Complete Installation of Developers Machine

1. OS Installation

1.1 How to Create a Bootable Pen drive

- First format the Pen drive.
- Go to Rufus -> Double Click on it.
- Now a pop-up will arise , login with Shekhar user with password !!q2w3e.
- Now select create bootable disk using ->ISO Image ->next to there will be an icon to browse the folder were the ISO Image is present -> Select the required one ->ready-> click start -> ok -> ok , once done close it.

1.2 Install OS using Pen Drive or any bootable device

- First while rebooting the system press **F2 -> Boot -> Boot List Option ->Legacy -> F10**
- Insert USB which have Ubuntu 16.04.3 image.
- You should see a welcome screen prompting you to choose your language and giving you the option to install Ubuntu or try it from the CD.
- If your computer doesn't automatically do so, you might need to press the **F12 key** to bring up the boot menu, but be careful not to hold it down - that can cause an error message.

1.3 Language selection and installation of Ubuntu pop

- Click on Install Ubuntu.
- You will also be asked for third-party installation and soon don't select anything -> continue.
- You will be be asked the type of installation you want -> Select erase all and reinstall ubuntu which is the first option -> continue .
- Select your location -> continue.
- Select the Keyboard type.
- Enter your Login details and password details

  [username -> admin1   password -> !1q2w3e4   ] -> continue.

1.4 Learn more about Ubuntu while the system installs

- Now wait till the installation is complete after that you will see the below screen
- Now remove the Pen Drive and restart your computer by clicking on Restart Now.
- Now you can start using Ubuntu.

2. Packeges Needed on Developers Machine

2.1 Network Configuration

2.1.1 Language Support

- Go to System Settings -> Language Support .
- You will see Keyboard input method system -> select none -> close.

2.1.2 Wifi Additional Drivers

- Go to System Settings -> Software and Updates.
- Now Go to Additional Drivers -> If you see any wifi drivers select it -> Apply Changes ->Install.
- Now once check Wifi connections by connecting through wifi.

2.1.3 Network Configuration

- $sudo su
- #apt-get install vim -y (if vim is not installed)

2.1.3.1 Set hostname

- #vim /etc/hostname
- Add 'localhost' to it save and exit.
- #echo localhost>/etc/hostname
- #hostname localhost

2.1.3.2 Edit /etc/hosts

- #vim /etc/hosts

  add '127.0.1.1 localhost' save and exit.
- #sed -i '2c\127.0.1.1 localhost' /etc/hosts

2.1.3.3 Set DNS

- #vim /etc/network/interfaces

  There should be no dns entry made here it -> save and quit

2.1.3.4 edit /etc/nsswitch.conf

- #vim /etc/nsswitch.conf

  comment this line

  “hosts: files mdns4_minimal [NOTFOUND=return] dns”

  add new line as follows.

  hosts: files dns   -> save and edit

Now goto System & Updates -> If you see officially supported ->select it -> close->reload

2.1.3.5 Edit /etc/resolv.conf

- # vim /etc/resolv.conf

  only the below lines should be there

  nameserver 8.8.8.8

  search google.com -> :wq!

- # service network-manager restart

2.2 Add developers user

- #useradd -m -d /home/developers/ -s /bin/bash developers
- #passwd developers (password -> nciportal)

2.3 Add hadoop user

- #useradd -m -d /home/hadoop/ -s /bin/bash hadoop
- # passwd hadoop (password -> hadoop)

2.4 Create directory in /usr/local

- #cd /usr/local
- #mkdir EIPS2_CONF EIPS2_LOGS EIPS2_DB EIPS2_USERS EIPS2_CHECKPOINT SensorData
- #chmod -R 777 EIPS2_* SensorData
- #chown -R developers:developers EIPS2_* SensorData

2.5 Add following files in EIPS2_CONF directory

these files are kept on common(10.13.10.213) at path:

EIPS2_CONF/ EIPS2_CONF/

1.app.properties

2.consumer-config.xml

3.database.xml

4.producer-config.xml

5.server.key

6.server.crt

7.server.csr

make sure that these files are latest updated.

Or EIPS_CONF is at location /ansible in Raj's machine.

