Joshua Dunne

# Assignment 1

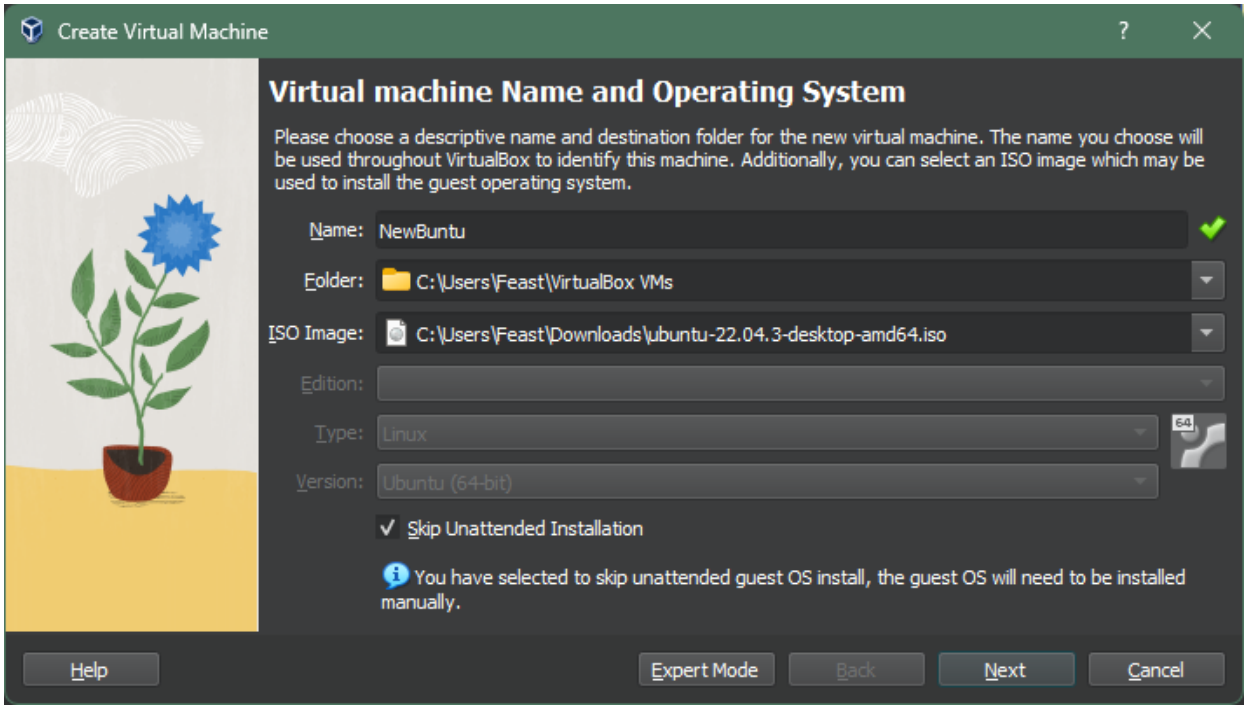
## Abstract

Install a persistent instance of Ubuntu desktop using Oracle Virtual box. Follow the setup guide, avoiding the unattended install option available in Virtual Box. Take screenshots of each step and use uname to print relevant system information.

