Ubuntu Server on VirtualBox

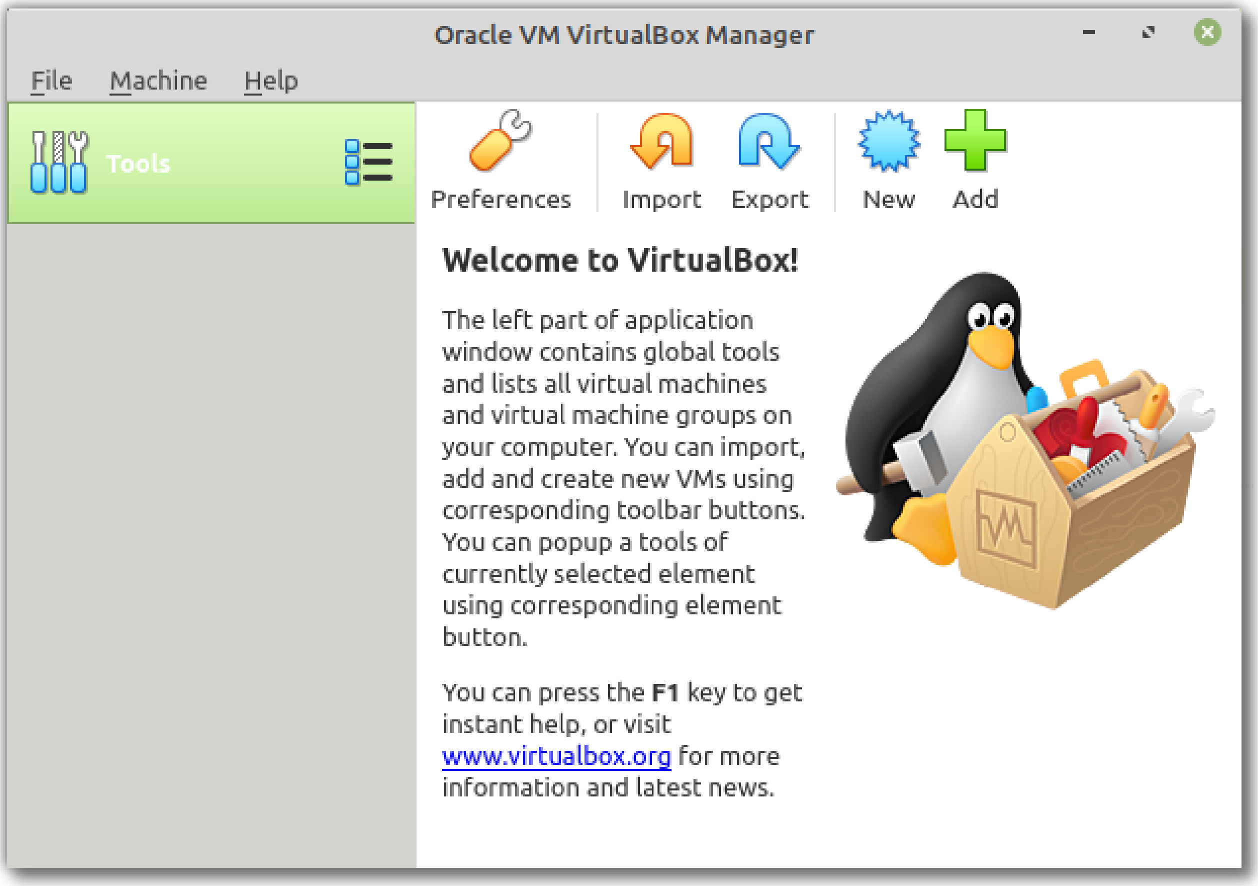
Download Ubuntu Server

The next thing to do is to grab a copy of Ubuntu Server. You can do this from their download page. This will download a 889MB iso file to your PC.

At the time of writing the current LTS version is Ubuntu Server 18.04.3 and this is what I'll be using. It's supported until April 2023 and is available as 64-bit only.

Create a New Virtual Machine

Start up VirtualBox. This should open the VirtualBox Manager, the interface from which you will administer all of your virtual machines.

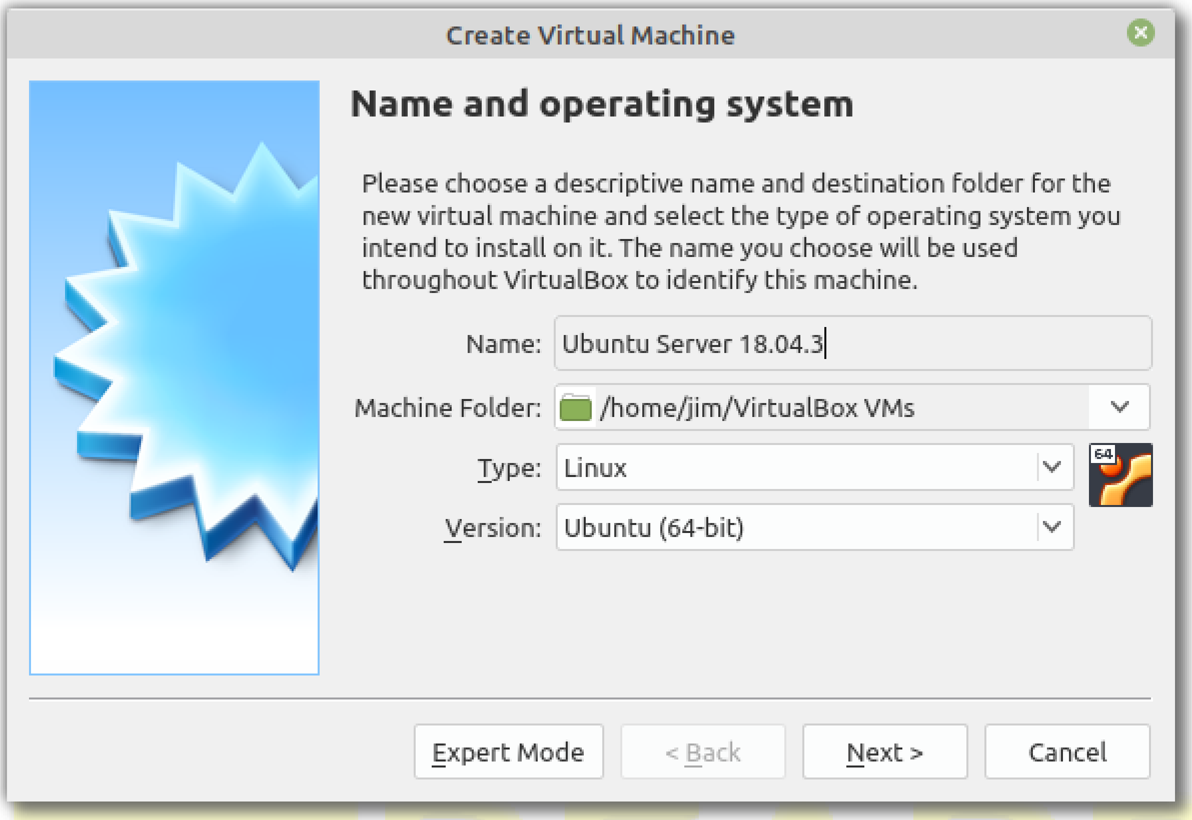


Welcome to VirtualBox!

The left part of application window contains global tools and lists all virtual machines and virtual machine groups on your computer. You can import, add and create new VMs using corresponding toolbar buttons. You can popup a tools of currently selected element using corresponding element button.

You can press the **F1** key to get instant help, or visit www.virtualbox.org for more information and latest news.

Next Click on *New* (in the top right of the VirtualBox Manager), give your virtual machine a name and the two drop down menus should automatically update.



Create Virtual Machine

Name and operating system

Please choose a descriptive name and destination folder for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

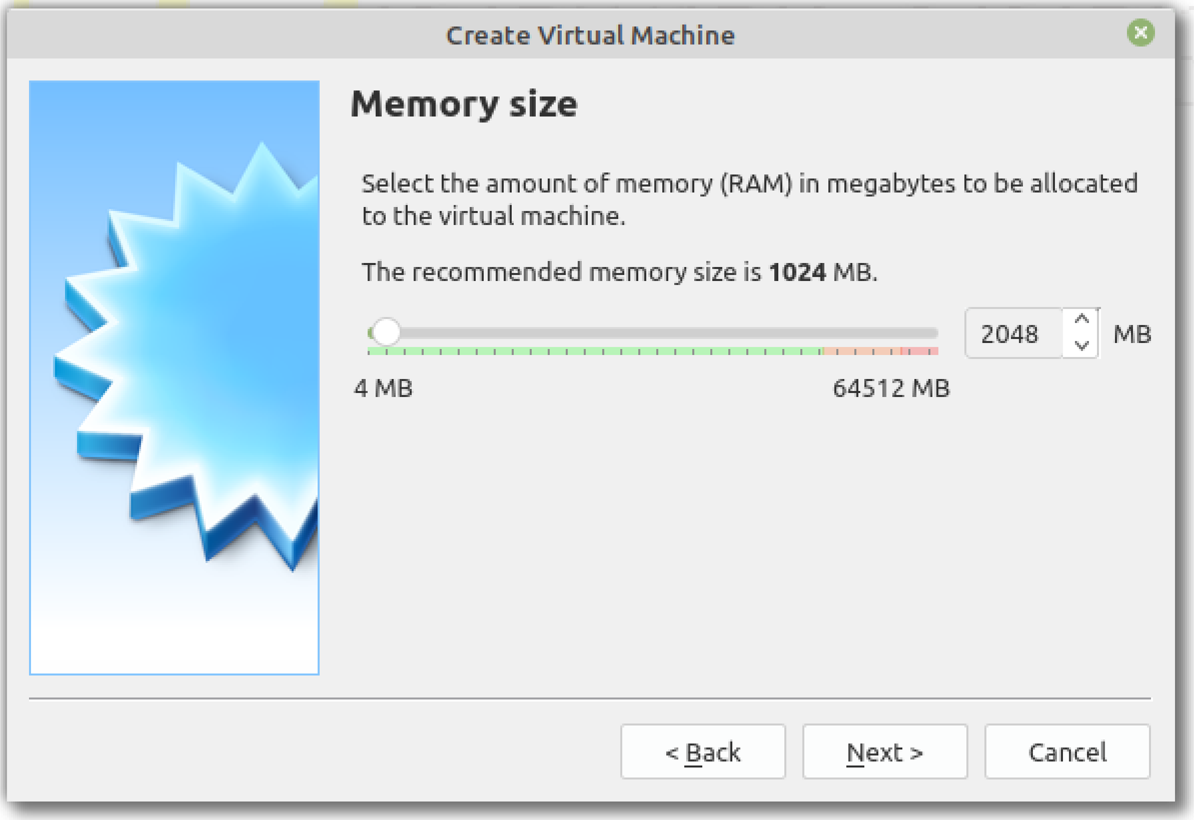
Name: Ubuntu Server 18.04.3

Machine Folder: /home/jim/VirtualBox VMs

Type: Linux

Version: Ubuntu (64-bit)

Click *Next*. The wizard will now ask you to select the amount of memory (RAM) in megabytes to be allocated to the virtual machine. I chose 2GB (2048 megabytes).



