Abstract

This document Shows how to setup a Linux Virtual Machine under windows, then install the development tools for Zephyr, this is the basis for all further Code development under Zephyr.

Setting up for Zephyr Development on Ubuntu

A step-by-step how-to guide

# 

IRCON.4

Lawrence.King’s

October 3, 2019

# Contents

[Contents 1](#_Toc21022925)

[1 Team members 4](#_Toc21022926)

[2 What is the Problem 4](#_Toc21022930)

[3 Coding Guidelines 5](#_Toc21022931)

[4 Generating initial secure keys in the device. 5](#_Toc21022932)

[5 Transcoding 5](#_Toc21022933)

[6 Whitebox security 5](#_Toc21022934)

[7 CSP – protecting the running system 5](#_Toc21022935)

[8 Demo 5](#_Toc21022936)

[9 Setup an Ubuntu machine 7](#_Toc21022937)

[10 Installing Zephyr and fetching the rc-demo git repository 24](#_Toc21022938)

[11 Setting up Device Firmware Update (DFU) 29](#_Toc21022939)

[12 Pushing New Images to the Asset over Bluetooth 30](#_Toc21022940)

[12.1 Install the GO programming language and MCUmgr 30](#_Toc21022941)

[12.2 Build a second Image, then upload to the MDK 32](#_Toc21022942)

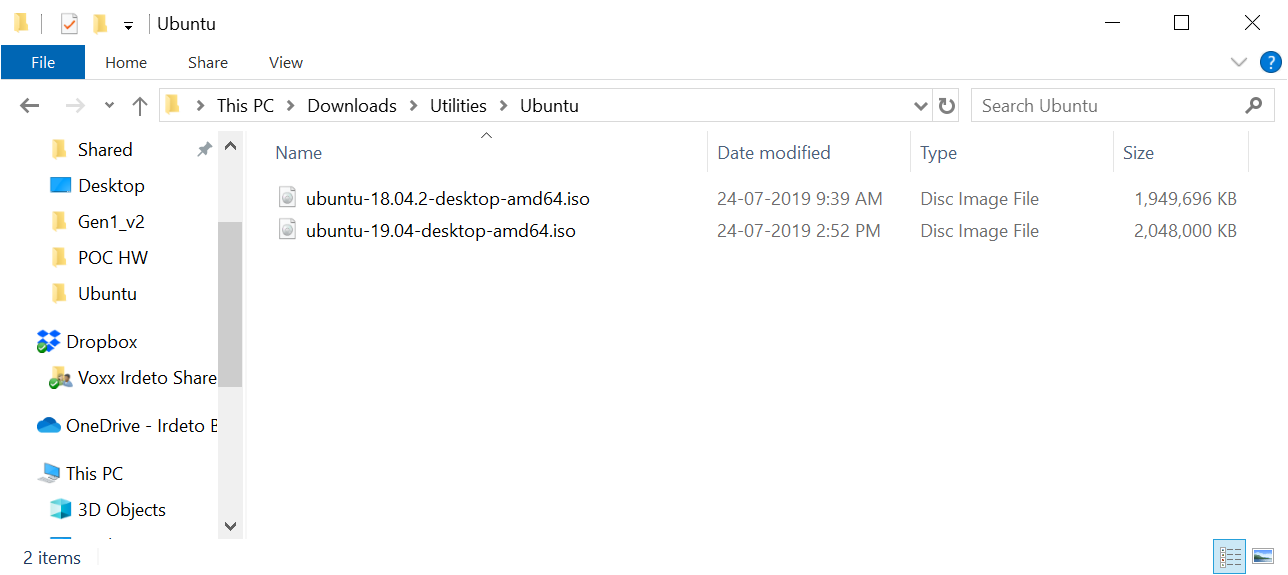
[13 References 32](#_Toc21022943)