## Introduction

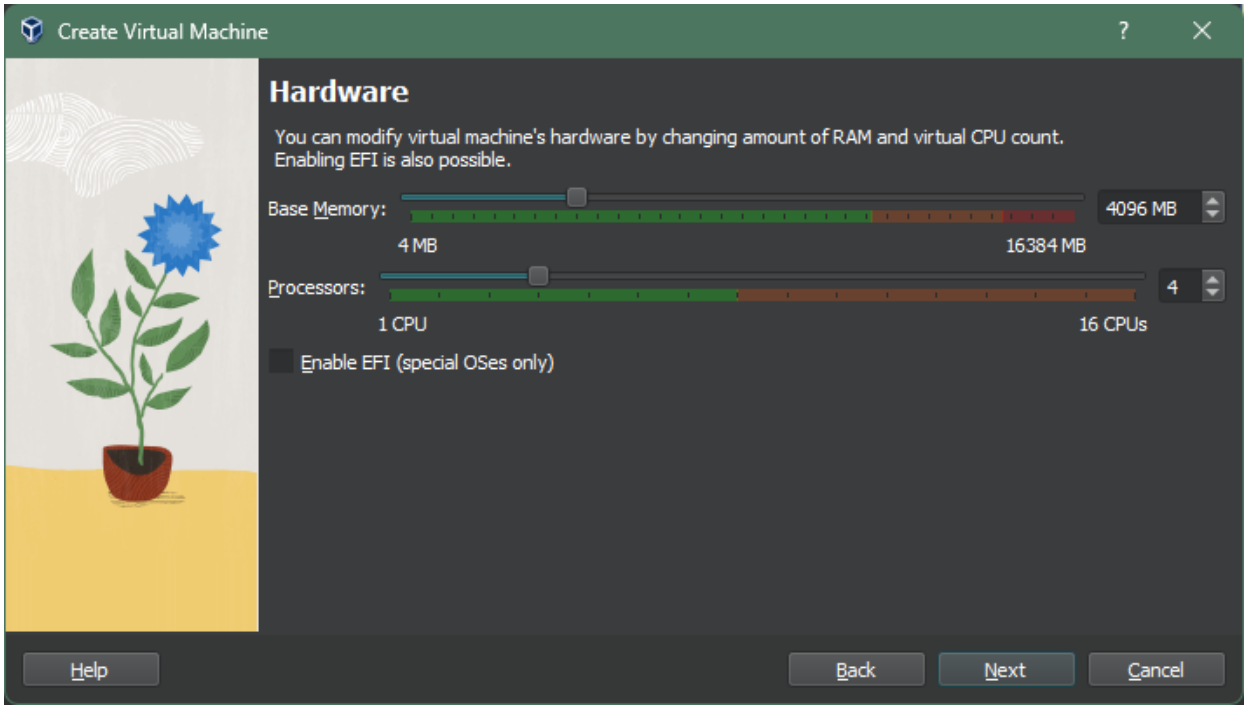
Needed is a very basic knowledge of permanence and the installation of linux. Most details here are taken care of by the wizard presented to the user first by Virtual Box, and later when installing Unix. We're not doing anything special, and these steps are those I'd have taken were I installing an ubuntu distro. The proof in the pudding is the uname results, as these will include all the particulars relevant.

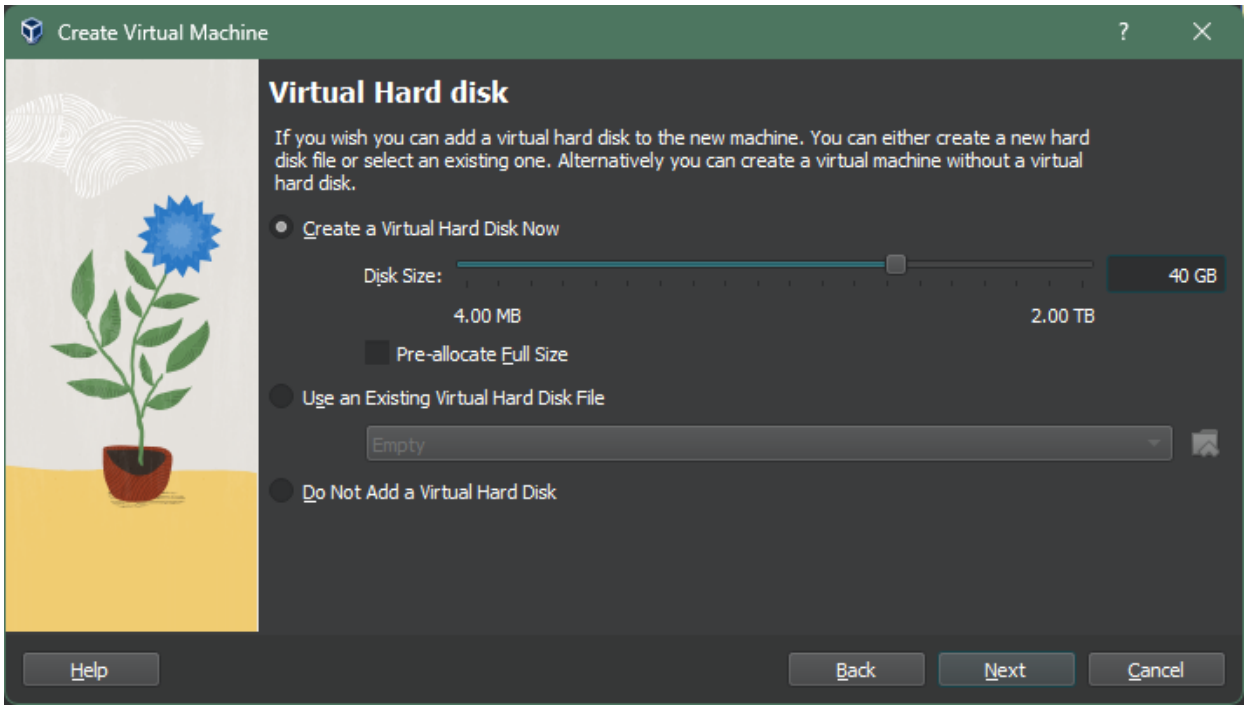
## Summary of Results



## Naming

In this step we're naming our Virtual machine, giving a place within the file system to create the instance, also, to note, we are skipping unattended installation. Were we not to, VirtualBox would manage the install, and we'd not need to go through the later steps in the setup wizard.

