

Day 10: Raspberry Pi OS Lite Configuration

Introduction

Today, you will learn many important things.

In the last chapter, I showed you how to install Raspberry Pi OS Lite. It was a short and easy chapter, but in this one you'll learn all of the basics about it in just one day!

Firstly, we'll configure the operating system the way you want it.

Then I'll give you other useful tips about passwords management, SSH access and packages management.

Let's go!

Basic configuration

Make sure that the Raspberry Pi OS Lite SD card is in your Raspberry Pi and start it up.

Then log in with the default login and password I gave you in the last chapter.

Now I'll show you how to configure each part of Raspberry Pi OS Lite.

They all use the same tool: *raspi-config*.

This tool exists only on Raspberry Pi OS, and allows you to do the basic configuration that you need to start (network, keyboard, language, etc.)

Localization options

The first thing you probably want to change is the keyboard layout (if needed).

It was a good experience to find the password with another layout, but to work seriously you need the most optimal layout for you.

- Start *raspi-config*

```
sudo raspi-config
```

It looks like this:

```
Raspberry Pi Software Configuration Tool (raspi-config)
1 System Options          Configure system settings
2 Display Options        Configure display settings
3 Interface Options      Configure connections to peripherals
4 Performance Options    Configure performance settings
5 Localisation Options   Configure language and regional settings
6 Advanced Options       Configure advanced settings
8 Update                 Update this tool to the latest version
9 About raspi-config     Information about this configuration tool
```

- Go to "Localisation Options"
- Then "Keyboard"
 - The wizard now asks you to answer a few questions to find the corresponding keyboard:
 - **Keyboard model:** probably the default one
 - **Keyboard layout:** use the arrows to choose "Other"
 - Then select your country from the list
 - Now select the keyboard layout from these country options
 - Finally, select a few other options about the keyboard configuration (AltGr key, Compose key)
You can keep the default options, as we don't need these keys for Raspberry Pi OS Lite
 - That's it, your keyboard is now properly configured!
 - Back to the "Localisation options" menu in *raspi-config*, you can also change other options if you need to: Locale, Timezone and WLAN country. They have intuitive submenus, so I think you'll be able to configure them fairly easily.

Network options

The second thing you'll probably need to configure before going further are the network options

If you want to use a wireless connection, you have to configure it now

- Open the Raspberry Pi OS configuration tool

```
sudo raspi-config
```

- Go to "System Options"
- Then select "Wireless LAN"

- If not already done, you may need to **select your country** (mandatory for the wireless configuration)
- Then, **enter your wireless network SSID (name) and password**
- You can now exit `raspi-config`

After a few seconds, the network should be up and ready to use.

If you need to use a static IP address, the easiest way is to fix it in your network router interface.

But generally you don't need it, and you can use the host name to access it later (the default host name is "raspberrypi" but you can change it in `raspi-config`).

I won't show you how to do this directly on Raspberry Pi, as it requires a few commands and concepts that we haven't yet come across.

If you need to, you can take a look at this post on RaspberryTips.com:

<https://raspberrytips.com/go/ip>

That's it for `raspi-config`.

This tool contains a lot more configuration options, but for now we don't need to use them. In the next three sections, I'll show you other configurations you have to do, and how you can do them with or without `raspi-config`.

Password management

Once you are finished with the basic configuration (Localization and Network), the first thing I recommend that you do is change the user password.

As you have seen, all Raspberry Pi units have the same default password.

If you don't want to allow anyone to access it, change the password now.

You can do this with a command, or in `raspi-config`.

As you are here to learn, I'll start with the command:

- In the Raspberry Pi OS Lite terminal, **type the following command:**

```
passwd
```

- Hit "Enter" to start the tool:

```
pi@raspberrypi:~ $ passwd
Changing password for pi.
Current password:
New password:
Retype new password:
passwd: password updated successfully
```

- Enter the current password (raspberrypi)
- Enter the new password
- And once again to confirm
- That's it, your new Raspberry Pi password is set

If you don't remember the command to do this, you can use *raspi-config* to start it

- Open *raspi-config*

```
sudo raspi-config
```
- Select "System options" (first option), then "Password"
- The same tool will start in the terminal

SSH remote access

When I started playing with a Raspberry Pi, I was using it on my TV with an old keyboard and mouse, while reading tutorials on my smartphone.

So, I had to decrypt mysterious commands and type them, making lots of mistakes.

But there is a way to be more comfortable while using your Raspberry Pi; you can connect to it from your computer. With Raspberry Pi OS Lite it doesn't change anything as you'll get the same terminal screen, and this will allow you to copy and paste commands directly from a web page, without errors.

SSH server

The most common way to do this is to use SSH.

SSH stands for Secure SHell.

It's a type of software called a secured protocol used to access remote computers.

SSH is installed by default on Raspberry Pi OS Lite, so there is almost nothing that you need to do, you just need to start the service.