Create Virtual Machine

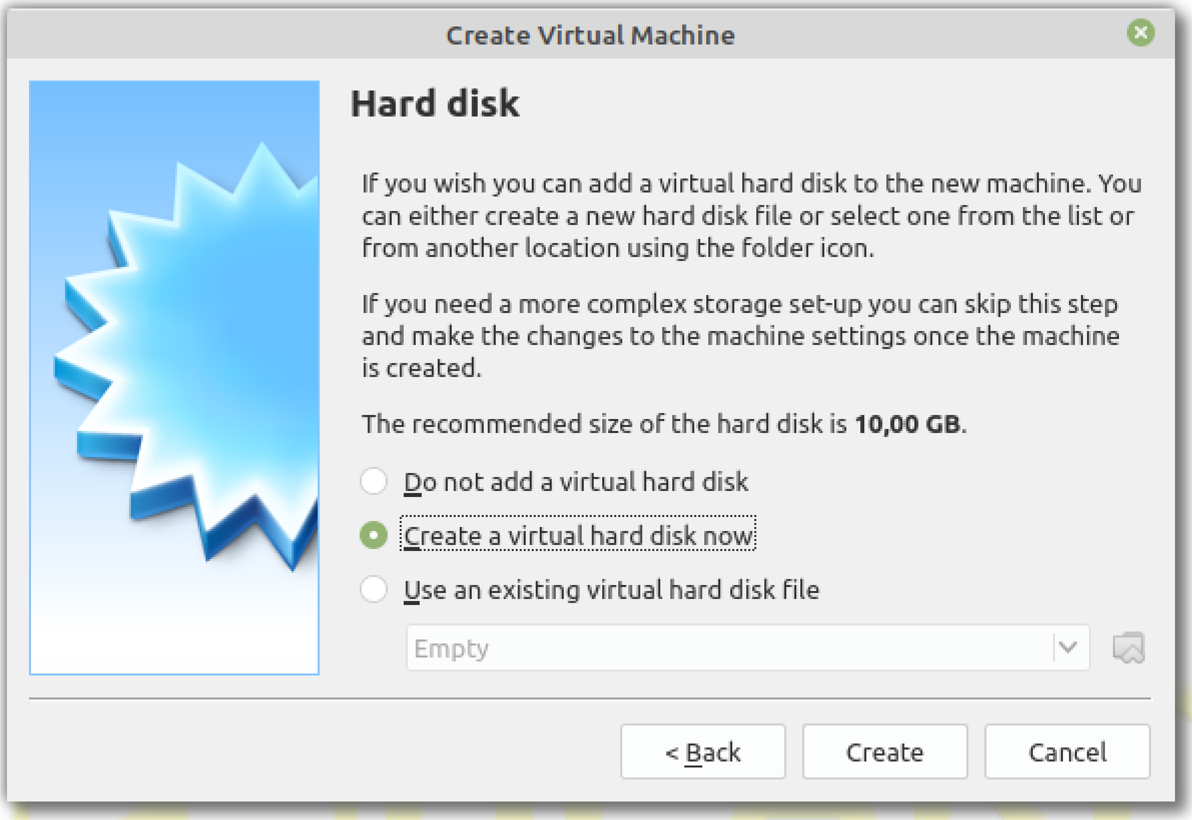
Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **1024** MB.

2048 MB

Click *Next* and you will be prompted to add a virtual hard disk to the new machine. Make sure that *Create a virtual hard disk now* is selected, then press *Create*.



Create Virtual Machine

Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

The recommended size of the hard disk is **10,00 GB**.

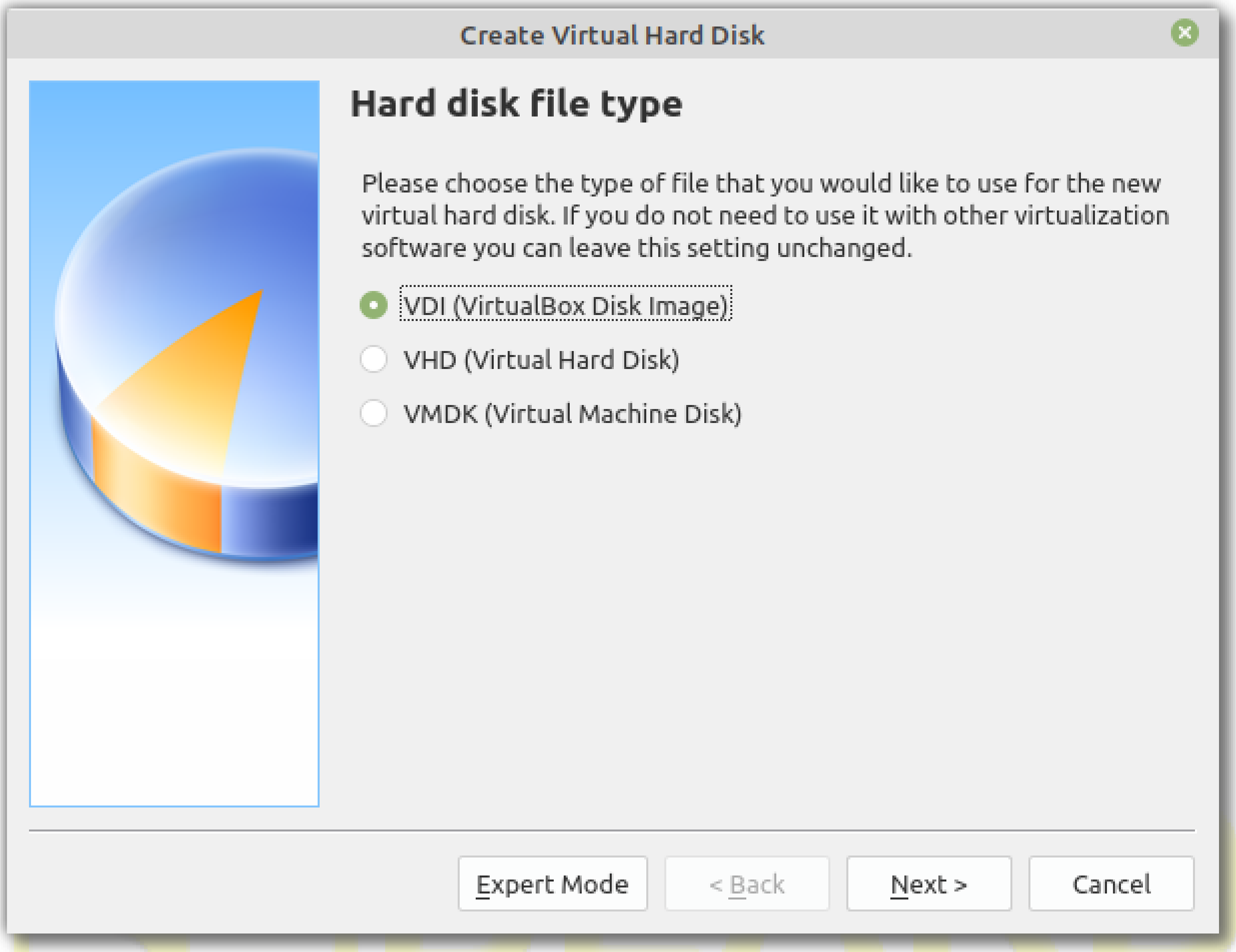
○ Do not add a virtual hard disk

● Create a virtual hard disk now

○ Use an existing virtual hard disk file

Empty

Now we need to choose the file type for the new virtual hard disk. Make sure that *VDI (VirtualBox Disk Image)* is checked and press *Next*.