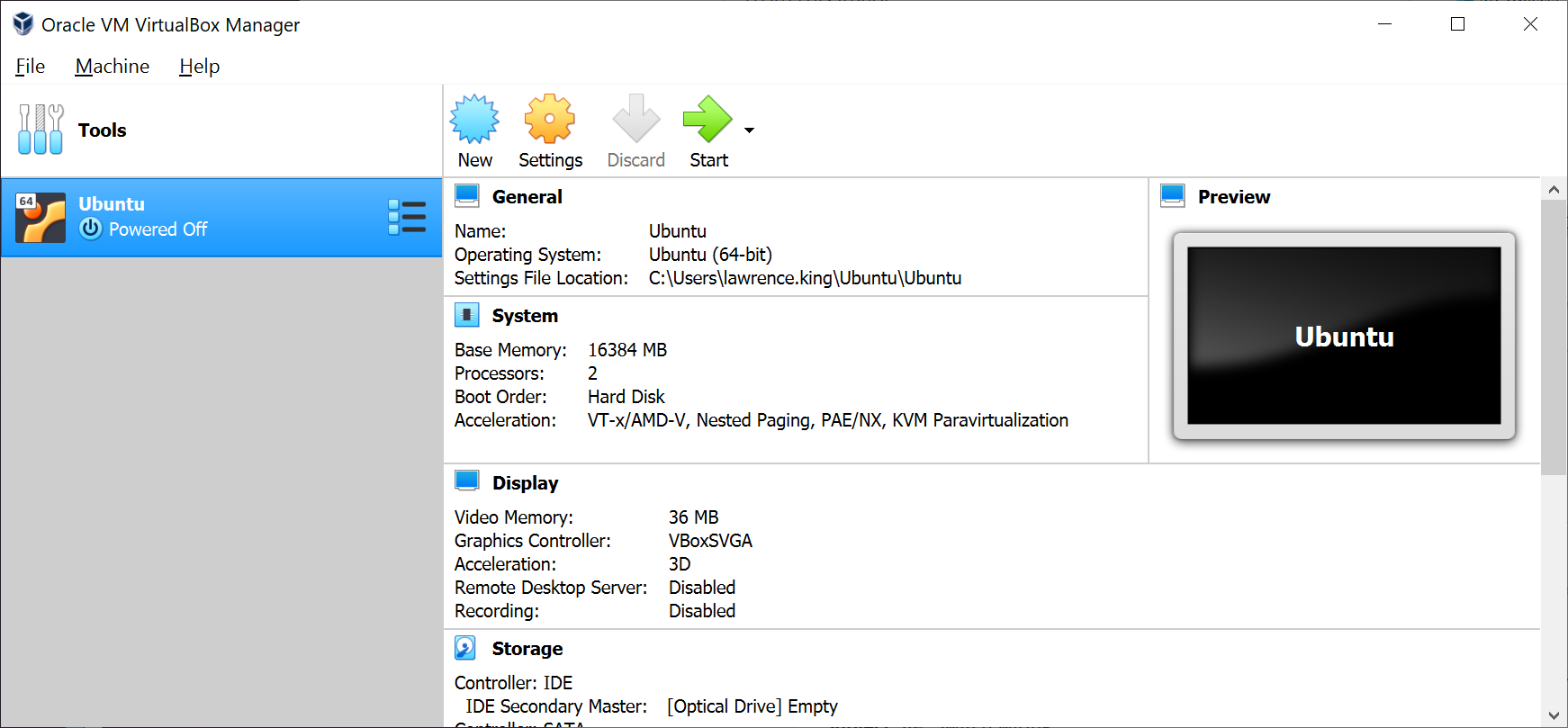
# Setup an Ubuntu machine

This step is optional, and not necessary if you already have an Ubuntu machine, or virtual machine. Skip to Section 2.

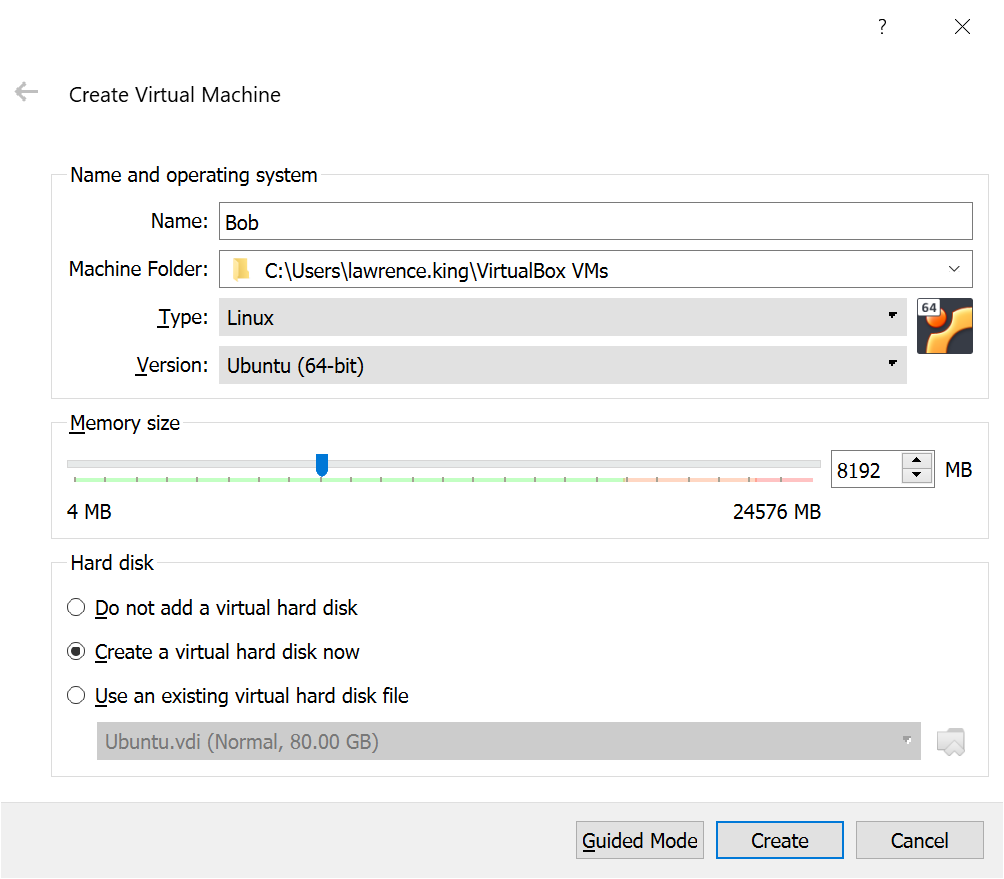
Notes:

* Ubuntu 16.04 is too old, the Zephyr code failed to build under Ubuntu 16.04. Ubuntu 18.04 and 19.04 have been tested and work.
* do not use Windows WSL, or WSL2 since these virtual machines cannot (currently) access the USB ports and needed in later steps.

