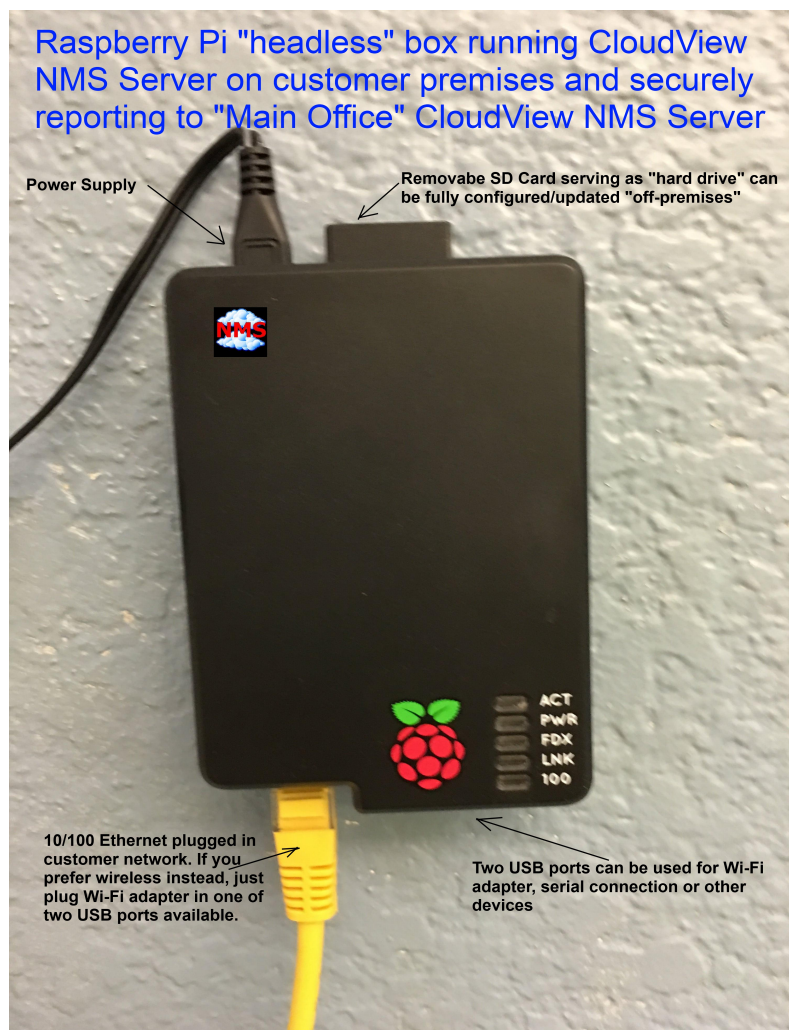




CloudView NMS Server running on Raspberry Pi

/step-by-step manual/

www.cloudviewnms.com

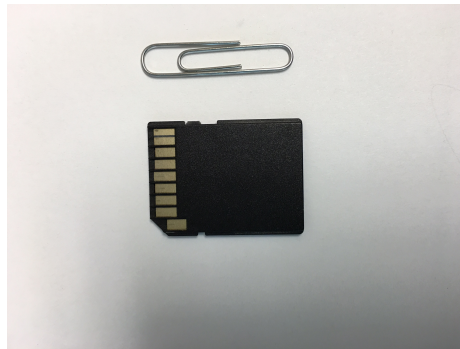


CloudView is a standards-based network management (monitoring) system (NMS). It can discover and perform many functions with virtually any vendor SNMP or TCP/IP devices/appliances. It provides you with the consistent view of your whole network, helps with its configuration, monitoring and troubleshooting.

CloudView NMS software can run on many modern platforms. One of the supported platforms is **Raspberry Pi** which is low cost (\$35), credit-card sized computer. Service Providers can pre-configure "headless" (no keyboard/monitor) **Raspberry Pi** box and plug it in the customer's network so it will monitor the network. The box securely (over SSL) reports the monitoring info/alerts to the "Central Office" CloudView NMS. No re-configuration of customer's router/firewall is required. The process of "Raspberry Pi" configuration is as simple as writing configuration files in the removable SD card which serves as "Raspberry Pi" hard drive.

This manual provides step-by-step instructions how to install and run CloudView NMS Server on Raspberry Pi.

Raspberry Pi comes with standard removable SD card, which serves as "hard drive" for the Raspberry Pi



computer. It usually comes with NOOBS (<https://www.raspberrypi.org/blog/introducing-noobs/>) which is good for beginners. However for our purpose (since we do not want anything extra taking space on this card) we want this card to contain pure and minimal **Raspbian** Linux (recommended Raspberry Pi OS). It is also good to use bigger capacity SD Card.

The detailed guide is here:

<https://www.raspberrypi.org/documentation/installation/installing-images/README.md>

To save space on SD Card, we recommend to download **RASPBIAN JESSIE LITE**, which is Minimal image based on **Debian Jessie**: <https://www.raspberrypi.org/downloads/raspbian/>

Using this link: <https://www.raspberrypi.org/documentation/installation/installing-images/README.md> you will find instructions how put the OS on the card. The procedure depends on which system you use pre-configuration (Linux, Windows or Mac OS). We used Linux, so the instructions are here. <https://www.raspberrypi.org/documentation/installation/installing-images/linux.md> .

After the above procedure, you must have bootable SD Card containing RASPBIAN JESSIE LITE. However, do not put the SD Card into your Raspberry Pi box yet. In order to communicate with the box remotely without mouse/keyboard/monitor we need to configure IP Address inside the box. Fortunately, it is very easy.

1. Plug the SD Card into your desktop computer or laptop (Windows/Linux/Mac OS). If your desktop or laptop does not have SD card port, it must have SD card reader connected.
2. Navigate to the corresponding drive. In the **boot** section of the drive, find file `cmdline.txt`. This file contains only one line of several "name=value" parameters divided by space. Using text editor add the following parameter in the end of the line (after space!) "ip=192.168.2.49". The line will look like

```
dwc_otg.lpm_enable=0 console=serial0,115200 console=tty1 root=/dev/mmcblk0p2  
rootfstype=ext4 elevator=deadline fsck.repair=yes rootwait ip=192.168.2.49
```

It is important to make sure that you do not add another line. It is still one line, though it is a long one.

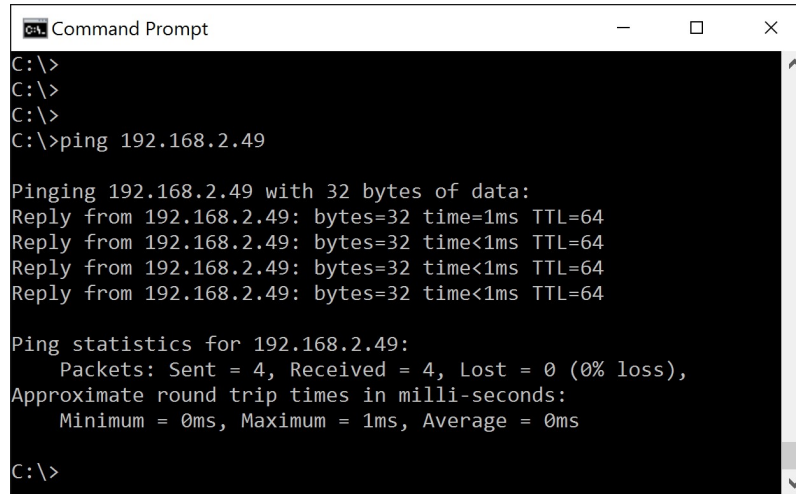
The above line assigns Private IP Address 192.168.2.49 to the Raspberry Pi box after it boots (starts). The corresponding IP Subnet is 255.255.255.0 which is clear because the IP address is of class C. If you do not understand all these details, it is OK. It is simply important to set the IP Address **similar** to the IP Address of your laptop/desktop, for example if your laptop IP Address is 192.168.2.48, you may assign IP Address 192.168.2.49 to the Raspberry Pi box provided that no other device in your network is using the same IP Address.

