Setting Up Linux for CSE775 - DO

Jim Fawcett

CSE775 – Distributed Objects

Spring 2019

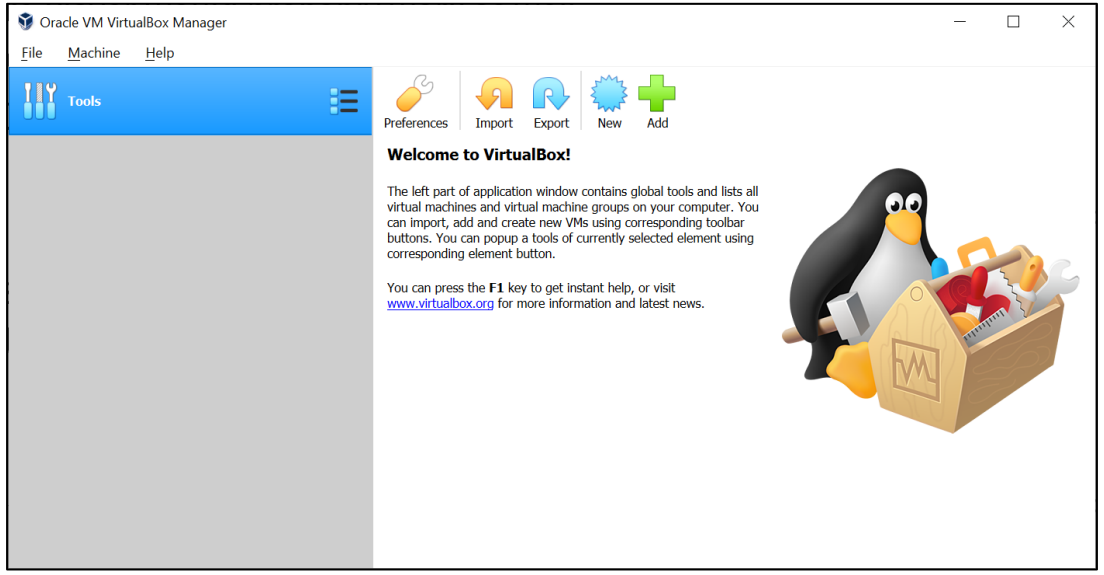
Tasks:

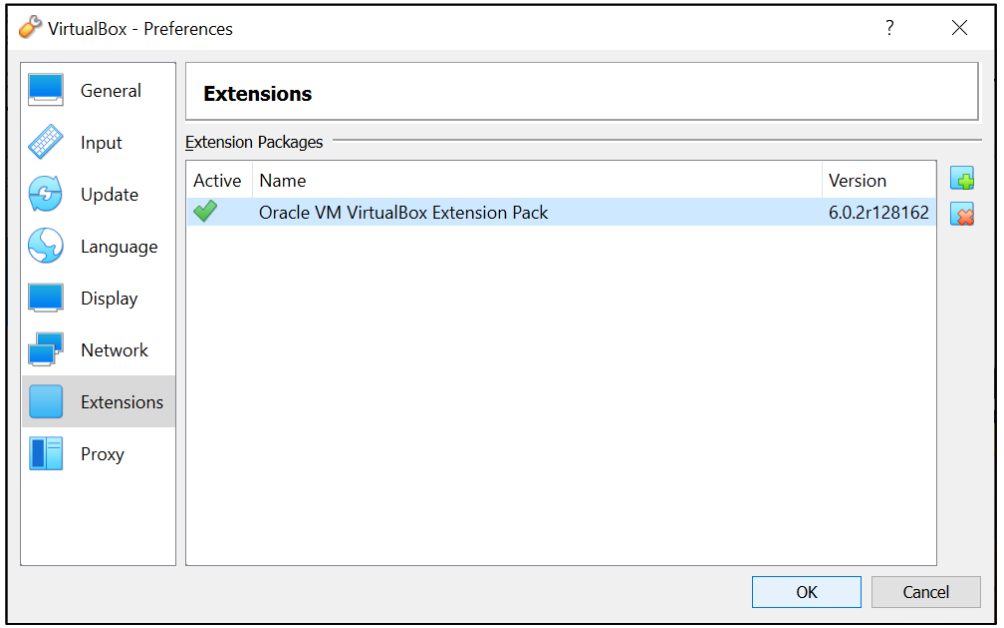
- Install VirtualBox
  - VMware Player would probably work as well
  - Download: https://www.virtualbox.org/wiki/Downloads
- Install a Debian Linux
  - I'll use Ubuntu 18.04
- Install "build essentials" tools
- Install Visual Studio Code (VS Code)
  - Setup build and launch tasks
- Optional Installs

