

Computer Systems Linux Server I



Contents

1	Introduction	2
2	Installing Ubuntu Server 18.04 LTS	2
2.1	Minimum Requirements	2
2.2	Installation	2
2.3	Network Configuration	9
3	Server administration: programming task	11
3.1	Create a programmed task	11



1 Introduction

As it is well known, Ubuntu is a free and open-source Linux distribution based on Debian. One of the editions of Ubuntu is Server and it is able to run in a Virtual Machine. In spite of existing a most updated version from 2019, it is going to be used 2018 version (Ubuntu Server 18.04 LTS (“Bionic Beaver”)) because it is supported until 2023 under public support and until 2028 as a paid option. . Ubuntu is released every 6 months but the LTS letters means that every two years with Long-Term support (LTS).

The edition can be downloaded from the following link:

<https://ubuntu.com/download/server>

From the last link, it is possible to download Ubuntu Server 18.04.4 live-server 64 bits. This version doesn't have a graphic environment, so every action may be done from the terminal.

2 Installing Ubuntu Server 18.04 LTS

2.1 Minimum Requirements

It is understood that when it is in the process of installing some software in a computer, the first task is making sure about the software requirements. It is not much different than an operating system is concerned. The minimum requirements for this version of Ubuntu server can be checked in the following link:

<https://help.ubuntu.com/lts/serverguide/preparing-to-install.html>

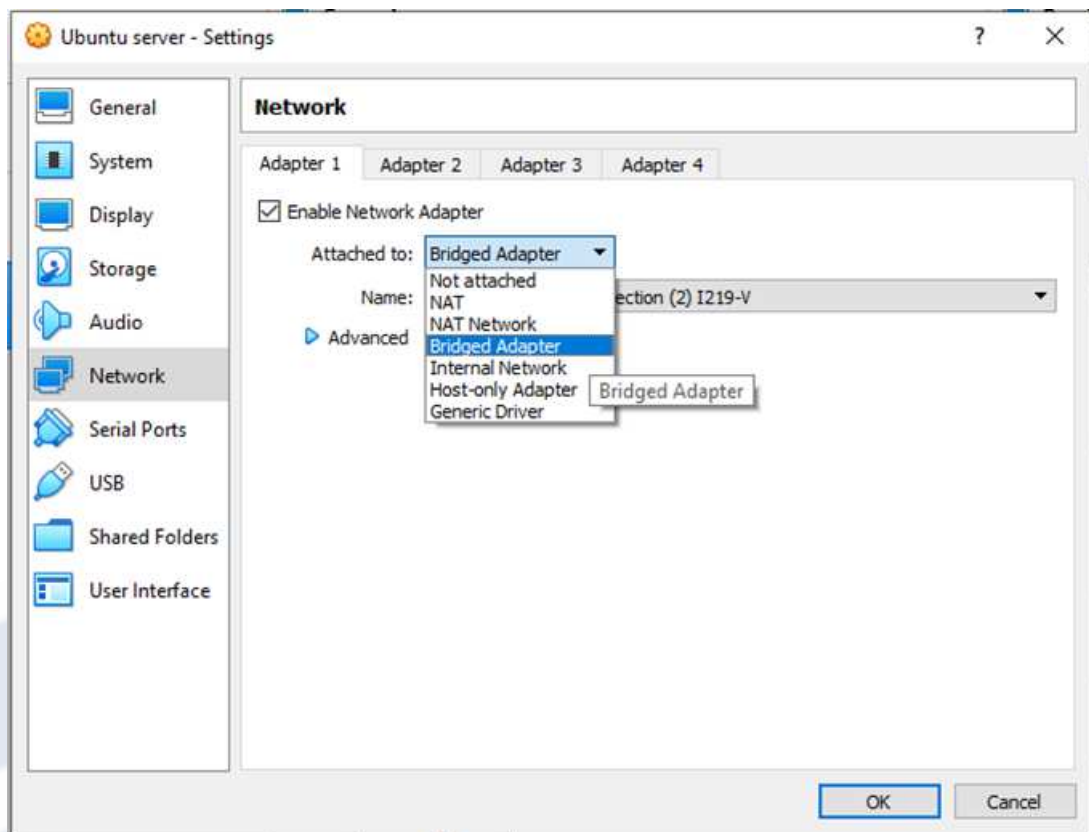
Processor	1 GHz
RAM	512 MB
Hard Disk	2.5 GB