Going further, for initial setup, it is important that your Raspberry Pi has access to the Internet (for example to install Java and MySQL, see below). The access will be provided automatically if your private IP subnet 192.168.2.x has access to the Internet (i.e. your desktop/laptop with 192.168.2.48 has Internet connection).

Remove SD Card from your desktop/laptop and plug it fully in the Raspberry Pi Box. Connect the Ethernet cable (which is plugged into your network) to the 10/100Mbps port of your Raspberry Pi. Finally connect the power supply which will start the boot-up process. It should take no longer than 2 minutes for the Raspberry Pi to boot-up.

Check if your Raspberry Pi box is up. From the command-line terminal of your desktop/laptop issue ping command:

```
ping 192.168.2.49
```



```
C:\>
C:\>
C:\>
C:\>ping 192.168.2.49

Pinging 192.168.2.49 with 32 bytes of data:
Reply from 192.168.2.49: bytes=32 time=1ms TTL=64
Reply from 192.168.2.49: bytes=32 time<1ms TTL=64
Reply from 192.168.2.49: bytes=32 time<1ms TTL=64
Reply from 192.168.2.49: bytes=32 time<1ms TTL=64

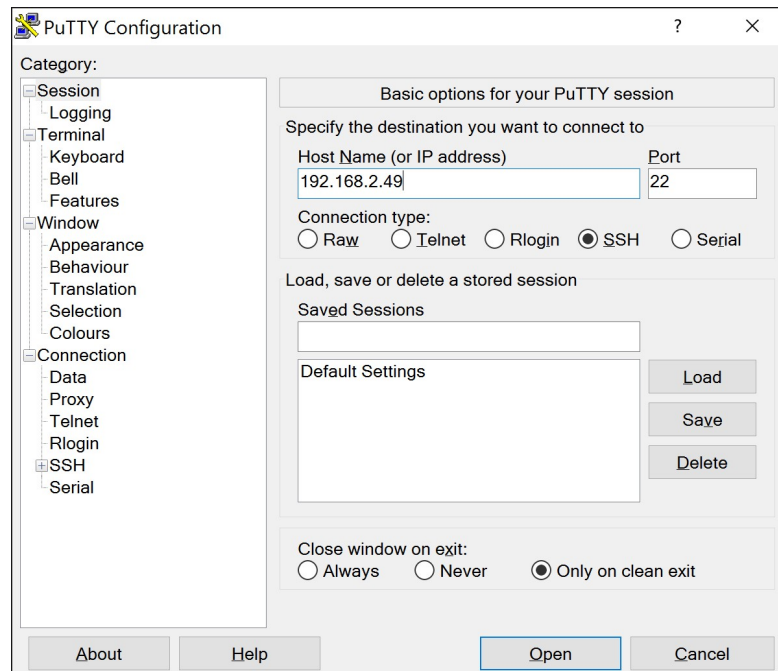
Ping statistics for 192.168.2.49:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

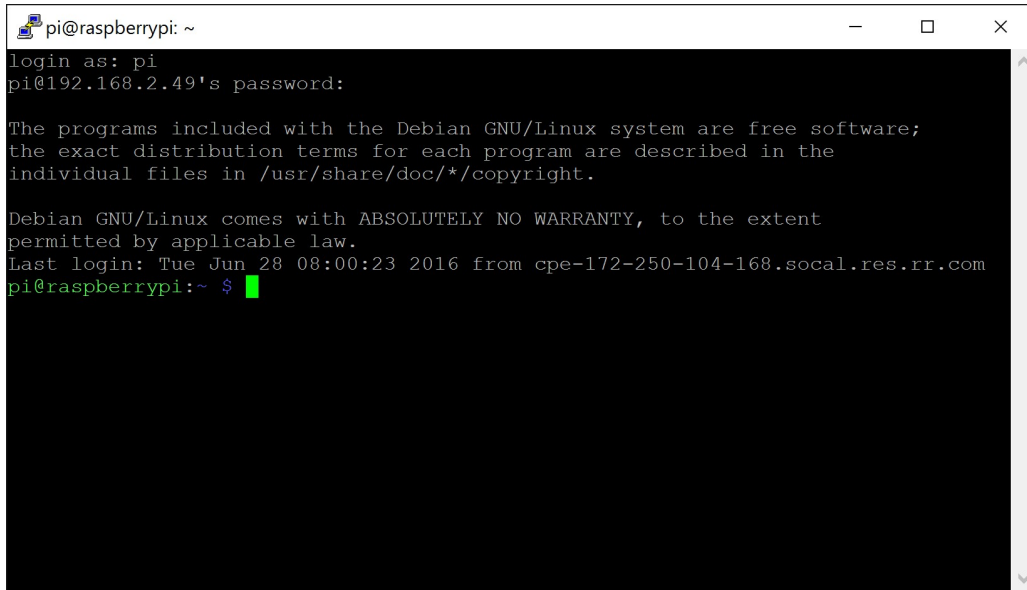
C:\>
```

If everything is OK, the above must show that the box is up. Now let us connect via SSH to the box. On Linux, Mac OS use command-line `ssh`, on Windows use Putty SSH (<http://www.putty.org/>). The default user name for Raspberry pi is "**pi**". The default password is "**raspberrypi**". Later you will have to change the defaults via corresponding Linux commands for security reasons. For now let us connect with default credentials. The corresponding Linux command to connect is:

```
ssh pi@192.168.2.49
```

With windows Putty:





```
pi@raspberrypi: ~
login as: pi
pi@192.168.2.49's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jun 28 08:00:23 2016 from cpe-172-250-104-168.socal.res.rr.com
pi@raspberrypi:~ $
```

We are connected and let us start configuration. First check that our Raspberry Pi has Internet connection. We will need it for initial configuration.

```
ping www.yahoo.com
```

If the above ping succeeds, the internet connection is OK.

If the above fails, you do not have internet access. To make sure Internet is reachable, you can set Static IP Address Configuration via editing file `/etc/dhcpd.conf` (`sudo nano /etc/dhcpd.conf`). Add the following lines in the end of the file:

```
# Custom static IP address for eth0.

interface eth0

static ip_address=192.168.2.49/24

static routers=192.168.2.1

static domain_name_servers=8.8.8.8
```

Of course, adjust the IP as you wish. The 192.168.2.1 IP address is referring to your router/gateway. "8.8.8.8" is the IP Address of Google DNS server on the public Internet which anybody can use. After this modification, restart the network interface or just perform a reboot (`sudo reboot now`).

