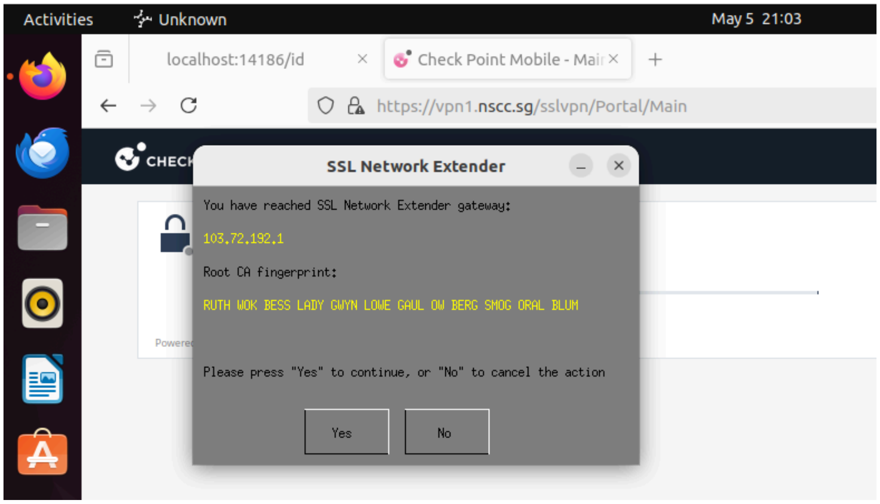
d. Click on **Connect** button.



Once connected you will be able to use your usual applications.

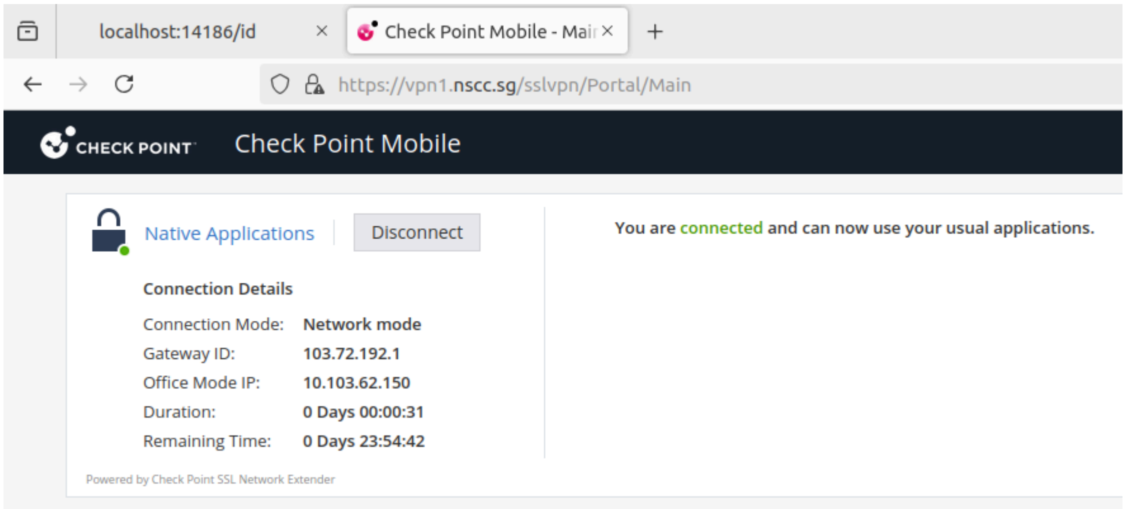
Powered by Check Point SSL Network Extender

e. Confirm the connection by selecting the **YES** button.



```
You have reached SSL Network Extender gateway:

103.72.192.1

Root CA fingerprint:

RUTH WOK BESS LADY GWYN LOWE GAUL OW BERG SMOG ORAL BLUM


Please press "Yes" to continue, or "No" to cancel the action
```

Connection is now established.



You are connected and can now use your usual applications.

