Live installation

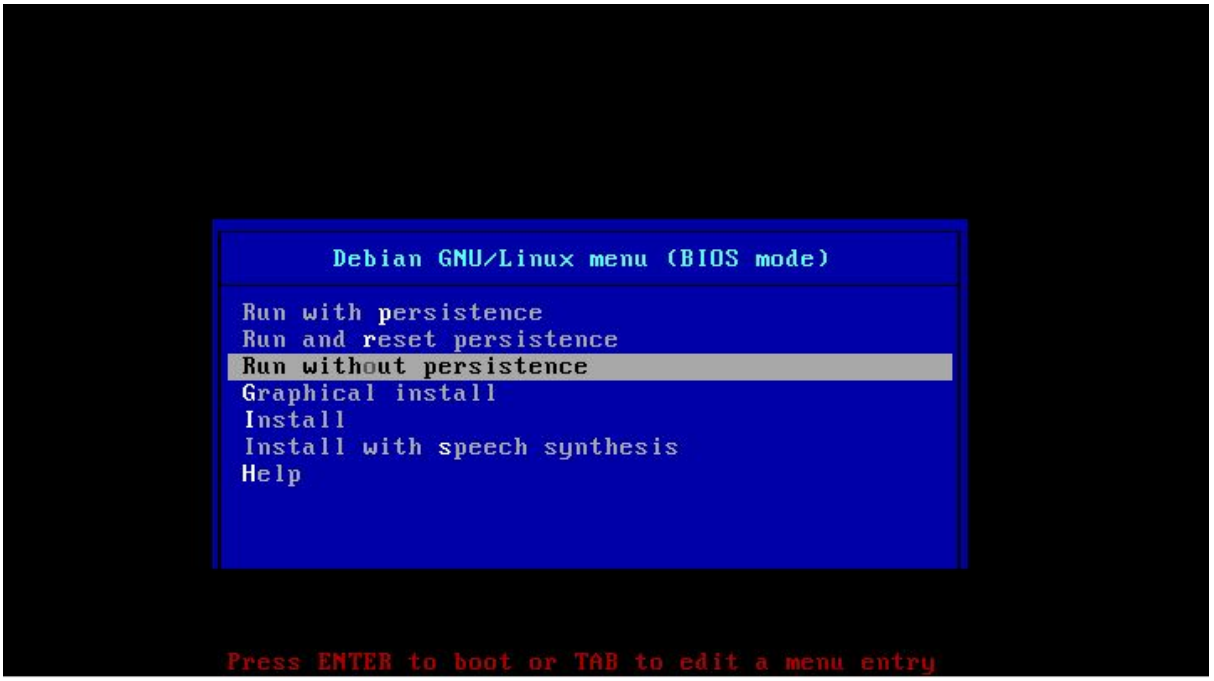
Download Raspbian live iso image from this page:
https://www.raspberrypi.org/software/raspberry-pi-desktop/
Or directly :
https://downloads.raspberrypi.org/rpd_x86/images/rpd_x86-2020-12-11/2020-12-11-raspios-buster-i386.iso

Install this image on a CD or USB device with the appropriate tool (Win32diskimager or similar tool)

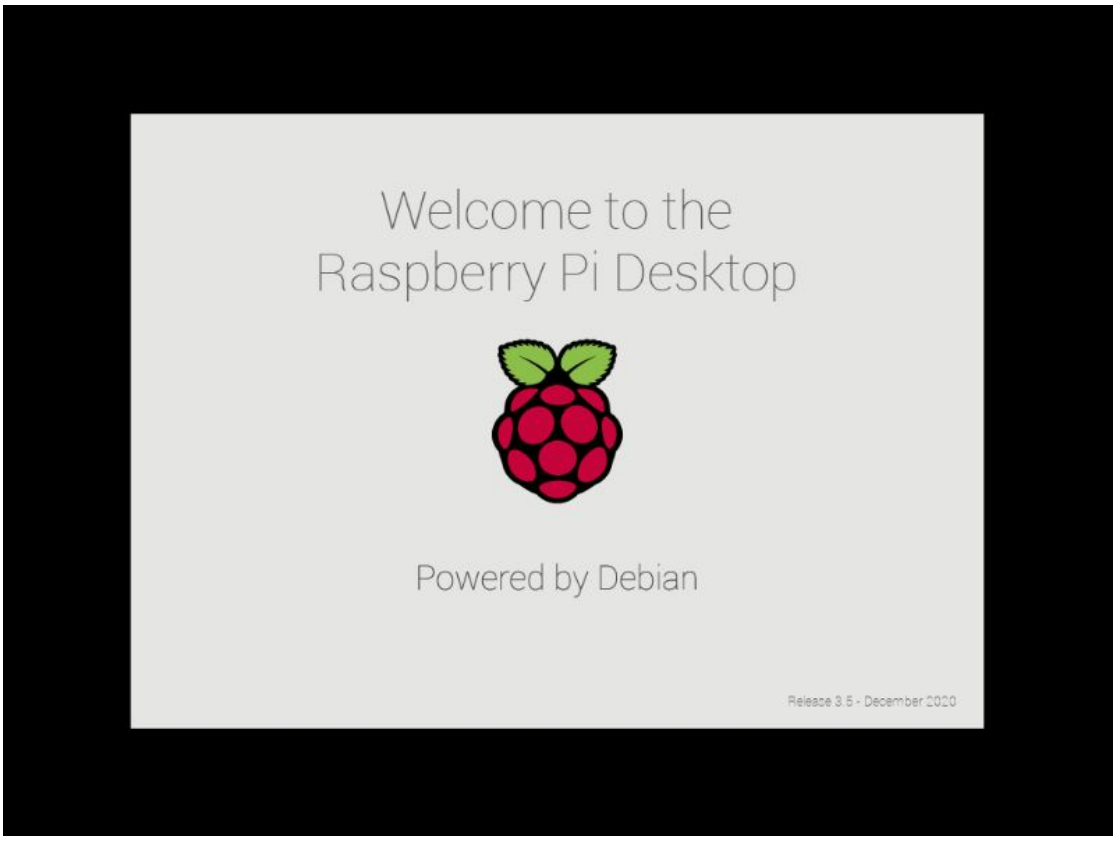
Or simply use this image in a virtual machine like Oracle VM VirtualBox

