Using a Raspberry Pi remotely

9/12/2020

You have a raspberry Pi at home so, now what? This document describes several methods you can use to get going with the Raspberry Pi at home, on a local network system (with an apartment or home router), and with the wider, networked world. The document is organized in local to remote order which may also appear to be simple to complex order. However, none of these methods is terrifically complex although, some of the network concepts are pretty mind-bending and amazing!

Definitions:

There will be breaks in this document to introduce definitions for terms as they are needed. Be on the lookout!

Keyboard, Mouse, HDMI

One of the simplest methods for running the Raspberry Pi in your remote location is the one we have been using in ECE5725 lab. If you have access to a USB keyboard, USB mouse, and HDMI monitor, simply plug these devices in and start the Raspberry Pi. Input will be from the keyboard and mouse, output will be to the HDMI monitor. Clearly, you can run a command line interface to the monitor or issue the startx command to start the desktop. Remember, you can exit the desktop with 'ctrl-alt-backspace' so you can travel between the desktop and command line window.

Note that many Televisions have HDMI inputs for attaching devices such as Bluray players or Roku devices. These HDBI connections can also be used to connect your Raspberry Pi. Simply find a free HDMI port on your TV and connect a cable between the RPi and TV.

PiTFT

Note that if you are using the Raspberry Pi setup from the conclusion of ECE5725 Lab3, the system should boot, with a login screen, to the PiTFT. With only a USB keyboard, you can login and control the RPi from this tiny command line window. This tiny screen isn't too optimal, but it does give us a way to crack into the system to get started with a few commands needed a bit later in this document.

Desktop on piTFT (see 3/20 notes in Lab F19 manual)

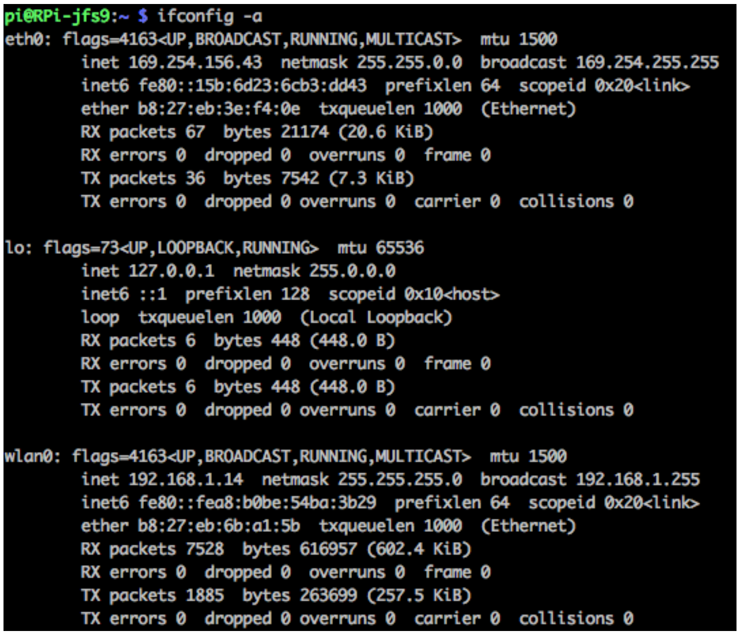
Definitions:

Terminal emulator: Program that displays a command line window on your laptop. Common programs include Putty and Mobaxterm for windows machines and the 'terminal' program included with the OS and iTerm2 application for Macs

Ethernet cable

Another really simple way to control the raspberry Pi is with an Ethernet cable connected to your laptop. In their continued wisdom and eye towards good design, the RPi developers designed a system that enables the use of a standard Ethernet cable (as opposed to a more specialized, reverse wired cable) as a way into the system. Here's how you use an Ethernet cable.

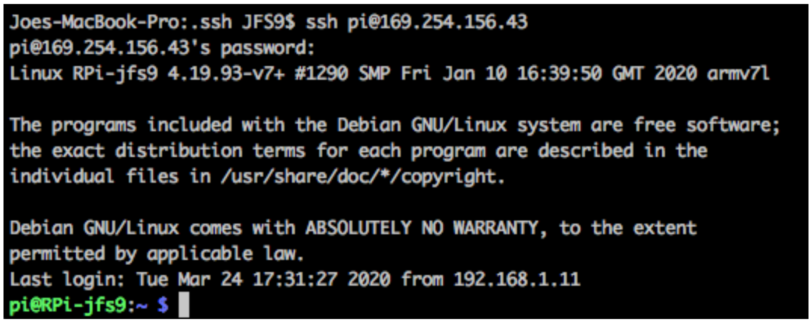
- Find, buy, borrow a spare Ethernet cable
- Connect one end to the Ethernet port on the RPi. Connect the other end to your Laptop
- If you have a USB keyboard connected to your Pi, type ‘ifconfig –a’ and take note of the ip address of eth0.
- Here is an example from my home system:



```
pi@RPi-jfs9:~ $ ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 169.254.156.43  netmask 255.255.0.0  broadcast 169.254.255.255
        inet6 fe80::15b:6d23:6cb3:dd43  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:3e:f4:0e  txqueuelen 1000  (Ethernet)
        RX packets 67  bytes 21174 (20.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 36  bytes 7542 (7.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 6  bytes 448 (448.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6  bytes 448 (448.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.14  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::fea8:b0be:54ba:3b29  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:6b:a1:5b  txqueuelen 1000  (Ethernet)
        RX packets 7528  bytes 616957 (602.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1885  bytes 263699 (257.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 1: ifconfig –a output on RPi

With the Ethernet cable plugged in, note that the eth0 address is 169.254.156.43. Using a terminal emulator, ssh into the RPi. An example on my system:



```
Joes-MacBook-Pro:.ssh JFS9$ ssh pi@169.254.156.43
pi@169.254.156.43's password:
Linux RPi-jfs9 4.19.93-v7+ #1290 SMP Fri Jan 10 16:39:50 GMT 2020 armv7l

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 Mar 24 17:31:27 2020 from 192.168.1.11
pi@RPi-jfs9:~ $ 
```

Figure 2: ssh from laptop to RPi over Ethernet cable connection

And, that’s it, I am logged into the Raspberry Pi and able to use the familiar command line window.

Here’s a photo of the connections; RPi connected to a Macbook using an Ethernet cable:

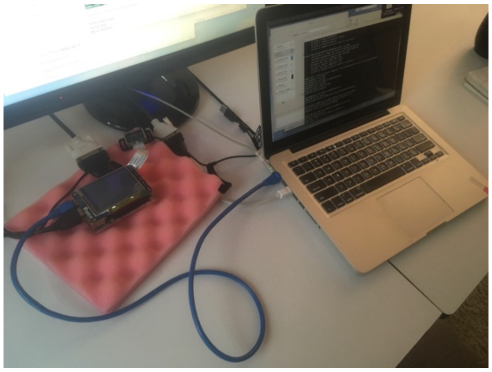


Figure 3: photo of RPi and Macbook connected using Ethernet cable

Definitions:

Virtual Network Computing (VNC): A graphical desktop sharing program that allows control of a remote computer using a remote, shared desktop [1]. Using VNC, you will be able to access the Raspberry Pi desktop on your laptop; pretty amazing! In order to get this going, you will start a VNC server on the RPi and a VNC viewer on your laptop.

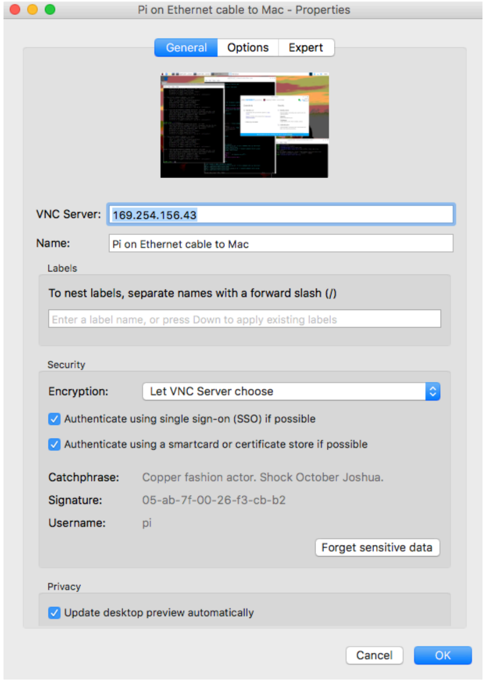
Ethernet cable, continued

To enable this capability on the RPi, run 'sudo raspi-config' select Interfacing options, and enable VNC. Make sure you save your changes. This will start a VNC server on your RPi.

You will also need to start a VNC viewer on your laptop. VNCConnect by RealVNC works well with the VNC server on the RPi. Visit this site:

https://www.realvnc.com/en/connect/dowclearnload/viewer/macos/

To download the VNC viewer for your specific platform. Once you install and run the VNC viewer, you will have to define a new connection. From the tabs at the top of the application, select 'file', 'new connection' and fill in the details. Here are the settings for my RPi connected to the laptop using an Ethernet cable:



Pi on Ethernet cable to Mac - Properties

VNC Server: 169.254.156.43

Name: Pi on Ethernet cable to Mac

Labels

To nest labels, separate names with a forward slash (/)

Enter a label name, or press Down to apply existing labels

Security

Encryption: Let VNC Server choose

☑ Authenticate using single sign-on (SSO) if possible

☑ Authenticate using a smartcard or certificate store if possible

Catchphrase: Copper fashion actor. Shock October Joshua.

Signature: 05-ab-7f-00-26-f3-cb-b2

Username: pi

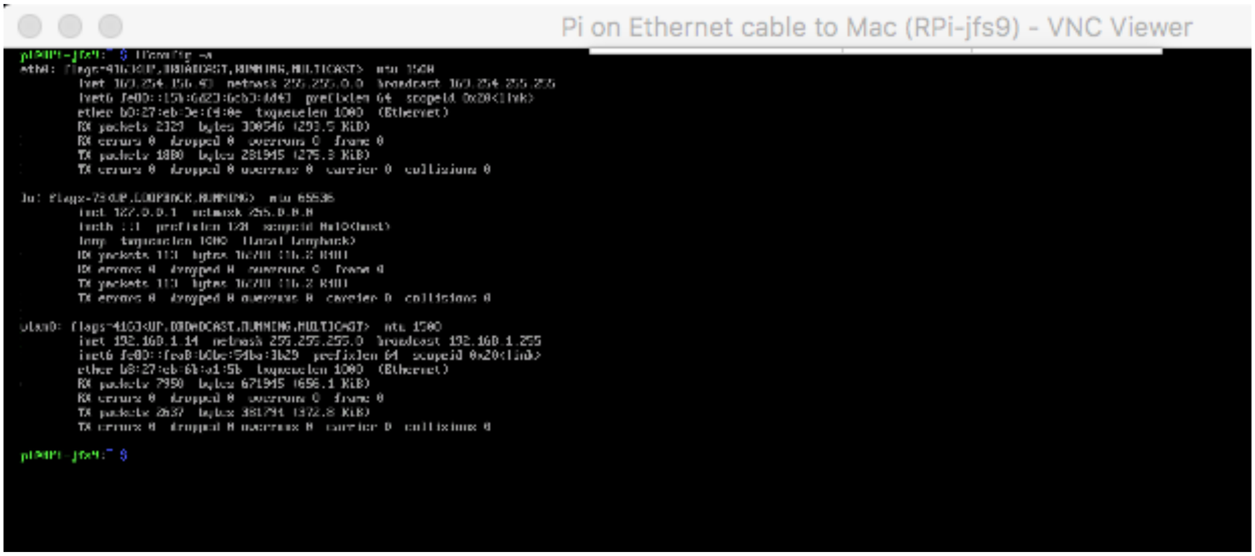
Privacy

☑ Update desktop preview automatically

Figure 4: properties screen in VNC viewer on the laptop

The only details I completed were the ‘VNC Server’ field which is the address of eth0 on the RPi and the Name field (I came up with a name I would remember). All other fields remain as defaults. Note, you can always check settings for any connection file by selecting ‘file’ then ‘properties’

This operation will create an icon in the VNC connect window. Double click the icon and you should see the current screen on you RPi. An example:



Pi on Ethernet cable to Mac (RPi-jfs9) - VNC Viewer