Task #1 – Install Virtualbox

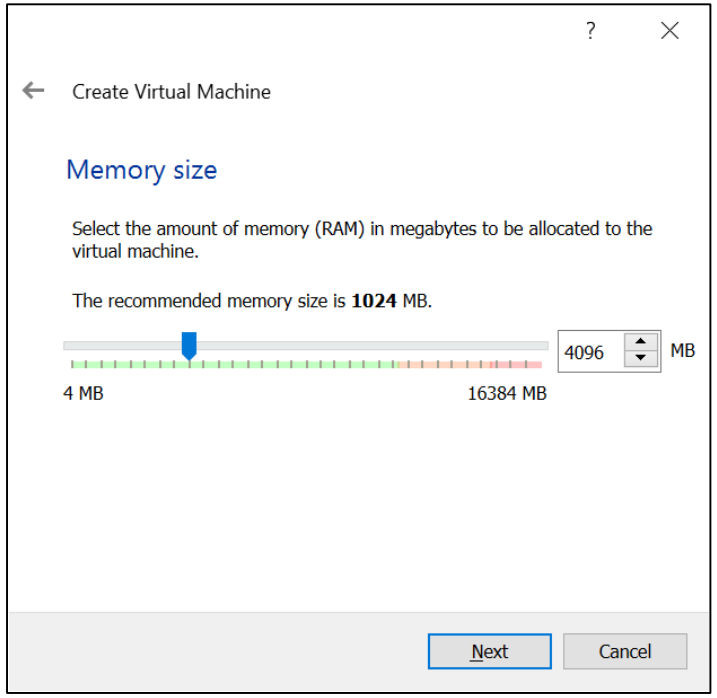
Installing Virtual Box

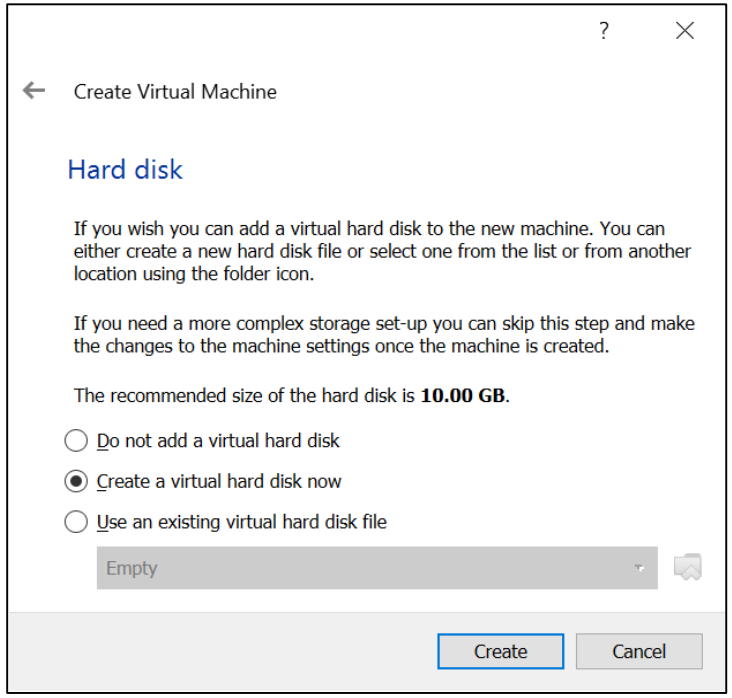
- I started by uninstalling Java and an old version of Virtualbox.
- Download and installation took about 2 minutes.
- https://www.virtualbox.org/wiki/Downloads
- Download Virtualbox extension pack and install