For CloudView NMS we need the following packages: Java and MySQL. But first let us update our Raspbian with the latest first, run:

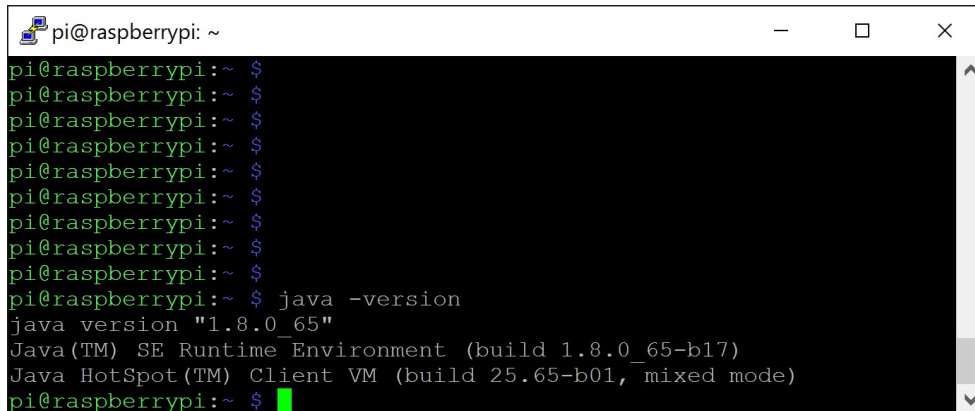
```
sudo apt-get update
```

To install Java run:

```
sudo apt-get install oracle-java8-jdk
```

After Java installation is done, to check it run:

```
java -version
```



```

pi@raspberrypi: ~
pi@raspberrypi:~$ 
pi@raspberrypi:~$ 
pi@raspberrypi:~$ 
pi@raspberrypi:~$ 
pi@raspberrypi:~$ 
pi@raspberrypi:~$ 
pi@raspberrypi:~$ 
pi@raspberrypi:~$ 
pi@raspberrypi:~$ 
pi@raspberrypi:~$ java -version
java version "1.8.0_65"
Java(TM) SE Runtime Environment (build 1.8.0_65-b17)
Java HotSpot(TM) Client VM (build 25.65-b01, mixed mode)
pi@raspberrypi:~$ 

```

Let us install MySQL, run:

```
sudo apt-get install mysql-server
```

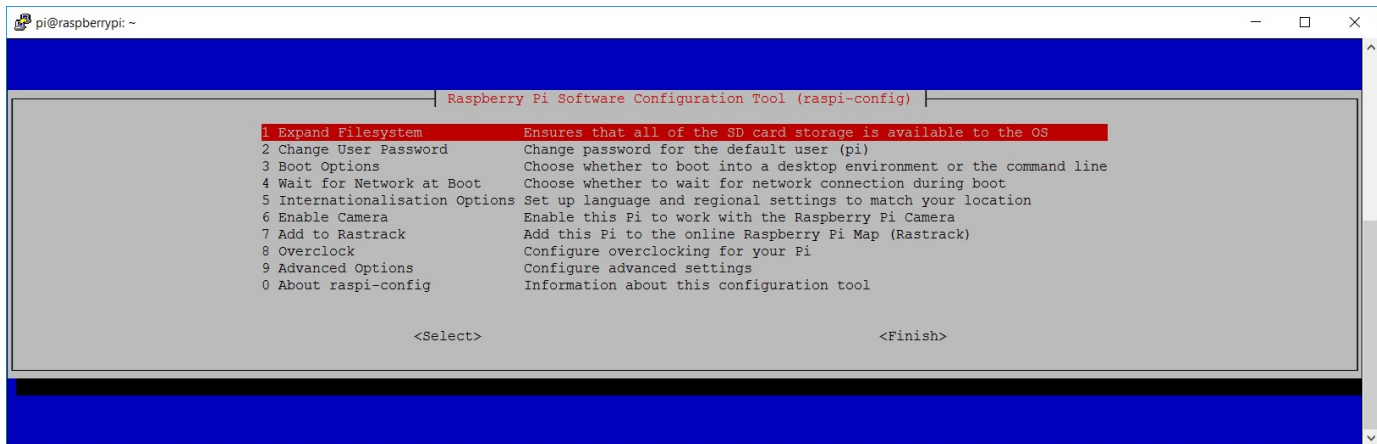
This will work for some time and (at certain moment) will ask you to enter MySQL root password (note that MySQL root password does not have anything to do with Linux root password). Please enter MySQL root password as **meta** . You will be asked to repeat the password again. You can change the password later via CloudView NMS web-GUI or CloudView NMS independent GUI client.

Now it would be a good time to check how much disk space left. Run `df` command. You may find that there is not enough space left in `/dev/root` (Mounted on `/`) to continue. It may show that `/dev/root` is used for 90% which is already not enough to continue.

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/root	7.8G	1.1G	6.4G	95%	/
devtmpfs	224M	0	224M	0%	/dev
tmpfs	229M	0	229M	0%	/dev/shm
tmpfs	229M	4.6M	224M	2%	/run
tmpfs	5.3M	4.1k	5.3M	1%	/run/lock
tmpfs	229M	0	229M	0%	/sys/fs/cgroup
/dev/mmcblk0p1	63M	21M	43M	34%	/boot

There is an easy way to increase disk space available for the Raspberry Pi file system. Run

```
sudo raspi-config
```



Using TAB key, select the first item "expand file system") then <Finish>. The reboot will be suggested. The above method should increase the available space dramatically.

However, if the above method does not work for you, there are other ways. The following links may help you to deal with the issue.

<http://raspberrypi.stackexchange.com/questions/41208/raspberry-pi-out-of-disk-space>

Here is another link:

<http://raspi.tv/2016/how-to-free-up-some-space-on-your-raspbian-sd-card-remove-wolfram-libreoffice>

As it is noted there, the command:

```
dpkg-query -Wf '${Installed-Size}\t${Package}\t${Priority}\n' | egrep
'\s(optional|extra)' | cut -f 1,2 | sort -nr | less
```

will show all the packages installed. The packages are sorted by size. You can choose to remove a couple of packages to free some disk space. For example we removed some packages connected to "docs". Also we removed compilers g++, c++.

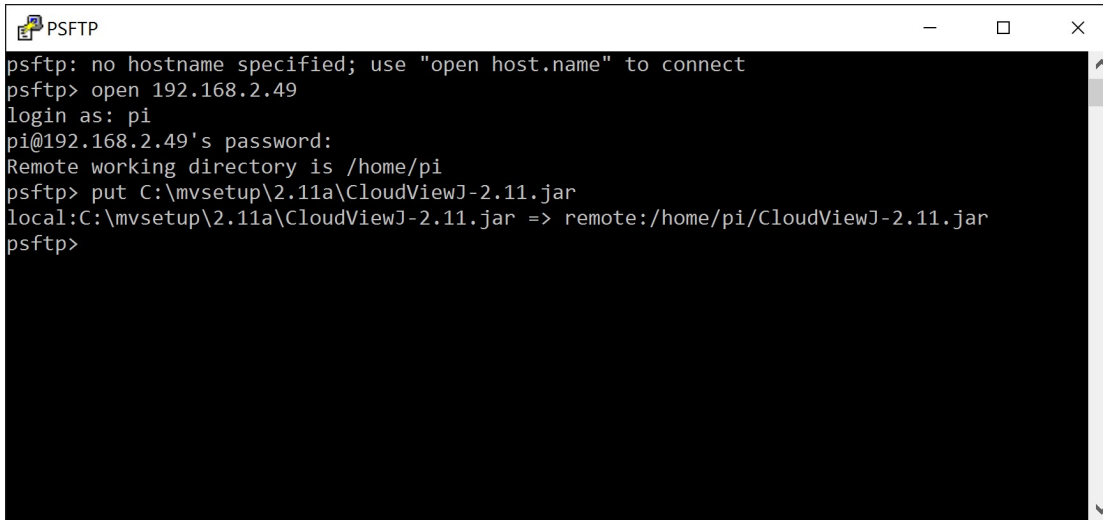
Let us install CloudView NMS Server software now. Download it from www.cloudviewnms.com download page. The current version file name is CloudViewJ-2.18.jar which reflects the version 2.18. The version number will change in the future. How to put the file on your Raspberry Pi? There are multiple methods described below (the screenshots were captured for older version 2.11, not the latest 2.18)

First method: download the file to your desktop/laptop and:

In Linux, Mac OS use:

```
scp <path to file>/CloudViewJ-2.18.jar pi@192.168.2.49:CloudViewJ-2.18.jar
```

In MS Windows MS, use WinSCP (free GUI App, find it on the Internet) or PSFTP:



```
PSFTP
psftp: no hostname specified; use "open host.name" to connect
psftp> open 192.168.2.49
login as: pi
pi@192.168.2.49's password:
Remote working directory is /home/pi
psftp> put C:\mvsetup\2.11a\CloudViewJ-2.11.jar
local:C:\mvsetup\2.11a\CloudViewJ-2.11.jar => remote:/home/pi/CloudViewJ-2.11.jar
psftp>
```

The above will put the CloudViewJ-2.18.jar file in the home directory of user pi /home/pi.