For this tutorial, I used Oracle VM VirtualBox

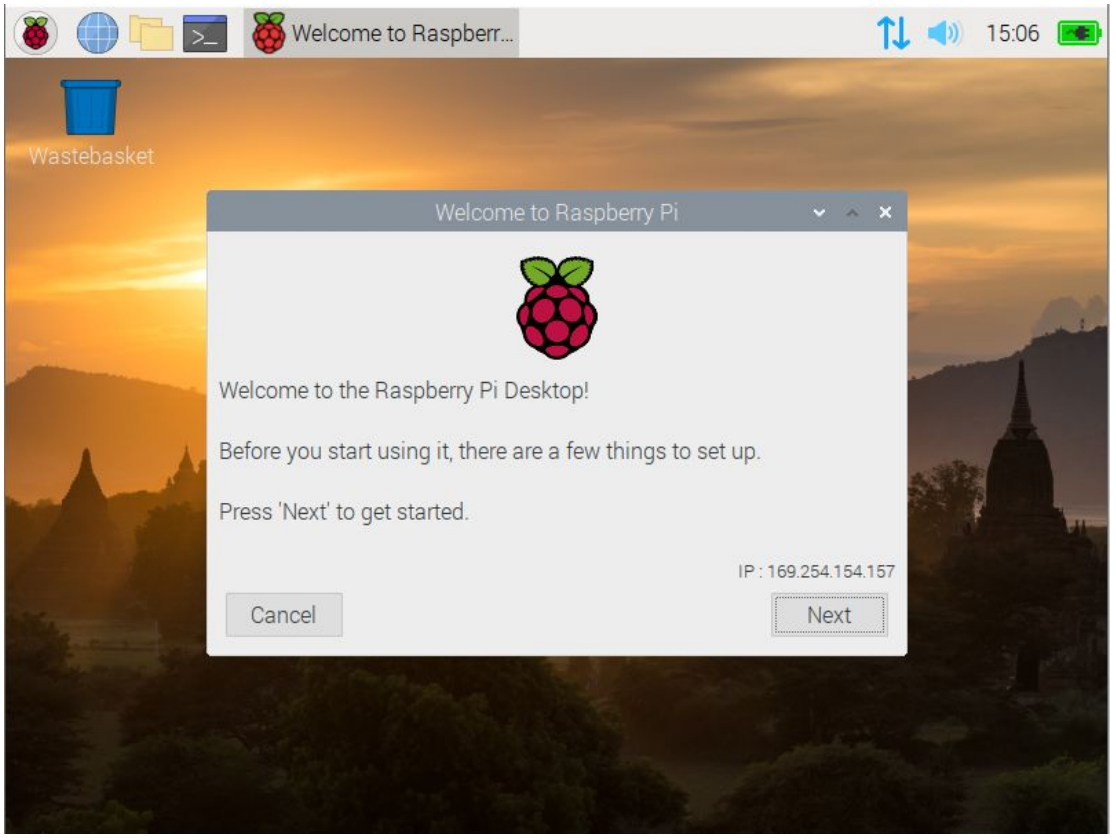
Here is the boot screen :
Select “Run without persistence” (but you can choose another)



The system will start as follow :