For above files clone following git into the */usr/local/*

git clone http://10.13.10.20:3000/CLOUD/EIPS2_configuration.git

username :- redux.team   password :- nciportal

Change name of the EIPS2_configuration to EIPS2_CONF

2.6 Create packages directory in /usr/local/share

- #cd /usr/local/share/
- #mkdir packages

2.7 Prerequisites

- # apt-get update
- # apt-get install ntp dnsutils lzop gcc g++ make automake autoconf vim wget openssh-client openssh-server net-tools pkg-config dialog dconf-tools bridge-utils software-properties-common maven -y

2.7.1 Java installation

- Add the java repository

  ```
  # add-apt-repository ppa:openjdk-r/ppa
  ```

- Update all repositories

  ```
  # apt-get update
  ```

- Install java8 and java7

  ```
  # apt-get install openjdk-8-jdk

  # apt-get install openjdk-7-jdk
  ```

- Assign JAVA_HOME environmental varialble

  ```
  # vim /root/.bashrc

     export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/
  ```

- Select alternative java

  ```
  # update-alternatives --config java

     select java-8
  ```

2.8 Eclipse Installation and copy ncplind.conf and vin0-fix.sh

2.8.1 Eclipse Luna Installation

- Download eclipes luna from https://www.eclipse.org/downloads/packages/release/luna/
- **`# mkdir /home/developers/eclipes-luna/`**
- Extract tar using following command replace dir name as /home/developers/eclipse-luna

  ```
  # tar -xvzf eclipse-java-luna-SR2-linux-gtk-x86_64.tar.gz -C /home/developers
  # mv eclipse/ eclipse-luna/
  ```

- `# chmod -R 777 /home/developers/eclipse-luna`
- `# chown -R developers:developers /home/developers/eclipse-luna`

2.8.2 Eclipse Oxygen Installation

- Download eclipes oxygen from https://www.eclipse.org/downloads/packages/release/oxygen
- # mkdir /home/developers/eclipse-oxygen
- Extract tar using following command replace dir name as /home/developers/eclipse-oxygen

```
# tar -xvzf eclipse-jee-oxygen-3a-linux-gtk-x86_64.tar.gz -C /home/developers
# mv eclipse/ eclipse-oxygen/
```

- #chmod -R 777 /home/developers/eclipse-oxygen
- #chown -R developers:developers /home/developers/eclipse-oxygen

2.8.3 Eclipse Photon Installation

- Download eclipes photon from https://www.eclipse.org/downloads/packages/release/photon
- # mkdir /home/developers/eclipse-photon/
- Extract tar using following command and replace dir name as /home/developers/eclipse-photon

```
# tar -xvzf  eclipse-jee-photon-R-linux-gtk-x86_64.tar.gz
# mv eclipse/ eclipse-photon/
```

- #chmod -R 777 /home/developers/eclipse-photon/
- #chown -R developers:developers /home/developers/eclipse-photon

2.8.4 Copy ncplind.conf and vin0-fix.sh

- #cd /home/admin1
- #cp ncplind.conf /home/developers/
- #chmod 777 /home/developers/ncplind.conf
- #chown developers:developers /home/developers/ncplind.conf
- #cp vino-fix.sh /home/developers
- #chmod 777 /home/developers/vino-fix.sh
- #chown developers:developers /home/developers/vino-fix.sh

2.9 Nodejs:

- Get nodejs source file from http://nodejs.org/dist
- Download nodejs source file – (We are currently using and testing node version- node v8.9.3 . You can get any version of nodejs from path- http://nodejs.org/dist/. Some node module dependencies will be there when you use any other version.)

  ```
  # wget https://nodejs.org/download/release/v10.16.3/
  ```

- Extract the file-

  ```
  # tar -xvzf node-v10.16.3-linux-x64.tar.gz -C /usr/local/share/packages/
  ```

- # ln -s /usr/local/share/packages/node-v10.16.3-linux-x64/lib/node_modules /usr/local/lib/node_modules
- Get package.json (ask senior for Package.json) file and place in usr/local/lib/
- # chmod -R 777 /usr/local/lib/node_modules /usr/local/lib/package.json
- # chown -R developers:developers /usr/local/lib/node_modules /usr/local/lib/package.json
- # echo export PATH=/usr/local/share/packages/node-v10.16.3-linux-x64/bin:\$PATH >> /home/admin1/.bashrc
  # echo export PATH=/usr/local/share/packages/node-v10.16.3-linux-x64/bin:\$PATH >> /home/developers/.bashrc
  # source /home/developers/.bashrc
  # source /home/admin1/.bashrc

