Installing and Setting up Ubuntu

Pieter P

Installation

1. Download the Ubuntu Server image from https://ubuntu.com/download/raspberry-pi

```
$ wget -P /tmp/ http://cdimage.ubuntu.com/releases/focal/release/ubuntu-20.04-preinstalled-server-arm64+raspi.img.xz
```

2. Then write the disk image to your SD card

```
$ xzcat /tmp/ubuntu-20.04-preinstalled-server-arm64+raspi.img.xz | sudo dd of=/dev/mmcblk0 bs=32M
```

   Be careful when using `dd`, if you specify the wrong output (`of=...`), it'll overwrite everything on that device, so it's easy to accidentally destroy your computer.

3. Install the SD card into the Raspberry Pi, connect it to a network using Ethernet, and power it up.
   If you don't have an Ethernet network to connect the Pi to, you can configure the WiFi first, as explained here: WiFi Setup

4. Use your favorite method to find its IP address. You can find it in your router's settings, with an app like Fing, or by using a command line tool like `nmap`. If you have a router that automatically adds the hostnames of the devices on the local network to its IP records, you might be able to just use the hostname `ubuntu`, without needing to find its IP address manually.

5. Connect to the Raspberry Pi over SSH

```
$ ssh ubuntu@192.168.1.100
$ ssh ubuntu@ubuntu # if you have a smart DNS server in your router
```

   Replace the IP address with the one you found in the previous step.
   You'll be prompted a password, the default one is `ubuntu`.

6. Follow the instructions to change the default password and connect again with the new password.

SSH Configuration

1. Set the hostname:

```
pi $ sudo hostnamectl set-hostname rpi3
```

2. Install the `avahi-daemon` package to enable mDNS:

```
pi $ sudo apt install avahi-daemon
```

3. If you already had the `avahi-daemon` installed, you have to restart it to use the new hostname:

```
pi $ sudo service avahi-daemon restart
```

4. Close the SSH connection:

```
pi $ exit
```

5. You should now be able to reach the Pi using its mDNS hostname:

```
$ ping rpi3.local -c3
```

6. Create an SSH configuration for the Pi on your computer, so you can connect to it without having to specify the hostname or username:

```
$ cat >> ~/.ssh/config << 'EOF'
  Host RPi3
      HostName rpi3.local
      User ubuntu
  EOF
```

7. Add your public key to the Pi's `authorized_keys`, so you can connect to it without entering the password each time:

```
$ ssh-copy-id -i ~/.ssh/id_rsa.pub RPi3
```

If you don't have an SSH key pair yet, you can follow these instructions on how to create one: DigitalOcean - How to Set Up SSH Keys.

8. You can now try to connect to it without having to specify the hostname or username, and without having to enter your password:

```
$ ssh RPi3
```

9. For security reasons, it's a good idea to disallow password login. Edit the `/etc/ssh/sshd_config` file:

```
pi $ sudo nano /etc/ssh/sshd_config
```

Locate the line `PasswordAuthentication yes`, and replace it with `PasswordAuthentication no`. Then save the file and exit the editor using `Ctrl+X`.
Finally, restart the SSH server to apply the settings:

```
pi $ sudo service ssh restart
```

Now you'll only be able to log into the Pi using the SSH key we installed in step 7.