Connection Details

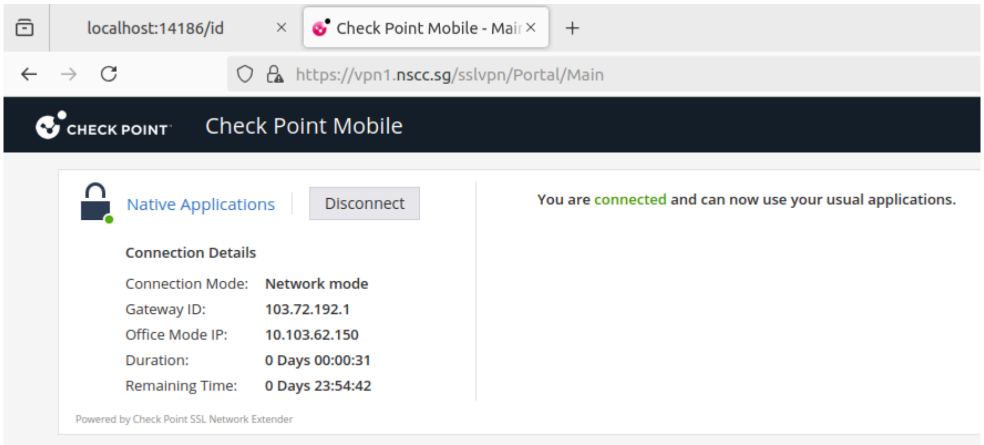
Connection Mode:	Network mode
Gateway ID:	103.72.192.1
Office Mode IP:	10.103.62.150
Duration:	0 Days 00:00:31
Remaining Time:	0 Days 23:54:42

Powered by Check Point SSL Network Extender

To disconnect the VPN:-

a) Open the browser and go to the ASPIRE2A+ SSL VPN page. https://vpn1.nscc.sg/sslvpn

b) Click on disconnect.



You are connected and can now use your usual applications.

Connection Details

Connection Mode:	Network mode
Gateway ID:	103.72.192.1
Office Mode IP:	10.103.62.150
Duration:	0 Days 00:00:31
Remaining Time:	0 Days 23:54:42

Powered by Check Point SSL Network Extender

Second Method: Using ChrootVPN

This method requires a script/wrapper maintained by a third-party. Please read our Disclaimer.

1. Install and configure ChrootVPN wrapper.

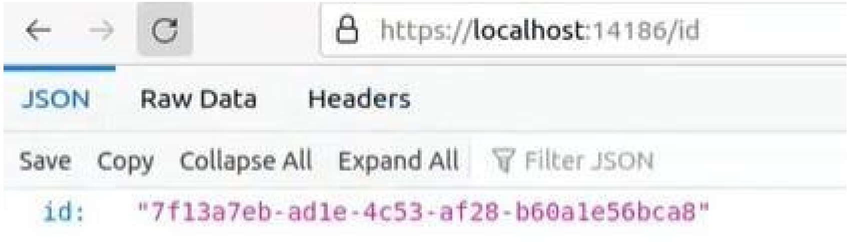
a. Download and extract the ChrootVPN wrapper package.

```
$ wget https://github.com/ruyrybeyro/chrootvpn/archive/refs/heads/main.zip
$ unzip main.zip
```

b. Setup the VPN client configuration. ASPIRE2A+ VPN Server is `vpn1.nscc.sg`. This step requires sudo privilege.

```
$ cd chrootvpn-main
$ sudo ./vpn.sh -i --vpn=vpn1.nscc.sg
$ sudo ./vpn.sh -i --vpn=vpn1.nscc.sg
[sudo] password for user:
ID=ubuntu
ID_LIKE=debian
DEB=true
Debian family setup
Hit:1 http://sg.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://sg.archive.ubuntu.com/ubuntu jammy-updates InRelease
[114 kB]
Get:3 http://sg.archive.ubuntu.com/ubuntu jammy-backports
InRelease [106 kB]
Hit:4
https://download.docker.com/linux/ub ..
..
chroot setup done.
..

Policy installed: /etc/firefox/policies/policies.json
No Firefox policy installed
open browser with https://localhost:14186/id to accept new
localhost certificate


..
LAUNCHER> Starting CShell...
LAUNCHER> CShell Started

Accept localhost certificate anytime visiting
https://localhost:14186/id
If it does not work, launch vpn.sh in a terminal from the
X11 console open browser at https://vpn1.nscc.sg to
login/start  VPN
```

c. In some cases, the self-signed SSL certificate is not recognized by the cshell. In this case, please open the web browser and go to `https://localhost:14186/id` to accept the certificate.