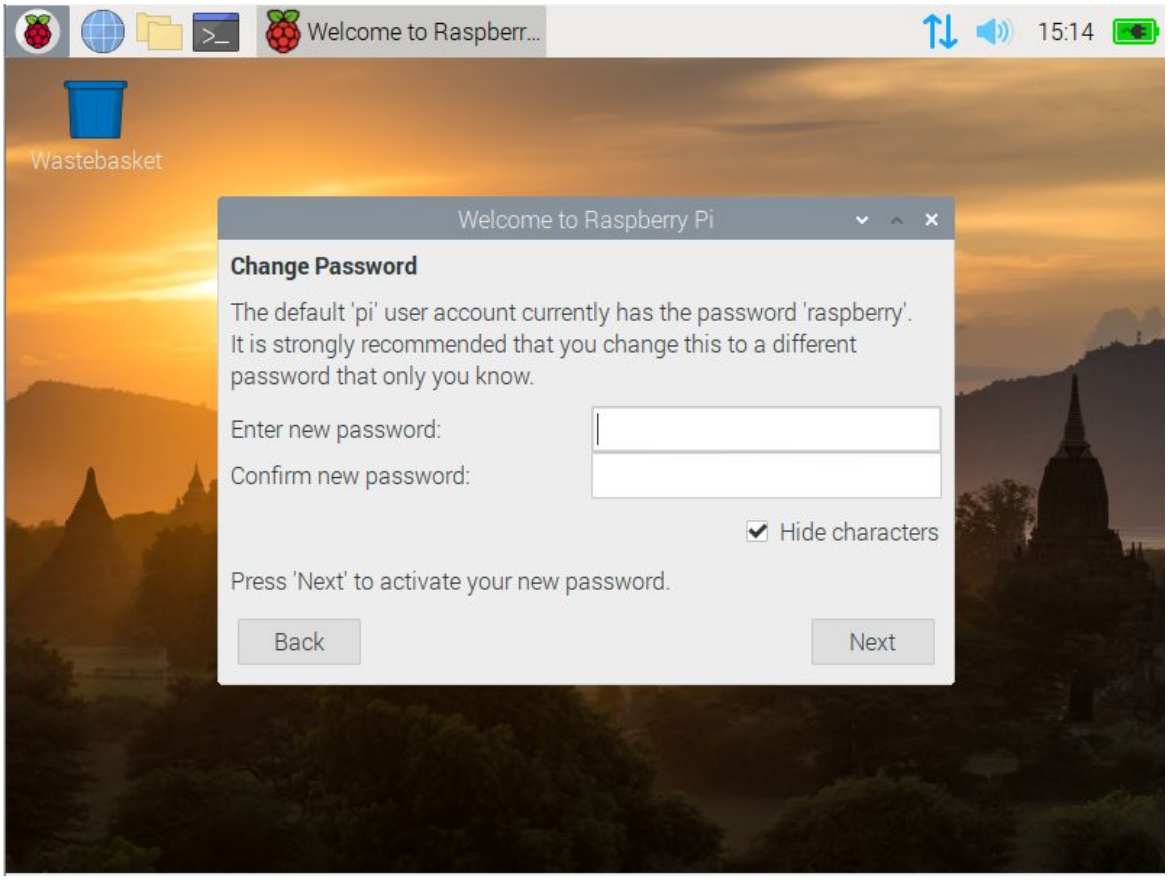


And you will have this screen :