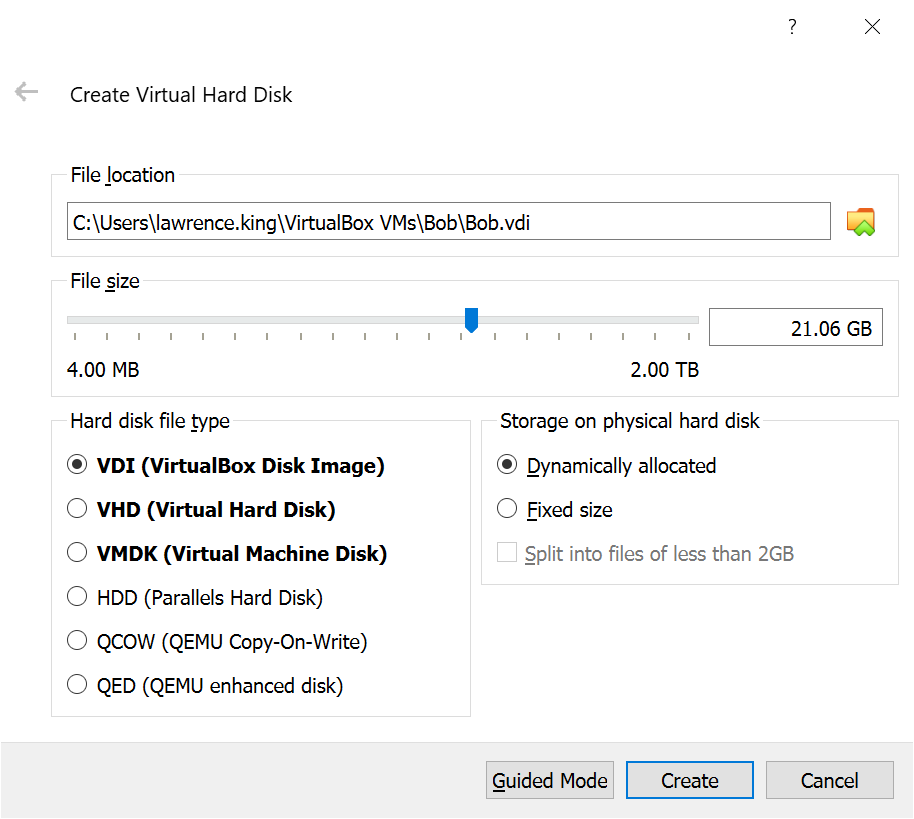
1. Download Ubuntu AMD64 ISO desktop image from <http://releases.ubuntu.com/18.04/> (or preferably the newer 19.04 version) and save it somewhere easy to find on your computer. 
2. Install Oracle VM VirtualBox Base Package and Extension Pack from <https://www.oracle.com/technetwork/server-storage/virtualbox/downloads/index.html> After installation start the Manager



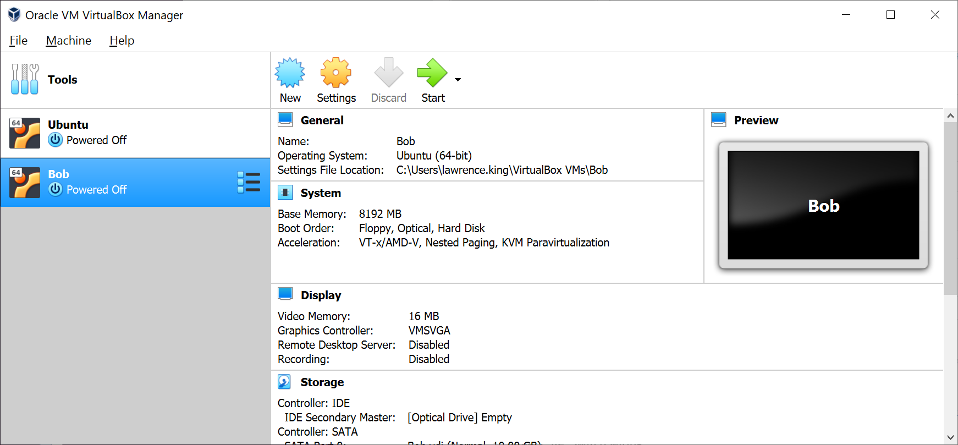
1. Click New to install a new virtual machine. In this case we will call it “Bob”, but any name is acceptable. Select type Linux, Version Ubuntu (64-bit) and set the memory size to 8GB and click **Create**.



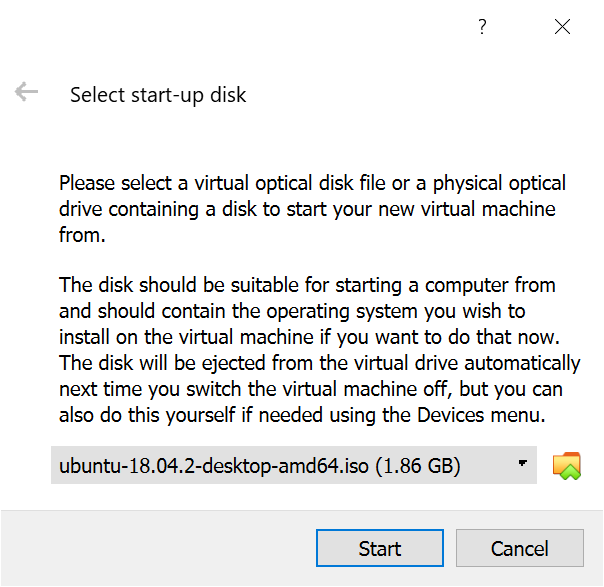
1. **Create** a virtual hard disk, the default location, the size needs to be about 20GB!.