On the other hand, in order to ensure an effective performance, the recommended requirements are:

RAM	1 GB
Hard Disk	10 GB
Network Adapter	Bridge Mode

2.2 Installation

Once the virtual machine is ready, it is possible to start with the installation; but first of all, the network card will be set up as a bridge:

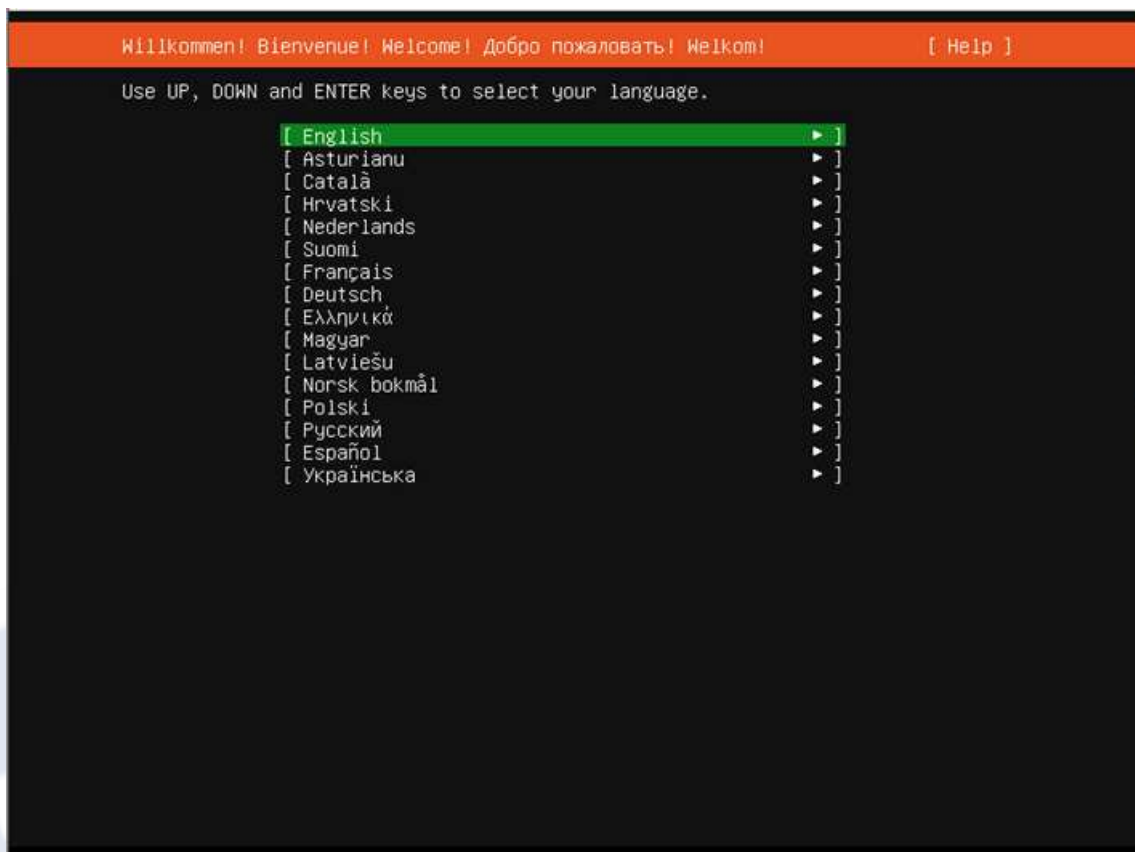


After configuring the network card, it is possible to start with the installation by choosing the image of the Ubuntu Server and the option to install.