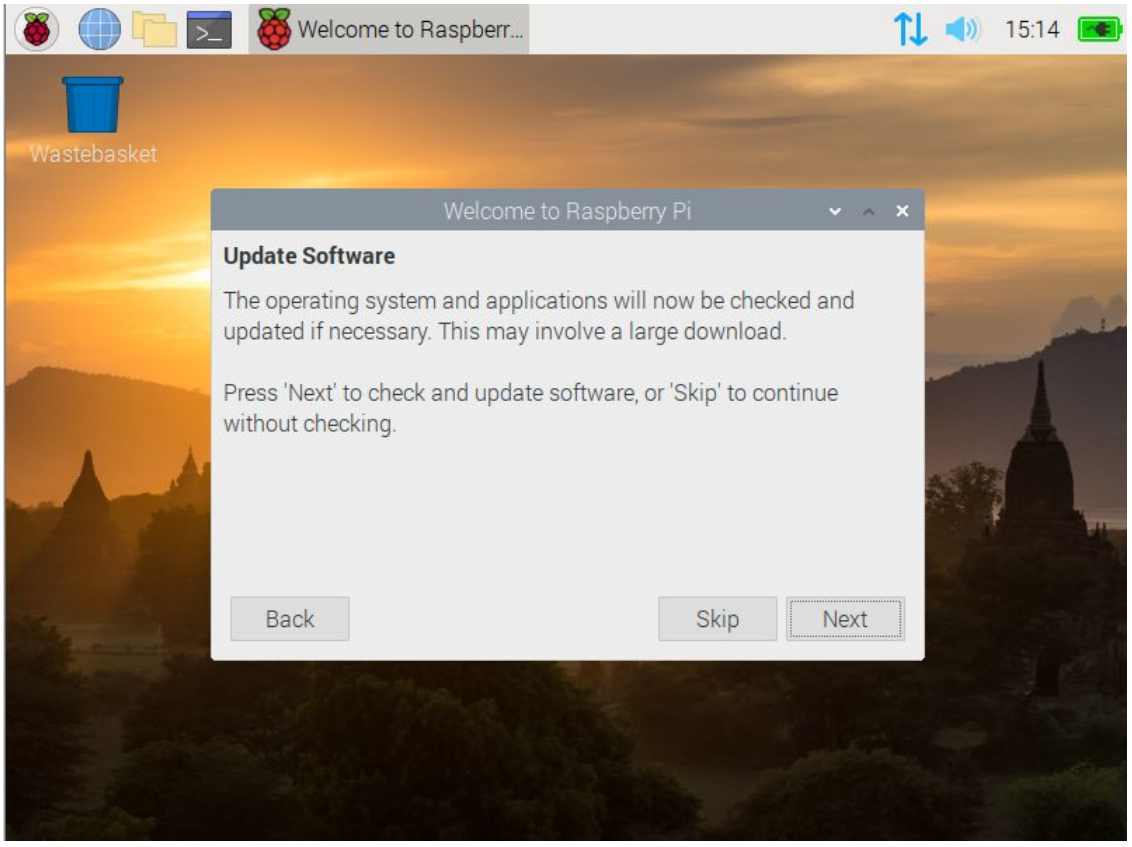


Select Next and choose your location informations:

You will asked to change the pi user password :

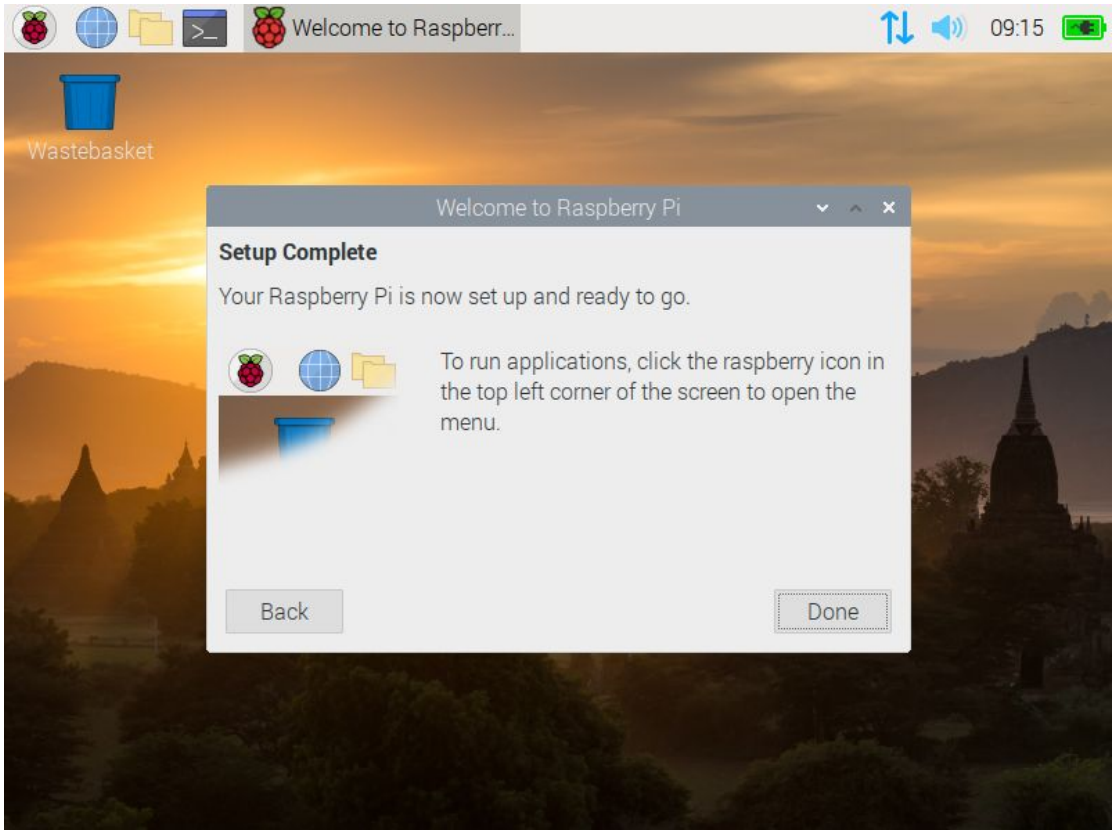


You will asked to update softwares like that :