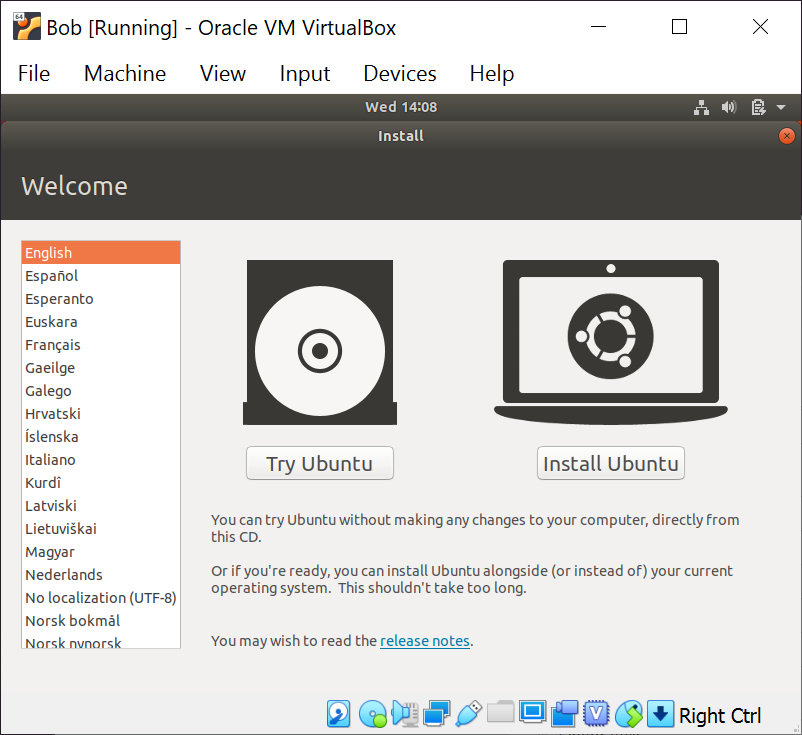
1. Now that you have created Bob, select Bob and click the **Start** arrow in the manager



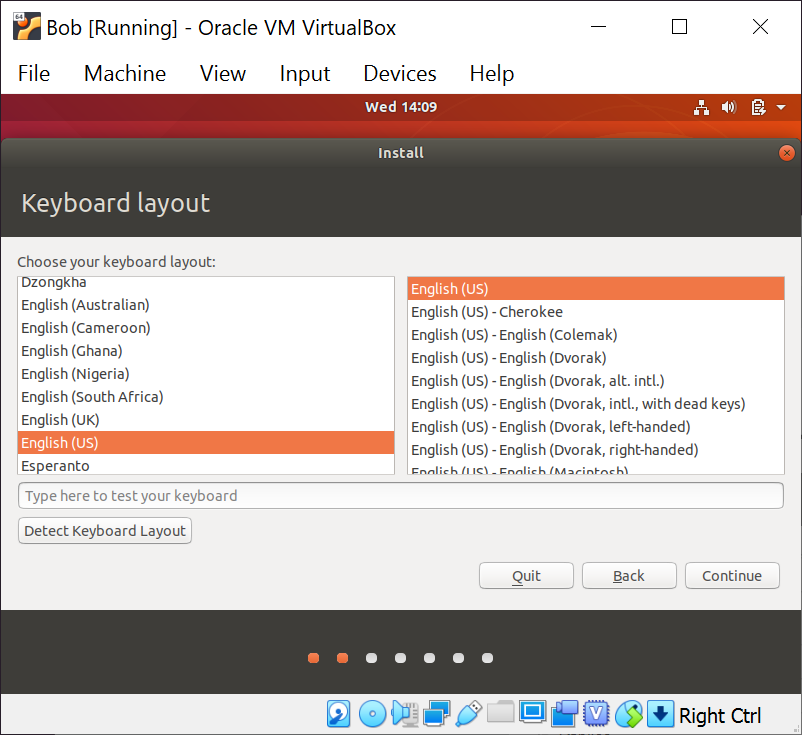
1. Bob will start and the very first question will be asking for the location of the .iso image you downloaded in step 1). Locate the .iso image and click **Start**.



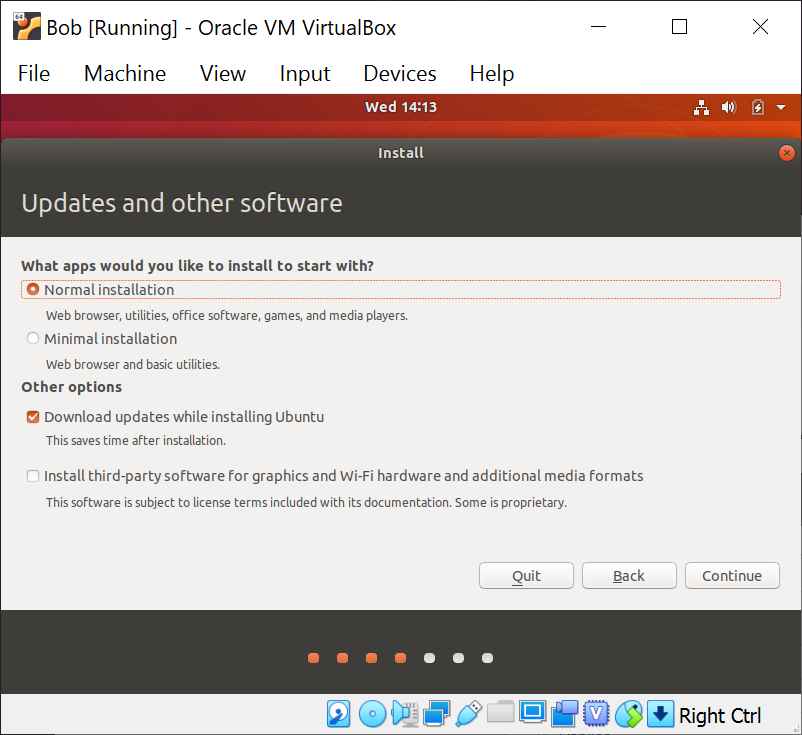
1. Ubuntu will start in a minute or so and ask you if you want to install Ubuntu, click **Install Ubuntu**