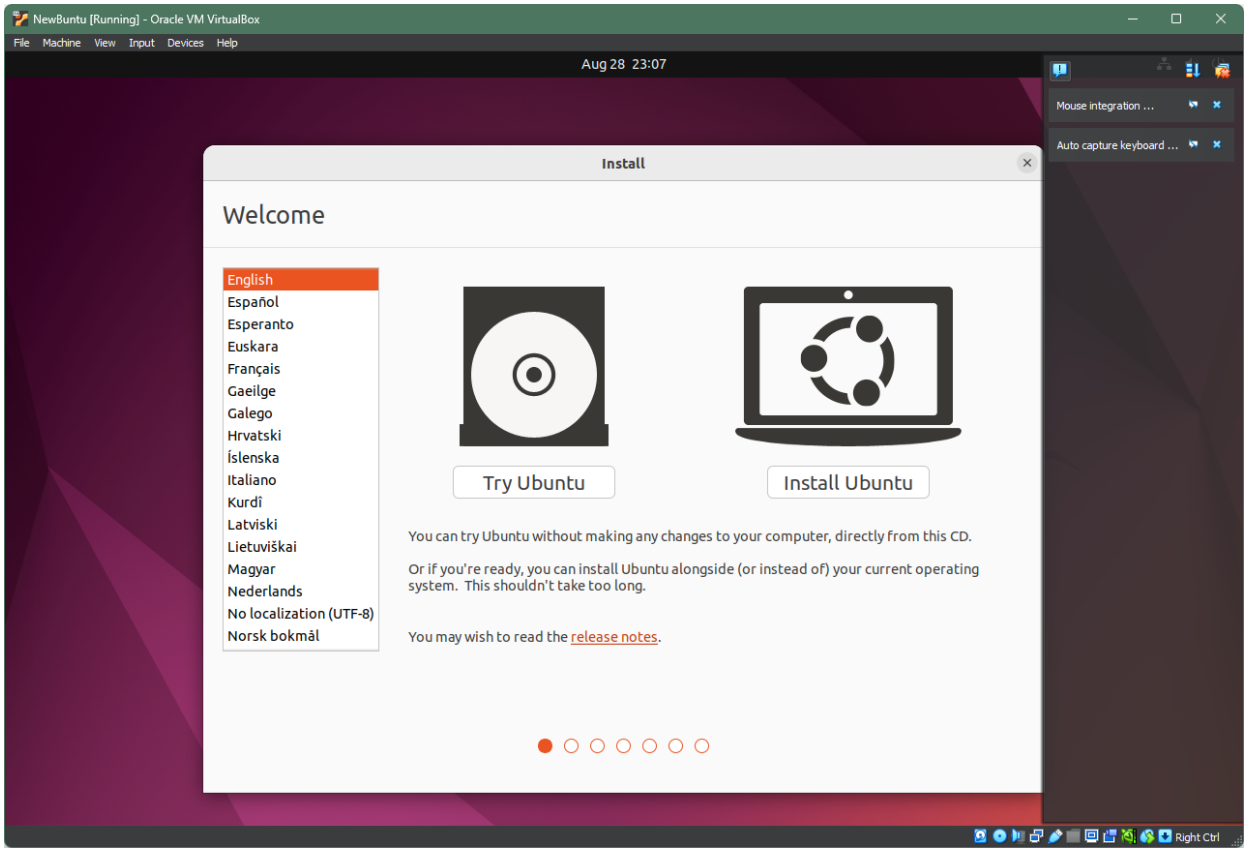


## Allocating diskspace and other resources

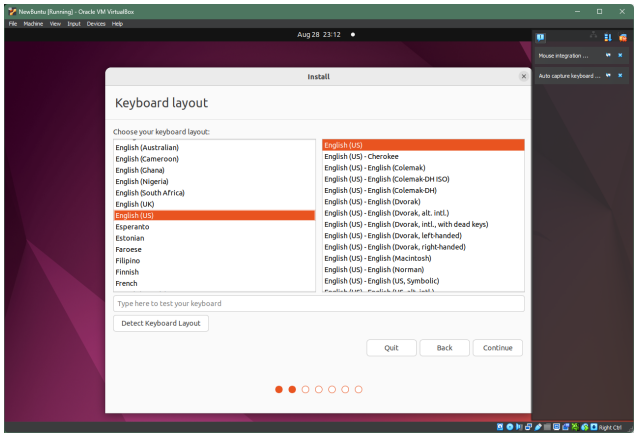
Here, first, we're giving our VirtualBox instance computational resources. As the computer the instance is on has 16GB allocated, 4GB of memory is reasonable, and more than enough to run the desktop manager, and various other tasks that a linux distro will undertake in the background. Too, as the CPU has multiple cores, we can allow our instance to run on several. Lastly, though the initial Ubuntu instance is small, this quickly snowballs. 40GB is a reasonable amount of space to give our Ubuntu instance to breathe, and will allow for the installation of many tens to hundred of programs and features to be later included.