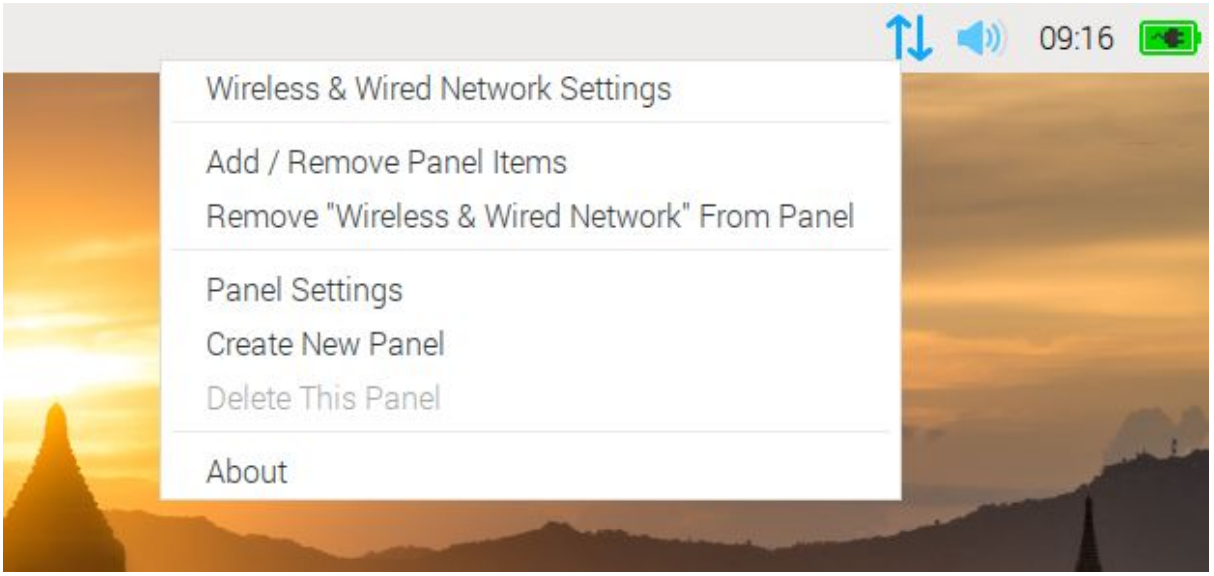
Click the skip button

And finish by clicking “Done”

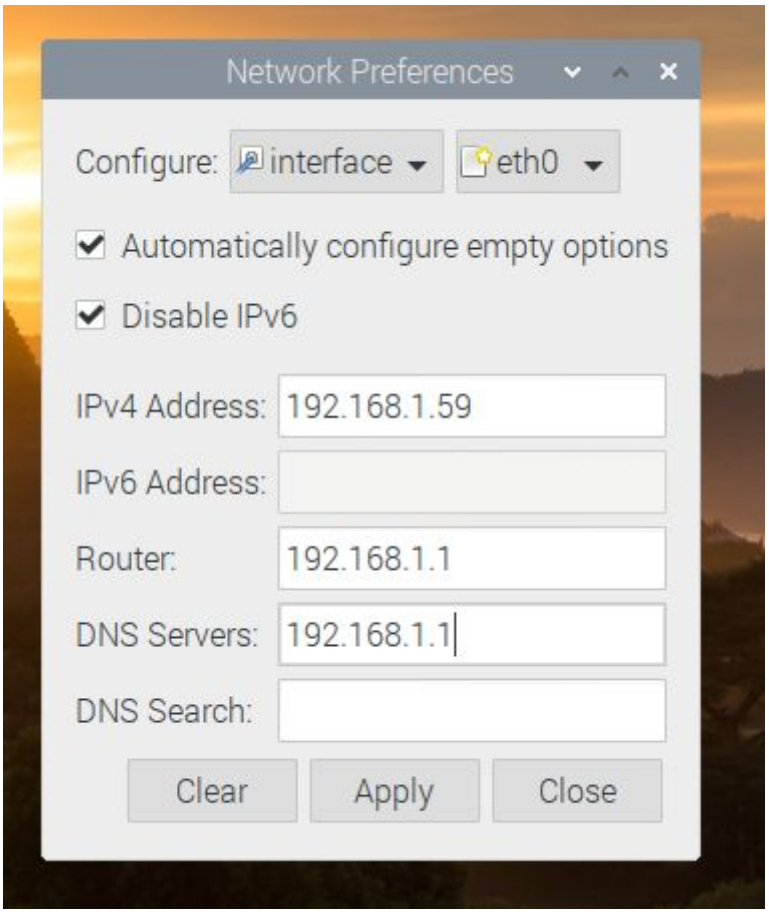


Now configure the network:

Right click on top right icon with blue arrows and click “Wireless & Wired Network Settings”



Select your interface and configure (Here below is just an example):



Click apply and close.
In terminal, try a ping to castles-backup.com
If it is a success, all is ok.
If it fails, you may need to try a different method like this one :
In a terminal edit /etc/dhcpcd.conf file