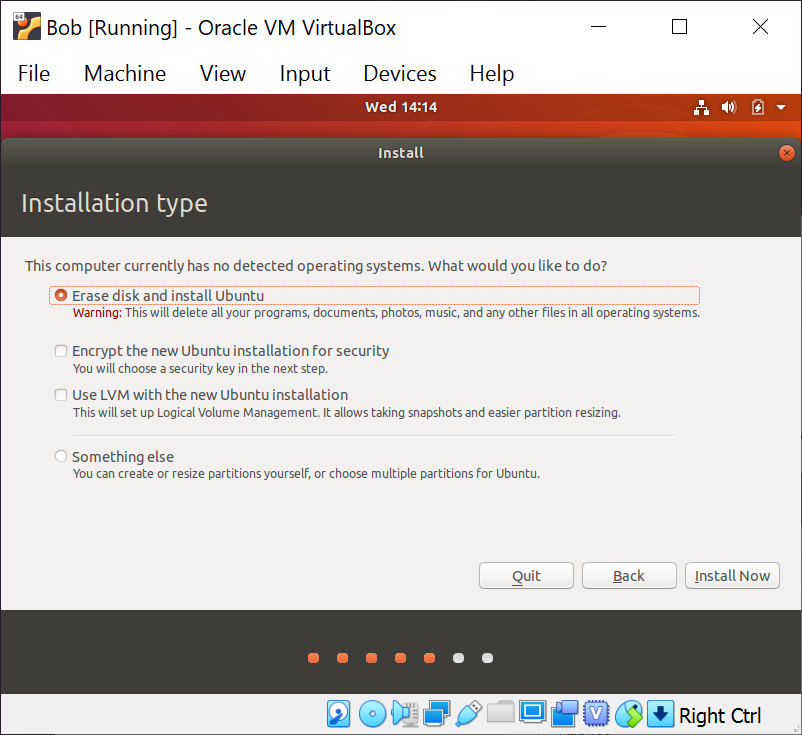
1. Select your language and keyboard layout and **Continue**



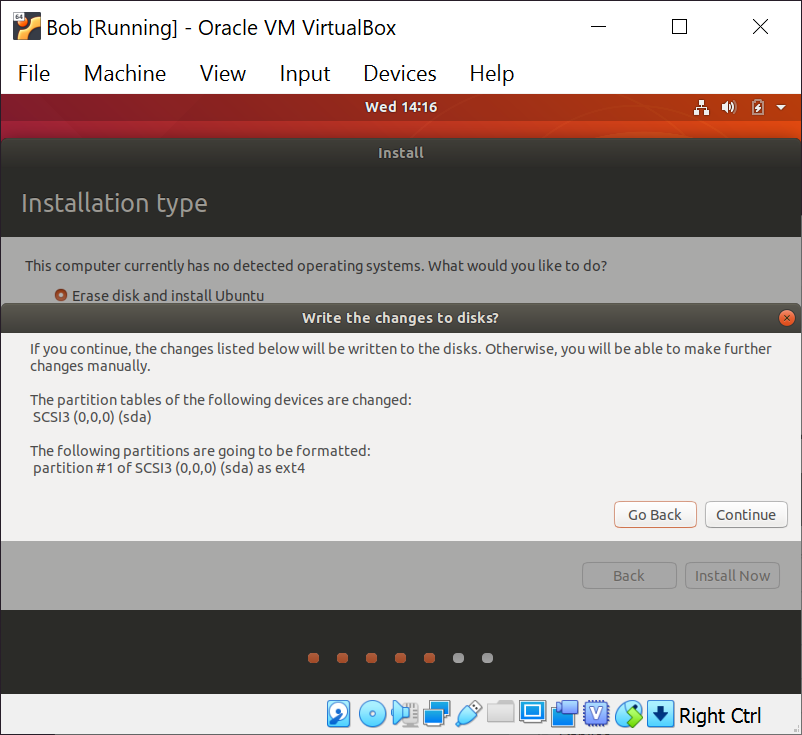
1. Select Normal Installation and **Continue**.



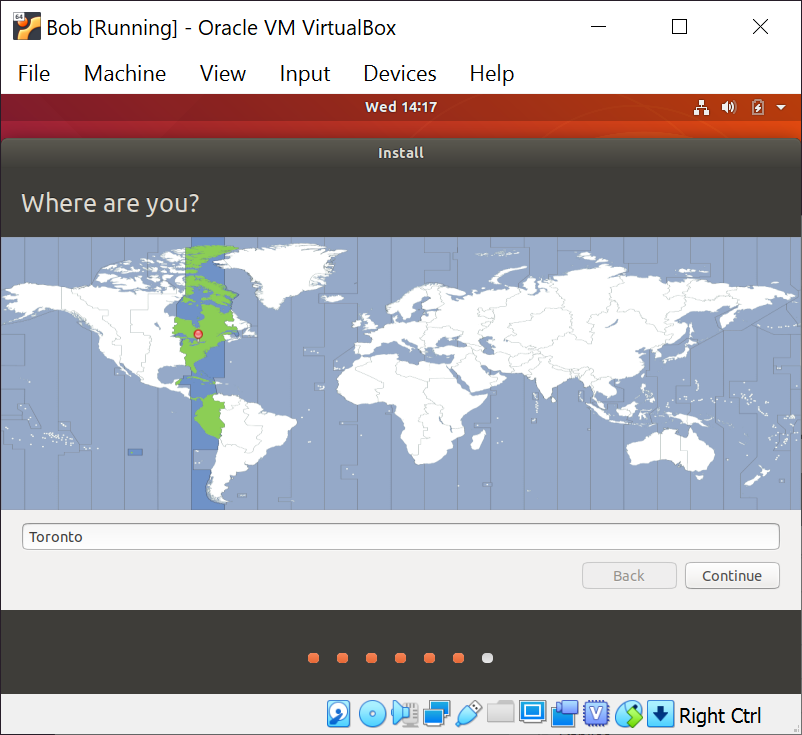
1. Select Erase Disk and **Install Now**.