Create Virtual Hard Disk

Hard disk file type

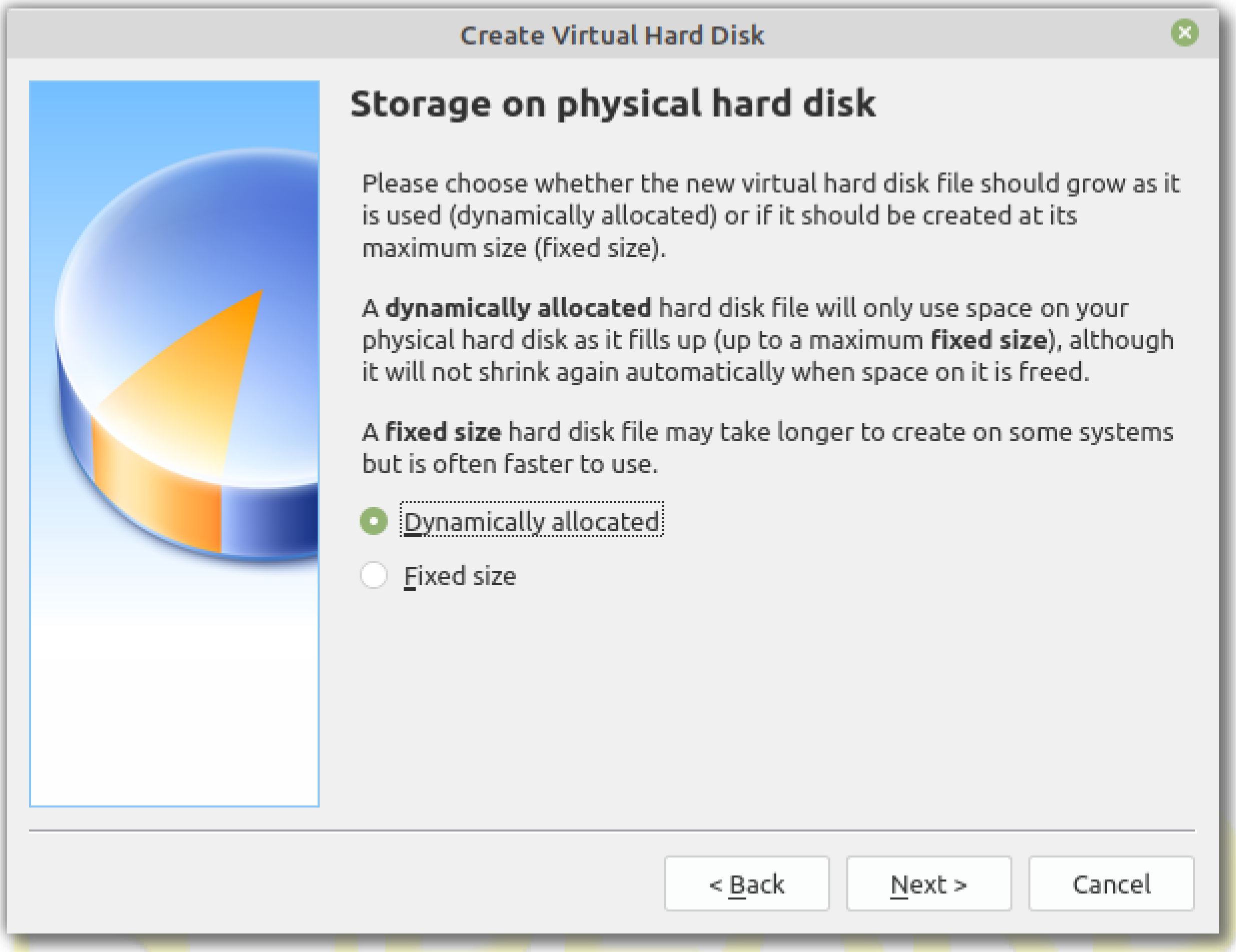
Please choose the type of file that you would like to use for the new virtual hard disk. If you do not need to use it with other virtualization software you can leave this setting unchanged.

● VDI (VirtualBox Disk Image)

○ VHD (Virtual Hard Disk)

○ VMDK (Virtual Machine Disk)

On the next screen you will be asked whether the new virtual hard disk should grow as it is used (dynamically allocated) or if it should be created at its maximum size. Make sure that *dynamically allocated* is selected, then click *Next*.



Create Virtual Hard Disk

Storage on physical hard disk

Please choose whether the new virtual hard disk file should grow as it is used (dynamically allocated) or if it should be created at its maximum size (fixed size).

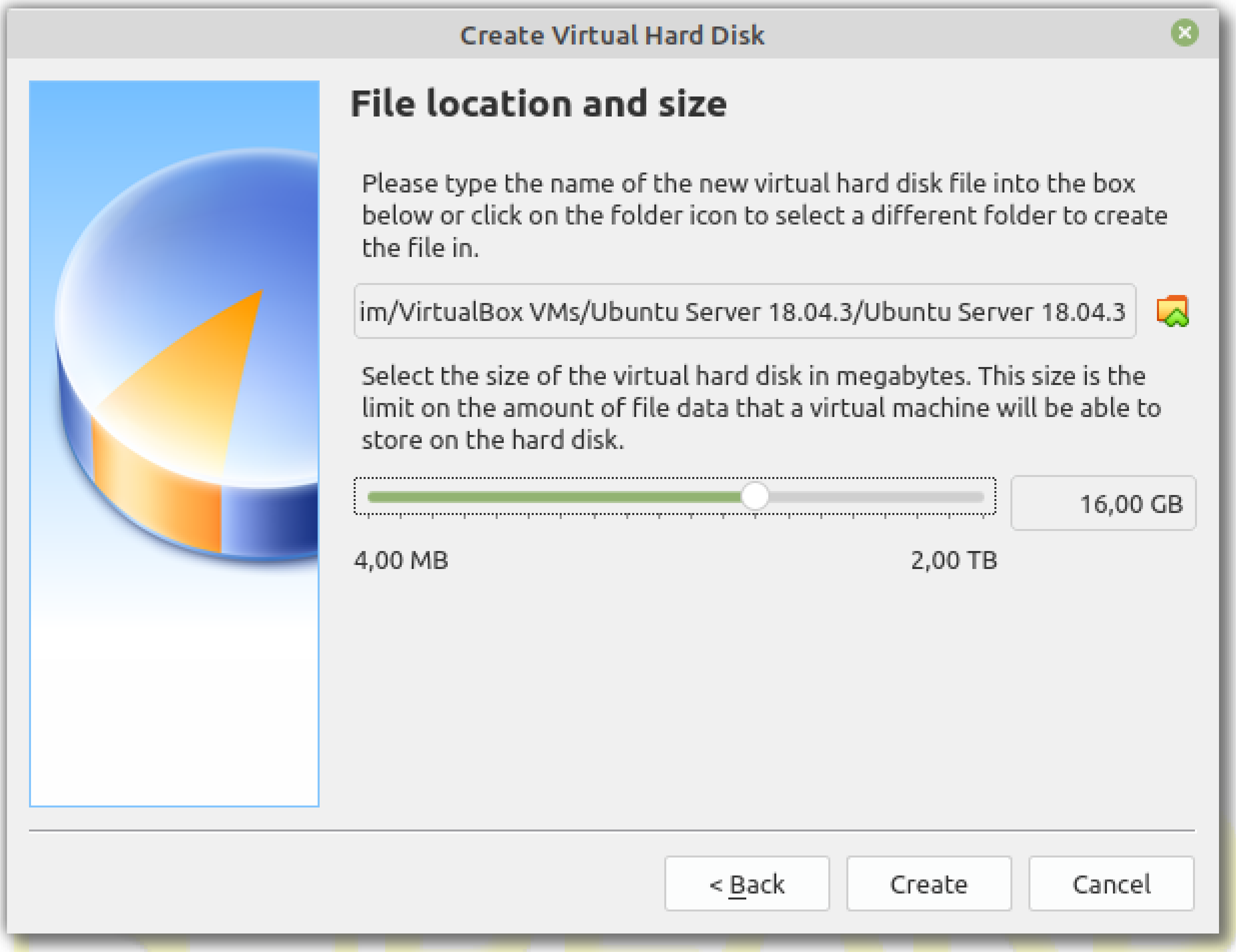
A **dynamically allocated** hard disk file will only use space on your physical hard disk as it fills up (up to a maximum **fixed size**), although it will not shrink again automatically when space on it is freed.

A **fixed size** hard disk file may take longer to create on some systems but is often faster to use.

● Dynamically allocated

○ Fixed size

Finally, select the size of the virtual hard disk in megabytes. The default size of 10GB should be plenty, but feel free to increase this as you see fit. Then click *Create*.



File location and size

Please type the name of the new virtual hard disk file into the box below or click on the folder icon to select a different folder to create the file in.

im/VirtualBox VMs/Ubuntu Server 18.04.3/Ubuntu Server 18.04.3

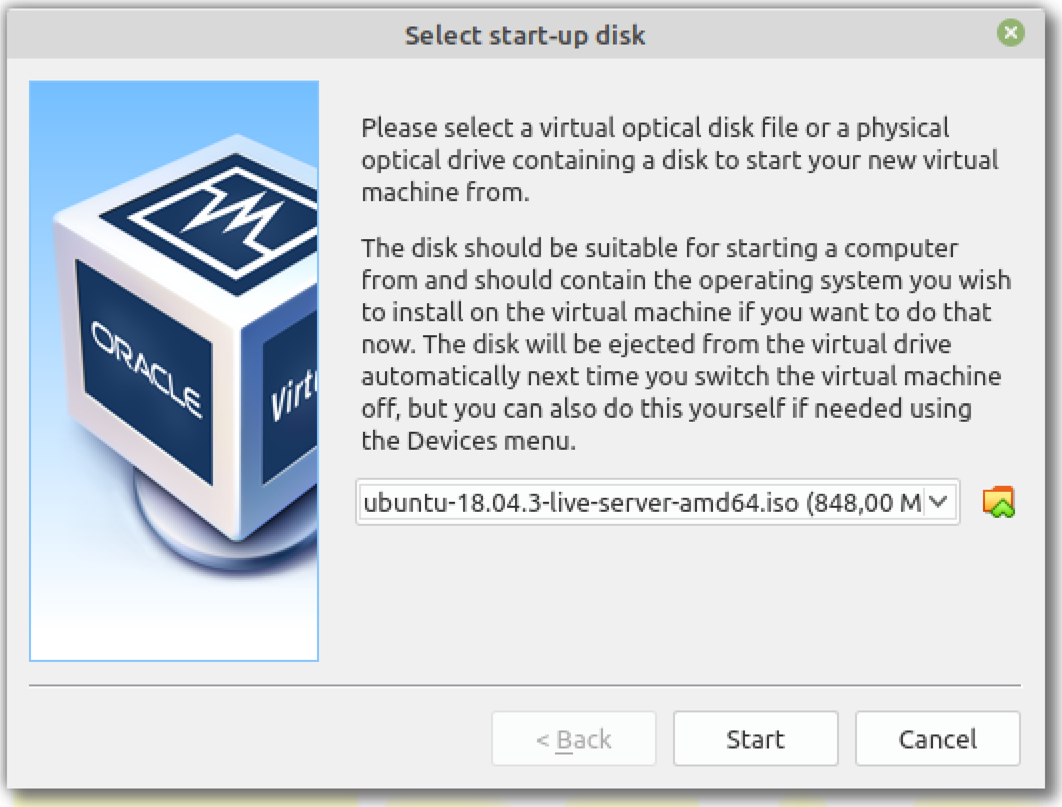
Select the size of the virtual hard disk in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard disk.