Another method to get the CloudViewJ-2.18.jar file is to use wget command to retrieve the file directly from the CloudView NMS web site:

```
wget www.cloudviewnms.com/files/CloudViewJ-2.18.jar
```

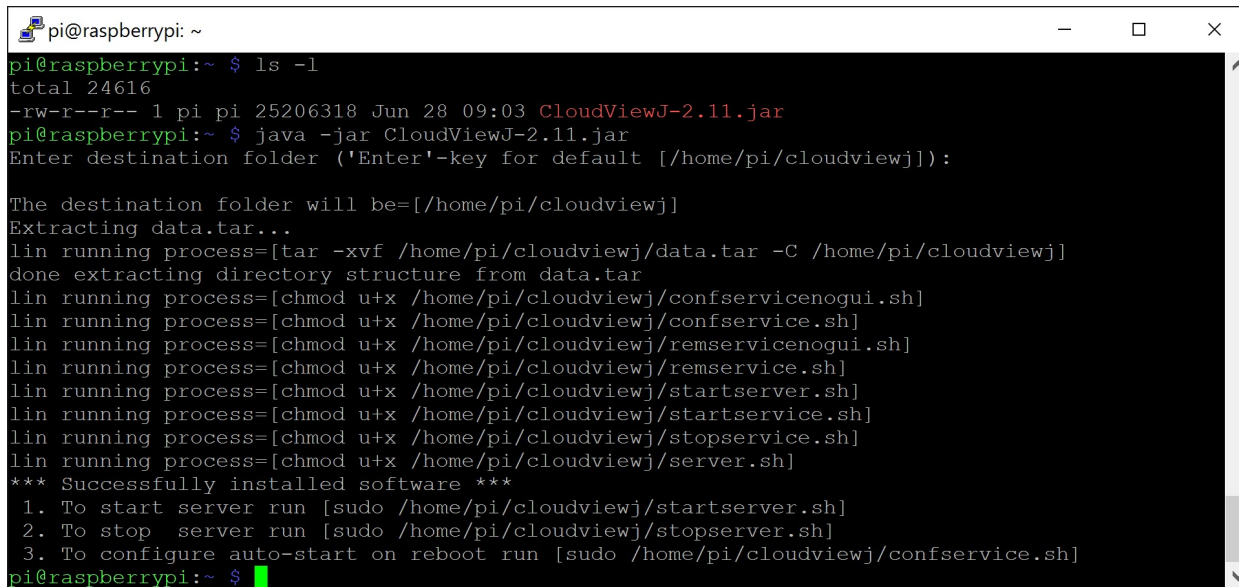
So we have CloudViewJ-2.18.jar file in the home directory of user pi /home/pi .



```
pi@raspberrypi: ~
pi@raspberrypi:~ $
pi@raspberrypi:~ $
pi@raspberrypi:~ $ ls
CloudViewJ-2.11.jar
pi@raspberrypi:~ $
```

It is self extractable jar file. To start installation run:

```
java -jar CloudViewJ-2.18.jar
```



```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ ls -l  
total 24616  
-rw-r--r-- 1 pi pi 25206318 Jun 28 09:03 CloudViewJ-2.11.jar  
pi@raspberrypi:~ $ java -jar CloudViewJ-2.11.jar  
Enter destination folder ('Enter'-key for default [/home/pi/cloudviewj]):  
  
The destination folder will be=[/home/pi/cloudviewj]  
Extracting data.tar...  
lin running process=[tar -xvf /home/pi/cloudviewj/data.tar -C /home/pi/cloudviewj]  
done extracting directory structure from data.tar  
lin running process=[chmod u+x /home/pi/cloudviewj/confservicenogui.sh]  
lin running process=[chmod u+x /home/pi/cloudviewj/confservice.sh]  
lin running process=[chmod u+x /home/pi/cloudviewj/remservicenogui.sh]  
lin running process=[chmod u+x /home/pi/cloudviewj/remservice.sh]  
lin running process=[chmod u+x /home/pi/cloudviewj/startserver.sh]  
lin running process=[chmod u+x /home/pi/cloudviewj/startservice.sh]  
lin running process=[chmod u+x /home/pi/cloudviewj/stopservice.sh]  
lin running process=[chmod u+x /home/pi/cloudviewj/server.sh]  
*** Successfully installed software ***  
1. To start server run [sudo /home/pi/cloudviewj/startserver.sh]  
2. To stop server run [sudo /home/pi/cloudviewj/stopserver.sh]  
3. To configure auto-start on reboot run [sudo /home/pi/cloudviewj/confservice.sh]  
pi@raspberrypi:~ $
```

Note you **should not run above as "sudo"**. Also we suggest that you simply press "Enter" when asked about destination folder. This will set the destination folder to default value /home/pi/cloudviewj

The CloudView NMS Server software is installed. To start the server run:

```
cd /home/pi/cloudviewj
```

```
sudo ./startserver.sh
```

See the output in below screenshot. Since it is first start the software will request license password. The license password can be received by eligible customers via e-mail from support@cloudviewnms.com.


```

pi@raspberrypi: ~/cloudviewj
permitted by applicable law.
Last login: Tue Jun 28 08:20:29 2016 from 192.168.2.48
pi@raspberrypi:~ $ ls
cloudviewj  CloudViewJ-2.11.jar
pi@raspberrypi:~ $ cd cloudviewj
pi@raspberrypi:~/cloudviewj $ ls *.sh
confservicenogui.sh  remservicenogui.sh  server.sh          startservice.sh
confservice.sh       remservice.sh       startserver.sh     stopservice.sh
pi@raspberrypi:~/cloudviewj $ sudo ./startserver.sh
Headless=true
Start app: Java Version=1080; OS=Linux,nver=1080 32-bit service=true headless=t
rue,rootdir=/home/pi/cloudviewj
No file [/home/pi/cloudviewj/server.ini], it willbe created
HTTP Web Server Started on TCP port 80
*****License Setup*****
**
First start, the computer specific string:
adePotreoonyoPrgoyoTPloildaahdrw%efrst&bb4&e**%/
To receive your license password/file, copy the above string with no extra space
s and email it to CloudViewNMS
If you already have your license password please enter/paste it below and press
'Enter'
/to run as Trial enter 'trial' as a license password/:

```

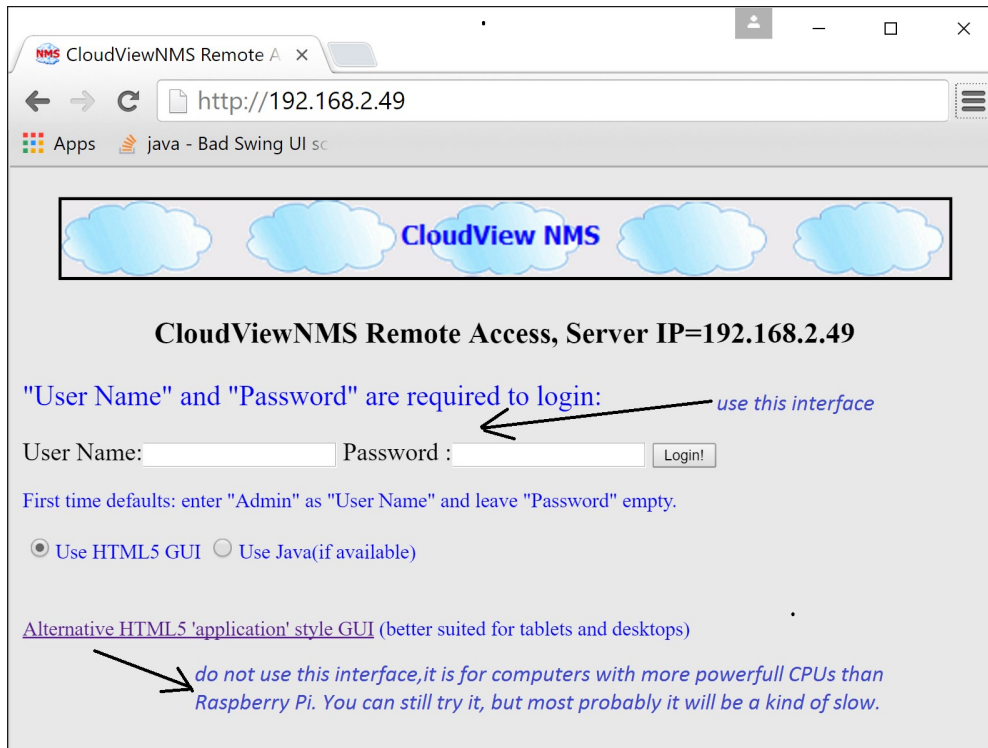
To run as Trial enter 'trial' as license password.

```

pi@raspberrypi: ~/cloudviewj
pi@raspberrypi:~/cloudviewj $ sudo ./startserver.sh
Headless=true
Start app: Java Version=1080; OS=Linux,nver=1080 32-bit service=true headless=t
rue,rootdir=/home/pi/cloudviewj
No file [/home/pi/cloudviewj/server.ini], it willbe created
HTTP Web Server Started on TCP port 80
*****License Setup*****
**
First start, the computer specific string:
adePotreoonyoPrgoyoTPloildaahdrw%efrst&bb4&e**%/
To receive your license password/file, copy the above string with no extra space
s and email it to CloudViewNMS
If you already have your license password please enter/paste it below and press
'Enter'
/to run as Trial enter 'trial' as a license password/:
trial
Ok, will run CloudViewNMS service/or restart/ as 'trial' for 2 hours; restart=fa
lse
Use web browser, web app or indepengent GUI client to connect via mentioned HTTP
/HTTPS TCP port
contOpen : should be only once....

```

The CloudView NMS Server started on TCP port 80. Now you can go back to your desktop and connect to the Raspberry Pi CloudView via Web Browser, the corresponding URL is <http://192.168.2.49> which is an equivalent of <http://192.168.2.49:80>



Notice that we used HTTP default TCP port 80 which is default web port. Later below, for security, it will be shown how to configure CloudView NMS Web Server to use HTTPS (HTTP over SSL) and another TCP port. The default HTTPS port is 443, but for security we recommend using another port like 4430. The corresponding URL will look like <https://192.168.2.49:4430>

There are two type of GUI interfaces available in the login page. We recommend using the first one shown in the above screenshot. The second interface is more powerful, but it requires CPUs more powerful than Raspberry Pi. You can still try it but the chances are it will be a kind of slow.

So let us login. The default "User Name" is "admin" (it is not case sensitive). The default password is empty string.

CloudViewNMS Remote Access, Server IP=192.168.2.49 , User="Admin"

Refresh Logout Views... Device/Submap... Network... Config...

Devices List, Current Map="main" , Map Health/SLA=100.000 %

Select	Device Name	Device Type	IP Address	Comm. Status	Alarms Status	Health/SLA %	Notepad
Total Number of devices on this map: 0							

Refresh Interval(sec): 60 Change Refresh Interval

Enter DeviceName/IP Address/Notepad (use wildcard '*' for partial text): Locate Device on Maps

Current Events Log

ACK	DupNumb/Date/Time	Map	Device	Severity	Event Message
***	Wed Jun 29 05:20:06:	Not Apply	Server	INFO	*** Web Client session opened on 172.250.104.168 by Admin ***
***	Wed Jun 29 04:43:42:	Not Apply	Server	INFO	Traps initialized, listening on port=162
***	Wed Jun 29 04:22:19:	Not Apply	Server	INFO	BSD sysLog Protocol Server started on UDP port 514
***	Wed Jun 29 04:22:19:	Not Apply	Server	INFO	TL1/Telnet Server started on TCP Port 3082
***	Wed Jun 29 04:22:19:	Not Apply	Server	INFO	Time Protocol /RFC 868/ server started on TCP port 37
***	Wed Jun 29 04:22:19:	Not Apply	Server	INFO	SNMP agent server started on UDP port 161
***	Wed Jun 29 04:22:18:	Not Apply	Server	INFO	HTTP Web Server Started on TCP port 80
***	Wed Jun 29 04:22:15:	Not Apply	Server	INFO	*** CloudViewNMS started ***
N/A	Wed Jun 29 04:22:15:	Not Apply	Server	INFO	MySQL Started/Restarted, number of events in History Events DB : 1153

Displayed 9 events of 9 in the current log.

The combo boxes above represent menus. All the functionality is accessed via the menu items.

Views... Device/Submap... Network... Config...

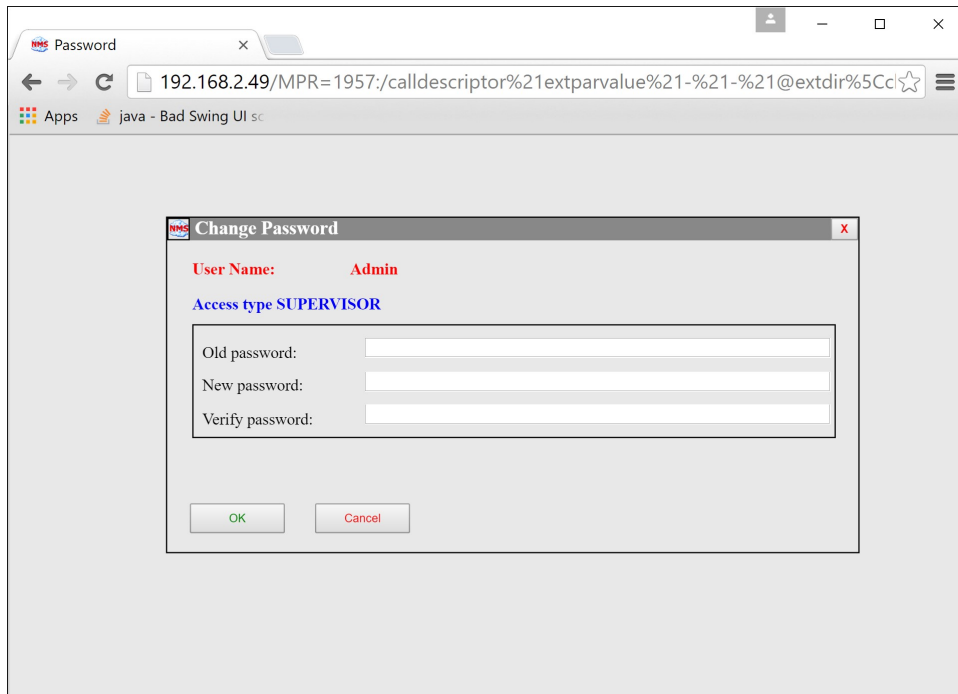
Views...
 Devices List View
 Map View
 Maps List View
 Files View
 Current Events View
 Current Events Window
 History Events View
 History Events Window
 Clear+Acknowledge All Events
 Acknowledge All Events
 Help View

Device/Submap...
 Add Device/Submap
 Add Link
 Delete Current Device
 Describe Current Device
 Autodiscovery
 Move Current Icon
 Current Device TCP/IP Services
 SNMP MibView
 Build Custom Mib App
 Results of Custom Mib App
 Set Map Background Image
 Save Maps
 Open Upper Map