Go to the end of the file
and manually add these lines :

```
interface eth0
static ip_address=192.168.1.59
static routers=192.168.1.1
static domain_name_servers=192.168.1.1
noipv6
```

Save & Close the file and restart daemon :

```
pi@raspberrypi:~ $ sudo vi /etc/dhcpcd.conf
pi@raspberrypi:~ $ sudo systemctl restart dhcpcd
pi@raspberrypi:~ $
```

Check again with a ping to castles-backup.com

```
pi@raspberrypi:~ $ ping castles-backup.com
PING castles-backup.com (82.65.93.57) 56(84) bytes of data.
64 bytes from 82-65-93-57.subs.proxad.net (82.65.93.57): icmp_seq=1 ttl=64 time=4.48 ms
64 bytes from 82-65-93-57.subs.proxad.net (82.65.93.57): icmp_seq=2 ttl=64 time=10.5 ms
^C
--- castles-backup.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 4.478/7.490/10.503/3.013 ms
pi@raspberrypi:~ $
```

Now, your system is OK

You can download the free script to make your backup system
wget https://www.castles-backup.com/castles-backup_live.tar.gz

```
pi@raspberrypi:~ $ wget https://www.castles-backup.com/castles-backup_live.tar.gz
--2021-01-08 09:27:46--  https://www.castles-backup.com/castles-backup_live.tar.gz
Resolving www.castles-backup.com (www.castles-backup.com)... 82.65.93.57
Connecting to www.castles-backup.com (www.castles-backup.com)|82.65.93.57|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1994 (1.9K) [application/x-gzip]
Saving to: 'castles-backup_live.tar.gz'

castles-backup_live.tar. 100%[=================================>]   1.95K  --.-KB/s    in 0s

2021-01-08 09:27:46 (5.51 MB/s) - 'castles-backup_live.tar.gz' saved [1994/1994]

pi@raspberrypi:~ $
```

You can now uncompress file with :
tar zxvf castles-backup_live.tar.gz

```
pi@raspberrypi:~ $ tar zxvf castles-backup_live.tar.gz
castles/
castles/rsync.sh
castles/2-cb_script_conf.sh
castles/restore.sh
castles/1-cb_script_install.sh
castles/config.txt
castles/dir2Rsync
castles/read_conf_cron.sh
pi@raspberrypi:~ $
```

What are these files :
1-cb_script_install.sh : script to prepare the system
config.txt : configuration file to edit and configure server to backup informations
2-cb_script_conf.sh : script to finish installation
dir2Rsync : configuration file to edit and add list of folders to backup
read_conf_cron.sh : script which check changes in config file and modify cron entry
rsync.sh : script to make backup
restore.sh script to make restore

So now, go to castles directory and execute 1-cb_script_install.sh
cd castles
sh 1-cb_script_install.sh

```
pi@raspberrypi:~ $ cd castles/
pi@raspberrypi:~/castles $ sh 1-cb_script_install.sh
```

You will get some errors, it's normal, it is not an optimized script :-)

```
 Castles-Backup live script
>>>> Create mount directory

rm: cannot remove '/media/disk': No such file or directory
*/15 * * * * root /home/pi/read_conf_cron
# UNCONFIGURED FSTAB FOR BASE SYSTEM
overlay / overlay rw 0 0
tmpfs /tmp tmpfs nosuid,nodev 0 0
# a swapfile is not a swap partition, no line here
#   use  dphys-swapfile swap[on|off]  for that
umount: /dev/sda1: no mount point specified.
umount: /dev/sdb1: no mount point specified.
umount: /dev/sdc1: no mount point specified.
umount: /dev/sdd1: no mount point specified.
Disk for backup has been prepared in previous boot (y/n) ? :
```

If it is your first installation you will say "n" otherwise, if you disk has been prepared in previous play with the solution, say "y"

After saying "n", you'll be asked to connect your device :
Connect the device you want to use to make backup
Don't forget to tell your virtual machine to mount the device ! otherwise it won't be found !

```
Disk for backup has been prepared in previous boot (y/n) ? : n

Connect disk in usb port and press Entrer :
```

Now select wich device :

```
Connect disk in usb port and press Entrer :

NAME  FSTYPE     SIZE MOUNTPOINT       LABEL            MODEL
loop0 squashfs   2.1G
sda                8G                                  VBOX_HARDDISK
sdb   vfat     501.5M                                  disk
sr0   iso9660      3G /run/live/medium Debian RPD M-A 1 VBOX_CD-ROM
sr1             1024M                                  VBOX_CD-ROM

what is the device ? (sda ? sdb ?) :
```