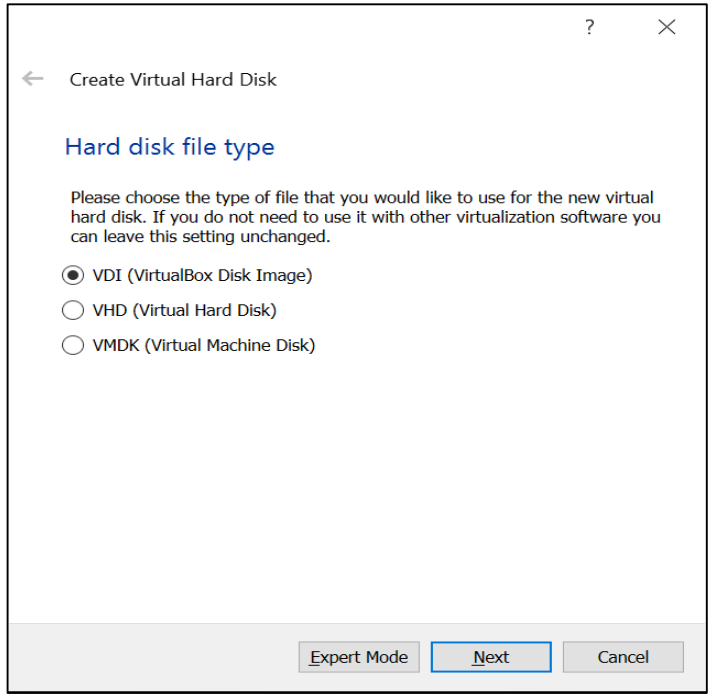




Setting VM Guest Properties

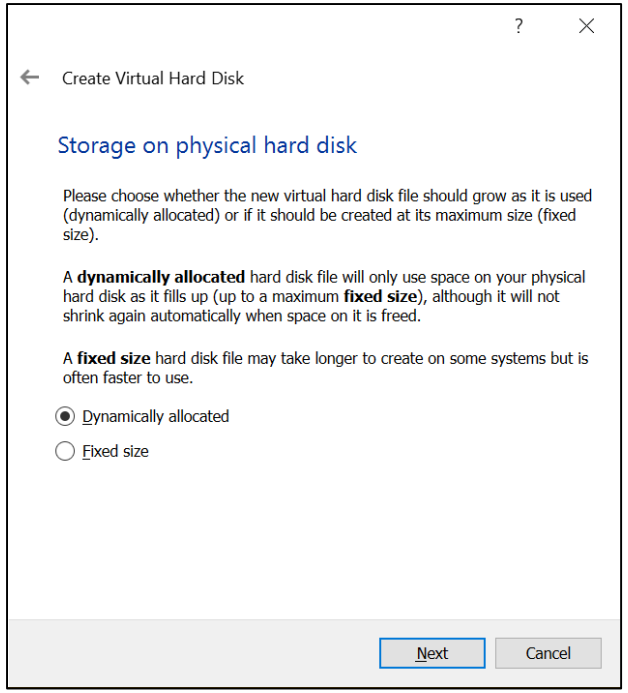


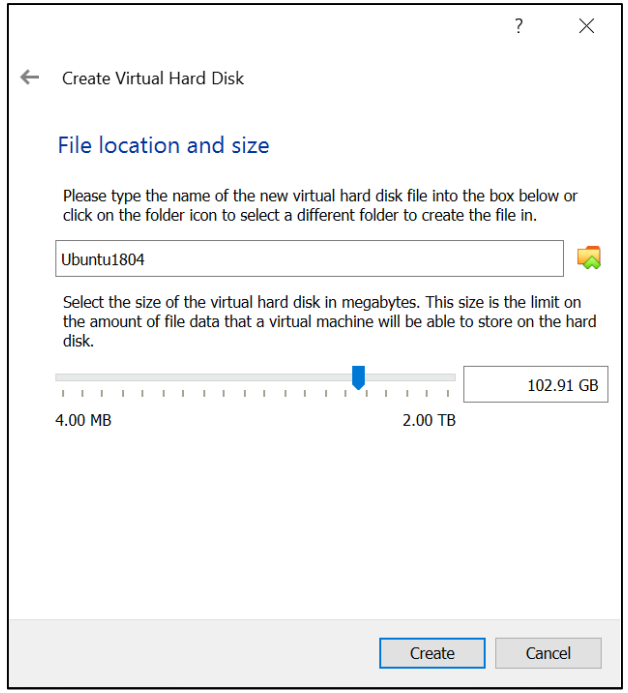


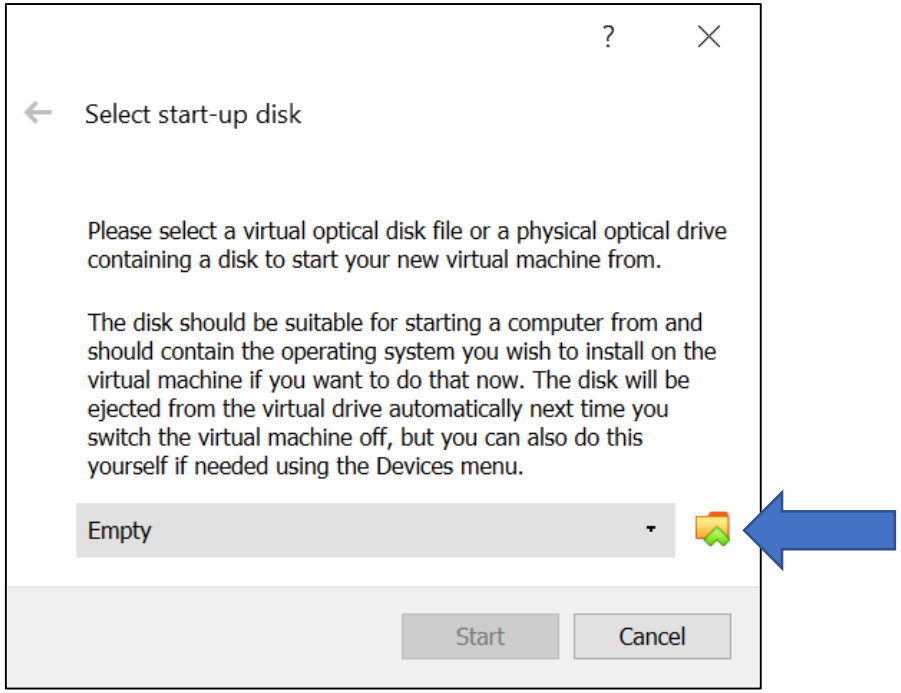


Need more RAM than recommended