```
pi@RPi-jfs9:~ $ ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 169.254.156.43  netmask 255.255.0.0  broadcast 169.254.255.255
        inet6 fe80::15b:6d23:6cb3:dd43  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:3e:c4:0e  txqueuelen 1000  (Ethernet)
        RX packets 2329  bytes 300546 (293.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1880  bytes 281945 (275.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 113  bytes 16700 (16.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 113  bytes 16700 (16.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.14  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::fea8:b0be:54ba:3b29  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:6b:a1:5b  txqueuelen 1000  (Ethernet)
        RX packets 7950  bytes 671945 (656.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2637  bytes 381794 (372.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

pi@RPi-jfs9:~ $
```

Figure 5: screenshot from laptop of VNC screen on RPi

Pretty standard, as this is yet another command line window. But, if you go into this window and type 'startx' you get:

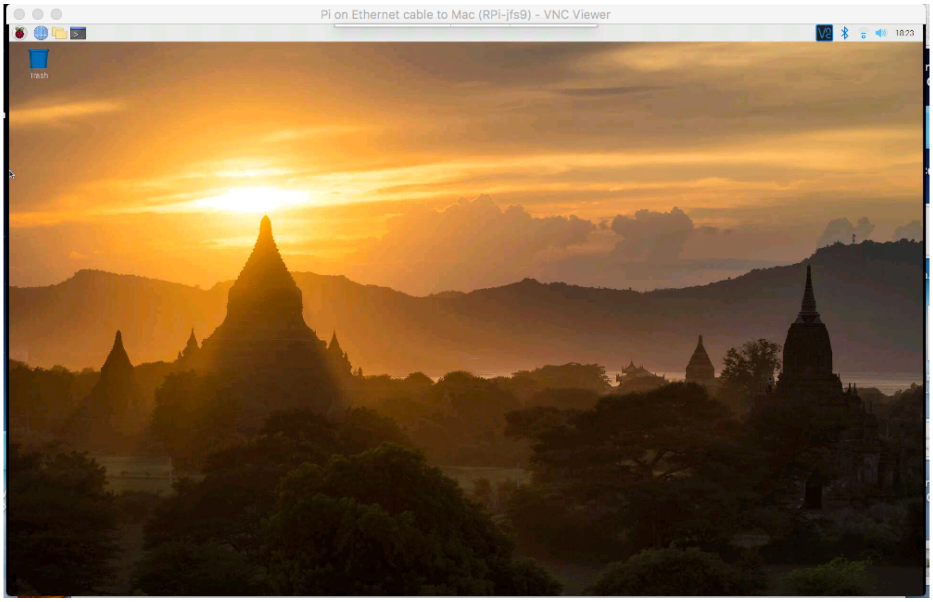


Figure 6: screen shot of VNC screen showing RPi desktop after running startx

Which is the RPi desktop…running and controllable on my Laptop! From this window, you can perform any task that you would normally run on the RPi desktop. This IS the RPi desktop, accessed remotely using VNC on the Pi and laptop.

Notes:

- The laptop keyboard and mouse are used for input to the RPi. The monitor window is displayed on the laptop screen.
- If you run VNC while, simultaneously, having a monitor connected to the RPI, any action you take on the desktop will be shown on the RPi monitor. Also, any action you take on the RPi desktop will be shown on the VNC laptop screen. The RPi desktop is mirrored between the monitor and VNC laptop window.
- This is an amazing feature however, it comes at a cost of speed and display resolution. Since video frames are being sent between the two devices, there is a lot more network traffic associated with VNC that with ssh into a command line window.
- How to take advantage of these two techniques:
    - Use ssh in a terminal emulator to open command line windows for code development and debug
    - Use VNC to run prototype examples of your system; for example, a pygame window running on the desktop displaying a GUI.

PiTFT wifi script

More to come

Part 2: External WIFI

All of the above techniques are important and useful as they cover techniques for using the RPi on a remote, home network system. Next, we consider how to interact with the RPi when across different remote network islands. One issue with using a RPi across networks is that local routers may introduce firewalls which prevent devices connected to the network from being accessed outside this network. In general, this is why you may not be able to ssh to a RPi connected to your home network from outside this network. Try it: hook up the RPi to your network and you should be able to use all the techniques above to access the RPi from your laptop; as long as the laptop and RPi remain on the home network. Now, if you connect your laptop to an external network (like Starbucks, in normal times!), you probably won’t be able to access the RPi connected to your home network at all.

So, we need to do a bit more to access the RPi from outside the home network. There are techniques involving opening ports on your home router, but these are complicated with important security issues. If you miss these, you run the risk of opening your home network to attack.

There is one method that presents itself that is straightforward to implement and takes advantage of existing class infrastructure.

Definitions:
SSH Tunneling: Using a secure SSH connection to transport arbitrary network data.

SSH Reverse Tunneling: Reverse SSH tunneling allows a connection between a remote computer and a local computer when the established SSH connection is between the local to remote computer. Confusing! A diagram may help:

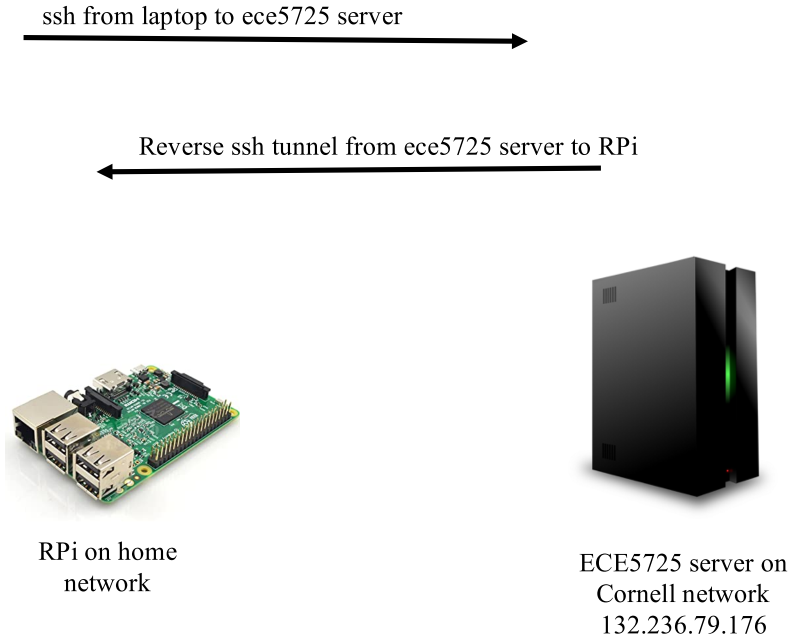

ssh from laptop to ece5725 server
Reverse ssh tunnel from ece5725 server to RPi
RPi on home network
ECE5725 server on Cornell network 132.236.79.176

Figure 7: RPi, ece5725 server and their respective connections. ssh connection from RPi to server and ssh reverse tunnel from server to RPi are illustrated

OK, even with the figure this is still a bit mind-bending! Some notes:

- Everyone in ece5725 can ssh to the ece5725-f20 server:
    - I setup the server with a static Cornell Ethernet address.
    - Our colleagues at CIT assigned an address (132.236.79.205) that is accessible both on-campus and remotely.
    - From your remote location, you should be able to log into the server and access your account and files. You have probably already logged into the server from home using your laptop connected to the home network. If not, try it!
    - Once your RPi is connected to the home network, you should be able to login to the ece5725 server from the RPi. Try this as well!
- OK, so far, pretty straightforward: this is all known from previous work in the class.
- What you will not be able to do, at the moment, is to log into your RPi (or laptop) connected to your home network FROM the ece5725 server.
- You can try this once you are logged into the server. Try