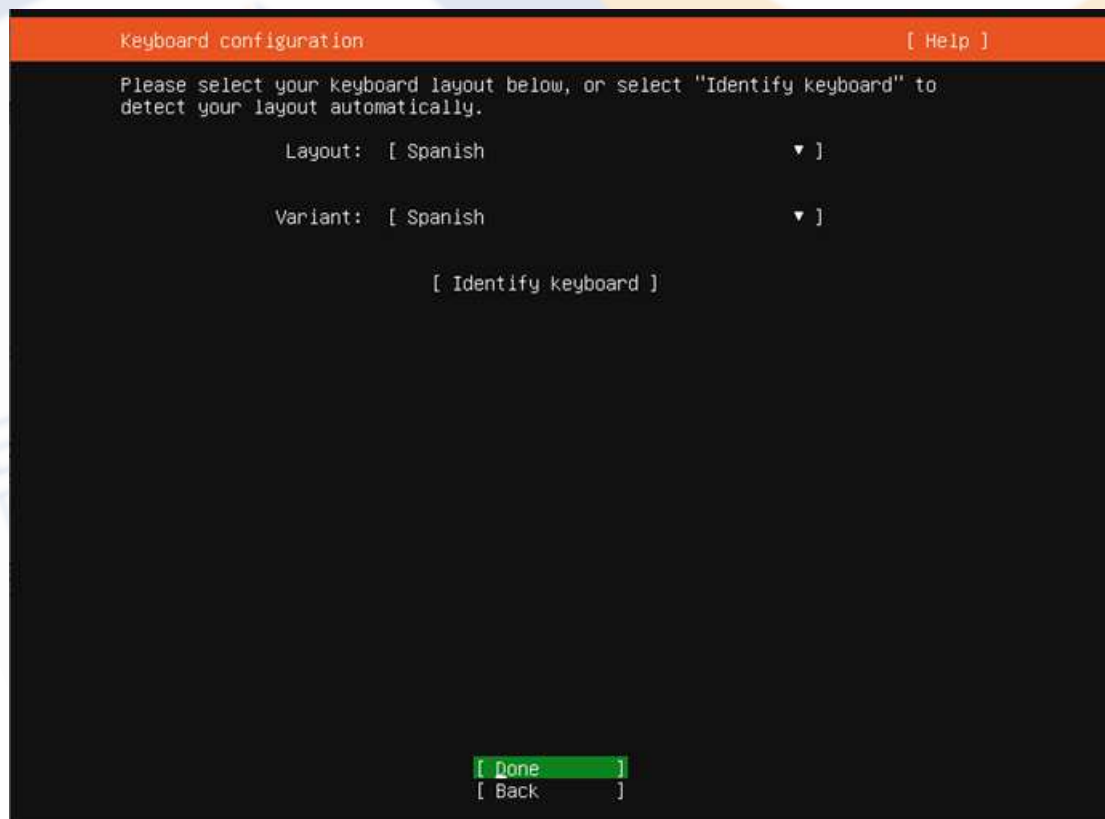
In order to move by the different screens, the used keys are:

- <Tab>: it is used for moving by the different options.
- <Space>: to select options
- <Intro>: it is used to activate the selected button.
- <Directions arrows>: it is used for moving through the different options.

The first point is choosing the language:



In the following screen, the keyboard should be selected:



In the next screen, the network should be configured. It is possible that some network has been detected; but if not, the message will be “autoconfiguration failed”. Later in the unit, the network will be configured so it should be selected by default.

Network connections

[Help]

Configure at least one interface this server can use to talk to other machines, and which preferably provides sufficient access for updates.

NAME	TYPE	NOTES
[enp0s3	eth	- ▶]
DHCPv4	192.168.1.11/24	
08:00:27:96:8b:61 / Intel Corporation / 82540EM Gigabit Ethernet Controller (PRO/1000 MT Desktop Adapter)		
[Create bond ▶]		

[Done]

[Back]

Now, a proxy address should be chosen; but in this case it is not necessary, so selected “done” option.

Configure proxy

[Help]

If this system requires a proxy to connect to the internet, enter its details here.

Proxy address:

If you need to use a HTTP proxy to access the outside world, enter the proxy information here. Otherwise, leave this blank.

The proxy information should be given in the standard form of "http://[[user] [:pass]@]host[:port]/".

