

Setting Up Ubuntu Core on a Raspberry Pi with ROS

Laboratory of Robotics - LaR

Universidade Federal da Bahia - 2019

This tutorial assumes you have a machine running Ubuntu. Everything can be done in a Windows/Mac machine, but then the software to create the image and maybe the commands to access the Ubuntu Core may differ.

1 Ubuntu Core Image on a SD Card

You can download the OS image in [here](#).

2 Download Ubuntu Core

Get the correct Ubuntu Core image for your board:

- [Ubuntu Core 18 image for Raspberry Pi 2](#).
- [Ubuntu Core 18 image for Raspberry Pi 3](#).
- You can verify the integrity of the files using the [SHA256SUM](#) and [SHA256SUM.gpg](#) files.

Figura 1: Screenshot from the download page from Ubuntu One

After downloading it, make sure to extract the image from the compressed file. Make sure to grab your SD card, connect to the computer and use the *Startup Disk*

Creator from Ubuntu and create a bootable SD card with the image you downloaded.

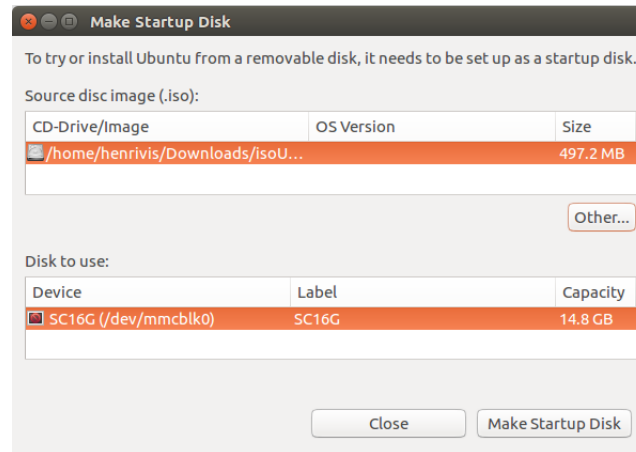


Figura 2: StartUp Disk Creator menu

You could do with the command line as well or other software as you wish.

1.1 Configuring your SSH

Ubuntu Core has no graphical interface, so in the long run in order to access it we're going to use SSH, *i.e* command line. If you haven't your SSH keys already set up you will have to:

```
$ ssh-keygen -t rsa
```

This will generate the following output, press just enter if you don't want to specify the name:

Generating public/private rsa key pair.

Enter file in which to save the key (/home/henrivis/.ssh/id_rsa):

And following:

Enter passphrase (empty **for** no passphrase):

Enter same passphrase again:

If you want a password when logging into your *RPi*, then insert something on your passphrase for your SSH keys.

Open the generated key with any text editor you want:

```
$ cd ~/.ssh/
```

```
$ gedit id_rsa.pub
```

Copy the whole key (everything) from the file, go to your Ubuntu One account, if you don't have one, just create it. Go to the following [link](#) and after creating your account, paste the public key from your clipboard into *Public SSH Key* as follows:



Figura 3: Screenshot from Ubuntu One Website

Make sure to press **Import SSH key**. After importing the key everything should be alright and ready for the next step.

1.2 Initial Ubuntu Core Setup

Now insert the SD card on your *RPi*, grab an ethernet cable, HDMI, mouse and keyboard and plug 'em all on your *RPi*, after plugging everything you may turn your *Rpi* on. We do that because usually the *RPi* tries to work on low power mode and turns the screen mode off, meaning the cable has to be connected on startup to ensure the HDMI port to be indeed read.

Wait for the black screen to show "**press enter to configure.**", enter the configure mode and check the **Network connections** section to see if the ethernet *eth0* cable has been identified, if so then your initial setup will be successful. I recommend you to setup

the *Wi-Fi* later because usually this initial setup is not enough and you have to setup it later anyways.

On the **Profile setup section**, make sure to insert the **e-mail registered in Ubuntu One account**. That done since your *RPi* is connected to the internet through the ethernet cable, it should retrieve the SSH public key that you provided before and successfully set up your Ubuntu Core. **Write the command down** returned from the Setup Screen to access the Ubuntu Core with SSH! Usually after the setup such screen appears:

```
Ubuntu Core 16 on 192.168.1.46 (tty1)

The host key fingerprints are:

RSA      SHA256:UniasRCgAoKVsnH/7Ma1ZKT0D5msXep0BDcM2Mw1B9I
DSA      SHA256:exIfb1moS4I8wTs0rthOmFDF9/ISYXpqhBgKIS8Ag0w
ECDSA    SHA256:sAAfvrIiDsMdfKqQA/U16L34IdpK0468rP9AUNvu18o
ED25519  SHA256:3mIrGTiLdaPukHdYeagyx12V/Q4hq21EPYJR21I064w

To login:

ssh thinger@192.168.1.46

Personalize your account at https://login.ubuntu.com.
```

Figura 4: Screenshot from Ubuntu One Website

Restart your your *RPi* and from the computer you generated the SSH keys try to SSH your Ubuntu Core by:

```
$ ssh henrivis@192.168.1.30
$ ssh USERFromRpi@IPfromRpi
```

You should be prompted with a question about being sure to continue with the connection, **type 'yes'** and the given IP should be added to the known hosts. Now you should be inside your *RPi*! Realize that it has no Documents, Pictures etc folders since it is focused on robotics applications.