16,00 GB

The hard disk should now be created and after a short while you should find yourself back in the VirtualBox Manager. You should be able to see your newly created virtual machine listed on the left.

Install Ubuntu Server in the Virtual Machine

Make sure your virtual machine is selected and press *Start*. VirtualBox Manager will ask you to select a virtual optical disk file or a physical optical drive to start the virtual machine from. Select the iso file you downloaded previously and press *Start*.



Select start-up disk

Please select a virtual optical disk file or a physical optical drive containing a disk to start your new virtual machine from.

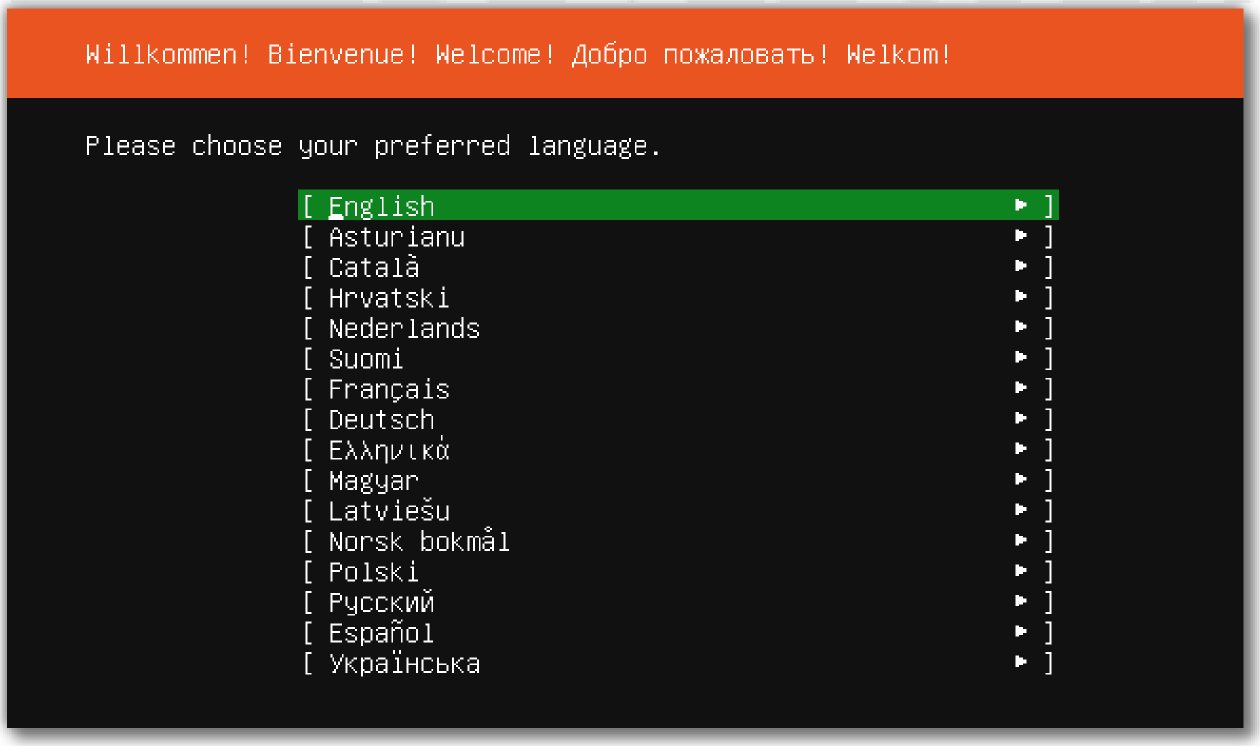
The disk should be suitable for starting a computer from and should contain the operating system you wish to install on the virtual machine if you want to do that now. The disk will be ejected from the virtual drive automatically next time you switch the virtual machine off, but you can also do this yourself if needed using the Devices menu.

ubuntu-18.04.3-live-server-amd64.iso (848,00 M

The Ubuntu installation process will now begin. It consists of thirteen steps and is quite painless.

The Welcome Screen

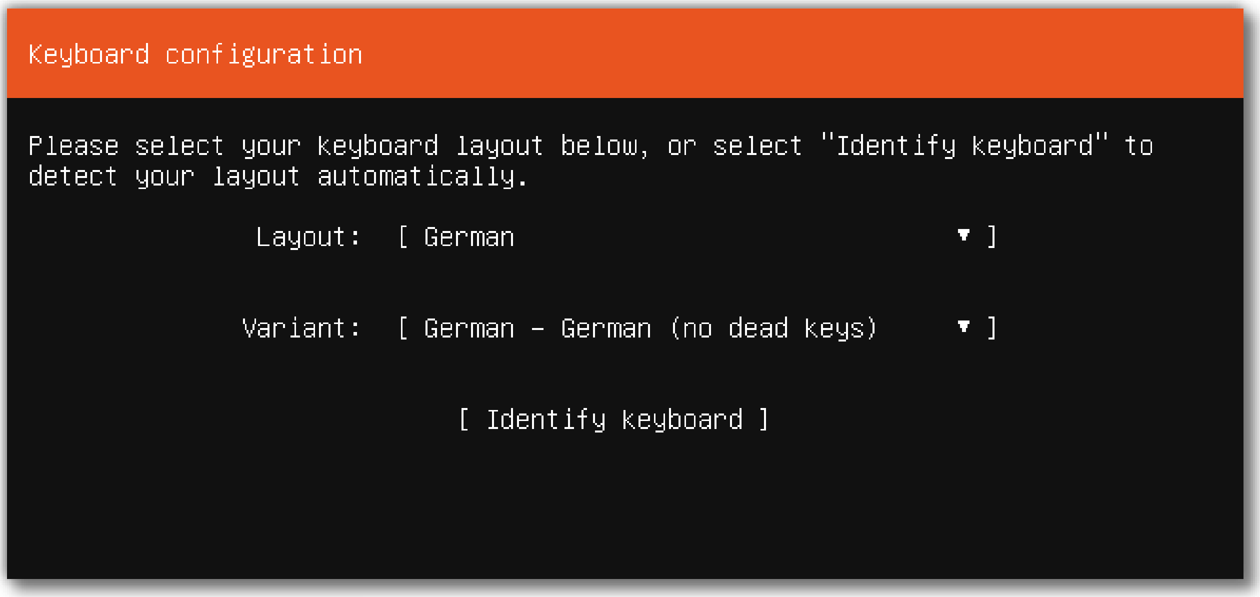
Here you should select your preferred language. I'm using English.



```
Willkommen! Bienvenue! Welcome! Добро пожаловать! Welkom!

Please choose your preferred language.

                [ English                                     ► ]
                [ Asturianu                                   ► ]
                [ Català                                      ► ]
                [ Hrvatski                                    ► ]
                [ Nederlands                                  ► ]
                [ Suomi                                       ► ]
                [ Français                                    ► ]
                [ Deutsch                                     ► ]
                [ Ελληνικά                                    ► ]
                [ Magyar                                      ► ]
                [ Latviešu                                    ► ]
                [ Norsk bokmål                                ► ]
                [ Polski                                      ► ]
                [ Русский                                     ► ]
                [ Español                                     ► ]
                [ Українська                                  ► ]
```

The Keyboard Configuration Screen

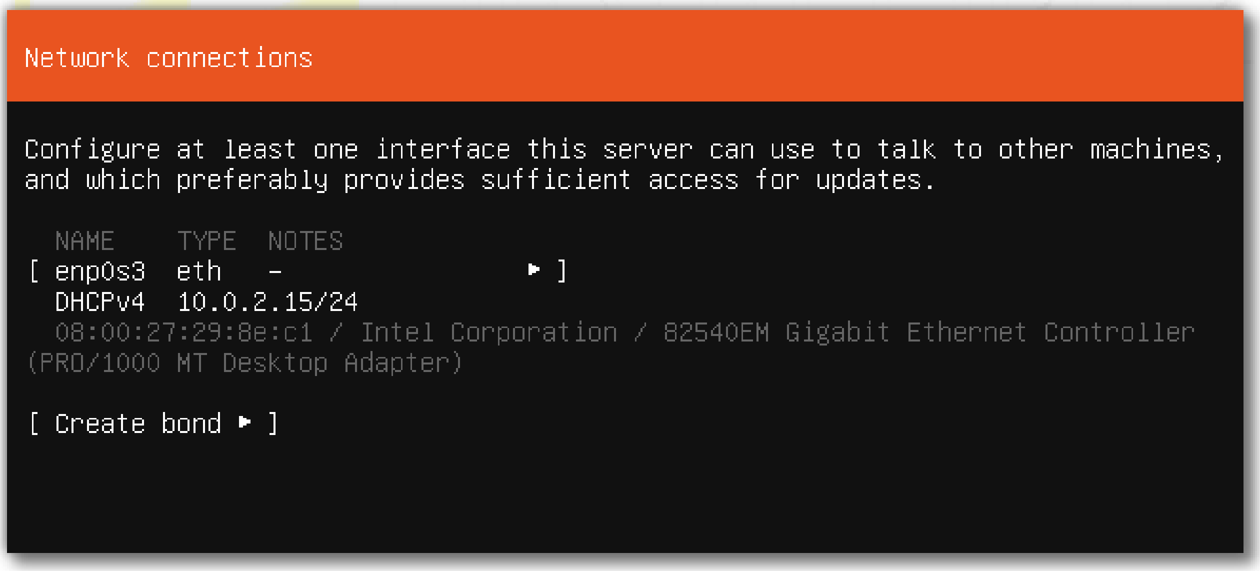
Here you should select a keyboard layout. As I'm using a German keyboard, I asked Ubuntu to detect my layout, which it did with a couple of simple questions.



```
Keyboard configuration

Please select your keyboard layout below, or select "Identify keyboard" to
detect your layout automatically.

               Layout:  [ German                         ▾ ]

              Variant:  [ German - German (no dead keys)  ▾ ]

                              [ Identify keyboard ]
```