[Done]

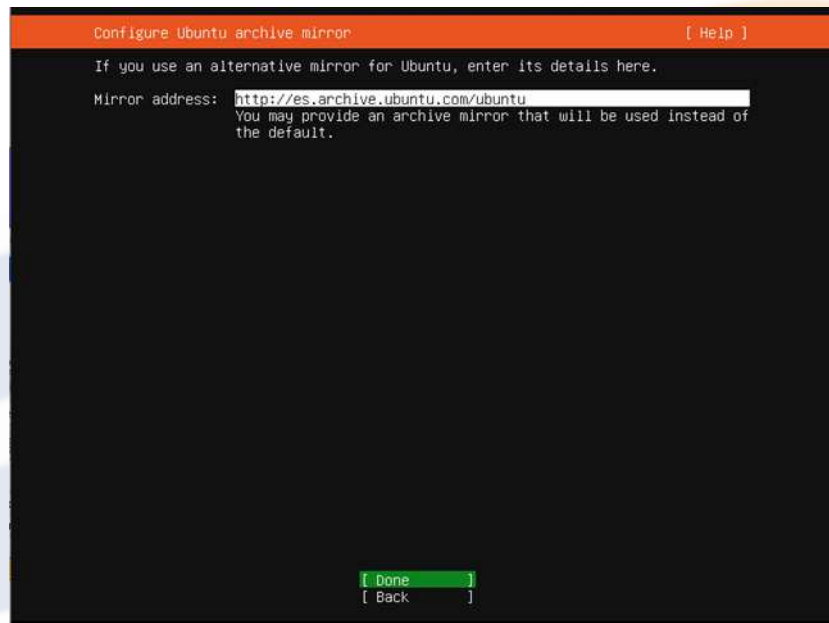
[Back]



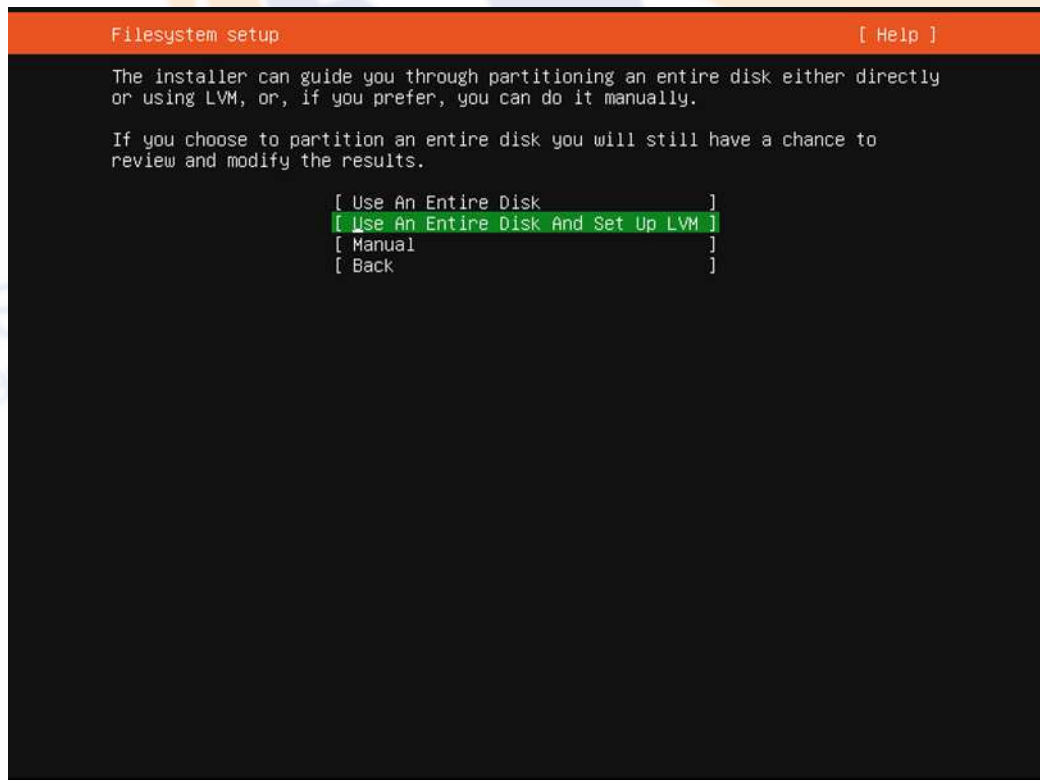
Proxy

Proxy is a server which is between a server who offer some services and a client. That configuration enables to control access, traffic registering and traffic constraints, enhanced performance, web cache, etc. . . .