https://localhost:14186/id

id: "7f13a7eb-ad1e-4c53-af28-b60a1e56bca8"

d. Check and verify that cshell is running.

```
$ ./vpn.sh status | grep -i cshell
```

```
$ ./vpn.sh status | grep -i cshell
CShell running
CShell - installed version      80,0,0070,42
CShell - available for download 80,0,0070,42
CShell localhost self-signed CA certificate
```

If the cshell process is not running, start the cshell.

`$ ./vpn.sh status | grep -i cshell`

```
$ ./vpn.sh status | grep -i cshell
argCommands->showStatus: CShell not running
```

`$ ./vpn.sh start`

```
$ ./vpn.sh start
non-network local connections being added to access control list
LAUNCHER> Starting CShell...
LAUNCHER> CShell Started

Accept localhost certificate anytime visiting
https://localhost:14186/id If it does not work, launch
vpn.sh in a terminal from the X11 console open browser at
https://vpn1.nscc.sg to login/start  VPN
```

`$ ./vpn.sh stop`

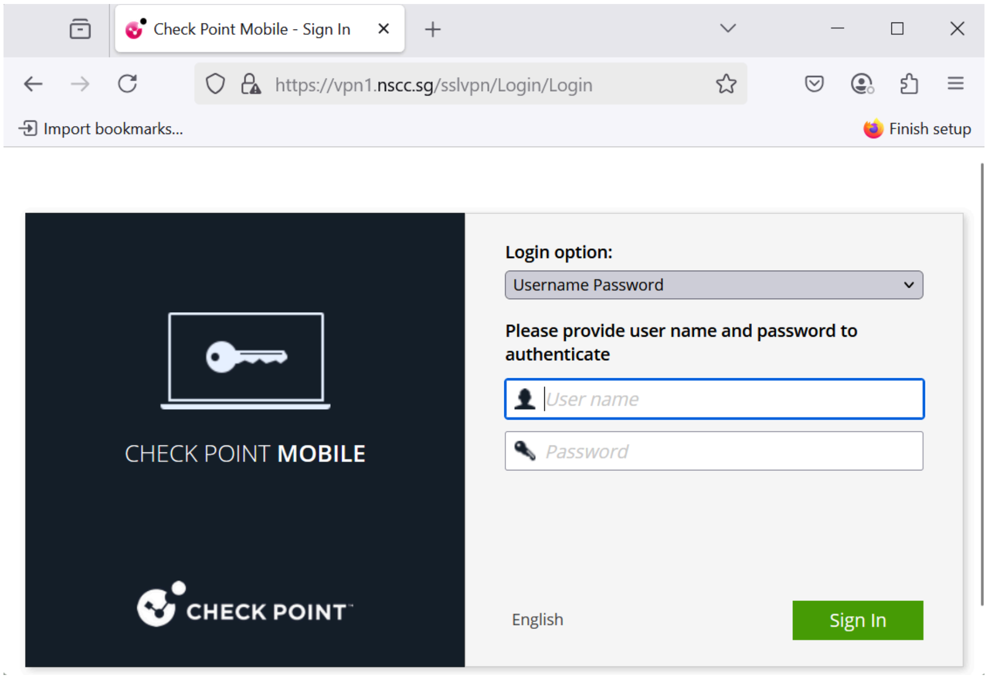
```
$ ./vpn.sh stop
CShell stopped
```

2. Connect to ASPIRE2A+ VPN via Checkpoint Mobile Access Portal.

NOTE: Please make sure the cshell is running before proceeding below instructions. Refer to step 1.d to check and verify.

a. Open the browser and go to the ASPIRE2A+ SSLVPN page.

