Networking

Useful References

1. [YouTube]***Introduction of Computer Networks***: https://youtu.be/3QhU9jd03a0
    - *"Crash Course Computer Science"* series are well designed. You are also encouraged to watch other videos of the channel.
2. **!!!! Google**, Baidu, ...

IP Basics [*IPv4* vs. *IPv6*]

- When you connect to the Internet, you are designated with an IP (*Internet Protocol*). It is like your ID card number, which helps the other to identify who you are and allows you to access the Internet with it.

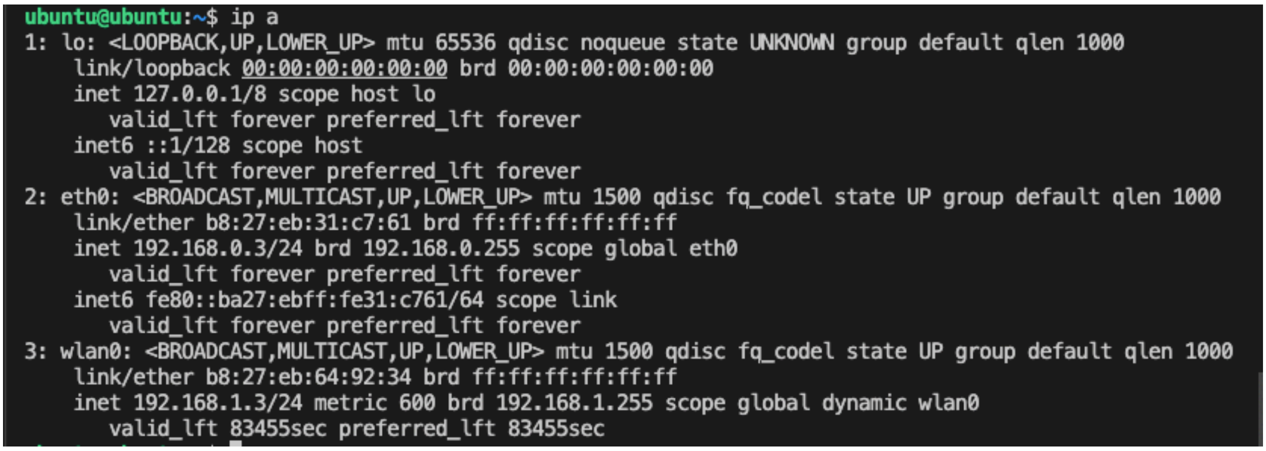
- IPv4 is the Internet Protocol of the 4th version, which has the format of 4 * 8-bits number; the numbers are separated by `.` (***32 bits / 4 bytes*** per address in total). It allows 4.19 billion (2^32) unique addresses for usage. For the convenience and simplicity, each number is represented in the decimal form (e.g. 192.168.1.1).
    - When the device connects to the Internet, it can be automatically assigned by DHCP or mannually given by the users.
- IPv6 is the 6th version, which is created for modern applications and has the format of 218 bits number. It allows 2^218 unique addresses for usage, much more than IPv4 can provide. This binary form is converted into hexadecimal form, where every 4 hexadecimal numbers are separated with a `:` (e.g. 3ffe:1900:fe21:4545:0000:0000:0000:0000, where each number is hexadecimal).
    - It supports automatic network-configuration
    - It is also more expensive and requires more space to operate and store the addresses.

- Although IPv6 provides more available addresses and more applied to modern devices, it is not feasible when we apply IPv6 to some older architectures. Thus IPv4 is still in use, because it is more managible when you are trying to build upon something more basic rather than degrading something more advanced.
- Refer to https://community.fs.com/blog/ipv4-vs-ipv6-whats-the-difference.html (webpage), https://www.youtube.com/watch?v=aor29pGhlFE (YouTube) and other websites to know more details. They are not the core focus of this workshop, but we still encourage you to learn more!

[Hand-on] Communication between two Respberry Pi

- **WARNING**: You will need two Raspberry Pi in this section.
- If You cannot ssh to any of the device and physical connection is not applicable, we suggest you to find a screen and keyboard to manipulate

Find the dynamic IP for SSH

1. Connect a Raspberry Pi board to your computer with a physical cable
2. The preset static IP for ethernet is `192.168.0.8`
3. Open terminal and enter `ssh ubuntu@192.168.0.8` and enter the password (given in the previous section)
4. Enter `ip a` or `sudo ifconfig`, you will see:
    - 