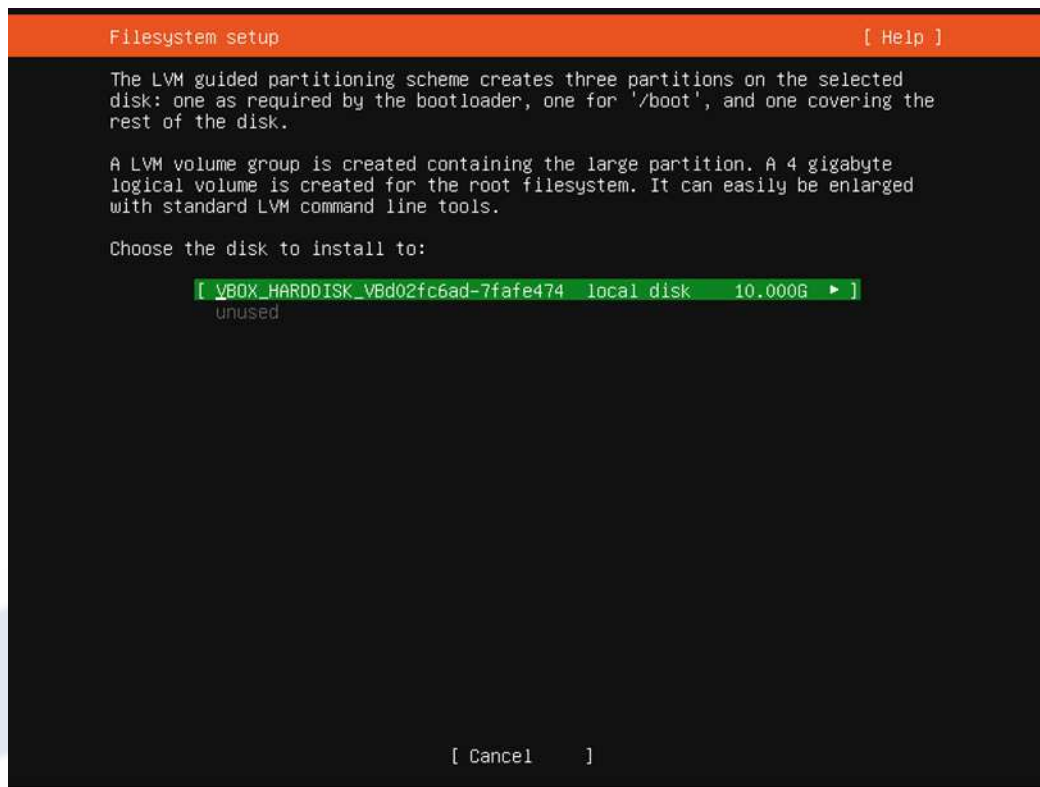
The following screen is those where the repository will be chosen, and the default option will be selected.