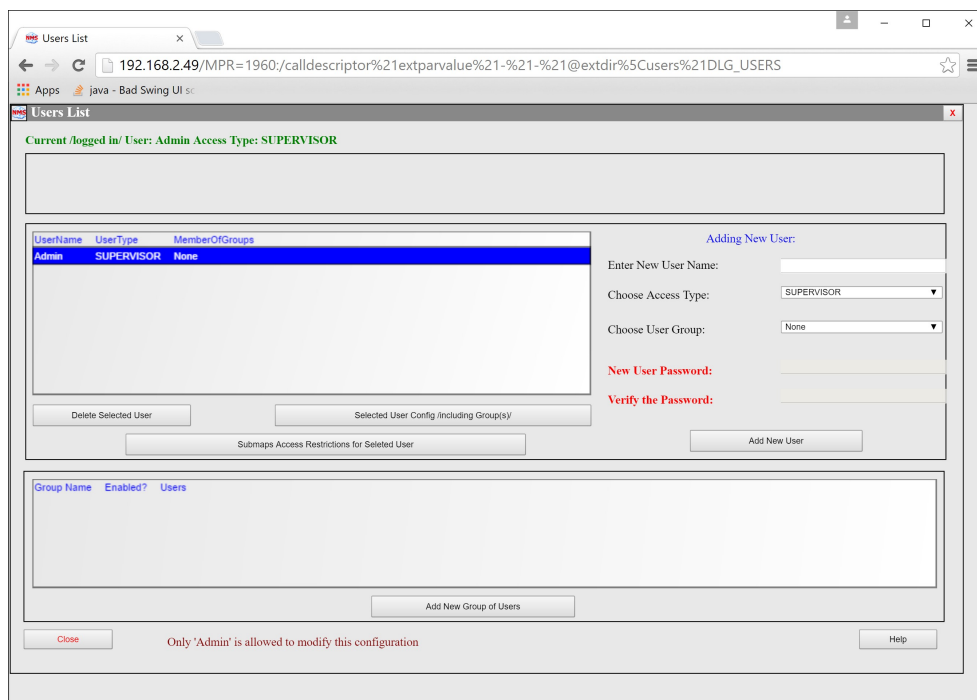
Network...
 Network Inventory
 Links Inventory
 Top 10 Ports
 Current Events Summary
 Connected Agents
 Agents Bandwidth
 Autodiscovery
 Services Monitor
 Add Device/Submap
 Arrange 'Equally'
 Arrange 'NoOverlap'
 Save Maps
 Locate Device on Maps
 Find Port/Interface on Maps
 Global Operations Tree

Config...
 Change Password
 Add/Delete Users
 Events Config
 Email Config
 Web Server Config
 SNMP Agent Config
 TL1 Config
 Submap Polling
 Connect Upper NMS
 Manage Maps
 Delete All Maps
 Set License
 Message to Log
 About

Let us start from changing Admin's password. The corresponding menu item is *Config->Change Password*



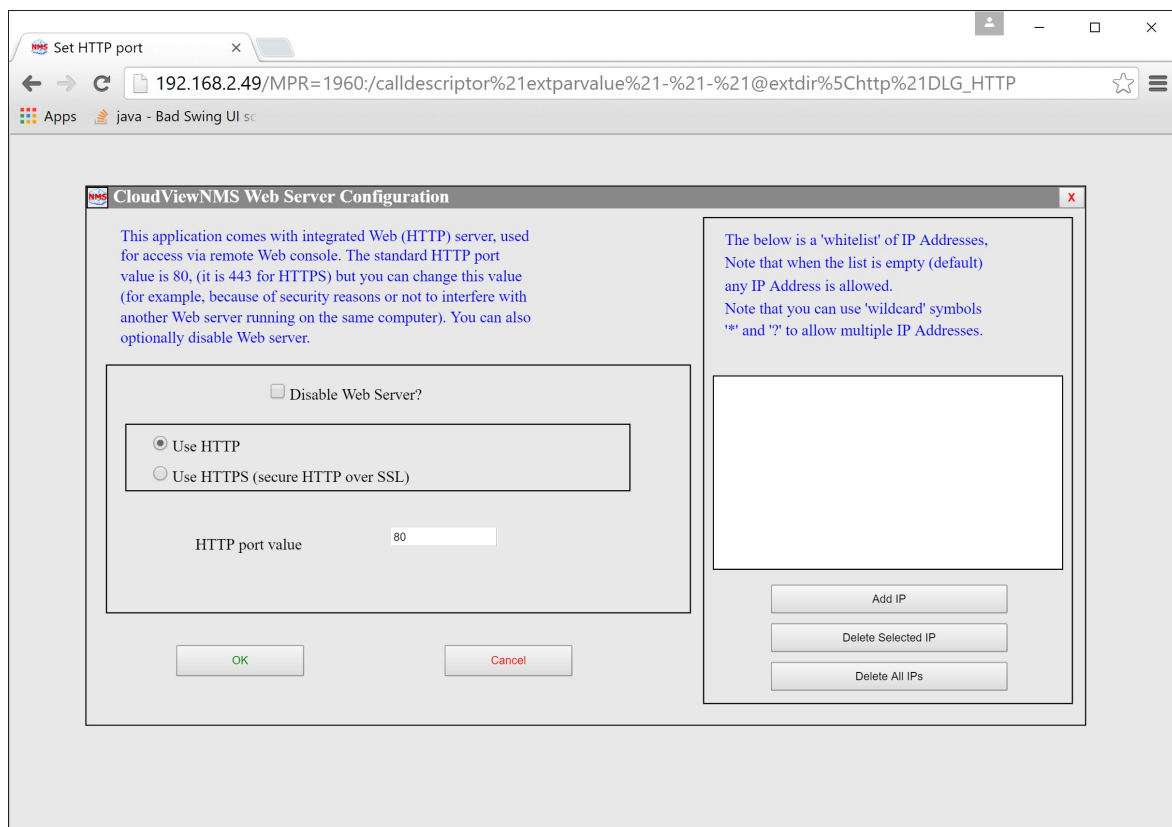
Use *Config->Add/Delete Users Agents* to add more users.



Going forward, notice the Central Office CloudView NMS has the same dialog to configure user accounts. Actually it is the same software, simply Raspberry Pi CloudView serves a "slave" and Central Office CloudView