  ```
  ssh pi@my_RPi_home_address
  ```

  from the window on the server. This will probably time out after trying, and failing, to make a connection.

- BUT, we know there is a secure, ssh connection from the RPi on the home network to the ece5725 server.
- Is it possible to take advantage of this secure connection….in reverse? Can we use this secure channel to allow the server to access the RPi on the home network?
- YES, with reverse tunneling. This is shown in the diagram. ssh allows us to take advantage of the secure channel to access the RPi on the home network

This is a lot of description for a few commands (sorry!). Let's set it up:

- From a command-line window on the RPi, I issue the command:

  ```
  ssh –R 1066:localhost:22 jfs9@132.236.79.205
  ```

  - you will recognize this as an ssh command into the server using my netid, jfs9, to signon. In your case, you would use your own netid in this ssh command
  - the –R flag is the 'reverse option'. This allows a new ssh session from the server to establish a session on the RPi
  - `1066:localhost:22` states that an ssh session on the ece5725 server connecting to port 1066 will be forwarded to port 22 on the RPi.
  - Port 22 is the port used to connect ssh sessions.
  - Port 1066 is an example unused port in Linux, so we can grab it for the reverse ssh tunnel. Please see 'Notes' below for a further discussion of free ports.

- Next, on your laptop, use a terminal window to login to the ece5725 server
- Once logged into the server, issue the command:

  ```
  ssh pi@localhost -p 1066
  ```

  This command will log into the RPi using port 1066

Example from the Terminal window:

```
# from the laptop terminal window, log into the ece5725 server

Joes-MacBook-Pro:~ JFS9$ ssh jfs9@132.236.79.205
jfs9@132.236.79.205's password:
Linux ece5725-s20 4.19.93-v7+ #1290 SMP Fri Jan 10 16:39:50 GMT 2020
armv7l

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: Wed Mar 25 13:32:21 2020 from 67.251.100.246
```

```
# Once logged into the ece5725 server, use the reverse ssh
# command to log into the RPi

jfs9@ece5725-s20:~ $ ssh pi@localhost -p 1066
pi@localhost's password:
Linux RPi-jfs9 4.19.93-v7+ #1290 SMP Fri Jan 10 16:39:50 GMT 2020
armv7l