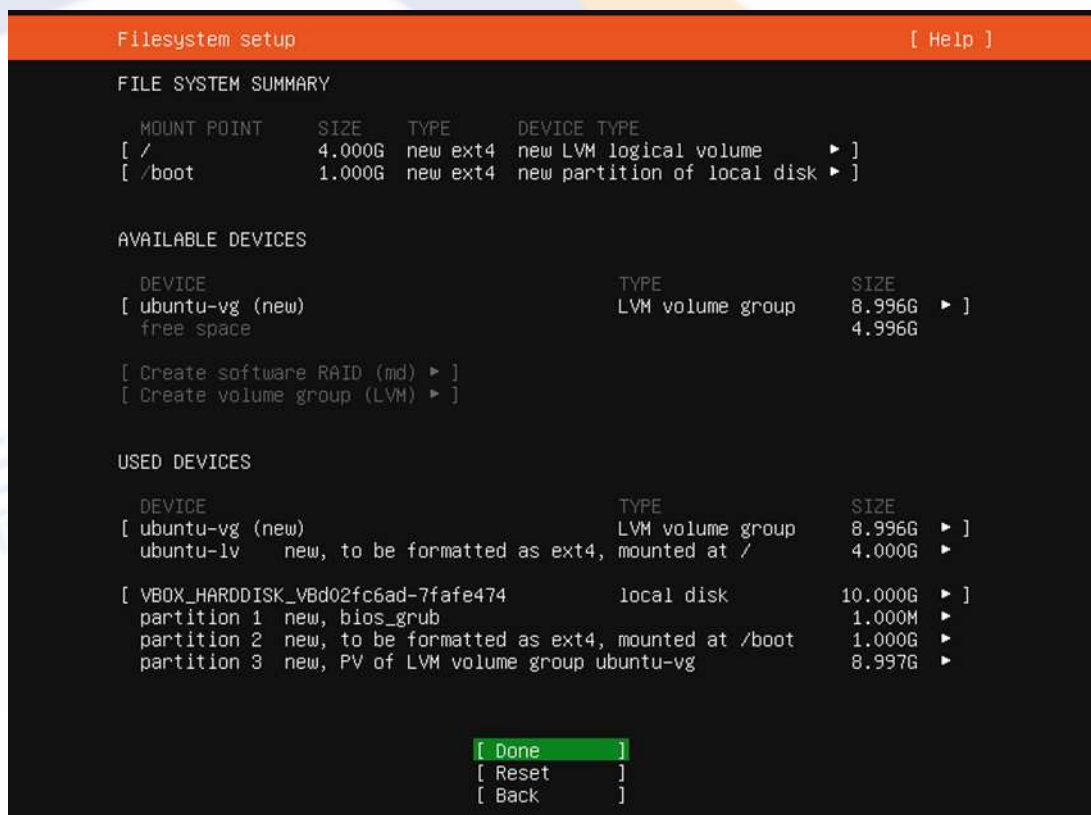
In the following steps, the hard disk will be configured. “Use an Entire Disk and Set Up LVM” option will be selected, a manual configuration is not necessary.

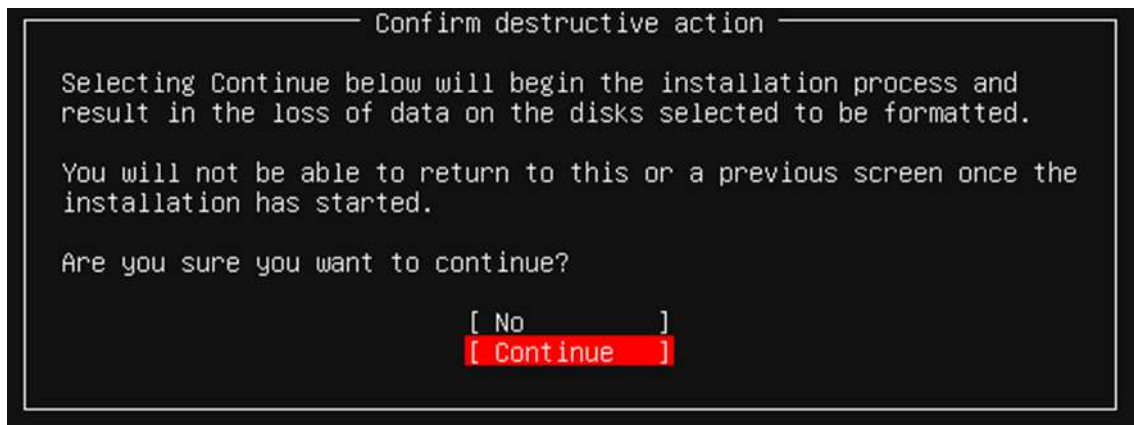


The virtual disk will be shown up, and it should be selected.