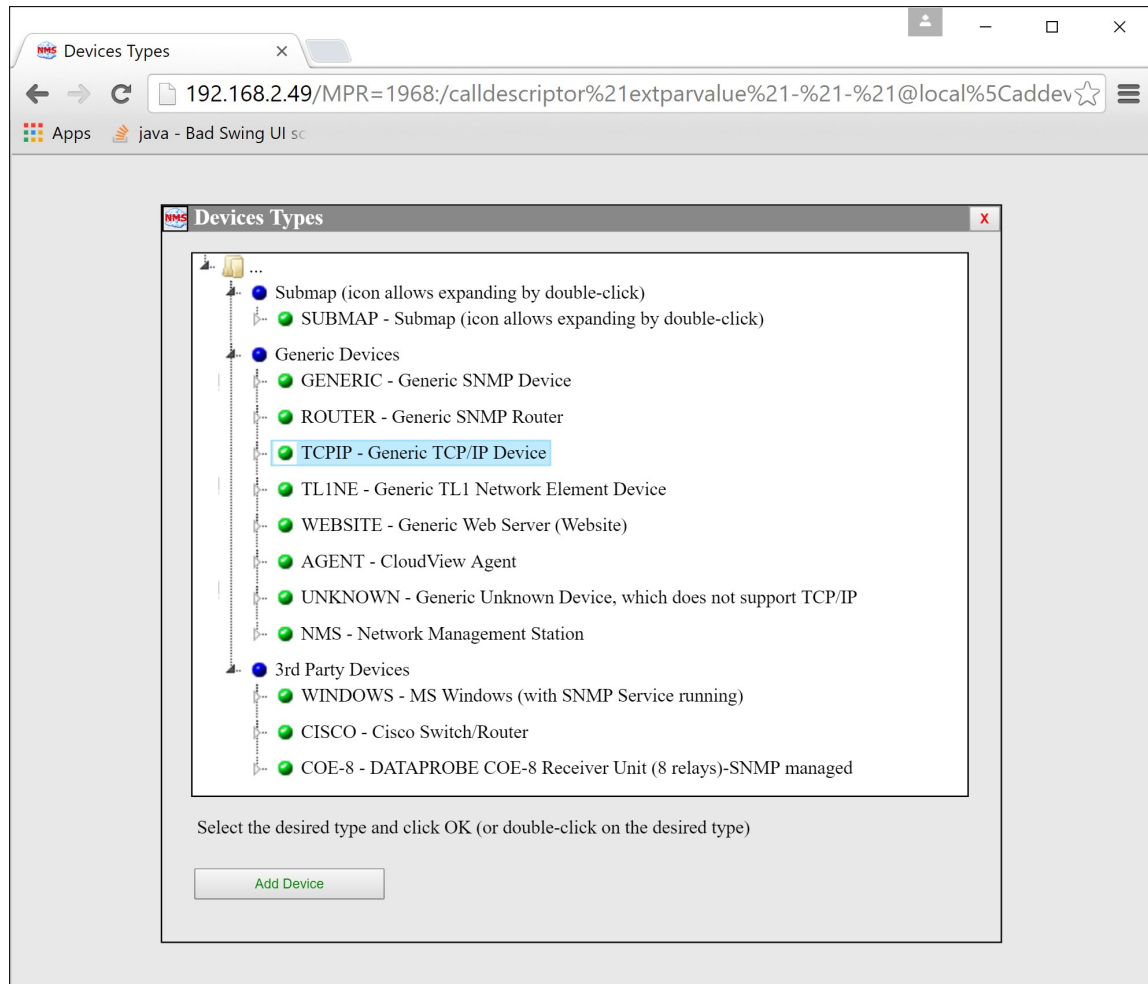
serves as “master” (in menus it is also called “upper NMS”). When connecting to the Central Office CloudView (Upper NMS), Raspberry Pi CloudView must provide credentials (i.e. user name and password) to be accepted by the Central Office CloudView as managed entity. The user name and their corresponding passwords are configured on CloudView NMS Server via **Main Menu->Options->Add/delete Users** Dialog. You can have one special configured user (i.e. username/password) for ALL remote agents or you can still use the same “admin” account for everything. To connect use *Config->Connect Upper NMS* menu item, the corresponding dialog will be discussed later.

Use *Config->Web Server Config* to configure web server for security. We recommend using HTTPS and some other TCP port like 4430.



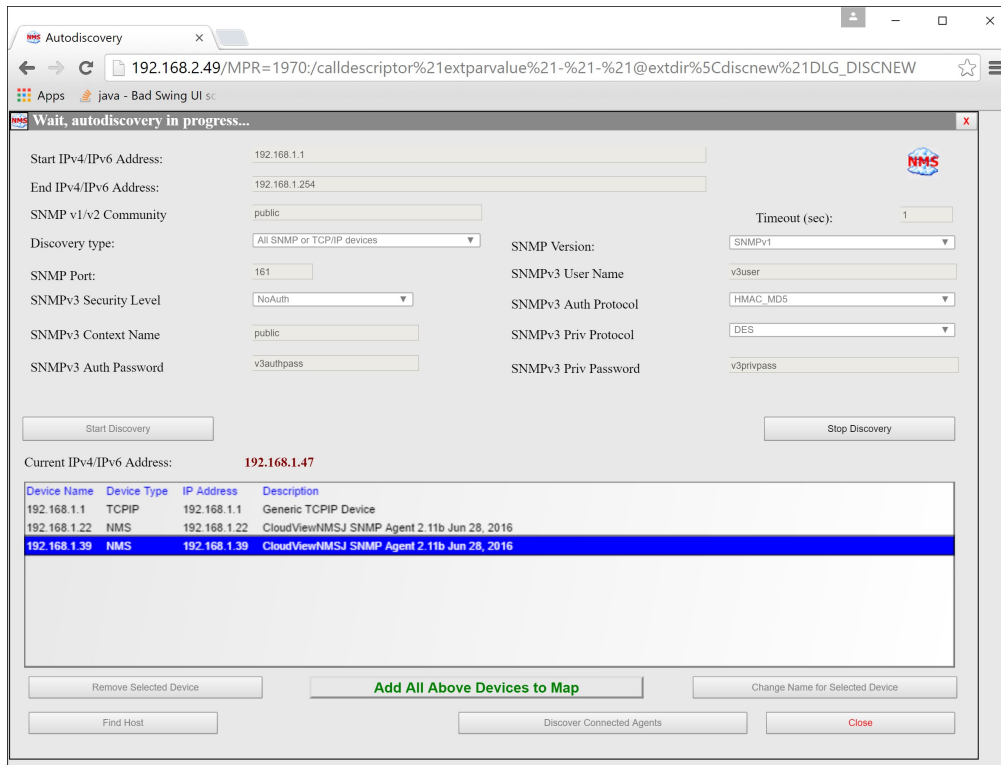
Now let us populate CloudView maps with monitored/managed devices. You can have a complex multiple hierarchical maps structure to represent your network. There are two ways to populate the maps.

The first method is via *Device/Submap->Add Device/Submap* menu item. See the screenshot below.

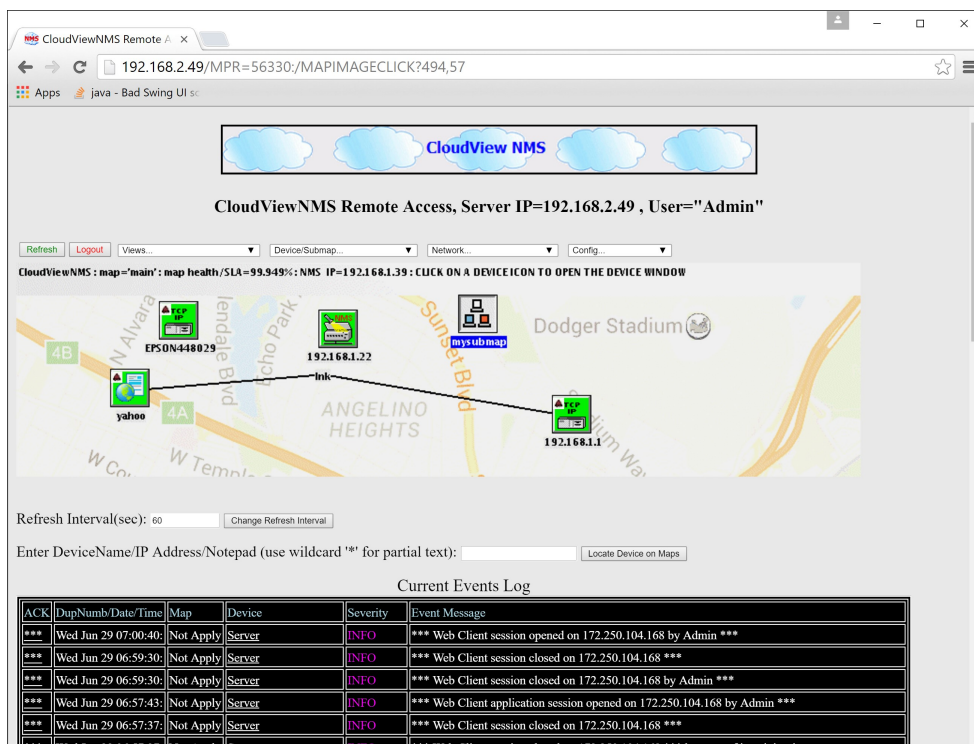


Select the desired device type and click on "Add Device" button. The next dialog will ask for the communication parameters and finally the device icon will be added to the map and the device will be added to the list. See Views->MapView menu item to see the map. Note that you can add background images from Google Maps via Device/Submap->St Map Background Image menu item.