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: Wed Mar 25 13:08:52 2020
pi@RPi-jfs9:~ $ ls
```

After this final step, I have successfully logged into the RPi on my home network from the ece5725 server

The Server hop

So, big deal. This may seem like a lot of jumping between the RPi, the server and the laptop for no good reason. But, consider that the last experiment allowed you to log into the RPi on your home network from the server. Here’s an updated diagram:

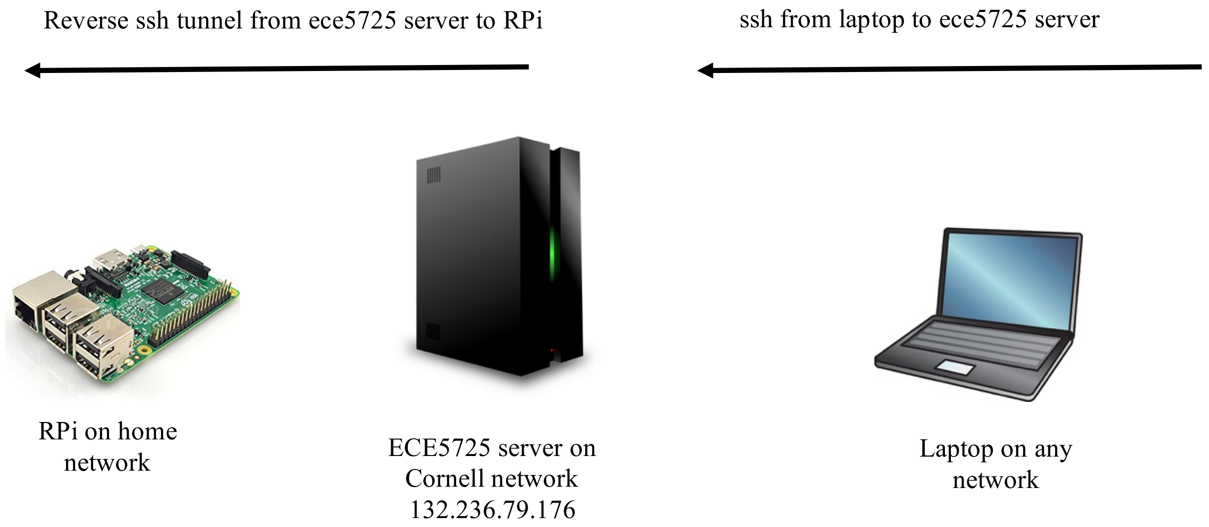

Reverse ssh tunnel from ece5725 server to RPi
ssh from laptop to ece5725 server
RPi on home network
ECE5725 server on Cornell network 132.236.79.176
Laptop on any network

Figure 8: RPi, laptop, ece5725 server and their respective connections. ssh connection from laptop to server and ssh reverse tunnel from server to laptop are illustrated

This means that, once the reverse ssh tunnel is setup between the RPi and the ece5725 server, a laptop on any network can ssh into the server (server hop number 1) and then ssh from the ece5725 server into the RPi using the reverse tunnel connection (server hop number 2)

In short, this gives us the ability to log into an RPi connected remotely to a home network using a laptop connected to a completely different remote home network. So, if you are working with your RPi connected to your home network in Ithaca, your lab partner, in another state, can log into the RPi remotely using this method

Notes:

- This is possible because:
    - The ece5725 server is connected to a publically available, static IP address
    - ssh give us the capability to establish a reverse ssh tunnel between the ece5725 server and the RPi
    - A remote laptop connected on a different home network has the ability to reach the server
- The example port address, 1066, is not the only free port available in Linux. In fact, each unique reverse ssh tunnel should use a unique, unassigned port address.
- We will be using the port range between 38000 and 39999.