Setup Virtual Disk and install Ubuntu







You will need a lot more disk space than the default setting

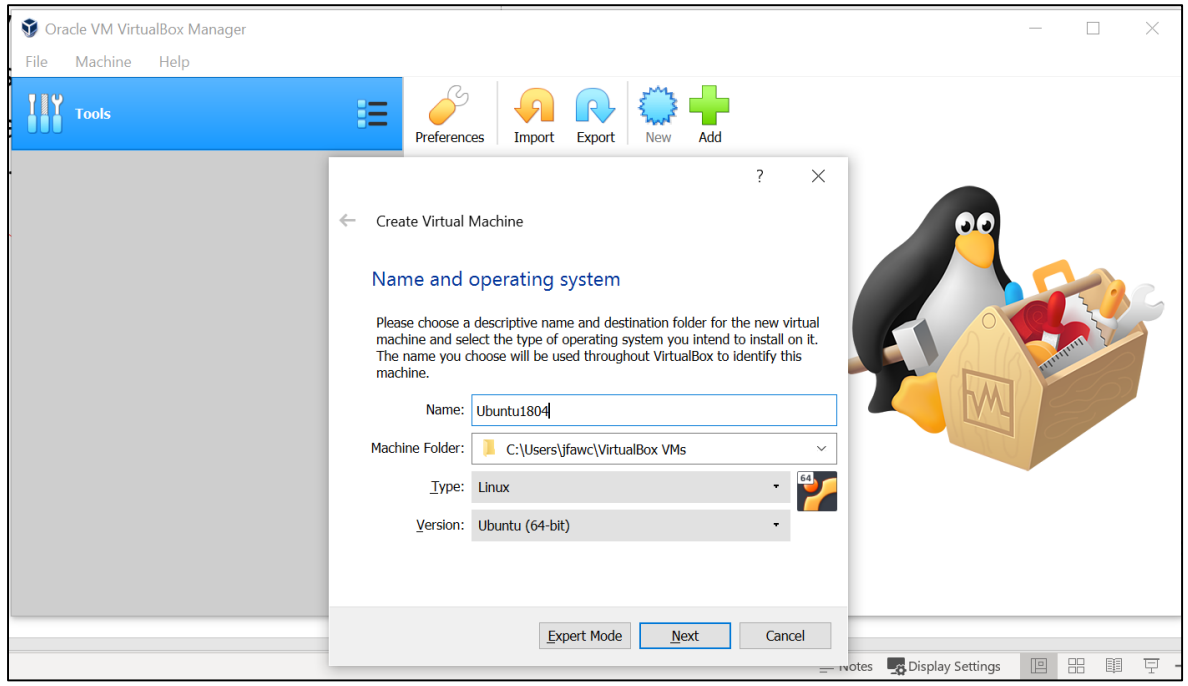
Here's where you select the Ubuntu iso file after downloading.