2.10 Redis Installation:-

- Update ubuntu package and install the prerequisites

  ```
  # apt-get update && apt-get install tcl -y
  ```

- Download 3.0.7 version of redis from link – http://download.redis.io/releases/ and untar it by using following commands

  ```
  # wget http://download.redis.io/releases/redis-3.0.7.tar.gz
  ```

  ```
  # tar –xvzf  redis-3.0.7.tar.gz -C /usr/local/share/packages
  ```

(Note: If you want to install another version of redis make changes in link eg.**http://download.redis.io/releases/redis-3.0.7.tar.gz)**

- Change into redis-3.0.7/src directory and install redis using “make” and “make test” commands:

  ```
  # cd /usr/local/share/packages/redis-3.0.7/
  ```

  ```
  # make install
  ```

```
# make test
```

- Start redis-server process to check if redis is working properly or not using below command:

```
# cd src/
# ./redis-server  --loglevel verbose
```

2.11 hbase

- Download hbase of version 7.2.6 from official website

# wget http://archive.apache.org/dist/hbase/2.0.0/hbase-2.0.0-bin.tar.gz (**Ask senior for updated hbase Tar**)

```
# tar -xvzf hbase-2.0.0-bin.tar.gz -C /home/hadoop
# chmod -R 777 /home/hadoop/hbase/
# chown hadoop:hadoop /home/hadoop/hbase/
# cp /home/hadoop/hbase/habse/bin/st*.sh  /home/hadoop/
# chmod 777 /home/hadoop/st*.sh
# chown hadoop:hadoop /home/hadoop/st*.sh
```

2.12 Kafka

- Download Kafka from the official site

# wget http://www-eu.apache.org/dist/kafka/2.0.0/kafka_2.11-2.0.0.tgz (**Ask senior for updated kafka Tar**)

- Go to the directory were the Kafka tar is present

```
# tar -xvzf kafka.tgz -C /home/hadoop/
# chmod -R 777 /home/hadoop/kafka/
# chown hadoop:hadoop /home/hadoop/kafka/
```

2.13 Zookeeper

Download Zookeeper from the official site

Goto the directory were the Zookeeper tar is present (**Ask senior for updated zookeeper Tar**)

- # tar -xvzf zookeeper.tar.gz -C /home/hadoop/
- # chmod -R 777 /home/hadoop/zookeeper/
- # chown hadoop:hadoop /home/hadoop/zookeeper/

2.15 VLC Installation

2.2 Installing VLC Media player from ubuntu software centre

- Open Ubuntu Software center
- Type VLC in search box and press enter.
- Select VLC from list and click on install.
- Type your super user password. Super user is the user which we create during the installation.

Install VLC from command line

Press ALT+CTRL+T to open the terminal. Run following command to add VLC repository.

- #sudo add-apt-repository ppa:videolan/master-daily
- #sudo apt-get update
- #sudo apt-get install vlc
- # sudo apt-get install -f

2.16 Google Chrome Installation

- Installing Google Chrome latest version via Command Line use following commands

```
wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb

dpkg -i google-chrome-stable_current_amd64.deb
```

OR

- By editing source.lists file

```
# vim /etc/apt/sources.list and add
deb http://dl.google.com/linux/chrome/deb/ stable main
# wget https://dl.google.com/linux/linux_signing_key.pub
# apt-key add linux_signing_key.pub
# apt update
# apt-get install google-chrome-stable
```

2.17 Ubuntu-tweak Installation

```
* add-apt-repository universe
  apt-get install gnome-tweak-tool
```

2.18 TeamViewer Installation

- Download TeamViewer from the below link

  ```
  # wget https://download.teamviewer.com/download/linux/teamviewer_amd64.deb
  ```

- Now go to the directory were it is downloaded

  ```
  # dpkg -i teamviewer_14.1.18533_amd64.deb
  ```

got the following error:

```
“dpkg: dependency problems prevent configuration of teamviewer:
 teamviewer depends on libqt5x11extras5 (>= 5.5) | qt56-teamviewer; however:
  Package libqt5x11extras5 is not installed.
  Package qt56-teamviewer is not installed.
 teamviewer depends on qml-module-qtquick-controls (>= 5.5) | qt56-teamviewer; however:
  Package qml-module-qtquick-controls is not installed.
  Package qt56-teamviewer is not installed.
 teamviewer depends on qml-module-qtquick-dialogs (>= 5.5) | qt56-teamviewer; however:
  Package qml-module-qtquick-dialogs is not installed.
  Package qt56-teamviewer is not installed.”
```

```
# apt install -f ./teamviewer_14.1.18533_amd64.deb -y
```