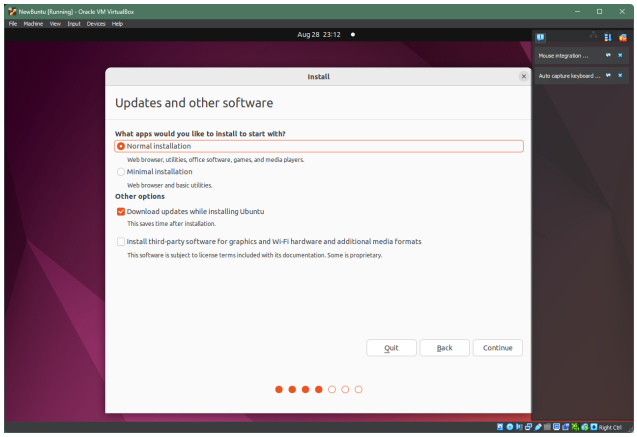
## HelloBuntu

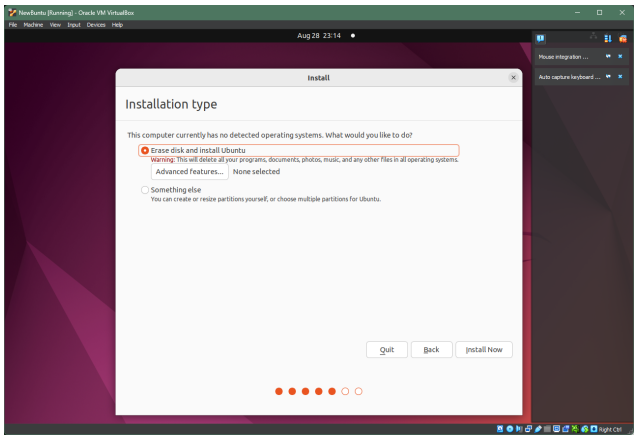


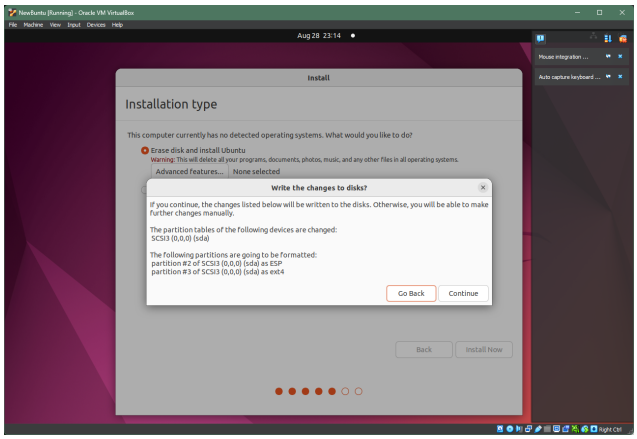

Here we see a live version of linux running. At the moment, as necessary, Virtual Box is drawing from resources located on the ISO we included to give Ubuntu the necessary features to install itself, which we will now do.