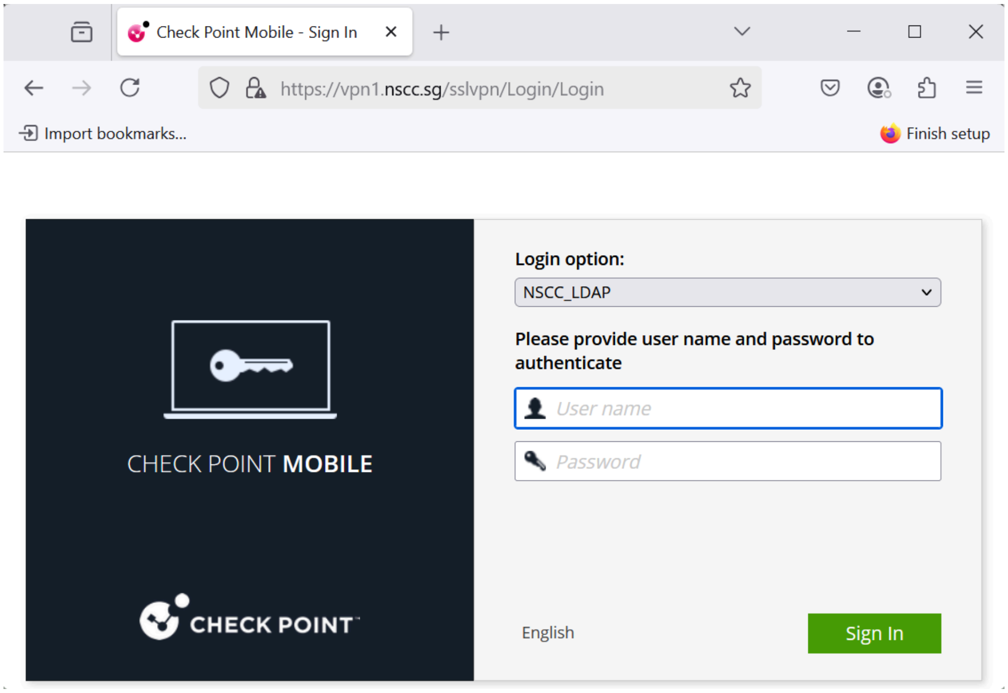
https://vpn1.nscc.sg/sslvpn



Login option: Username Password

Please provide user name and password to authenticate

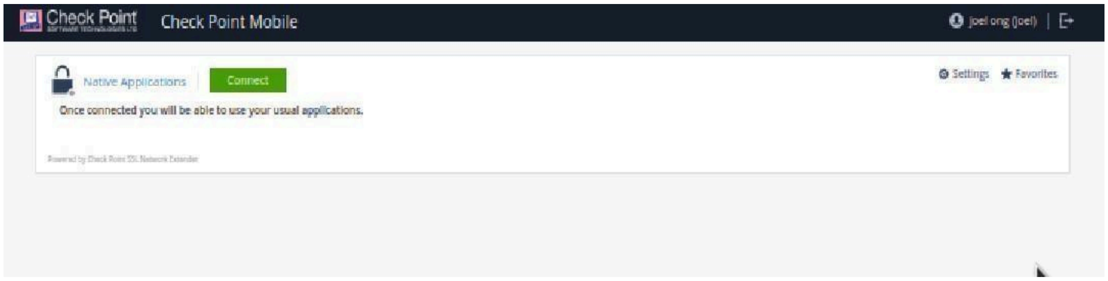
b. Set the login option as “**NSCC_LDAP**” and login with your ASPIRE2A+ credentials. Approve the DUO 2FA notification in your mobile device.