Task #2 – Install Ubuntu

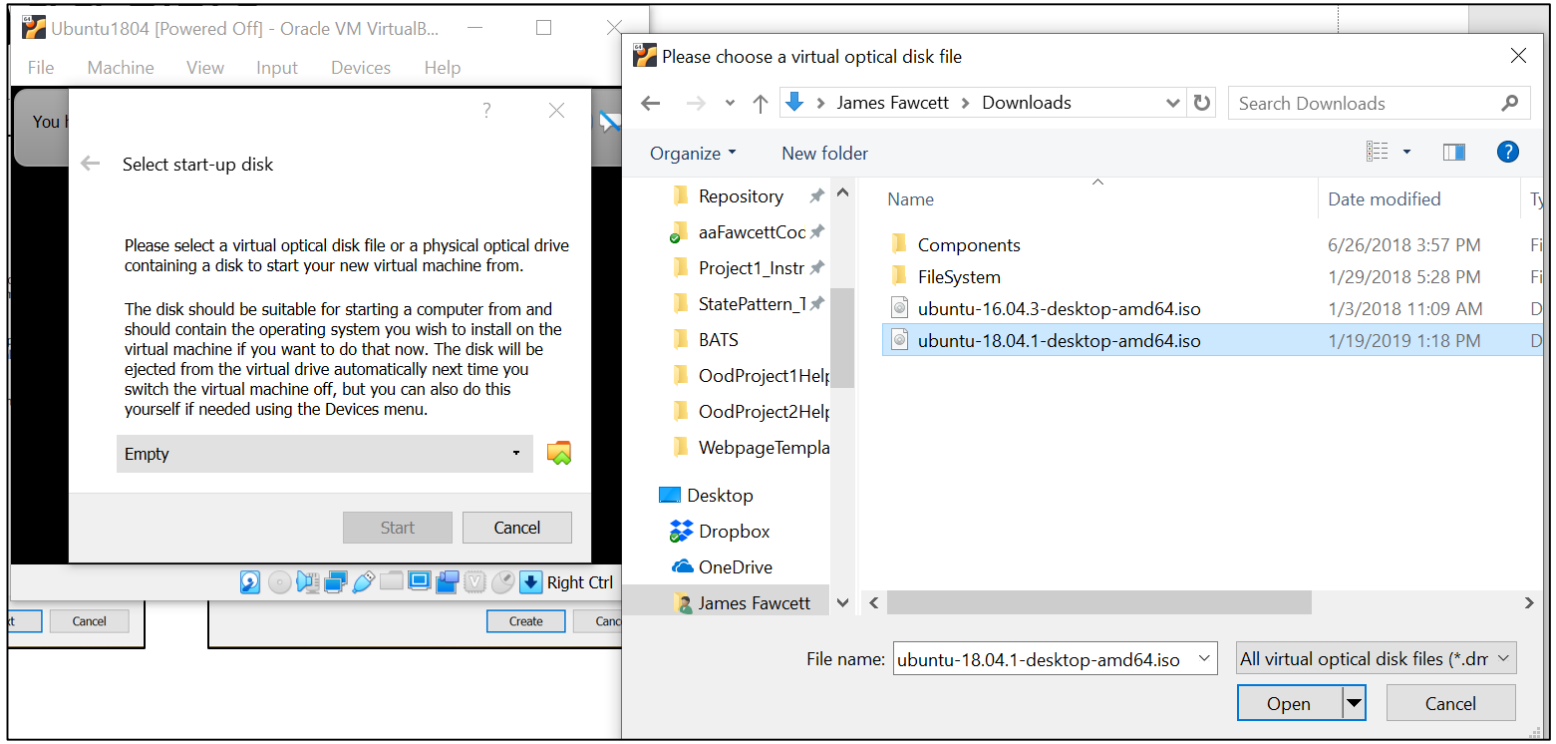
Download Ubuntu Desktop - 18.04.1 LTS

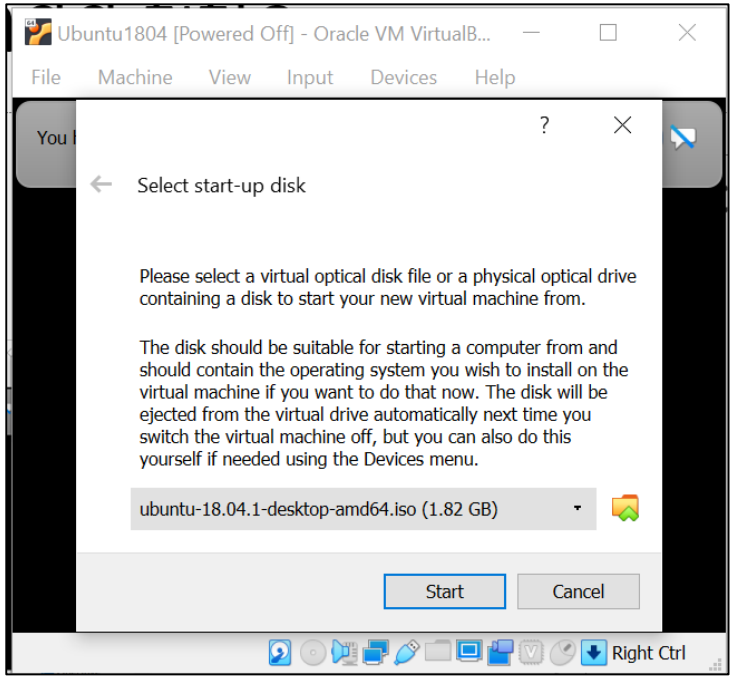
- Download Ubuntu (1.8GB)
  - https://www.ubuntu.com/download/desktop
  - Takes about 4 minutes
  - You get an iso image you install in Virtualbox.