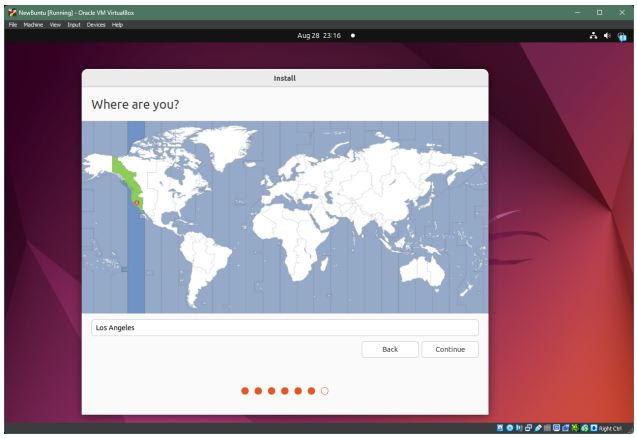
## Basic setup

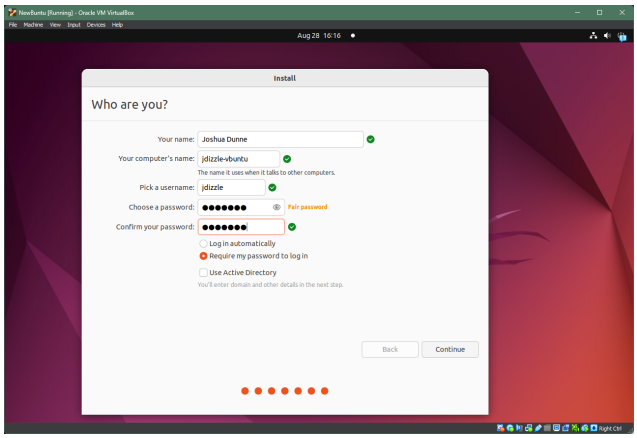


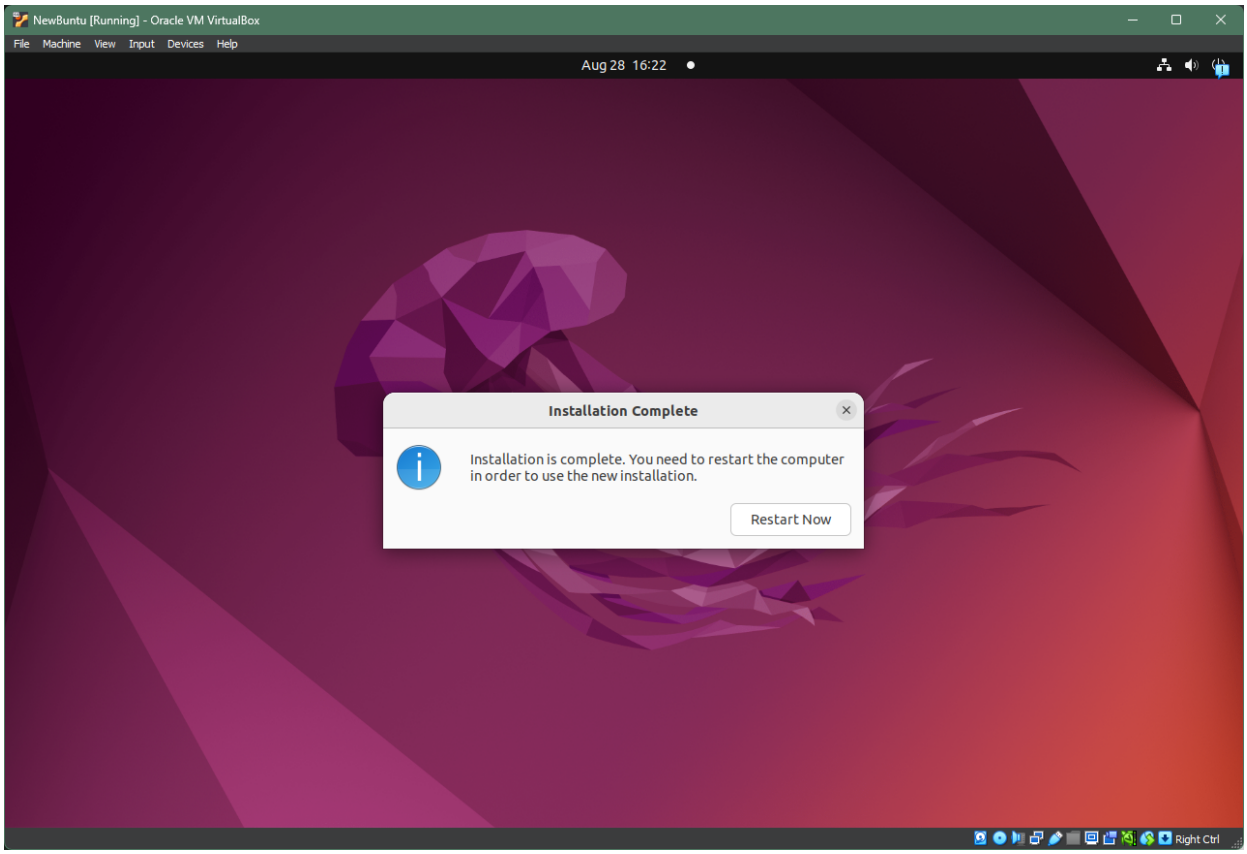




Next up are the steps which the wizard will take us through. We need to be sure that the keyboard is set up correctly, as with linux, if anything breaks, we'll be staring at a cli. Too, as updates are important for security, many more recent versions of linux distros will update themselves during installation to save time afterwards. After that, we're being asked to be double sure that we want to do this. As we specified that we do indeed want to make permanent changes, Ubuntu is about to rewrite the entire filesystem that was there and replace it with its own. Any files that were there are at chance of being overwritten. The time zone, like the