The second way to populate the maps is via Network->Autodiscovery menu item. See the screenshot below.



See the screenshot of the populated map view. Clicking on the submap icon allows navigating to another map. Device icons are colored per their communication status.



CloudView NMS GUI is very rich and you can find many useful functions investigating different menu items. Mostly the dialogs are self-explanatory. The GUI is similar or the same which is described in other CloudView NMS manuals.

One special dialog requires some more explanation. It is *Config->Connect Upper NMS* menu item which is used to connect to the Central Office CloudView NMS. The help text inside the dialog explains the concept.

Enter Server parameters:

This window configures/shows connection(s) to upper NMS Server(s), e.g. 'Central Office(s)'. Upon connection, this NMS appears as agent on upper NMS which allows remotely to see status of the devices, events from 'Central Office'NMS.

So you can create really distributed network management.

With such scenario this NMS can be hidden behind firewall/router/NAT IP and still give access to this NMS info from the 'Central Office'.

This feature is for Service Providers to monitor customer's networks without reconfiguring customer's routers.

***	Address/FQDN	Secure?	Port	ConnectAs
X	192.168.1.22	HTTPS	4430	myraspberrypi

Currently connected: 1

Setrver IP Address: 192.168.1.22
It can be also DNS/FQDN name (like www.company.com)

HTTP/HTTPS: HTTPS(SSL)

Server TCP Port: 4430

User Name: Admin

Password:

This PC Name: myraspberrypi

Connect! Close

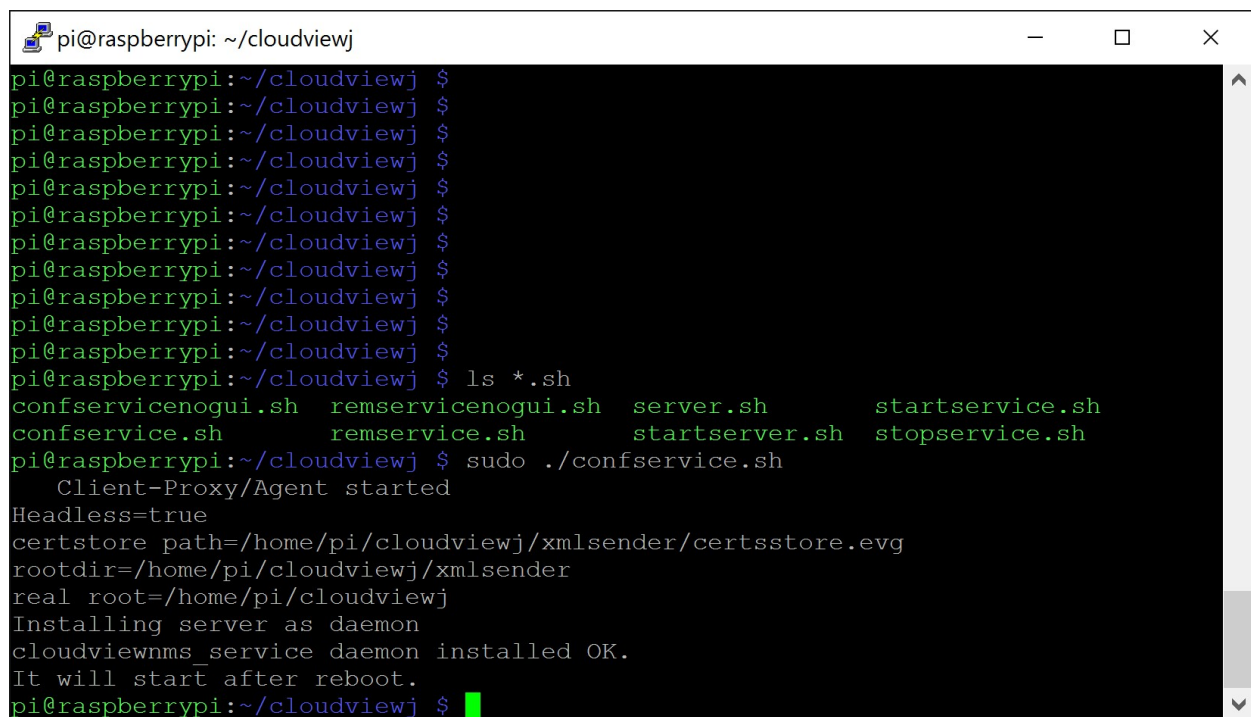
The above connection may be secure connection over public internet. Below is a little bit more info about security. Like most SCADA networks our customers are security sensitive. They are service providers which want securely monitor their remote customers networks. This is how it works. Every customer corporate LAN is protected by its firewall/router from INCOMING connections. The monitored computers, devices, appliances inside LAN use Private IP Address (NAT Address) like 192.168.1.x which is hidden from outside public Internet. However OUTGOING connections are still allowed. That is why people inside LAN can still

browse Internet. The privacy is guaranteed by the corporate firewall/router which "proxies" such connections. This is common scheme which is also used by CloudView NMS serving as agent to the upper CloudView NMS. No special configuration of customer's firewall/router is required: for the router it is no different than client-browser connection. This security concept is used by many reliable systems (including banking systems). For example, "Skype" also works this way. Additionally we use secure over SSL-encrypted connections and **white lists of IP Addresses** which are allowed to be connected (see Web Server Config Dialog).

The Central Office CloudView NMS IP Address/FQDN can be actually public IP Address of the Service Provider router. This router can be configured for TCP **port forwarding**, i.e. to forward port 4430 packets to the private IP Address of the CloudView NMS Server ("Central Office") inside Server Provider LAN. This way Service Provider can have the global unified picture from multiple remote customers on his "Central Office" CloudView NMS.

When you are done with all configuration, run the following script located in the /home/pi/cloudviewj directory

```
sudo ./confservice.sh
```



```

pi@raspberrypi: ~/cloudviewj
pi@raspberrypi:~/cloudviewj $
pi@raspberrypi:~/cloudviewj $
pi@raspberrypi:~/cloudviewj $
pi@raspberrypi:~/cloudviewj $
pi@raspberrypi:~/cloudviewj $
pi@raspberrypi:~/cloudviewj $
pi@raspberrypi:~/cloudviewj $
pi@raspberrypi:~/cloudviewj $
pi@raspberrypi:~/cloudviewj $
pi@raspberrypi:~/cloudviewj $
pi@raspberrypi:~/cloudviewj $
pi@raspberrypi:~/cloudviewj $ ls *.sh
confservicenogui.sh  remservicenogui.sh  server.sh          startservice.sh
confservice.sh       remservice.sh       startserver.sh     stopservice.sh
pi@raspberrypi:~/cloudviewj $ sudo ./confservice.sh
Client-Proxy/Agent started
Headless=true
certstore path=/home/pi/cloudviewj/xmlsender/certsstore.evlg
rootdir=/home/pi/cloudviewj/xmlsender
real root=/home/pi/cloudviewj
Installing server as daemon
cloudviewnms_service daemon installed OK.
It will start after reboot.
pi@raspberrypi:~/cloudviewj $

```

The above will ensure that CloudView NMS Server starts automatically every time your Raspberry Pi reboots. After reboot it automatically connects to the "Central Office" CloudView NMS (if configured like shown above).