In my case, I plugged a 500Mo USB drive. I can see it with “sdb” name
So I say sdb and press Enter

```
Create new linux partition on /dev/sdb
Checking that no-one is using this disk right now ... OK

Disk /dev/sdb: 501.5 MiB, 525860864 bytes, 1027072 sectors
Disk model: disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x657671ba

Old situation:

>>> Created a new DOS disklabel with disk identifier 0xd5bd90c6.
/dev/sdb1: Created a new partition 1 of type 'Linux' and of size 500.5 MiB.
Partition #1 contains a ext4 signature.
/dev/sdb2: Done.

New situation:
Disklabel type: dos
Disk identifier: 0xd5bd90c6

Device     Boot Start     End Sectors   Size Id Type
/dev/sdb1        2048 1027071 1025024 500.5M 83 Linux

The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
Format partition /dev/sdb1 in EXT4
mke2fs 1.44.5 (15-Dec-2018)
```

```
mke2fs 1.44.5 (15-Dec-2018)
/dev/sdb1 contains a ext4 file system
        last mounted on /media/disk on Thu Jan  7 13:58:34 2021
Creating filesystem with 512512 1k blocks and 128520 inodes
Filesystem UUID: 2bdaaa89-ebb1-4652-a475-9953301a0124
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729, 204801, 221185, 401409

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

Disk for backup is ready
Add disk to the system
UUID                                 NAME    FSTYPE     SIZE MOUNTPOINT  LABEL           MODEL
                                     loop0   squashfs   2.1G
                                     sda          8G                                     VBOX_HARD
                                     sdb        501.5M                                   disk
2bdaaa89-ebb1-4652-a475-9953301a0124 `-sdb1 ext4     500.5M
2020-12-11-13-39-06-00               sr0     iso9660      3G /run/live/m Debian RPD M-A 1 VBOX_CD-R
                                     sr1         1024M                                    VBOX_CD-R

Disk UUID :
```

We are now asked for the UUID of the new created partition

```
Add disk to the system
UUID                                 NAME    FSTYPE     SIZE MOUNTPOINT  LABEL           MODEL
                                     loop0   squashfs   2.1G
                                     sda          8G                                     VBOX_HARD
                                     sdb        501.5M                                   disk
2bdaaa89-ebb1-4652-a475-9953301a0124 `-sdb1 ext4     500.5M
2020-12-11-13-39-06-00               sr0     iso9660      3G /run/live/m Debian RPD M-A 1 VBOX_CD-R
                                     sr1         1024M                                    VBOX_CD-R

Disk UUID : 2bdaaa89-ebb1-4652-a475-9953301a0124
```

I press Enter and :

```
Add disk to the system
UUID                                 NAME    FSTYPE     SIZE MOUNTPOINT  LABEL           MODEL
                                     loop0   squashfs   2.1G
                                     sda          8G                                     VBOX_HARD
                                     sdb        501.5M                                   disk
2bdaaa89-ebb1-4652-a475-9953301a0124 `-sdb1 ext4     500.5M
2020-12-11-13-39-06-00               sr0     iso9660      3G /run/live/m Debian RPD M-A 1 VBOX_CD-R
                                     sr1         1024M                                    VBOX_CD-R

Disk UUID : 2bdaaa89-ebb1-4652-a475-9953301a0124
UUID=2bdaaa89-ebb1-4652-a475-9953301a0124 /media/disk ext4 defaults,auto,rw,nofail 0 0
pi@raspberrypi:~/castles $
```

Now, my device is ext4 formatted and mounted in /media/disk

First step is finished.

Now prepare the server to make backup
Edit config.txt file in /home/pi
vi /home/pi/config.txt

```
#Configuration file
#*****************************************************
ipServer2Backup="192.168.1.2"
userNameServer2Backup="[redacted]"
#*****************************************************

#Daly time to make backup
timeForDailyBackup="9"

#file containing directories to backup
dir2BackupPath="/home/pi/dir2Rsync"

~
~
~
~
~
```

Change the server ip address
Change the username to connect to through ssh
Actually, backup script start every day at 9 AM (you can change it)

After changes, save and close the file

Now launch 2-cb_script_conf.sh

```
pi@raspberrypi:~/castles $ sh 2-cb_script_conf.sh
```

A public ssh ley is going to be created:

```
pi@raspberrypi:~/castles $ sh 2-cb_script_conf.sh
chmod: cannot access '/home/pi/.ssh/id_rsa.pub': No such file or directory
When filename asked, leave empty
When password is asked, it is the password of the user we are going to connect throw ssh
Preparing ssh keys for server to backup

# 192.168.1.2:22 SSH-2.0-OpenSSH_6.0p1 Debian-4
# 192.168.1.2:22 SSH-2.0-OpenSSH_6.0p1 Debian-4
# 192.168.1.2:22 SSH-2.0-OpenSSH_6.0p1 Debian-4
Enter file in which to save the key (/home/pi/.ssh/id_rsa):
```