The Network Connections Screen

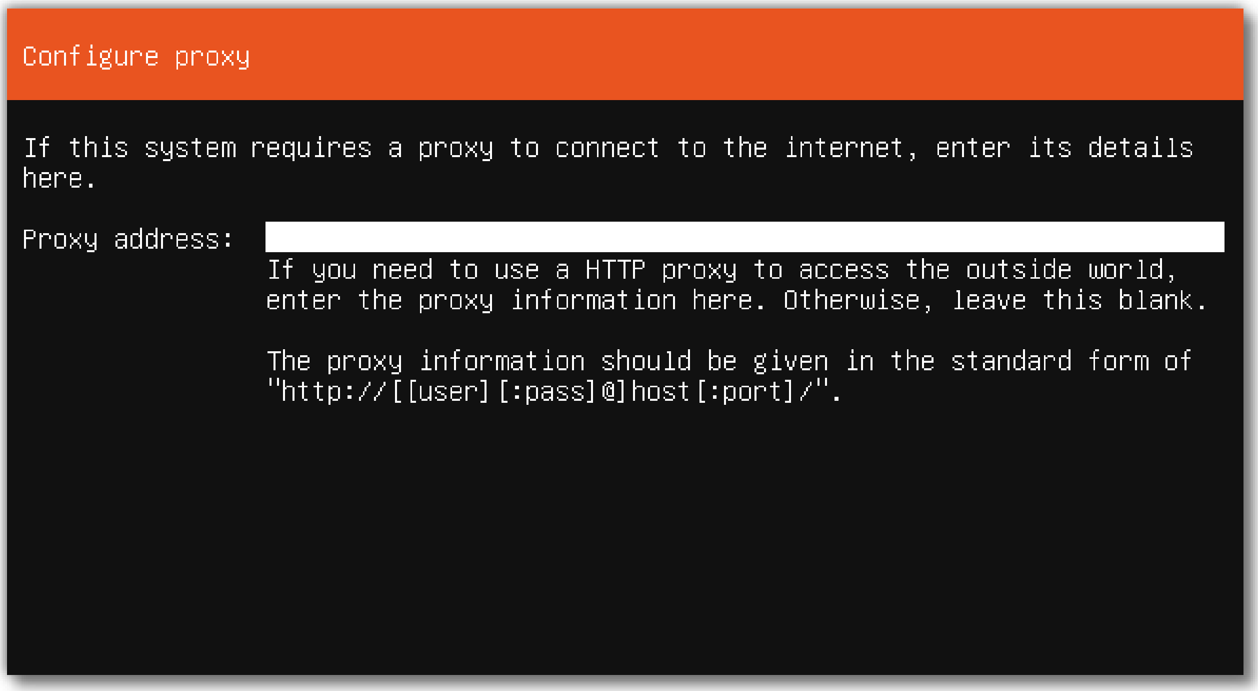
Here Ubuntu will attempt to configure the standard network interface. Normally you can just accept the default and select *Done*.



```
Network connections

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  10.0.2.15/24
  08:00:27:29:8e:c1 / Intel Corporation / 82540EM Gigabit Ethernet Controller
(PRO/1000 MT Desktop Adapter)

[ Create bond ▸ ]
```

The Configure Proxy Screen

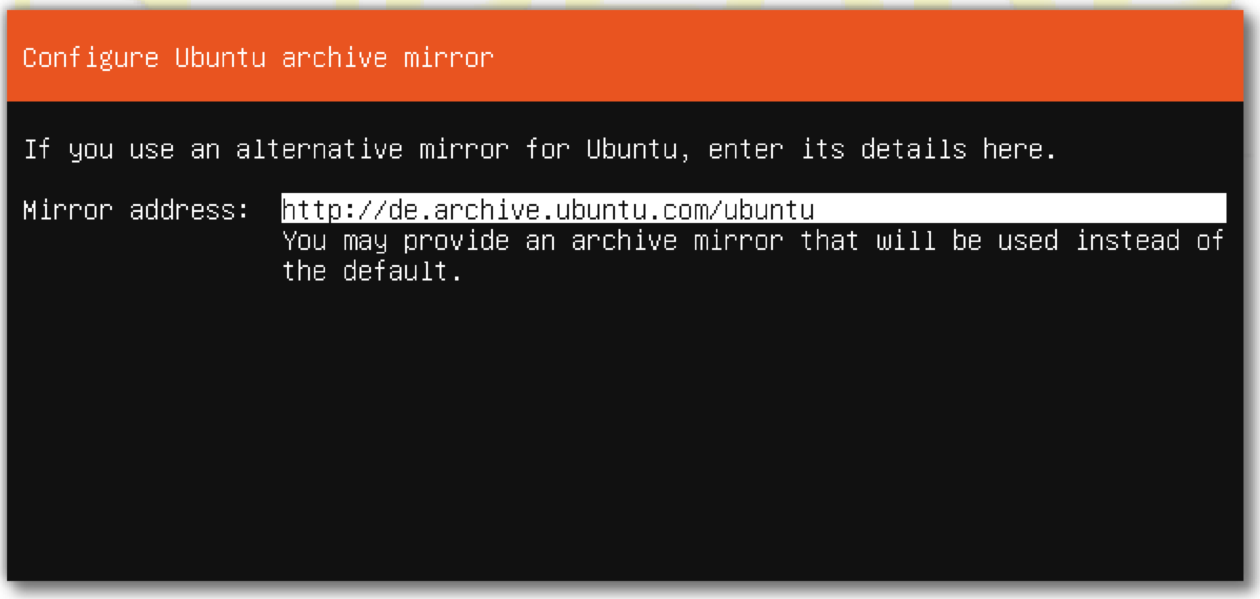
If your system requires a proxy to connect to the internet (mine doesn't), enter its details in the next dialogue. Then select *Done*.



```
Configure proxy

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]/".
```

The Ubuntu Archive Mirror Screen

If you wish to use an alternative mirror for Ubuntu, you can enter the details here. Otherwise accept the default mirror by selecting *Done*.

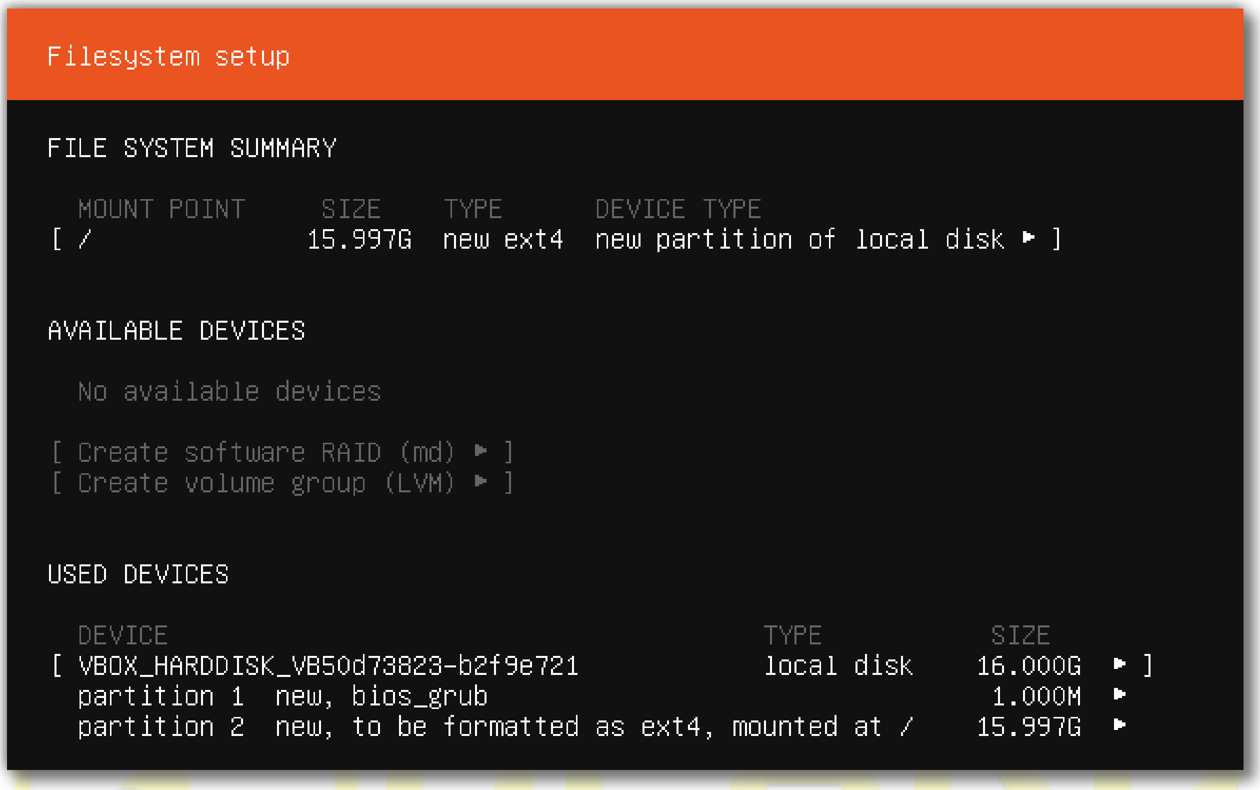


```
Configure Ubuntu archive mirror

If you use an alternative mirror for Ubuntu, enter its details here.

Mirror address:  http://de.archive.ubuntu.com/ubuntu
                 You may provide an archive mirror that will be used instead of
                 the default.
```

The Filesystem Setup Screen

The installer can guide you through partitioning an entire disk or, if you prefer, you can do it manually. If you choose to partition an entire disk you will still have a chance to review and modify the results before Ubuntu is installed. I selected *Use An Entire Disk*.