2.19 Maven Installation

```
# sudo apt-get install maven -y
# sudo apt-get install -f
#mvn -version
```

2.20 Git Installation

```
# sudo apt-get update

# sudo apt-get install git

# sudo apt-get update

# sudo apt-get install build-essential libssl-dev libcurl4-gnutls-dev libexpat1-dev gettext unzip -y

# git - -version
```

2.21 VPNC

- #apt-get install vpnc -y
- # apt-get update
- #apt-get install -f

2.22 AnyDesk

Download anydesk from the below link

- *wget https://download.anydesk.com/linux/anydesk_5.5.1-1_amd64.deb*
- *dpkg -i anydesk_5.5.1-1_amd64.deb*
- *apt-get install -f*

2.23 Nginx Installation

- # sudo add-apt-repository ppa:nginx/stable
- # sudo apt-get update
- # sudo apt-get install nginx -y
- # sudo apt-get update
- # sudo apt-get -f install

Testing Developers Machine

1. Take git clone from git

- # cd /home/developers/workspace/
- **NodeProject :** git clone http://node.user:nciportal@10.13.10.20:3000/node.adminNodeProject.git
- **cloudsStatic : git clone http://10.13.10.20:3000/UI/**cloudsStatic.git
- **cloudrelease** : git clone http://10.13.10.20:3000/CLOUD/cloudrelease.git
- **cloudigniteserver** : git clone http://10.13.10.20:3000/CLOUD/cloudigniteserver.git
- **Nginx files : git clone** http://10.13.10.20:3000/NGINX**_PROJECT/LOCAL.git**

  1) **replace default path**: /etc/nginx/sites-available/
  2) **replace nginx.conf** (**path**: /etc/nginx/nginx.conf)
  3) **replace mime.types** (**path**: /etc/nginx/mime.types)
  4) **put proxy.conf at location** (**path**: /etc/nginx/proxy.conf)

2. Installing Node modules

- apt install librdkafka-dev

  No need to install node modules in node-v10.

- # cd /usr/local/share/packages/node-v8.9.4-linux-x64/lib/
- Install these modules using npm

  # su

  # npm i @angular/cli@6.0.3 -g

  # npm i bower -g

  # npm i shelljs -g

  # npm i apidoc -g

  # npm i jsonpack -g

  # npm i mocha -g

  # npm i jsonfile -g

  # npm i chai -g

  # npm i chai-datetime -g

  # npm i prompt -g

```
# npm i hoek@4.0.1 -g

# npm i joi@9.0.4 -g

# npm i rimraf -g

# npm i qrcode@1.3.3 -g

# npm i crypto-js -g
```

- npm i @angular/cli@6.0.3 -g;npm i bower -g;npm i shelljs -g;npm i apidoc -g;npm i jsonpack -g;npm i mocha -g;npm i jsonfile -g;npm i chai -g;npm i chai-datetime -g;npm i prompt -g;npm i hoek@4.0.1 -g;npm i joi@9.0.4 -g;npm i rimraf -g;npm i qrcode@1.3.3 -g;npm i crypto-js -g

```
# su developers
```

- npm i xml2json@0.11.0 -g;npm i node-expat@2.3.15 -g
- **Creating node_modules softlink**

```
cd  /home/developers/workspace/

# ln -s /usr/local/share/packages/node-v10.16.3-linux-x64/lib/node_modules .
```

- # cd /home/developers/workspace/cloudsStatic/

```
# bower i
```

3. Starting Hbase services

- On 1st and 2nd tab run the below command

```
 # su – hadoop
```

- Get apache-pheonix-5.tar.gz and hbase_phoenix.tar.gz, hbase_start.sh and hbase_stop.sh
- On 1st tab run below commands

```
 # ls  -lrt

 # cd hbase/hbase/conf/

 # chmod -R 600 jmxremote*

 # cd /home/hadoop/

# ./hbase_start.sh
```

- On 2nd tab type

 # jps   (and check whether all the 8 services are running as shown below)

  - Jps
  - QuorumPeerMain

  - Kafka
  - HQuorumPeer
  - RESTServer
  - HRegionServer
  - Kafka
  - HMaster
- `# cd hbase/hbase/bin`
- `# ./hbase shell`