- A future step will be to assign free port ranges to teams so there are no ‘collisions’ using this technique.
- For the moment, this method gives you the capability of accessing the Pi remotely with a command line window.

Direct desktop using RealVNC signon

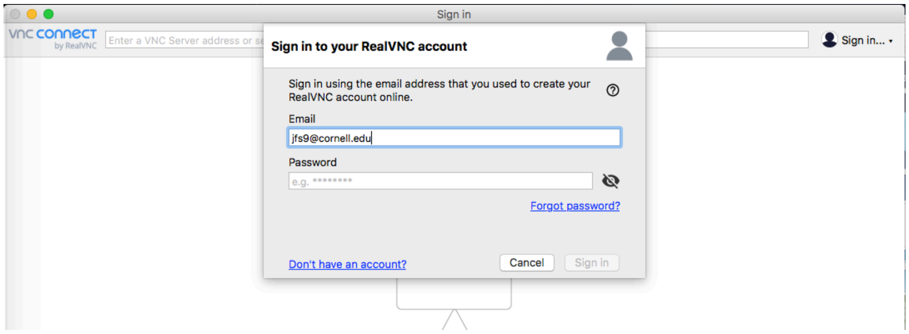
So far, techniques have covered connection to the RPi using direct methods (monitor or Ethernet cable), ssh and ssh reverse tunneling. All of these methods are quite secure as they depend only on local access or ssh which includes dual key encryption.

There is a solid method that uses the RealVNC techniques (discussed earlier in the document. If you skipped ahead, please go back and review the VNC section). By creating a RealVNC login, you can display the desktop on the Raspberry Pi using a remote laptop or desktop. It is very straightforward to get started. The steps are:

- Enable the VNC server on the Raspberry Pi (using sudo raspi-config)
- Visit realvnc.com and register for a signon using your favorite email address
- Once you have the signon and password established, go to the vnc icon on the RPi (upper right hand side of the startx desktop). Right click and go to ‘licensing’
- Enter the email address and PW you just setup in the pop-up box
- When prompted, make sure to click ‘Direct and Cloud Connectivity’

NEED SCREEN SHOTS OF STEPS HERE!; To be added

- On your laptop, using the VNC viewer, sign in using the credentials you created. Here's a snip from my laptop:



Sign in to your RealVNC account

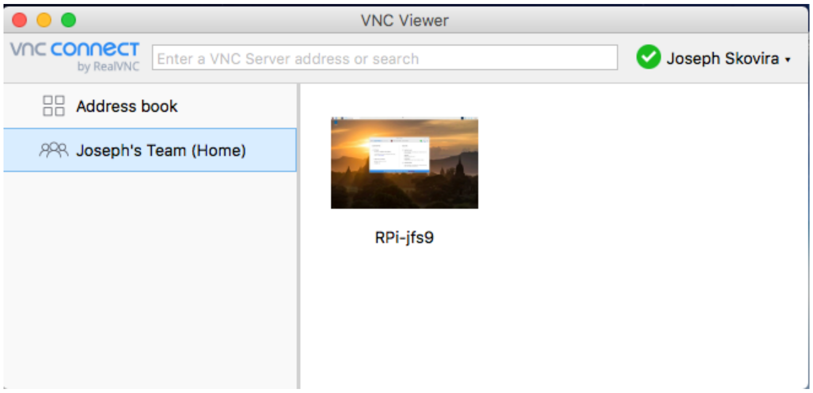
Sign in using the email address that you used to create your RealVNC account online.

Email: jfs9@cornell.edu

Password: e.g. ********

Note the 'Sign In' icon in the top right hand side of this screen. Your initial signon will result in some confirmation emails from the RealVNC service. Read these carefully to make sure all is well.

- Once you sign in, you will see a link to your RPi on your laptop VNC viewer. Here is an example:

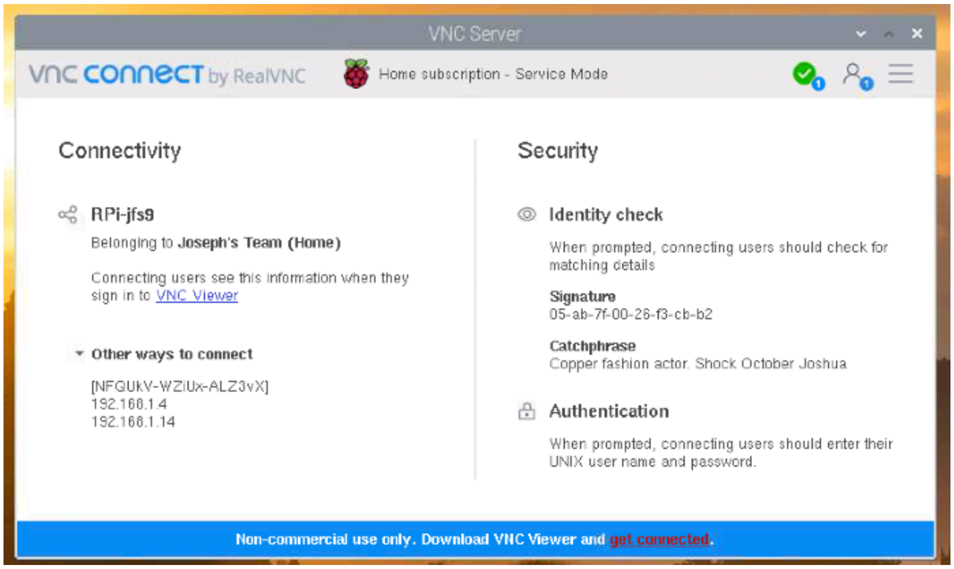