I was then prompted to select my virtual machine’s hard disk as the disk to install to, before being shown a summary of what the installer would do. As this is a “destructive action”. I was asked to confirm my choice with *Continue*.



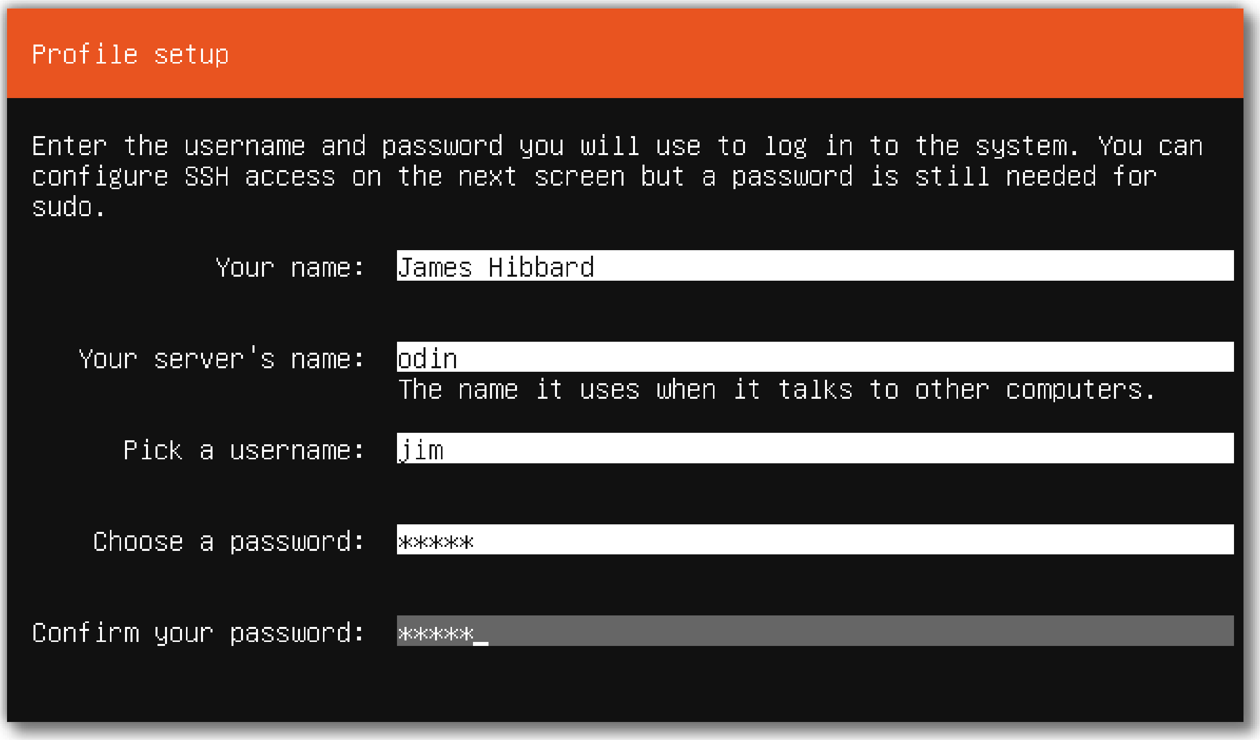
```
Filesystem setup

FILE SYSTEM SUMMARY

  MOUNT POINT     SIZE    TYPE      DEVICE TYPE
[ /              15.997G  new ext4  new partition of local disk ► ]


AVAILABLE DEVICES

  No available devices

[ Create software RAID (md) ► ]
[ Create volume group (LVM) ► ]


USED DEVICES

  DEVICE                                         TYPE          SIZE
[ VBOX_HARDDISK_VB50d73823-b2f9e721              local disk    16.000G  ► ]
  partition 1  new, bios_grub                                   1.000M  ►
  partition 2  new, to be formatted as ext4, mounted at /      15.997G  ►
```

The Profile Setup Screen

Here you are required to enter:

- Your (real) name
- Your server’s name
- Your username
- Password

Fill these details out as you see fit.

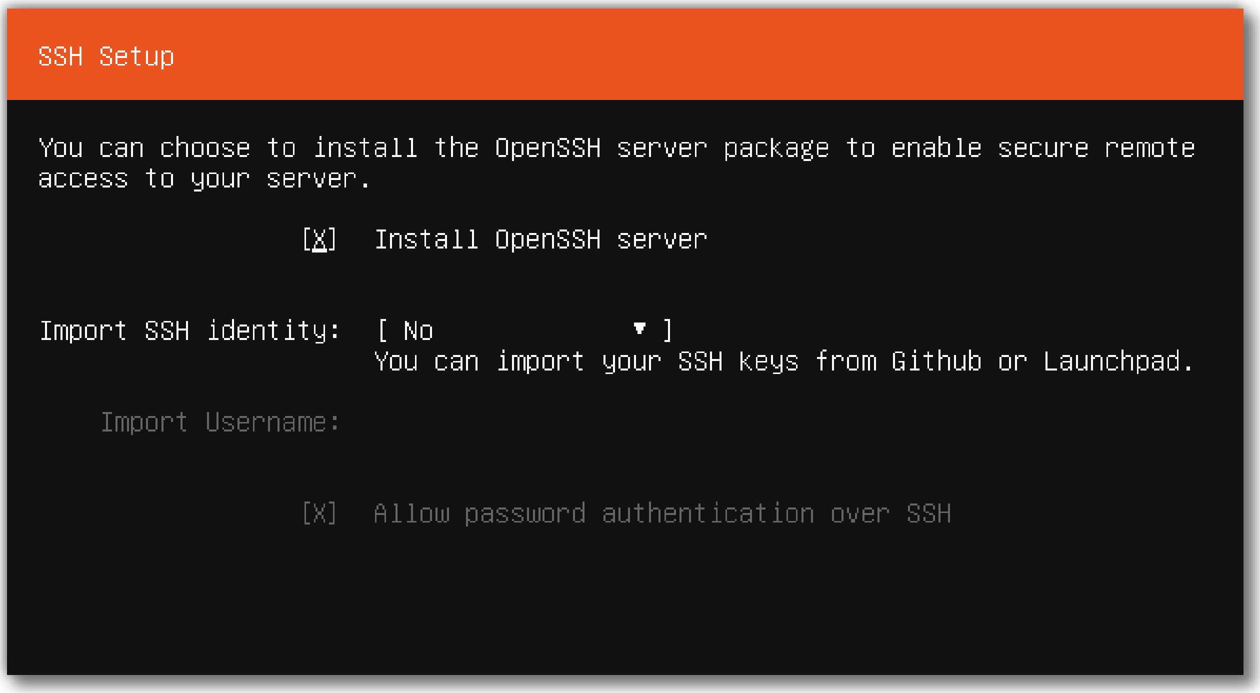


```
Profile setup

Enter the username and password you will use to log in to the system. You can
configure SSH access on the next screen but a password is still needed for
sudo.

              Your name: James Hibbard

       Your server's name: odin
                           The name it uses when it talks to other computers.

         Pick a username: jim

       Choose a password: ******

   Confirm your password: ******_
```

The SSH Setup Screen

Here we have a chance to install the OpenSSH server package. We'll need this to connect to the virtual machine via SSH later on, so ensure that you select it.

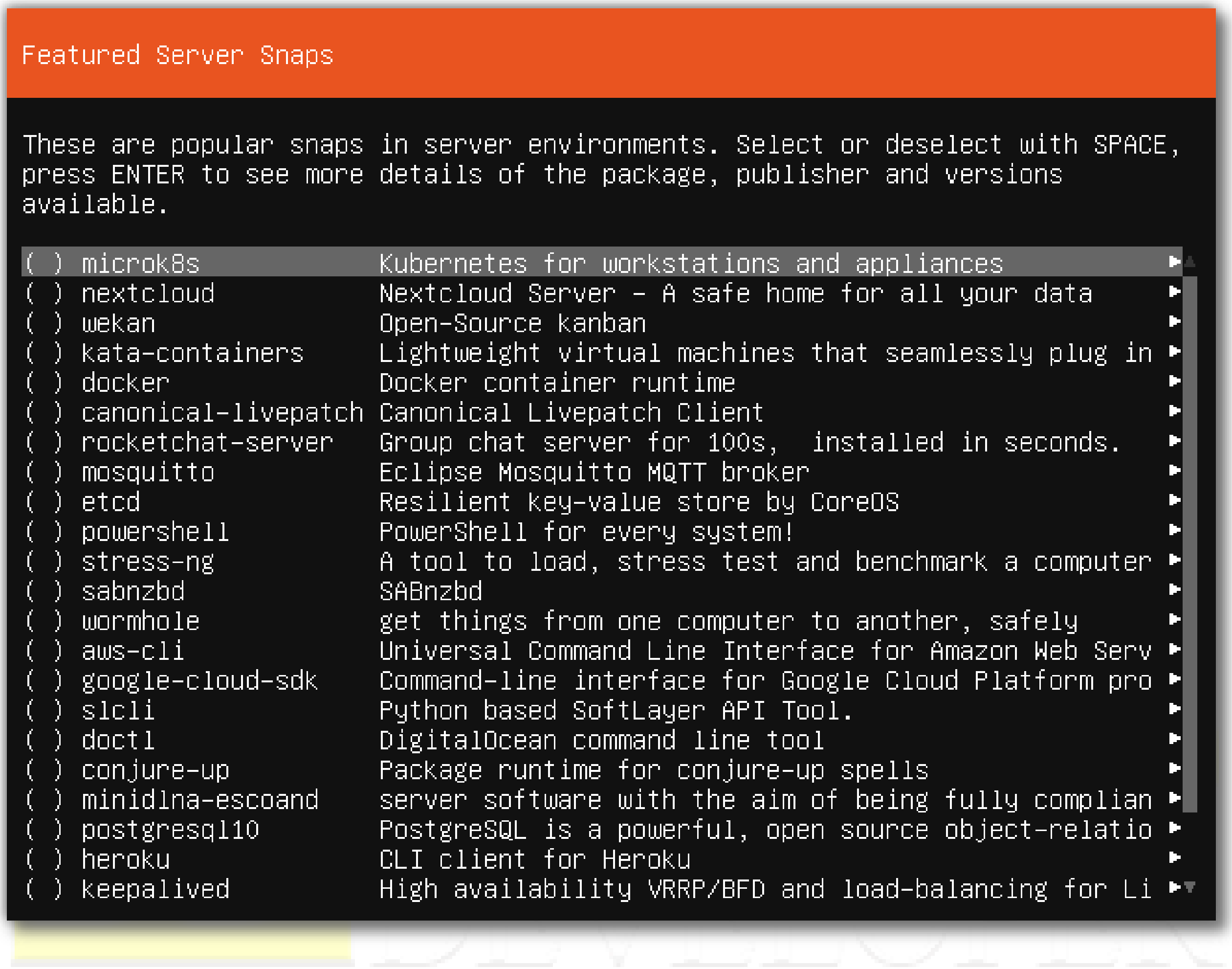
You also have the opportunity to import your SSH keys from GitHub or Launchpad. I selected *No* for this option.



```
SSH Setup

You can choose to install the OpenSSH server package to enable secure remote
access to your server.

                 [X]  Install OpenSSH server

Import SSH identity:  [ No              ▼ ]
                      You can import your SSH keys from Github or Launchpad.

    Import Username:

                 [X]  Allow password authentication over SSH
```