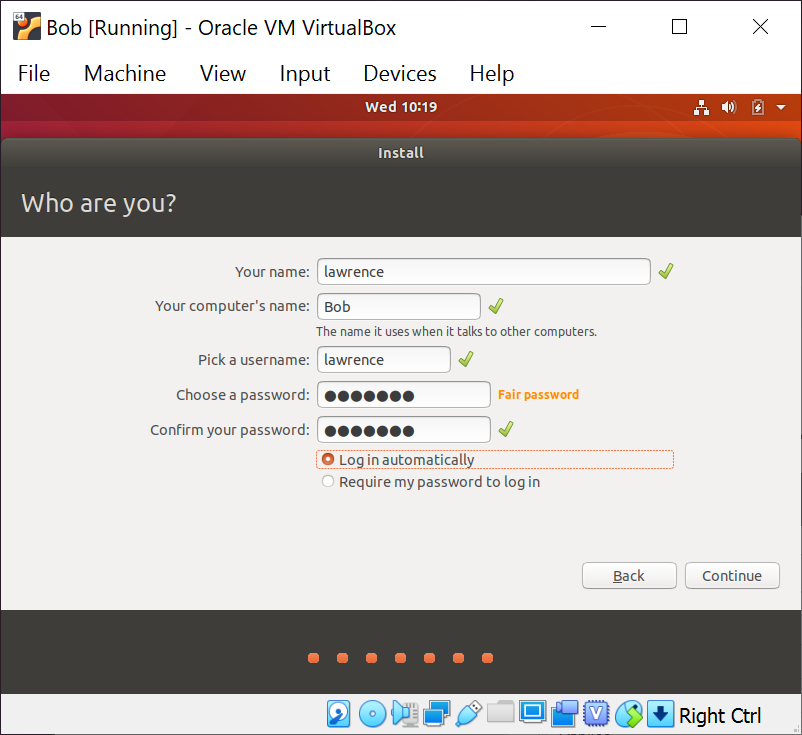


1. Yes, you are sure, **Continue**.

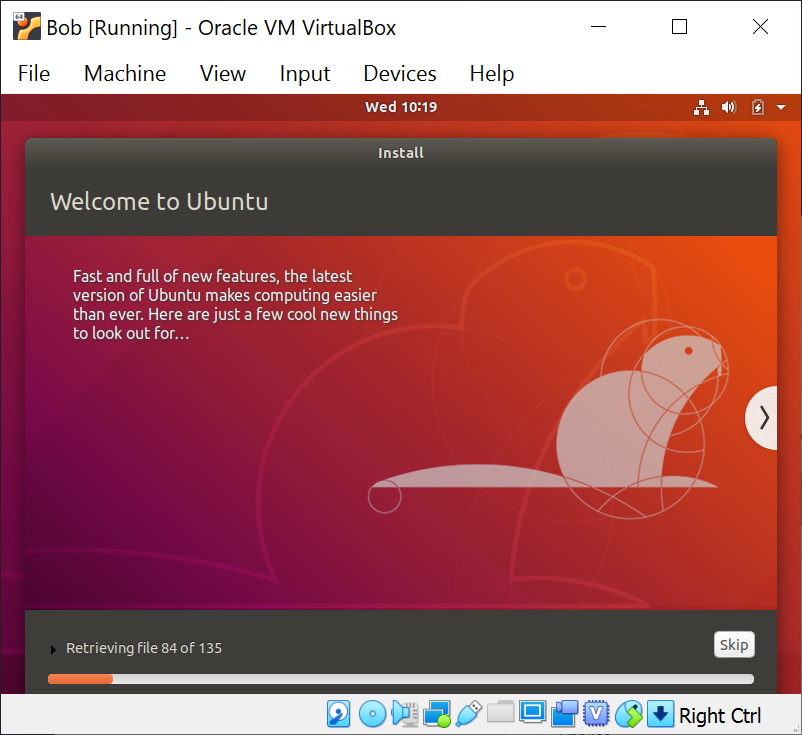


1. After a few minutes Ubuntu will have installed itself onto your virtual machine and start asking you a few questions, **Continue**.