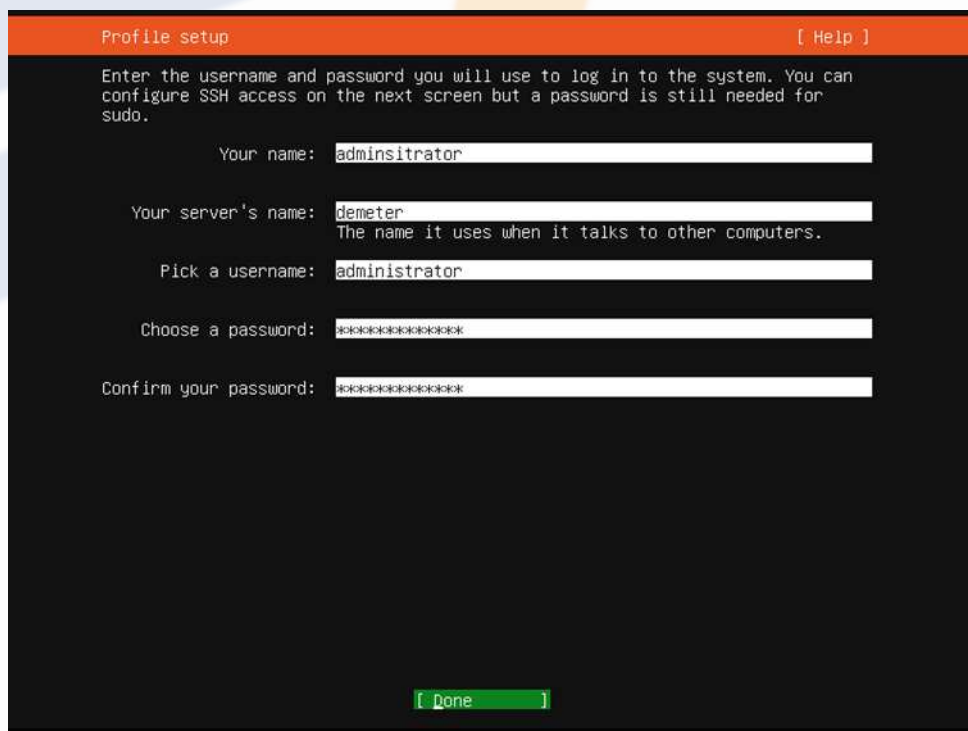


Information about the partitions of the hard disk will be shown. And at the end, a warning message will appear. The message informs that the hard disk is going to be fully erased.





Once the hard disk is configured, information about the machine and the administrator of the machine is asked. Many times, the server name is a Greek god, or something related with mythology. It is possible to choose: Atenea, Demeter, Zeus, Hera, Afrodita... or whatever is wanted.



Next step: enables to install the OpenSSH server package to enable remote access to the server. The option won't be selected.

After all these options are selected, the operating system will start the installation. And at the end, it will ask that the installation medium is removed.

Once the server has been installed, the system is rebooting, and the boot process starts. When all the process has been loaded, user and password are asked, and a welcome message will be shown.



Switch off

To switch off the server, it is possible to execute the command:

```
sudo poweroff
```

2.3 Network Configuration

From the 17.10 version, Canonical introduces a new network tool called NetPlan. The NetPlan performance is based on a file with the description of the net adaptors which will be defined. This file is a plane text file with a specific format following YAML specifications. The file can be found in the path “/etc/netplan”. More specifically, the file is “/etc/netplan/50-cloud-init.yaml”.

In order to display the content of the file it is possible to execute:

```
cat /etc/netplan/50-cloud-init.yaml
```

1. First of all, it is necessary to find the name of the active network interfaces that are going to configure. To do this, run the command:

```
ifconfig a
```

```
administrator@demeter:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.11  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::a00:27ff:fe96:8b61  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:96:8b:61  txqueuelen 1000  (Ethernet)
        RX packets 132  bytes 48838 (48.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 124  bytes 18676 (18.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop txqueuelen 1000  (Local Loopback)
        RX packets 112  bytes 8848 (8.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 112  bytes 8848 (8.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

2. In order to configure the file, it is possible to use nano editor:

```
sudo nano /etc/netplan/50-cloud-init.yaml
```

```
# This file is generated from information provided by the datasource. Changes
# to it will not persist across an instance reboot. To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    enp0s3: 1
      dhcp4: true 2
  version: 2
```

By default, the file gives the interface name (number 1 in the picture above) in this case the name is “enp0s3”. and on the other hand, that interface is using DHCP in order to get an IP address.

3. In order to configure static IP, the file should be modified. Bear in mind that the alignment of the file should be respected, don't use the tab key and use spaces. The final file would be:

```
# This file is generated from information provided by the datasource. Changes
# to it will not persist across an instance reboot. To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    enp0s3:
      dhcp4: no
      addresses:
        - 192.168.1.24/24
      gateway4: 192.168.1.1
      nameservers:
        addresses:
          - 8.8.8.8

version: 2
```

Where:

- dhcp: “no”, do not establish the static values:
 - Addresses: it establishes the static IP address. The address should belong to the network that the host is connected. The number after the address points out the mask.
 - Gateway4: it is the IP address of the device which is providing the Internet connection. It depends of the network which the host is connected.
 - Nameservers: it configures the address of the DNS server. In that case, the example shows the IP of the DNS server of google.
4. Once configured, it is necessary to update the interfaces. There are 2 ways of doing that, one way is using the netplan tool:

```
sudo netplan apply
```



Other way

The other way is restart the networkd daemon:

```
sudo systemctl status systemd-networkd
```

5. The last step, check the current network configuration with:

`ifconfig`

```
administrator@demeter:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.24 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::a00:27ff:fe96:8b61 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:96:8b:61 txqueuelen 1000 (Ethernet)
    RX packets 168 bytes 21852 (21.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 88 bytes 7824 (7.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

After configuring network, the system should be updated:

```
sudo apt update
sudo apt upgrade
```

3 Server administration: programming task

Ubuntu is able to run some programmed tasks in regular intervals. The service which is in charge of that is called cron. This service is formed by crond daemon and several tables in order to define the tasks. The crond daemon is started during the system boot and it is activated each minute in order to revise whether it has some task to execute or not. Tasks are managed through crontab command by each user.

3.1 Create a programmed task

In order to create a table for programming some task, the suitable command is :

```
crontab -e
```

If it is the first time in executing the command, a menu will appear in order to choose what editor is going to be used. Next time, the selected editor will be executed automatically.

The content of the file is just information about how the task should be programmed. Basically, each task should be written in a line with the following information:

- Minute: the value between 0 and 59
- Hour: a value between 0 and 23
- Day of the month: a value between 1 and 31
- Month of the year: a value between 1 and 12
- Day of the week: it could be expressed in two different ways:
 - As an integer number , where: 0 or 7 would be Sunday, 1 would be Monday and so on.
 - As a text: sun, mon, tue, wed, thu, fri, sat.

- The last field will be the command to be executed by Shell. Crontab doesn't check the command so the command will be sent to the Shell. Pay attention if the % character should be used because crontab will read it as the ending of line. So if % character is needed to use, the \ character should be before.

The first data can be expressed as:

- Individual values
- Ranges of values: the range will be expressed with a hyphen (-) between 2 values. For instance: mon-wed.
- List of values separated by comas.
- *: it points out all of the possible values.
- It could be expressed as an increment using /number. For instance, if some task should be executed from 5 to 15 o'clock, just the odd hours, it could be written 5-15/2.

Other way

Example: It will want to switch off the computer in 2 minutes in order to check if crontab works properly.

To do this, program this task every day in 2 minutes from current hour.

Solution ⇒ Assuming that it is now 15:30 h the task would be:

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
32 15 * * * /sbin/poweroff
```

Has the task been executed?

The problem is that the task will run with the privileges of the user account. In order that the task is executed, it is necessary to execute crontab -e with sudo before.

sudo crontab -e

After programming the task, check that the task is running.



Try it

| It is wanted to turn off the computer every working day at 15:30 h:

Some recommendations:

- When the command is written in crontab, it is recommended that it is written the command with the full path.
- If some task is programmed and the computer is turned off, the task won't be executed anymore.

Related commands:

- `crontab -l`: it displays the task list.
- `crontab -r`: it removes all the user programmed task .