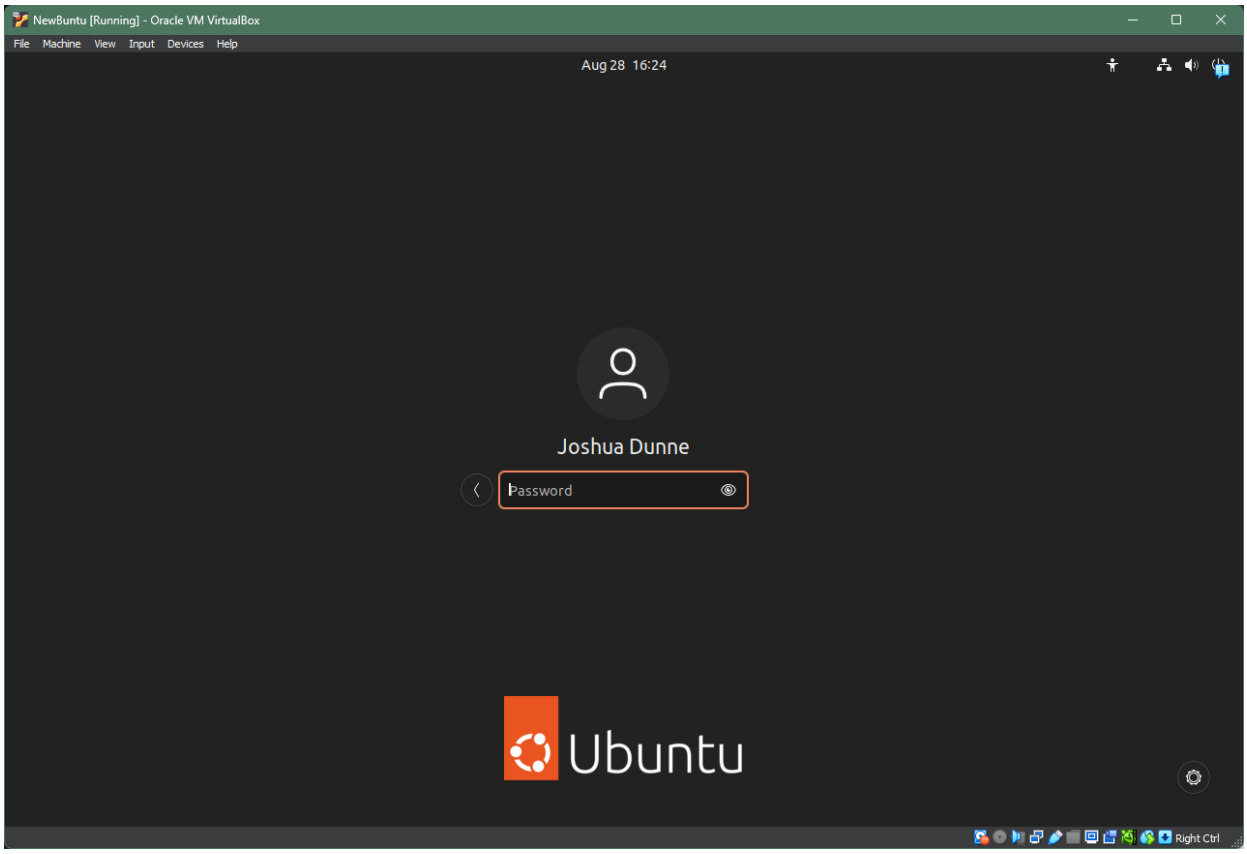
keyboard shortcut, is relevant for innumerable system related tasks and processes, thus, we're being asked to specify it. Lastly, as logging straight in as root is often undesirable in a desktop environment, we give the details to ubuntu it would need for "useradd". Ubuntu will also set up the home directory and assign that user to groups such as sudo.