      ```
      ubuntu@ubuntu:~$ ip a
      1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
          link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
          inet 127.0.0.1/8 scope host lo
             valid_lft forever preferred_lft forever
          inet6 ::1/128 scope host
             valid_lft forever preferred_lft forever
      2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
          link/ether b8:27:eb:31:c7:61 brd ff:ff:ff:ff:ff:ff
          inet 192.168.0.3/24 brd 192.168.0.255 scope global eth0
             valid_lft forever preferred_lft forever
          inet6 fe80::ba27:ebff:fe31:c761/64 scope link
             valid_lft forever preferred_lft forever
      3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
          link/ether b8:27:eb:64:92:34 brd ff:ff:ff:ff:ff:ff
          inet 192.168.1.3/24 metric 600 brd 192.168.1.255 scope global dynamic wlan0
             valid_lft 83455sec preferred_lft 83455sec
      ```
    - `eth0` indicates the interface for ethernet (the network that requires physical cable). In the following steps, we are trying to establish connection between `eth0` of 2 different Raspberry Pi. The IP address (IPv4) following the keyword `inet` (`192.168.0.3/24` in this picture) is the **static** IP we have already configured for you, which will require a slight modification later.
    - `wlan0` indicated the interface the wireless connection. It is the interface where your device can access the campus wifi. The IP address (IPv4) after the keyword `inet` (`192.168.1.3/24`) is one assigned by the DHCP (a network management protocol that automatically assign IP to a device once it connects to the Internet); as the following words - "**dynamic** wlan0" - suggests that the IP may be changed if the device reconnects to the Internet in the future.
5. You can use the dynamic `wlan0` IP to ssh the device remotely.

Get static IP prepared for each device

1. Use a physical cable to connect two Raspberry Pi boards.
2. Before any advanced steps, you should look up to network status of this device, enter: `sudo ip a` or `sudo ifconfig`
    - **Warning**: If you see `DOWN` in the description following `eth0` , use the command `sudo ifconfig eth0 up` to turn it up.
3. (**Option 1**) Enter the following command to edit a config file: `sudo vim /etc/netplan/50-cloud-init.yaml`:
    - **NOTE**: `vim` is a text editor, which has its own syntax to operate properly; you can refer to: https://www.geeksforgeeks.org/getting-started-with-vim-editor-in-linux/
    - Here are some commonly used commands:
        - `<shift> + i`: insert mode
        - Press `<esc>` to leave insert mode and shift back to visual mode
        - `q!` or `q` : exit without saving the modification

        - `wq`: save the modification and exit

    (**Option 2**) Or you can choose `nano`, another text editor that is more similar to *Word* or *Page* we commonly use; thus the command should be `sudo nano /etc/netplan/50-cloud-init.yaml`

    - Press `<esc>` to exit and there will be quick instructions on the bottom of the screen. You can simply follow it, but DO READ THE INSTRUCTIONS CAREFULLY
    - Here are some commonly used commands:
        - `<ctrl> + X` to exit
        - When you are asked whether to change the file name, you can change the name if you refer; press `<enter>` to confirm the name and exit

4. You will see the following content:

    ```
    network:
      version: 2
      render: networkd
      ethernets:
        <ethernet_name>:
          dhcp4: no
          addresses:
            - <ip_address of this device>/24
          routes:
            - to: default
              via: <the ip_address of another device you connect to>/24
          nameservers:
            addresses:
              - 8.8.8.8
              - 8.8.4.4
    ```

    - change the `<ip_address of this device>` and make sure that two devices have different IPs. It should be in **ipv4** format and will not change if you reconnect the cable.
    - In particular, we suggest you to use `192.168.0.*`, when `*` can be any number between 1 and 255, as long as it is different from your IP address of **wlan**
        - **NOTE**: for example, my **wlan0** has an IP address of `192.168.1.3`, my `<ip_address of this device` might be `192.168.0.3` for clarity.
    - **NOTE**: In this step, we meant to set up ***static IP addresses*** for both devices. It allows them to identify and connect to each other WITHOUT the Internet.

5. Save the file and exit;
6. Enter `sudo netplan try` to check the connection and potential syntax error
7. Enter `sudo netplan apply` to apply the configuration
8. Enter `sudo reboot` to restart the device
9. **After BOTH devices are configured**, `ping <ip_address of the other device` to check if the configurations work properly
    - For example, device 1 has a static ip `192.168.0.2` and device 2 has `192.168.0.3`
    - On the terminal of device 1, I enter `ping 192.168.0.3` to call the other device; on the terminal of

    device 2, I enter `ping 192.168.0.2` to call device 1.
    - If you want to exit, press `<ctrl> + C`
10. Enter `sudo wondershaper eth0 1024 1024` on both devices
    - To **fix the upload and download rate,** use `wondershaper` and fix the size to `1Mbps`
11. Use **iperf3** to send informatio between two devices:
    - It is similar to `ping`, and should be already installed
    - `man iperf3` or `man iperf` to check the installation
        - if it is not installed, enter `sudo apt-get install iperf3`
    - On device 1, enter `iperf3 -s -i 1 -p 1314` to let device 1 work as a server and waits for the packets transmitted.
    - On device 2, enter `iperf3 -c <IP address of eth0 of server> -i 1 -t 60 -p 1314`. So this device works as a client and send packets to the server device based on these parameters.
    - Now, on the server device, you can see the information of the packet transmitted if things work.
    - Refer to https://zhuanlan.zhihu.com/p/314727150 for more parameters and explanations; or you can simply `man iperf3`