If you have already set up your SSH keys such error could happen:

```
@@ WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY! @@
```

The solution is as follows:

```
ssh-keygen -R 192.168.3.10
```

Where the IP would be the the one from your *RPi* running Ubuntu Core.

1.3 Configuring Ubuntu Core Development Workspace

You can see from the first time logging into your *RPi* that Ubuntu Core is a little raw, the download of desktop images are possible if you want, but in order to maintain standards we're gonna follow with the no-graphical-interface setup.

Since Ubuntu Core comes with the snap packaging, which we're usually not accustomed with, there's a way to install the Ubuntu LTS classical apt packaging so we can manage the packages with apt like we normally do:

Logged with SSH in your *RPi*:

```
$ snap refresh
```

Install the developer mode:

```
$ snap install classic --edge --devmode
```

To enter the developer mode, i.e using apt we have **always** to run:

```
$ sudo classic
```

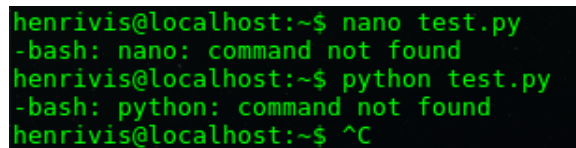
Install the basic packages (including git!):

```
$ sudo apt update
```

```
$ sudo apt install snapcraft build-essential git
```

1.4 Validating our Installation

Feel the Ubuntu Core's environment before doing anything, just create a Python file on the default home directory and run it to see if it works! As an example i tried to run *nano* (default terminal text editor from Ubuntu) and look what happens:

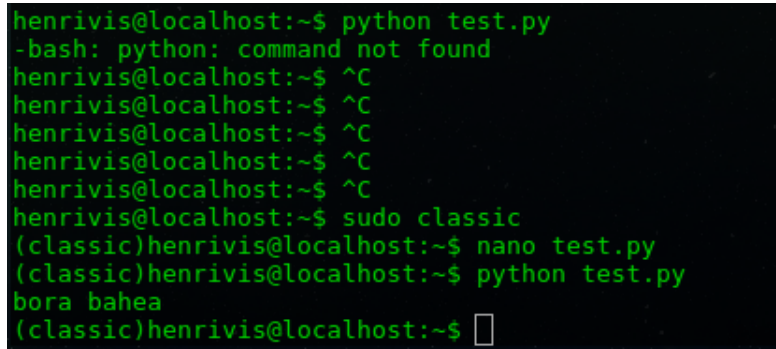


```
henrivis@localhost:~$ nano test.py
-bash: nano: command not found
henrivis@localhost:~$ python test.py
-bash: python: command not found
henrivis@localhost:~$ ^C
```

Figura 5: Screenshot inside Ubuntu Core

Ubuntu Core by default hasn't the Ubuntu settings as we saw before, so it recognizes

neither *python* nor *nano* commands. What we have to do is just call *sudo classic*:



```
henrivis@localhost:~$ python test.py
-bash: python: command not found
henrivis@localhost:~$ ^C
henrivis@localhost:~$ ^C
henrivis@localhost:~$ ^C
henrivis@localhost:~$ ^C
henrivis@localhost:~$ ^C
henrivis@localhost:~$ sudo classic
(classic)henrivis@localhost:~$ nano test.py
(classic)henrivis@localhost:~$ python test.py
bora bahea
(classic)henrivis@localhost:~$
```

Figura 6: Screenshot inside Ubuntu Core

1.5 Installing ROS

The whole system is already set up, the only thing left is ROS, to install it make sure you're in classic mode:

```
$ sudo classic
```

Add ROS Debian packages:

```
$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" >
/etc/apt/sources.list.d/ros-latest.list'
```

Add the repository key:

```
$ sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key
C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
```

Re-index all the repositories added:

```
$ sudo apt update
```

Install ROS (note we're downloading the smallest installation, which takes about 700MB), for some reason on Ubuntu Core g++ is still needed even though you want to program solely in Python.

```
$ sudo apt install g++ ros-kinetic-ros-base
```

Source ROS to your .bashrc permanently:

```
$ echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc
$ source ~/.bashrc
```

Small test to see if ROS was correctly installed:

```
$ roscore
```

1.6 Configuring Wi-Fi

Referências

- [1] <https://snapcraft.io/blog/your-first-robot-introduction-to-the-robot-operating-system-2-5>
- [2] <https://answers.ros.org/question/325039/apt-update-fails-cannot-install-pkgs-key-not-working/>
- [3] https://ubuntu.com/download/raspberry-pi?_ga=2.52615020.553336230.1575910913-1544619573.1575774377