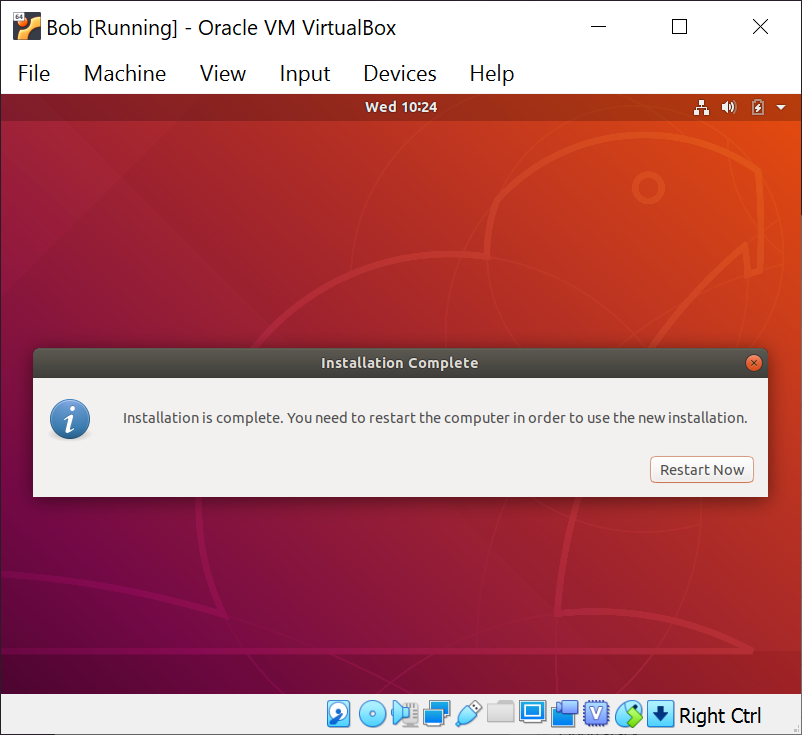


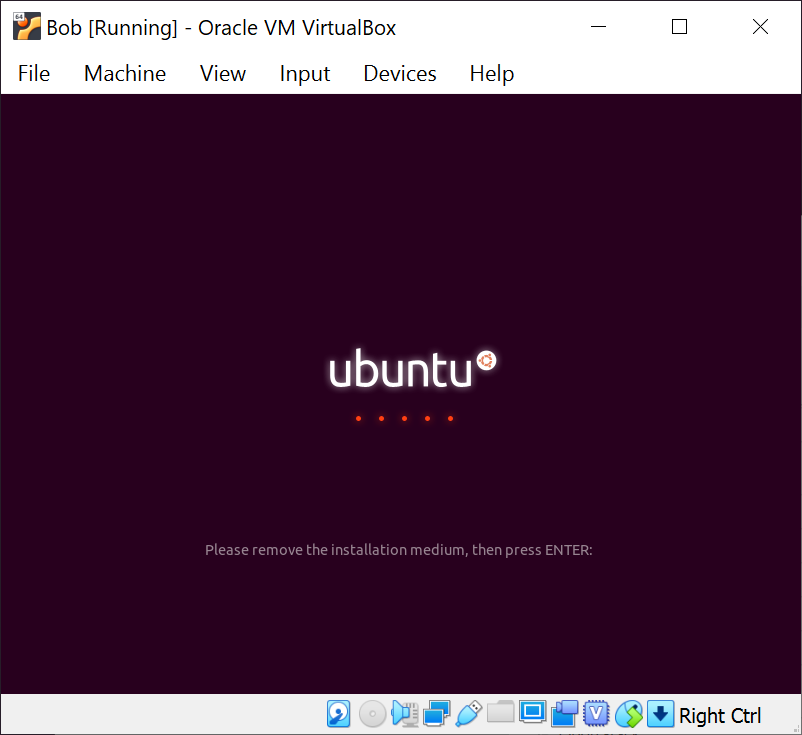


1. **Do Not** skip the updates, just let them run. This takes several minutes.

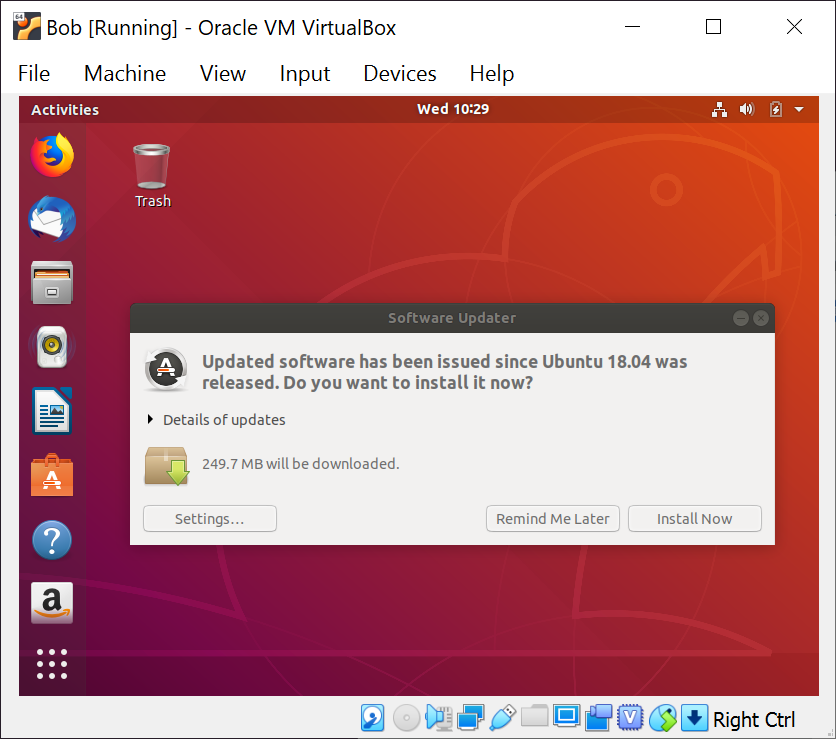


1. When Ubuntu is finished installing and the first round of updates you want to **Restart Now** and **Enter**.





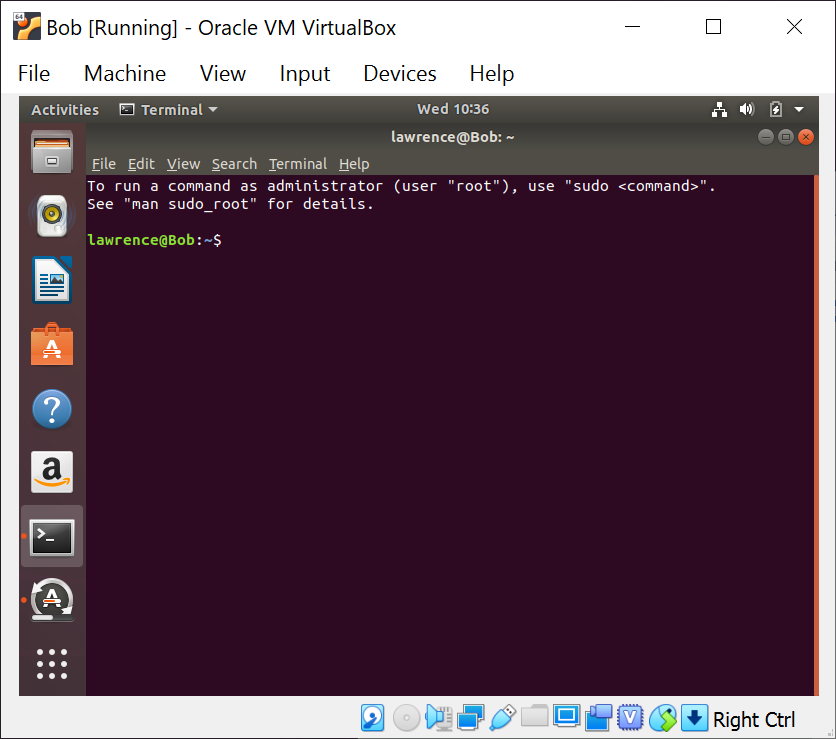
1. Ubuntu is now ready for use, **Install Now** any updates. You will need to provide the password you entered in Step 12) to enable installing the updates. You will need for the software updater to complete updates before you can run the commands in Section 2.