ANDDDDD……



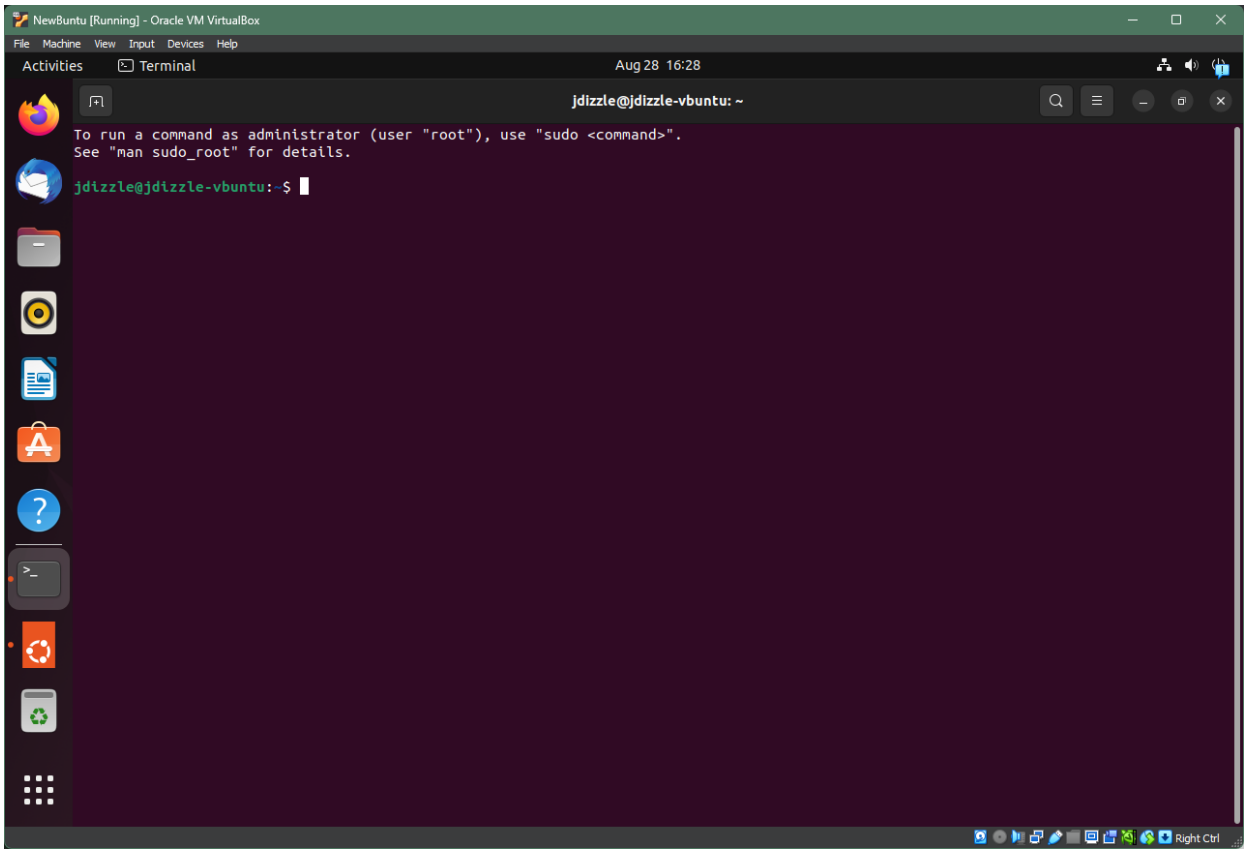

Tada. Now we clench. It SAYS it's complete, and as this isn't my first time installing a linux distro, this is the time where I pray to god that everything will work out of the box.