The Featured Server Snaps screen

Here you can select from a list of popular snaps to install on your system. Snaps are self-contained software packages that work across a range of Linux distributions. I didn't select any.



```
Featured Server Snaps

These are popular snaps in server environments. Select or deselect with SPACE,
press ENTER to see more details of the package, publisher and versions
available.

( ) microk8s              Kubernetes for workstations and appliances
( ) nextcloud             Nextcloud Server - A safe home for all your data
( ) wekan                 Open-Source kanban
( ) kata-containers       Lightweight virtual machines that seamlessly plug in
( ) docker                Docker container runtime
( ) canonical-livepatch   Canonical Livepatch Client
( ) rocketchat-server     Group chat server for 100s,  installed in seconds.
( ) mosquitto             Eclipse Mosquitto MQTT broker
( ) etcd                  Resilient key-value store by CoreOS
( ) powershell            PowerShell for every system!
( ) stress-ng             A tool to load, stress test and benchmark a computer
( ) sabnzbd               SABnzbd
( ) wormhole              get things from one computer to another, safely
( ) aws-cli               Universal Command Line Interface for Amazon Web Serv
( ) google-cloud-sdk      Command-line interface for Google Cloud Platform pro
( ) slcli                 Python based SoftLayer API Tool.
( ) doctl                 DigitalOcean command line tool
( ) conjure-up            Package runtime for conjure-up spells
( ) minidlna-escoand      server software with the aim of being fully complian
( ) postgresql10          PostgreSQL is a powerful, open source object-relatio
( ) heroku                CLI client for Heroku
( ) keepalived            High availability VRRP/BFD and load-balancing for Li
```

And that's it, the installation is complete. Ubuntu will ask you to remove the installation medium (which you can do via *Devices > Optical Drives > Remove disk from virtual drive > Force unmount*) and then reboot.

If this option is grayed out, you're good to go. Just reboot.

Up and Running with SSH

Once your virtual machine has rebooted and you have logged in, you'll probably notice that some packages can be updated.

Let's fix that:

```
sudo apt-get update
sudo apt-get upgrade
```

Now let's double check that SSH is installed (it should be if you selected the option *Install OpenSSH server* during instalation).

```
jim@odin:~$ ssh
```

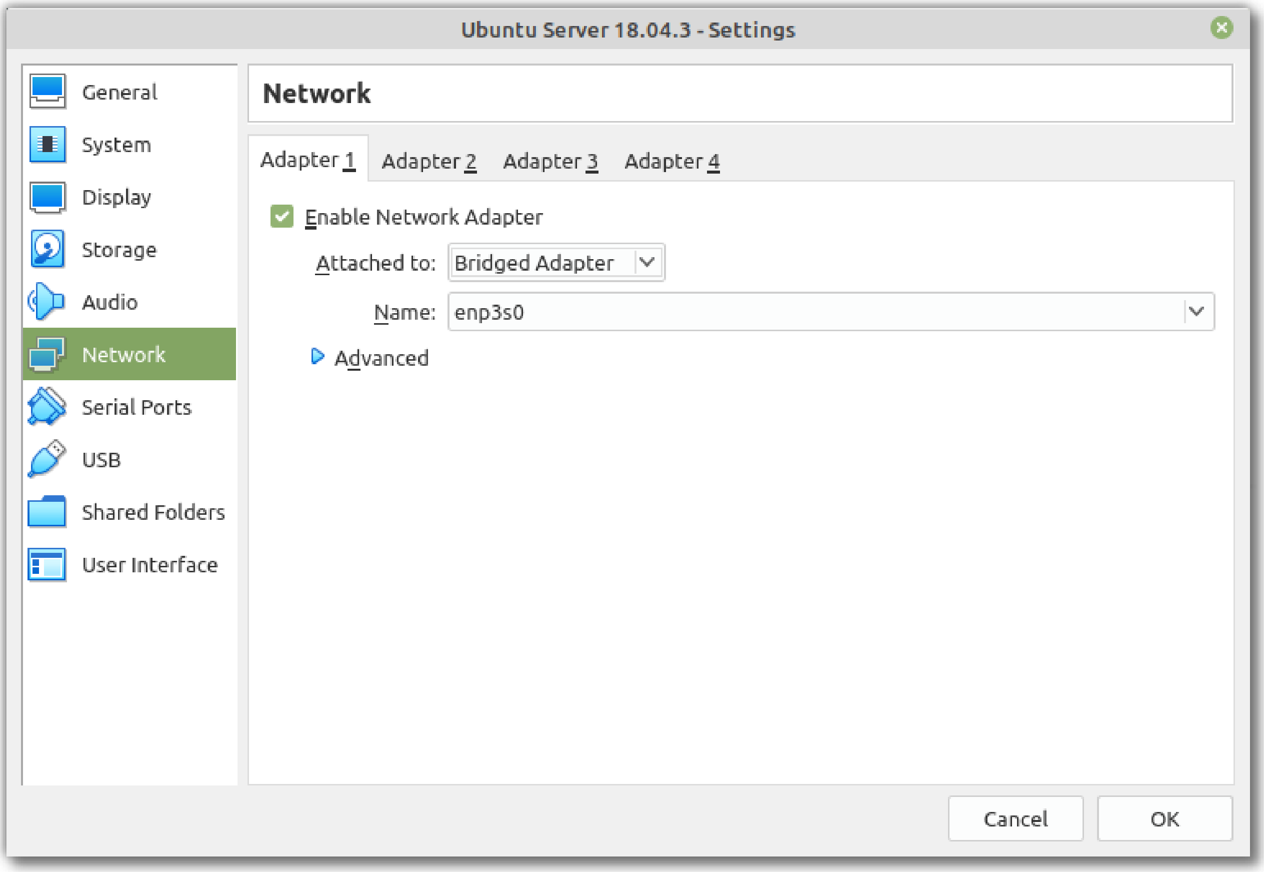
```
usage: ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-b bind_address] [-c cipher_spec]
           [-D [bind_address:]port] [-E log_file] [-e escape_char]
           [-F configfile] [-I pkcs11] [-i identity_file]
           [-J [user@]host[:port]] [-L address] [-l login_name] [-m mac_spec]
           [-O ctl_cmd] [-o option] [-p port] [-Q query_option] [-R address]
           [-S ctl_path] [-W host:port] [-w local_tun[:remote_tun]]
           [user@]hostname [command]
```

If you get a “command not found” error, you can install it with:

```
sudo apt-get install openssh-server
```

The next step is to give our Ubuntu server an IP address on our local network. To do this, power off the virtual machine using `sudo poweroff` or *Machine* > *ACPI Shutdown*.

Then, in VirtualBox Manager, click on the *Network* panel on the right. Change the setting *Adapter 1* > *Attached to* to “Bridged Adapter” and click *OK*.



☑ Enable Network Adapter

Attached to: Bridged Adapter

Name: enp3s0

Start up your virtual machine, then enter `ifconfig` (in the guest) and note the IP address assigned to your main network adapter. In my case this was `192.168.178.66`.

Starting and Stopping VirtualBox in Headless Mode

You might have noticed, working with the VirtualBox Manager and the guest OS is a bit of a pain. If you’re going to continue doing this, you should at least install the guest additions, as well as enable clipboard support.

There is a slightly nicer way however — you can start and stop the virtual machine using the VBoxManage command from your terminal.

To power on:

```
VBoxManage startvm "Ubuntu Server 18.04.3" --type headless
```

And to power off:

```
VBoxManage controlvm "Ubuntu Server 18.04.3" poweroff
```

Where “Ubuntu Server 18.04.3” is whatever you called your virtual machine (the name it has in the VirtualBox Manager GUI).

Connecting to the Ubuntu Server

Let’s go ahead and start the Ubuntu server in headless mode, before connecting to it via SSH.

Note: The following commands should be run on your host.