Let an empty name et press Enter
You're now asked for the SSH password of the user on the server to backup

```
pi@raspberrypi:~/castles $ sh 2-cb_script_conf.sh
chmod: cannot access '/home/pi/.ssh/id_rsa.pub': No such file or directory
When filename asked, leave empty
When password is asked, it is the password of the user we are going to connect throw ssh
Preparing ssh keys for server to backup

# 192.168.1.2:22 SSH-2.0-OpenSSH_6.0p1 Debian-4
# 192.168.1.2:22 SSH-2.0-OpenSSH_6.0p1 Debian-4
# 192.168.1.2:22 SSH-2.0-OpenSSH_6.0p1 Debian-4
Enter file in which to save the key (/home/pi/.ssh/id_rsa):
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/pi/.ssh/id_rsa.pub"
[redacted]@192.168.1.2's password:
```

Indicate your password and press Enter

```
pi@raspberrypi:~/castles $ sh 2-cb_script_conf.sh
chmod: cannot access '/home/pi/.ssh/id_rsa.pub': No such file or directory
When filename asked, leave empty
When password is asked, it is the password of the user we are going to connect throw ssh
Preparing ssh keys for server to backup

# 192.168.1.2:22 SSH-2.0-OpenSSH_6.0p1 Debian-4
# 192.168.1.2:22 SSH-2.0-OpenSSH_6.0p1 Debian-4
# 192.168.1.2:22 SSH-2.0-OpenSSH_6.0p1 Debian-4
Enter file in which to save the key (/home/pi/.ssh/id_rsa):
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/pi/.ssh/id_rsa.pub"
[redacted]@192.168.1.2's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh '[redacted]@192.168.1.2'"
and check to make sure that only the key(s) you wanted were added.

Test connexion
>>>Connexion is Ok
pi@raspberrypi:~/castles $
```

A test connexion is made and if it's Ok, you'll see the message "Connexion is Ok"

Now open the file dir2Rsync to configure folders to backup :

```
pi@raspberrypi:~/castles $ sudo vi /home/pi/dir2Rsync
```

Lines must end with "/" :

```
/home/john/
/home/will/
~
~
~
```

Save and close.

For this tutorial, I used this configuration :

```
/home/[redacted]/castles/
~
~
```

Backup

Now I can start the rsync script manually to make a test :

I have to move to /home/pi
cd /home/pi
sh rsync.sh

```
pi@raspberrypi:~ $ sh rsync.sh
receiving incremental file list
.d..t...... ./
>f+++++++++ 1-cb_script_install.sh
>f+++++++++ 2-cb_script_conf.sh
>f+++++++++ config.txt
>f+++++++++ dir2Rsync
>f+++++++++ read_conf_cron.sh
>f+++++++++ restore.sh
>f+++++++++ rsync.sh

sent 147 bytes  received 5,307 bytes  519.43 bytes/sec
total size is 4,789  speedup is 0.88
pi@raspberrypi:~ $
```

We can check the backup directory :

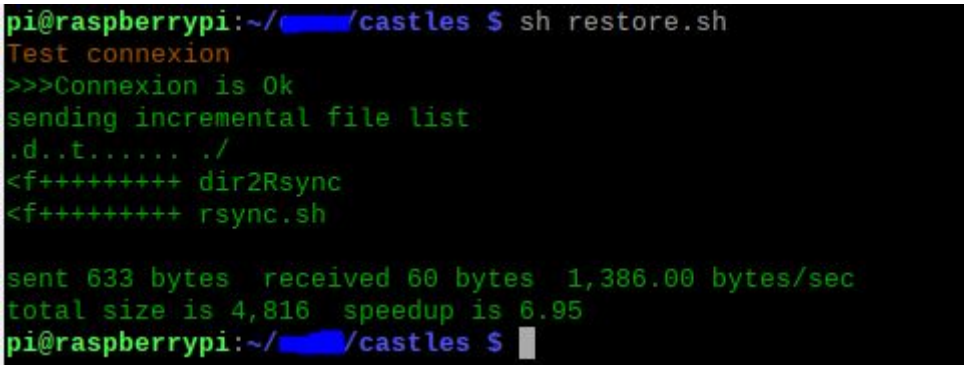
```
pi@raspberrypi:~ $ tree /media/
/media/
|-- disk
|   `-- home
|       `-- [redacted]
|           `-- castles
|               |-- 1-cb_script_install.sh
|               |-- 2-cb_script_conf.sh
|               |-- config.txt
|               |-- dir2Rsync
|               |-- read_conf_cron.sh
|               |-- restore.sh
|               `-- rsync.sh
`-- pi

5 directories, 7 files
pi@raspberrypi:~ $
```

All the files have been backup

Restore

To test restore script I ‘ve deleted 2 files on the server :
I deleted dir2Rsync ans rsync.sh files

Now, I launch restore.sh :
sh restore.sh



The 2 files have been restored !