From your Raspberry Pi OS Lite terminal, run the following command:

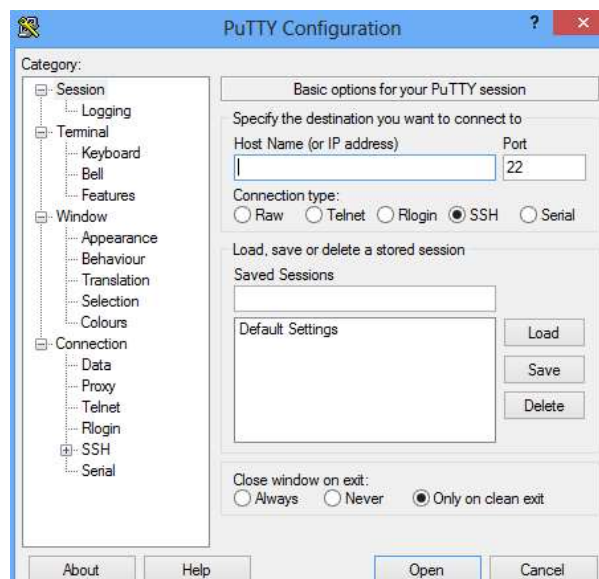
```
sudo service ssh start
```

And that's it, you have nothing else to do.

You can now connect to your Raspberry Pi from your computer.

SSH client

- **For Windows users:**
 - You need to install software to use SSH connections
 - I'm using MobaXterm (<https://mobaxterm.mobatek.net/>) because I have several servers and Raspberry Pi to manage. But the easiest way is probably to use Putty (<https://putty.org/>)
 - Download Putty for Windows from the official website
 - Once done, start it (double-click on it)
 - A window like this shows up:



- Enter the Raspberry Pi host name ("raspberrypi" by default)
- Then click on the "Open" button
- A black screen shows up, enter the username ("pi") and password to connect
- That's it, you now have access to the Raspberry Pi OS Lite terminal from your computer

- **From a Linux or macOS operating system:**

- On Linux systems (Debian, Ubuntu, macOS, other Raspberry Pi, etc.), SSH is preinstalled by default, so you have nothing more to install
- To connect to another system, you need to use a command line
- **Open a terminal** on your computer (it depends on your exact system, often it's in the Accessories or System submenu)
- Then **use this command** to connect to your Raspberry Pi:

```
ssh pi@raspberrypi
```

- Press "Enter" and enter your password to connect
- You are now connected to the Raspberry Pi, and can use this terminal to follow the next sections and chapters

Auto-start the SSH server

There is an issue with the command I gave you to start the SSH service. As soon as you stop or reboot the Raspberry Pi, the service stops and won't restart automatically.

If you have only one thing to do, it's OK, as it's safer than leaving the service open all the time.

But you'll probably use it every day to try the commands I'll give you in this book. You have two solutions to make it start automatically.

SSH auto-start with command lines

In your terminal (SSH or directly on your Raspberry pi), **enter these two commands:**

```
sudo update-rc.d ssh defaults
sudo update-rc.d ssh enable
```

After the next reboot, the SSH service will start automatically.

You can use these commands to auto-start any service (just replace ssh by the service name), so it can be useful to know them.

SSH auto-start with raspi-config

And if you don't remember them, there is also a way to do this in *raspi-config*:

- **Start raspi-config**

```
sudo raspi-config
```

- Go to “Interface options” and select “SSH”
- Answer yes to the question “Would you like the SSH server to be enabled?”

Close *raspi-config*, and that’s it; SSH service will start automatically on next boot.

Packages management

On Raspberry Pi OS Desktop, we saw some tools to update, install or remove packages.

It’s almost the same on the Lite version, without the interface :)

You can do everything with the “apt” command.

Here is the syntax:

```
sudo apt <action> <search string>
```

Then you have five actions to know:

- **Update**
 - Goal: Raspberry Pi OS works with a repository server that hosts all packages. Before using apt you need to download the last files inventory from the server. Use this command each time you want to use another apt action
 - Syntax: *sudo apt update*
 - There is no other option to add, just ask to update the repository files
- **Upgrade**
 - Goal: once apt knows the updated list of available files, it can suggest upgrades for your system. This action will download and install all the updated version of your software
 - Syntax: *sudo apt upgrade*
 - There is also no option to add
- **Search**
 - Goal: as you may have seen on Raspberry Pi OS Desktop, package names are not always obvious. The search action allows you to find the package name from a search string
 - Syntax: *apt search <string>*

- Example: `apt search sense-hat`

```
pi@raspberrypi:~ $ apt search sense-hat
Sorting... Done
Full Text Search... Done
python-sense-hat/testing 2.2.0-1 armhf
Sense HAT python library (Python 2)

python3-sense-hat/testing 2.2.0-1 armhf
Sense HAT python library (Python 3)

sense-hat/testing 1.2 all
Sense HAT configuration, libraries and examples
```

- This command displays a list of available packages containing the string you search for. You can then use the package name to install it.
- **Install**
 - Goal: as the name suggests, this action allows you to install a specific package
 - Syntax: `sudo apt install <package name>`
 - Example: `sudo apt install python3-sense-hat`
 - Make sure to use the exact package name you get with apt search
- **Remove**
 - Goal: you can use the remove action to uninstall a package, it works almost like the install action
 - Syntax: `sudo apt remove <package name>`
 - Example: `sudo apt remove python3-sense-hat`
 - After using the remove action, you can use the autoremove to also uninstall dependencies that are no longer needed

That's it, you now know how to manage packages on Raspbian (and any Debian-like operating systems).

I often use these commands, even on a desktop OS, as I don't like the slow speed of the graphical tools.

Homework

We have seen many things today, and for your homework I want you to reread this chapter and try everything you can.

These are the basics on Raspberry Pi, and we'll use them all the time in the next chapters:

- The *raspi-config* tool
- Remote access with SSH
- Packages management

Make sure you understood everything, try to use SSH to install new packages etc.

In the next chapter I'll teach you many more useful commands. So, you must really know how to use and understand those we have seen today before moving on to the next chapter.