- Install in Virtualbox

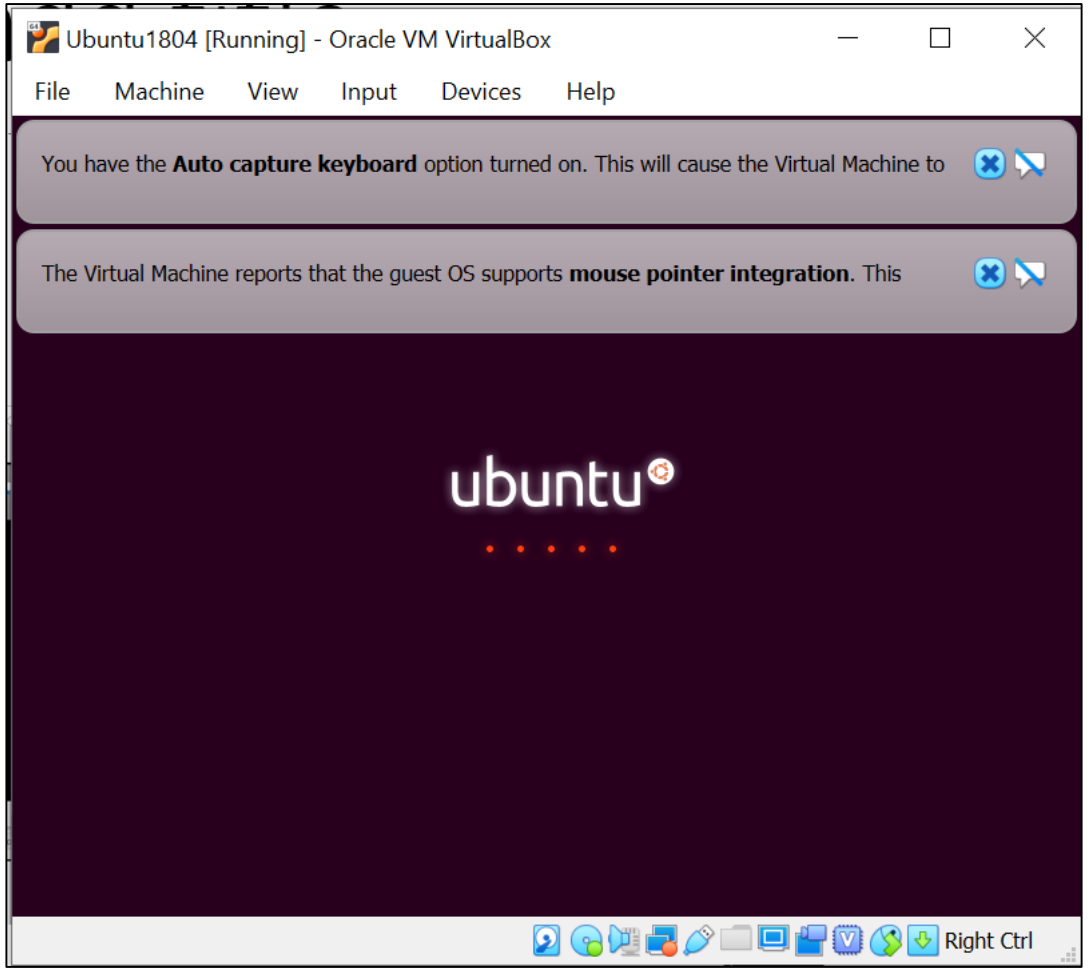