- Double click this icon and you should see the raspberry pi desktop.
- Note, the above steps were condensed from the Reference, [4]

Notes:

- This is a straightforward method to view the desktop on the RPi from a remote location
- One downside is that cloud connections are used through the RealVNC app so I suspect this is a bit less secure than the ssh methods discussed so far.
- Note the 'Identity Check' section. The signature and catch phrase will pop up when you use the VNC Viewer to connect to the RPi. These credentials SHOULD MATCH between the RPi and the VNC Viewer signon page! If they do not, you probably shouldn't proceed as your communication may be routed elsewhere (bad!)

Here is a snip from the VNC Server page on my RPi. Note the name of my Raspberry Pi (RPi-jfs9) and the 'identity check section'.



Connectivity

RPi-jfs9

Belonging to **Joseph's Team (Home)**

Connecting users see this information when they sign in to VNC Viewer

Other ways to connect

[NFGUkV-WZiUx-ALZ3vX]
192.168.1.4
192.168.1.14

Security

Identity check

When prompted, connecting users should check for matching details

Signature
05-ab-7f-00-26-f3-cb-b2

Catchphrase
Copper fashion actor. Shock October Joshua

Authentication

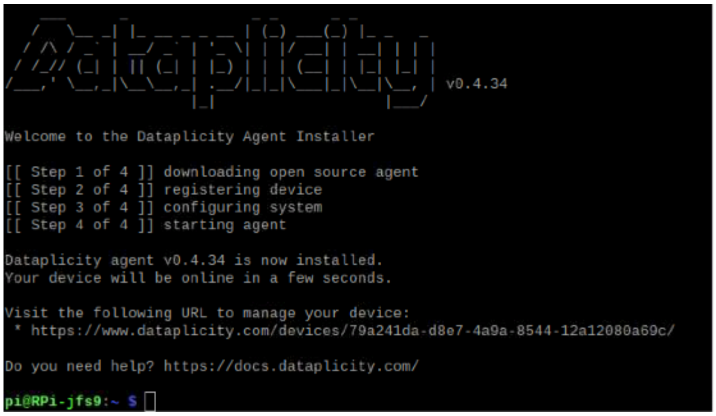
When prompted, connecting users should enter their UNIX user name and password.

Non-commercial use only. Download VNC Viewer and get connected.

Console Window Access with Dataplicity

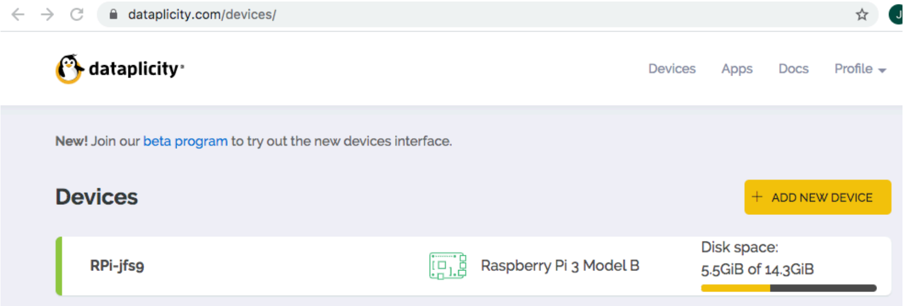
Similar to the cloud access techniques used with RealVNC to access the RPi desktop, another cloud provider, Dataplicity may be used to gain access to a remote RPi through console windows:

- Visit dataplicity.com
- Using your favorite email address, register for an account
- You will receive a verification email and must setup a password
- Once you do, you will be returned a command to cut and paste into the RPi
- Once you paste and run the command, Dataplicity will install and configure some code. Here is the result of the command on my RPi:



```
    ___      _              _ _      _ _
   /   \__ _| |_ __ _ _ __ | (_) ___(_) |_ _   _
  / /\ / _` | __/ _` | '_ \| | |/ __| | __| | | |
 / /_// (_| | || (_| | |_) | | | (__| | |_| |_| |
/___,' \__,_|\__\__,_| .__/|_|_|\___|_|\__|\__, | v0.4.34
                     |_|                   |___/

Welcome to the Dataplicity Agent Installer

[[ Step 1 of 4 ]] downloading open source agent
[[ Step 2 of 4 ]] registering device
[[ Step 3 of 4 ]] configuring system
[[ Step 4 of 4 ]] starting agent

Dataplicity agent v0.4.34 is now installed.
Your device will be online in a few seconds.

Visit the following URL to manage your device:
 * https://www.dataplicity.com/devices/79a241da-d8e7-4a9a-8544-12a12080a69c/

Do you need help? https://docs.dataplicity.com/

pi@RPi-jfs9:~ $
```

- Once this is complete, visit the Dataplicity device page: https://www.dataplicity.com/devices/
- Here, the RPi will be displayed. Click the RPi link and signon. Here is the device page for my RPi:

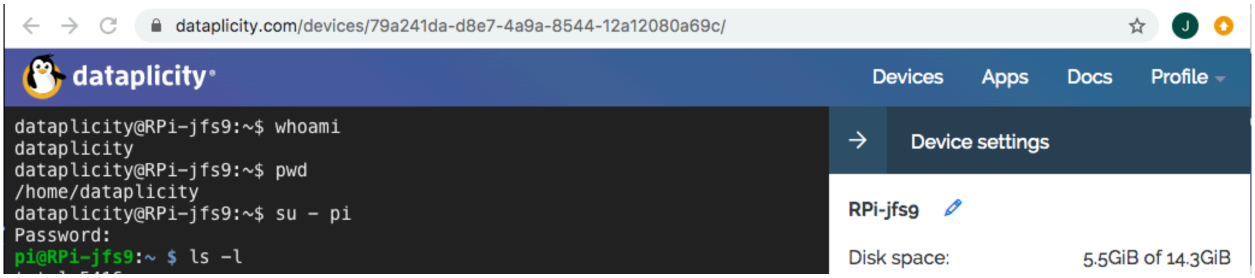


New! Join our beta program to try out the new devices interface.

Devices

RPi-jfs9 Raspberry Pi 3 Model B Disk space: 5.5GiB of 14.3GiB

- Once you click the device name, you will be logged in immediately with a ‘Dataplicity’ account. You will want to immediately run ‘su – pi’ to switch to the pi User on the RPi. An example:


dataplicity.com/devices/79a241da-d8e7-4a9a-8544-12a12080a69c/
dataplicity
Devices
Apps
Docs
Profile
Device settings

```
dataplicity@RPi-jfs9:~$ whoami
dataplicity
dataplicity@RPi-jfs9:~$ pwd
/home/dataplicity
dataplicity@RPi-jfs9:~$ su - pi
Password:
pi@RPi-jfs9:~ $ ls -l
```

RPi-jfs9

Disk space: 5.5GiB of 14.3GiB

And now you have access to the RPi through a console window. Note that Dataplicity also uses cloud services so some of your access information is leaking out into the cloud. Be careful to read and follow all of the confirmation messages from Dataplicity to protect access to your device. [4] [5]

Port Forwarding on home WIFI Router

If you have access to a home WIFI router and are comfortable with changing settings, you can use port forwarding to achieve remote access to the RPi. A link to these methods may be found here: https://www.lifewire.com/how-to-port-forward-4163829

Clearly, you will not be able to use WIFI outer port forwarding if you do not control you WIFI router (for those of you in apartments, for example). In addition, this technique comes with a number of security concerns that you must take into account. However, the benefits are that you can more easily access the RPi remotely by using the open port on the router. [6]

References and Notes:

[1] wikipedia.org/wiki/Virtual_Network_Computing
[2] Reference for VNC viewer download:
https://www.realvnc.com/en/connect/download/viewer/macos/
[3] Realvnc signon for cloud connection to local RPi
https://www.realvnc.com/en/raspberrypi/
[4] Real VNC and Dataplicity discussion:
https://lifehacker.com/how-to-control-a-raspberry-pi-remotely-from-anywhere-in-1792892937
[5] Thanks to ECE5725 students Iman Nandi and Tyler Sherman for independently suggesting Dataplicity
[6] Thanks to ECE5725 student Tyler Sherman for sharing techniques he has used to setup port forwarding on a home system.