Login option: NSCC_LDAP

Please provide user name and password to authenticate

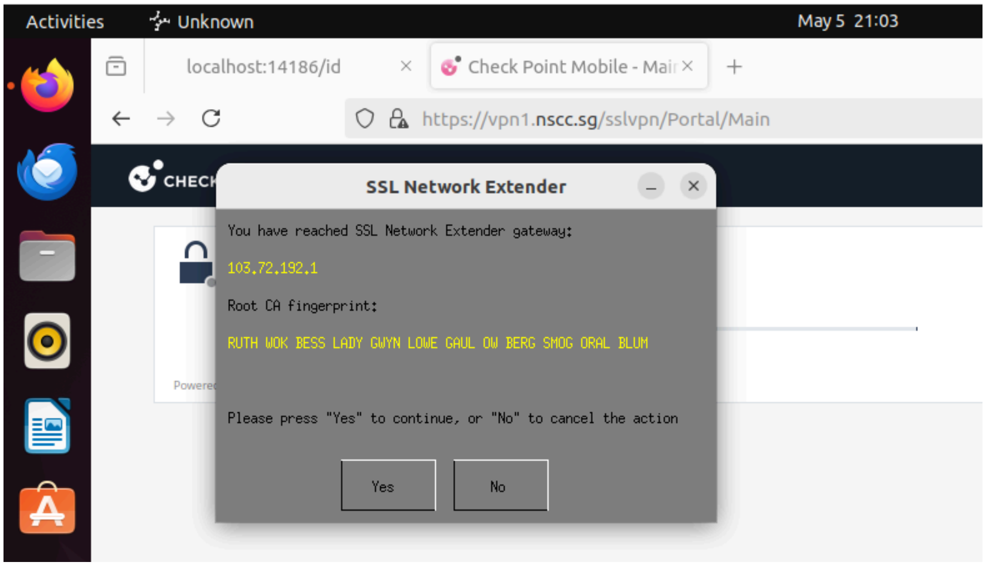
c. Click on **Connect** button.



Once connected you will be able to use your usual applications.

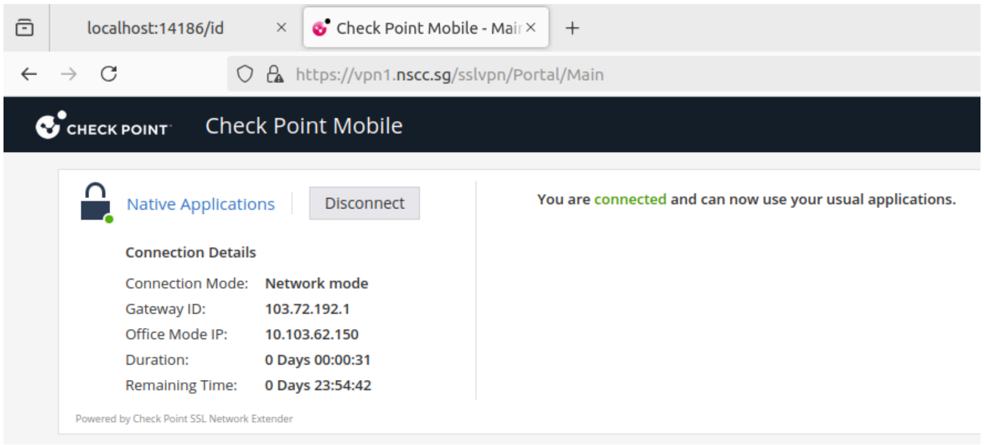
Powered by Check Point SSL Network Extender

d. Confirm the connection by selecting the **YES** button.



```
You have reached SSL Network Extender gateway:

103.72.192.1

Root CA fingerprint:

RUTH WOK BESS LADY GWYN LOWE GAUL OW BERG SMOG ORAL BLUM


Please press "Yes" to continue, or "No" to cancel the action
```

Connection is now established.



You are connected and can now use your usual applications.