## Login screen



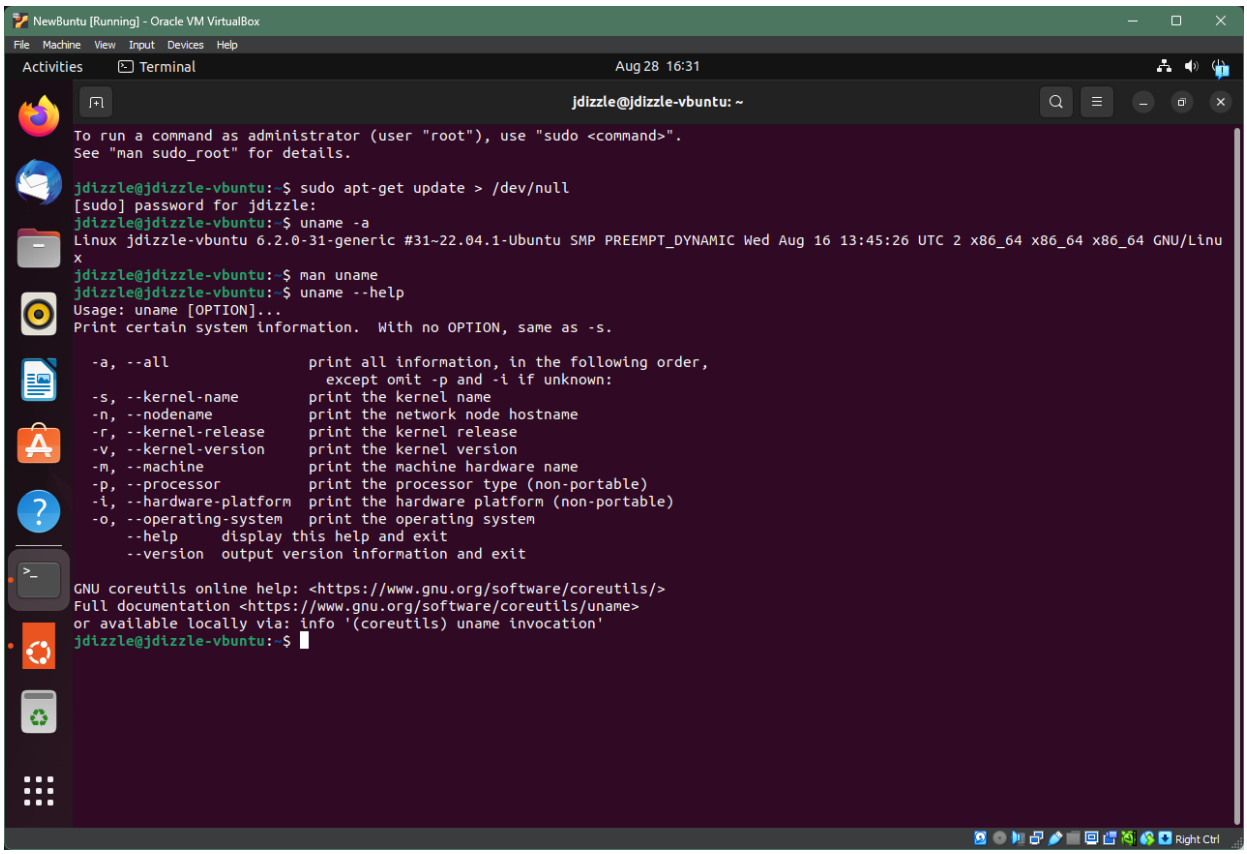

As we're given a desktop manager by default, we get a nice clean screen to login through. This is familiar to all desktop users, here our name is given to us as we provided it, and as we created the user, we know that our actual username is jdizzle, as it always is with me.

## Open a terminal




Like any good linux user, we ignore all the pretty icons and hit Cntrl+Alt+T, we're presented with a configured prompt, and a terminal that has so much of the legwork already done for us.

## Update repositories, display system information, check out unames manpage




Here we're doing a few things, all of which are advisable, amongst some others. Firstly we tell ubuntu to go ahead and update our repositories, with this, ubuntu will contact a long list of preconfigured servers and download the most current set of software that is available through our package manager apt. We print out uname -a, which as the help demonstrates, will gives us all the information that uname can spit back. Let's break this down

## Uname -a output

-a simply tells uname that we want all the data, but we can break things down easily by using the parameters to uname. I also went ahead and installed guest additions to make copying and pasting easier.

**Our original output of uname -a**
Linux jdizzle-vbuntu 6.2.0-31-generic #31~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Wed Aug 16 13:45:26 UTC 2 x86_64 x86_64 x86_64 GNU/Linux

| | | | |
|---|---|---|---|
| Uname -s | Kernel name | Linux | We're running a linux kernel, thus, the output of our kernel name is simply linux |
| Uname -n | Network node host name | jdizzle-vbuntu | Our hostname is how we're identified on the network |
| Uname -r | Kernel release | 6.2.0-31-generic | The version (build) of the linux kernel we're currently running |
| Uname -v | Kernel version | #31~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Wed Aug 16 13:45:26 UTC 2 | The kernel release, which adds specificity to the version. I'd assume this was the version of Ubuntu it was built for, and the time in which it was built, also, we specify SMP for symmetric multiprocessing. |
| Uname -m | Machine hardware type | x86_64 | We're primarily concerned with the architecture and instruction set when we built a kernel, fittingly, we're building for x86_64, 64 bit, x86 type |
| Uname -p | Processor type | x86_64 | Likewise, we're only really concerned with the processor architecture, thus this agrees with the hardware type |
| Uname -i | Hardware platform | x86_64 | I'm not even sure why this would be different than the machine or processor type, but, again, we're specifying specifics of the computer that might be relevant |
| Uname -o | Operating system | GNU/Linux | We're referring here more specifically to the type of operating system. GNU/Linux means that we are running Linux, but with all the added features that GNU added to it |

## Conclusion

In conclusion, there's several important differences between the installation of Ubuntu that we now have, and the previous instance where we were running it from a removable media. As the removable media was mounted as read only, and we did not allocate a permanent storage originally, in the first instance, any changes we might have made to the file structure would not have been persistent through rebooting. All changes we were making were to volatile memory, and would have been lost were we to reboot, or were the system to have lost power.
Unlike the original instance, once we allocate non volatile memory to ubuntu and go through the actual installation process, we have an instance in which changes we make to the file system are made permanent. This is the way most people are accustomed to running their operating system. Here, we can install software, configure it, and make changes knowing that, when we reboot, those changes will be persistent.
A good application for a non-persistent operating system is something like a sandbox. Here, none of our changes are persistent. If anything goes wrong, we can reboot and know that we'll simply be back where we started. Something like Kali linux is a good example of a good use. Because the changes we make will not persist, none of our user data is stored between reboots. We come back to a clean browser every time that hasn't retained any data, and are less likely to misconfigure the operating system, or make a mistake in leaking personal information.
Unlike Kali. Ubuntu desktop is much more meant to be installed than it is to be used from removable media. Here our settings are retained. Our configuration remains permanent. While Ubuntu Desktop can be run from removable media, this is often sold more so that a potential user might try out the distro before installing. Ubuntu Desktop is meant as a familiar alternative to the usual operating system experience a regular Windows/Mac use might expect. We can create and edit files. Download movies. Install games. When we reboot and come back to it later, these changes will persist as the installation resides in non volatile memory.