To finish installing all of the updates, restart the Virtual Machine.

# Installing Zephyr and fetching the rc-demo git repository

Start a terminal program on your Ubuntu Virtual machine.



Enter the following commands

sudo apt-get update

sudo apt-get -y upgrade

VirtualBox and Ubuntu have many features, here are hints on some you might want to turn on at this time:

If you are running 18.04 you need to install a compiler and some support for the guest additions:

sudo apt-get install -y build-essential module-assistant

<https://www.centennialsoftwaresolutions.com/post/enable-copy-paste-between-windows-7-and-ubuntu-16-04-3-on-oracle-vm-virtualbox> - common clipboard between Linux and Win10

<https://helpdeskgeek.com/virtualization/virtualbox-share-folder-host-guest/> - shared folder between Linux and Win10

Add the following to the end of your ~/.bashrc then restart your terminal window (<crtl>-D) or source ~/.bashrc.

#export BOARD=nrf52840\_mdk

export GIT\_SSL\_NO\_VERIFY=true

#export ZEPHYR\_TOOLCHAIN\_VARIANT=zephyr

export ZEPHYR\_SDK\_INSTALL\_DIR=/opt/zephyr-sdk

export PATH=~/.go/bin:~/.local/bin:$PATH

source ~/zephyrproject/zephyr/zephyr-env.sh 2>/dev/null

export ZEPHYR\_TOOLCHAIN\_VARIANT=gnuarmemb

export GNUARMEMB\_TOOLCHAIN\_PATH=/usr

export GOPATH=$HOME/.go

Now install basic Linux utilities

ssh-keygen #generate a key for yourself,

# and then add the key to gitlab <https://gitlab.ott.dev.intra/profile/keys>

sudo apt-get install -y --no-install-recommends git ninja-build gperf \

ccache dfu-util device-tree-compiler wget cmake python3-testresources\

python3-pip python3-setuptools python3-tk python3-wheel xz-utils file \

make gcc gcc-multilib libncurses5

if you have Ubuntu 18.04, upgrade device tree compiler and cmake

wget http://mirrors.kernel.org/ubuntu/pool/main/d/device-tree-compiler/device-tree-compiler\_1.4.7-1\_amd64.deb

sudo apt install ./device-tree-compiler\_1.4.7-1\_amd64.deb

sudo -H pip3 install --upgrade pip

pip3 install --user cmake

Restart your terminal window (<crtl>-D) again.

Now install the JLink and pyOCD tools so you can load code to your board.

From a browser (likely FireFox) open <https://www.nordicsemi.com/Software-and-Tools/Development-Tools/nRF-Command-Line-Tools/Download#infotabs> and select “Linux 64” and download nRFCommandLineTools1021Linuxamd64tar.gz or the latest version.

cd ~/Downloads

tar -vxf nRFCommandLineTools1021Linuxamd64tar.gz

sudo dpkg -i nRF-Command-Line-Tools\_10\_2\_1\_Linux-amd64.deb

The nRFtools depend on the Segger tools, but an old version of the Segger tools are embedded in the nRFtools. Get the latest, in the web browser go to <https://www.segger.com/downloads/jlink/#J-LinkSoftwareAndDocumentationPack> and download the SEGGER “J-Link Software and Documentation pack for Linux, DEB installer, 64-bit” then install the latest version of the Segger JLink tools

sudo dpkg -i Jlink\_Linux\_V648\_x86\_64.deb

And the pyOCD tools:

git clone https://github.com/mbedmicro/pyOCD.git

sudo cp pyOCD/udev/\*.rules /etc/udev/rules.d

sudo udevadm control --reload

sudo udevadm trigger

Now install the ARM cross compiler

sudo apt install -y gcc-arm-none-eabi

Zephyr build system uses it’s own private versions of some utilities (particularly OpenOCD). These are burried in the Zephyr SDK.

wget <https://github.com/zephyrproject-rtos/sdk-ng/releases/download/v0.10.3/zephyr-sdk-0.10.3-setup.run>

cd <sdk download directory>

chmod +x zephyr-sdk-0.10.3-setup.run

./zephyr-sdk-0.10.3-setup.run

Next install West, the Zephyr meta tool

cmake --version

# cmake must be version 1.13.3 or better!

pip3 install --user west

west init zephyrproject

cd zephyrproject; west update; cd ..

source zephyrproject/zephyr/zephyr-env.sh

pip3 install --user pyyaml~=5.1

pip3 install --user -r zephyrproject/zephyr/scripts/requirements.txt

Now that the SDK is installed we can setup the OpenOCD access rules:

sudo cp ${HOME}/opt/zephyr-sdk/sysroots/x86\_64-pokysdk-linux/usr/share/openocd/contrib \

/etc/udev/rules.d/

sudo udevadm control --reload-rules

sudo usermod -aG plugdev $USER

At this point you will have cloned zephyr into the zephyrproject directory, now clone the rc-demo repository into a directory called rc-demo and you will be ready to build code and load it onto the board.

cd zephtrproject/zephyr

west build -b frdm\_kl25z samples/basic/blinky

west flash

The board type can be one of 150 different possible boards. ‘west boards’ will list the available boards