4. Region Server Graphical Testing

- Check out localhost:60010

5. Start Redis

```
# cd /usr/local/share/packages/redis-3.0.7/src/

# ./redis-server  --loglevel verbose ( now wait till you see Successfully Done)
```

6. **Start Ignite**

- create .m2 directory in /home/developers
- `# cp settings.xml / home/developers/.m2` (Ask senior for settings.xml)
- `# chown developers:developers  home/developers/.m2/settings.xml`
- `# cd /home/developers/workspace/cloudigniteserver`
- `# ./startIgnite.sh` (now wait till you see Successfully Done)

7. Start Tiger

- `# cd /home/developers/workspace/cloud`
- `# ./startTiger.sh` (now wait till you see Successfully Done)

8. Please follow this procedure updated procedure for clouzer org setup:

1. Replace attached pom.xml of cloudrelease and cloudigniteserver.
2. Start Ignite and Tiger.
3. Run Initscript using the command on path cloudrelease/cloud/target/:

```
java -cp dependency/*:cloud.jar com.cloud.loginregister.script.operation.InitScript
```

4. Restart Ignite and Tiger two times.
5. Take updates of master branch of cloudsStatic project.
6. Run bower i on cloudsStatic project's path.
7. Clone purebluescript project from git using the command:

```
git clone http://10.13.10.20:3000/pureblue/purebluescript.git
```

username: redux.team
password: nciportal
Note: If you have already cloned it. You need take updates of purebluescript project.

8. Take checkout of version 0.1.5 of purebluescript project using command:

```
git checkout 0.1.5
```

9. Run command npm i on purebluescript project.
10. Copy script_all.sh and purebluescript.sh files on path /home/developers/. PFA
11. run command chmod +x ./script_all.sh on path /home/developers/
12. run command chmod +x ./purebluescript.sh on path /home/developers/
13. run below command on path /home/developers/ :

```
./script_all.sh <workspace> purebluescript
```

For e.x. your workspace is "workspace" run command:

```
./script_all.sh workspace purebluescript
```

14. Change your node's IP in app.json file at path purebluescript/util

15. Start node: go to path /home/developers/workspace/Nodeproject/

```
* git checkout 1.77.2

# node Hapiserver.js
```

16. Run command npm run crm on path purebluescript.
17. Enter option 12 for Clouzer Org setup and press enter.

18. After successfully done, change the pom.xml in ignite and tiger.

Microservices Setup :

Local :

1. Checkout clouds-msconfig project from repository http://10.13.10.20:3000/CLOUD/clouds-msconfig.git.

2. Switch to “local” branch.

3. Copy all files from the EIPS2_CONF directory of that project and paste into /usr/local/EIPS2_CONF directory of your workstation.

Microservice Tiger :

1. Checkout clouds-msrelease project from repository http://10.13.10.20:3000/CLOUD/clouds-msrelease.git.

2. Switch to the branch your working on(master or develop).

2.1 : Run “startStreamTiger.sh” to start stream services.

2.2 : Run “startRestTiger.sh” to start REST services.

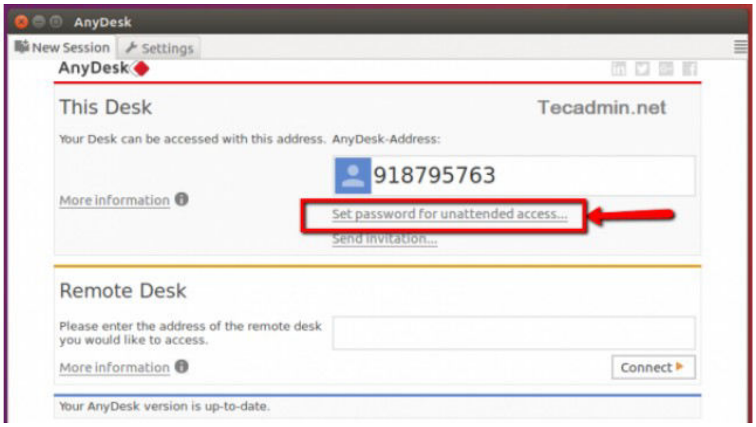
10. Giving sudo access to nginx and vpnc

- # vim /etc/sudoers   (as a root user)
- Add the below line to sudoers for sudo access and then save the file