Connection Details

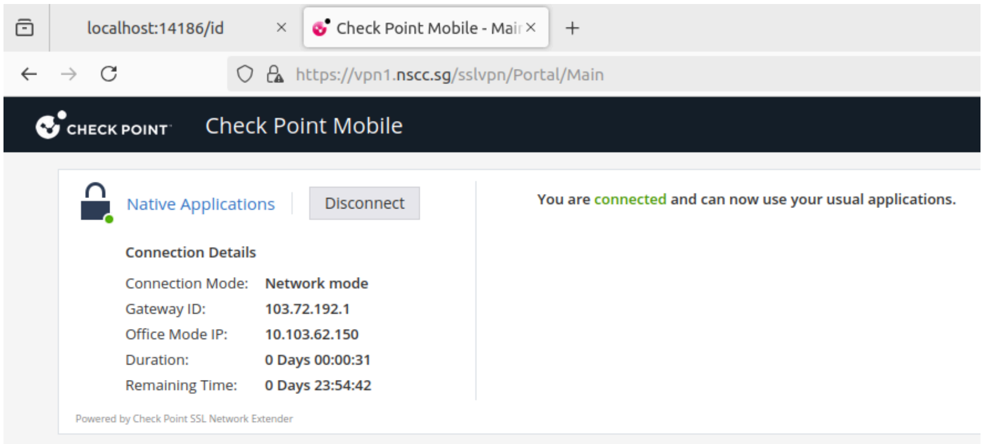
Connection Mode:	Network mode
Gateway ID:	103.72.192.1
Office Mode IP:	10.103.62.150
Duration:	0 Days 00:00:31
Remaining Time:	0 Days 23:54:42

Powered by Check Point SSL Network Extender

To disconnect the VPN:-

a) Open the browser and go to the ASPIRE2A+ SSL VPN page.
`https://vpn1.nscc.sg/sslvpn`

b) Click on disconnect.



You are connected and can now use your usual applications.

Connection Details

Connection Mode: Network mode
Gateway ID: 103.72.192.1
Office Mode IP: 10.103.62.150
Duration: 0 Days 00:00:31
Remaining Time: 0 Days 23:54:42

Powered by Check Point SSL Network Extender

Troubleshooting Advice

1. The snx error (Ubuntu-20.04)

```
$ sudo ./snx_install.sh Installation successful snx: error while
loading shared libraries: libpam.so.0: cannot open shared object
file: No such file or directory
```

Solution - Install below packages and rerun sns_install script.

To resolve this, run the following:

```
$ sudo apt-get install libstdc++5:i386 libpam0g:i386
```

Rerun snx =>

```
$ sudo ./snx_install.sh
Installation successful
```

Refresh the VPN URL and now it opens the extender.

```
$ sudo ldd /usr/bin/snx
```

2. Steps to start cshell automatically with system startup.

To start it automatically without sudo, run it in the .bashrc file, which will run with your normal permissions every time you open a terminal. Alternatively, you could run it once when you log in by placing it in your .profile file because everything in `.profile` runs as sudo, so the `launcher` is going to get stuck.

Create a simple log file to determine whether the launcher should be launched each time a terminal window is opened. To remove the log file each time you log in, add the following lines to the .profile file.

```
if [ -f "/home/[MY-USER-NAME]/cshell_launcher.log" ]; then
   rm /home/[MY-USER-NAME]/cshell_launcher.log
```

```
fi
```

On the `.bashrc` file should contain the following lines to check whether the launcher file should be run every time you open a terminal:

```
if [ ! -f "/home/[MY-USER-NAME]/cshell_launcher.log" ]; then
   /usr/bin/cshell/launcher >
/home/[MY-USER-NAME]/cshell_launcher.log
fi
```

Check if the log file we created is in your home directory by restarting your computer, logging in to your user account and typing:

```
$ ls | grep cshell

cshell_launcher.log
```

Then show its contents by using `cat` and it should be displayed as the following:

```
$ cat cshell_launcher.log

LAUNCHER> Starting CShell...

LAUNCHER> CShell Started
```

Now you should be all set. You should be able to connect every time you log on to your computer if you open your company's Check Point Mobile Access page. Don't forget to open a terminal window first.