On most *nix systems, the SSH client software should be part of the default installation. If you don’t have it available, you should be able to grab it from the repos, like so:

```
sudo apt install openssh-client
```

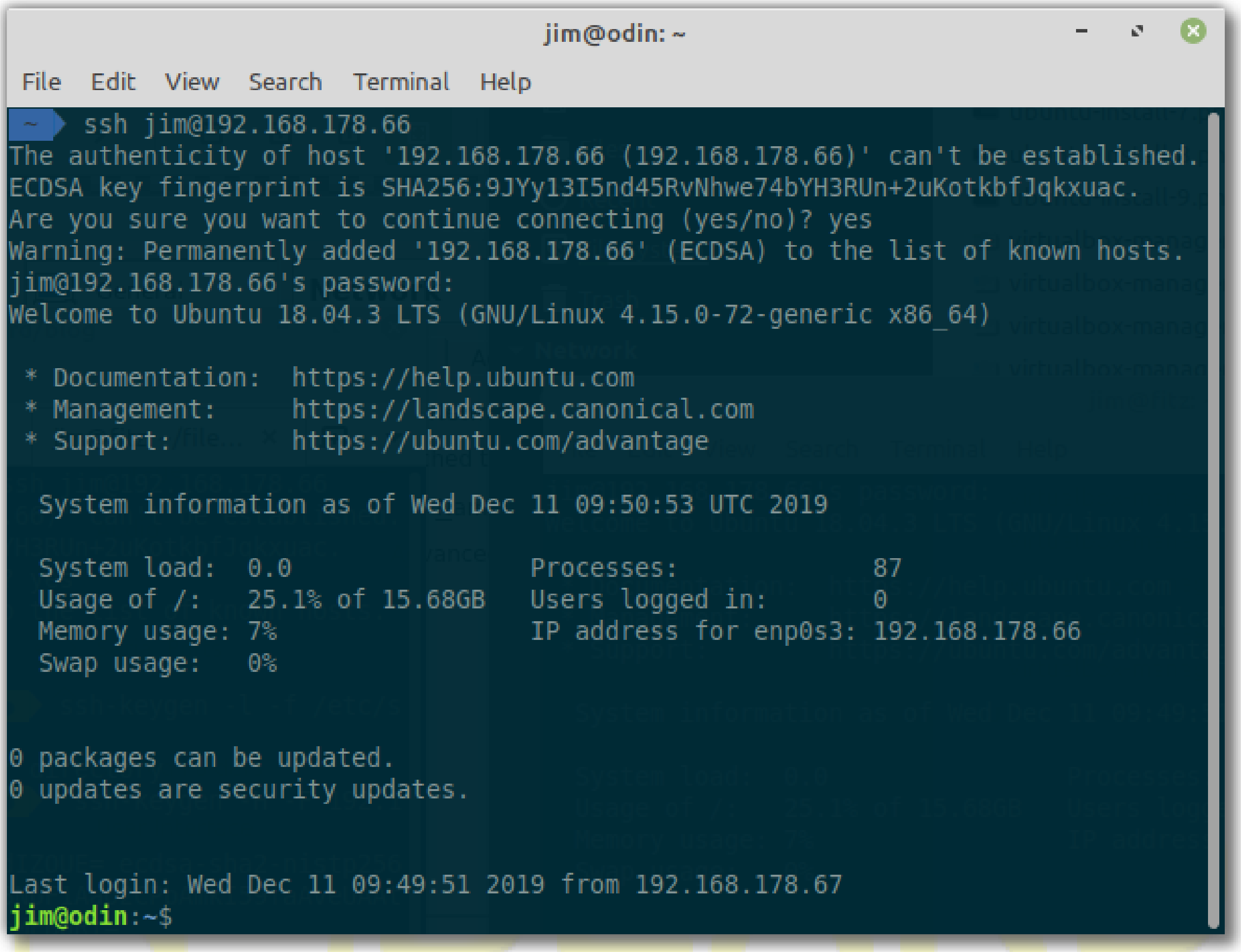
or just hit DuckDuckGo.

Then (ensuring that you replace “jim” and the IP address with your corresponding values) you can connect like so:

```
ssh jim@192.168.178.66
```

This will give you a warning that the host’s authenticity cannot be established and ask you if you want to continue connecting. Answer “yes”.

Next, it will prompt you for your password. Enter it and you will be connected to your Ubuntu server from your host OS.



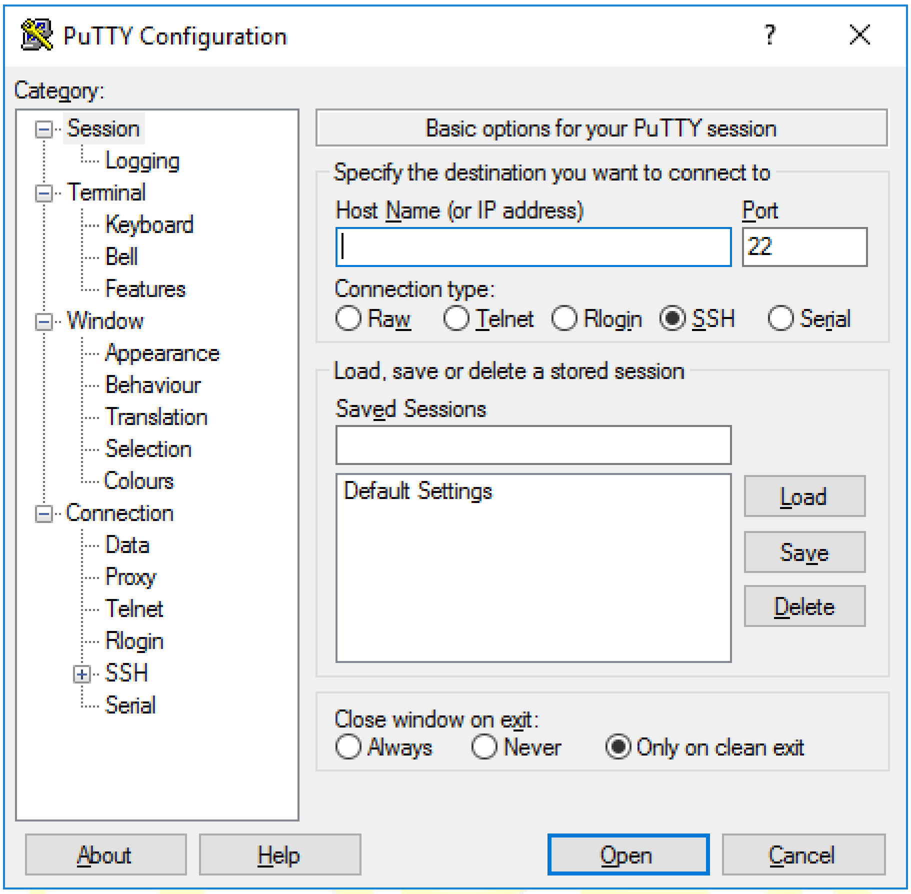
```
 ~  ssh jim@192.168.178.66
The authenticity of host '192.168.178.66 (192.168.178.66)' can't be established.
ECDSA key fingerprint is SHA256:9JYy13I5nd45RvNhwe74bYH3RUn+2uKotkbfJqkxuac.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.178.66' (ECDSA) to the list of known hosts.
jim@192.168.178.66's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-72-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Dec 11 09:50:53 UTC 2019

  System load:  0.0               Processes:             87
  Usage of /:   25.1% of 15.68GB   Users logged in:       0
  Memory usage: 7%                IP address for enp0s3: 192.168.178.66
  Swap usage:   0%


0 packages can be updated.
0 updates are security updates.


Last login: Wed Dec 11 09:49:51 2019 from 192.168.178.67
jim@odin:~$
```

For Windows Users

If you’re running Windows you’ll need to install a SSH client such as PuTTY.

When PuTTY starts, a window titled *PuTTY Configuration* should open. This window has a configuration pane on the left, a *Host Name* field and other options in the middle, and a pane for saving session profiles in the lower right area.

For simple use, all you need to do is to enter the IP address of the host you want to connect to in the *Host Name* field and click *Open*.



Specify the destination you want to connect to

Host Name (or IP address):

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions:

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Generate and Install a SSH Key Pair

SSH keys offer a secure manner of logging into a server without the need of a password.

In a nutshell, this depends upon you generating a public and a private SSH key pair. The private key is kept on your PC (and should be guarded carefully). The public key is copied over to the server you wish to connect to.

Generate the Keys

On *nix systems (Windows users see the next section), you can generate your key pair with the following command:

```
ssh-keygen -o -b 4096 -t rsa
```

The `-o` option instructs ssh-keygen to store the private key in the new OpenSSH format instead of the old (and more compatible PEM format). This is advisable, as the new OpenSSH format has an increased resistance to brute-force password cracking.

The `-b` option is used to set the key length to 4096 bits instead of the default 1024 bits for security reasons.

In the following dialogue you will be required to answer a couple of questions:

- Where to save the newly generated key pair
- Which passphrase to use

Here you can accept the default location and leave the passphrase blank by pressing `Return`.

ssh-keygen will then output a summary of what it has done:

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/jim/.ssh/id_rsa):
Created directory '/home/jim/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/jim/.ssh/id_rsa.
Your public key has been saved in /home/jim/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:sx5uJeVdH/cT/1+GxsSWYzmjf5hUaE33f/e57EbqBfY jim@fitz
The key's randomart image is:
+---[RSA 4096]----+
|                 |
|                o|
|               +o|
|          .  .+==|
|        So . =@oB|
|        .oo o*+BB|
|        oo  ..*EX|
|       o..   +=+=|
|       .o    ..+=+|
+----[SHA256]-----+
```

Copy the Public Key to the Ubuntu Server

To copy the public key to the Ubuntu server use:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub jim@192.168.178.66
```

Where `~/.ssh/id_rsa.pub` is the path to your public key, taken from the output above. And where `jim@192.168.178.66` should be altered to reflect your details.

The command will run and you should be asked for your server password. Enter it, then attempt to log into the server like so:

```
ssh jim@192.168.178.66
```

This time you should be in without a password.

For Windows Users

You should be able to use a tool like PuTTYgen to achieve the same thing. Here is a tutorial on using PuTTYgen to create a new key pair for authentication.

You will have a little more leg work when it comes to copying the key to the server, where you will need to add the public key to a `~/.ssh/authorized_keys` file.

You can do that like so:

```
cd
mkdir .ssh
cd .ssh
```

```
nano authorized_keys
```

This will create the appropriate file, then open the nano editor into which you can copy your newly generated public key.

When you’re done, press `Ctrl` + `X` to save your changes and exit nano.