  #developers ALL=/usr/sbin/vpnc,/usr/sbin/vpnc-disconnect,/etc/init.d/nginx

11. AnyDesk

- Goto to search and type anydesk as shown below window will open -> Click on the set password for unattended access by clicking on it



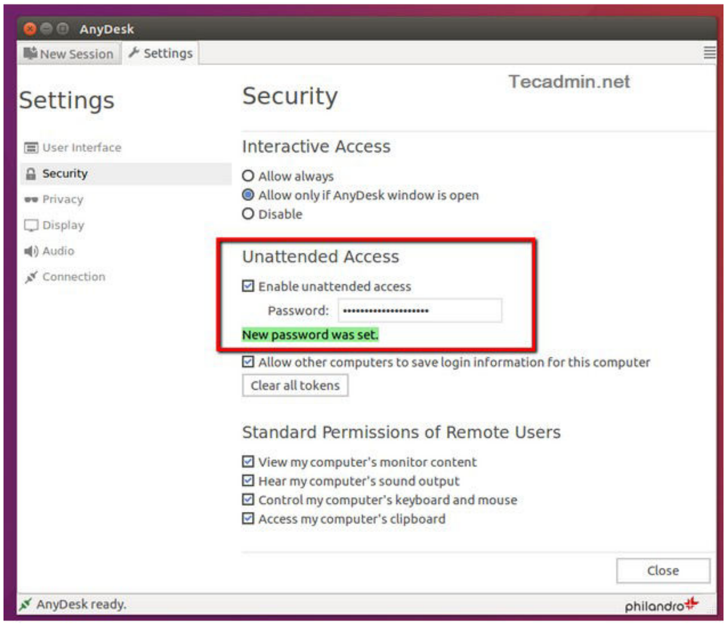
Your Desk can be accessed with this address.

918795763

Please enter the address of the remote desk you would like to access.

Your AnyDesk version is up-to-date.

- As shown above select Unattended Access and enter password -> nciportal close



AnyDesk

New Session | Settings

Tecadmin.net

Settings

User Interface
Security
Privacy
Display
Audio
Connection

Security

Interactive Access

○ Allow always
● Allow only if AnyDesk window is open
○ Disable

Unattended Access

☑ Enable unattended access

Password: ••••••••••••••••••••

New password was set.

☑ Allow other computers to save login information for this computer

Clear all tokens

Standard Permissions of Remote Users

☑ View my computer's monitor content
☑ Hear my computer's sound output
☑ Control my computer's keyboard and mouse
☑ Access my computer's clipboard

Close

AnyDesk ready.

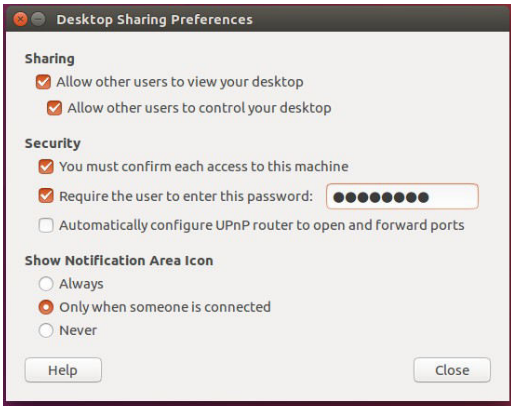
philandro

12. Desktop Sharing

- Goto search and type desktop sharing

  select Allow other users to view your desktop and also select Require the user

  to enter this password --> nciportal --> close



Desktop Sharing Preferences

Sharing

☑ Allow other users to view your desktop
☑ Allow other users to control your desktop

Security

☑ You must confirm each access to this machine
☑ Require the user to enter this password: ●●●●●●●●
□ Automatically configure UPnP router to open and forward ports

Show Notification Area Icon

○ Always
● Only when someone is connected
○ Never

Help | Close

13. VPNC Testing

- # sudo vpnc
  /home/developers/ncplind.conf

        - password --> nciportal
        - Enter password for ncpl.dev1@14.141.151.234 --> vpndev12ncpl
        - Now you will get an message that VPNC started in background
- # sudo vpnc-disconnect

14. Starting nginx service

- # sudo /etc/init.d/nginx start
- # sudo /etc/init.d/nginx status
- # sudo /etc/init.d/nginx stop

15. Stop all services you have started

- Stop the below service
    - HapiServer
    - Redis
    - startTiger
    - startIgnite
    - hbase