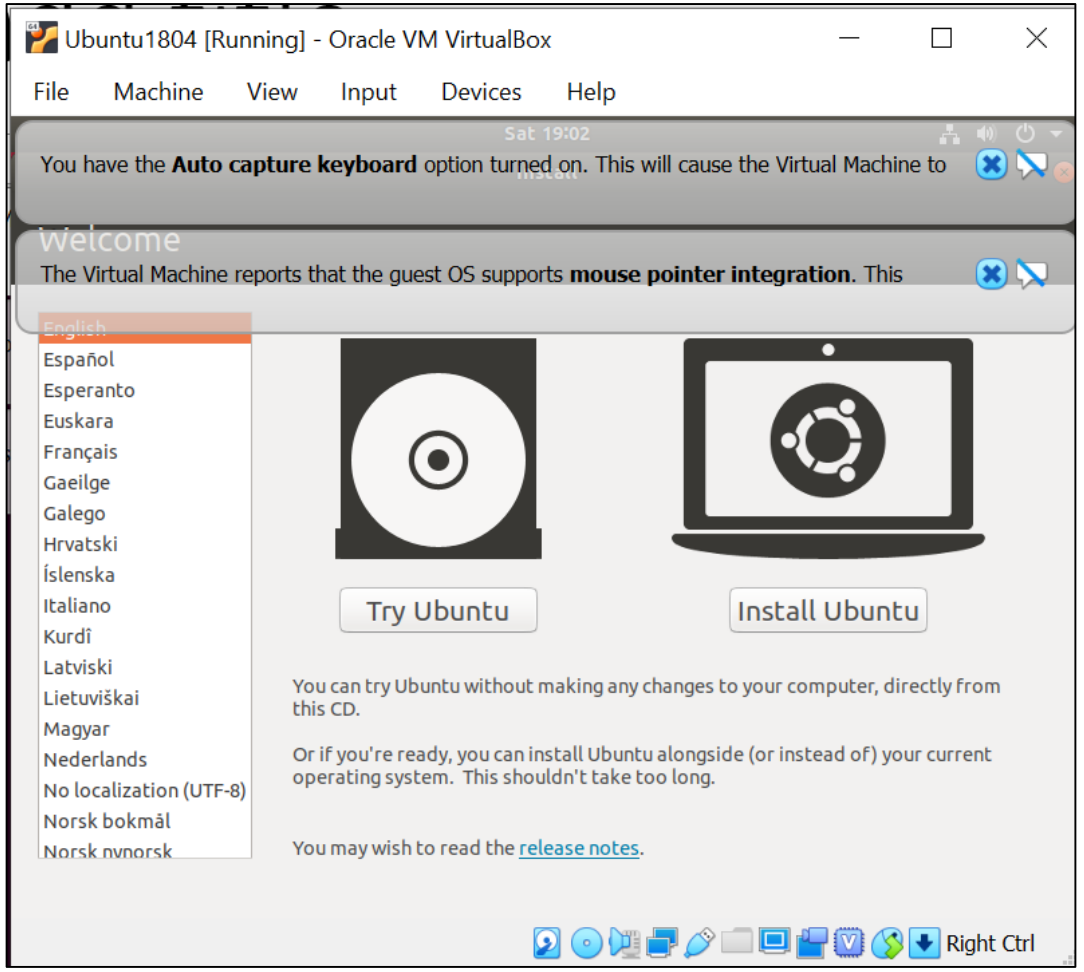
Select Ubuntu from Downloads folder



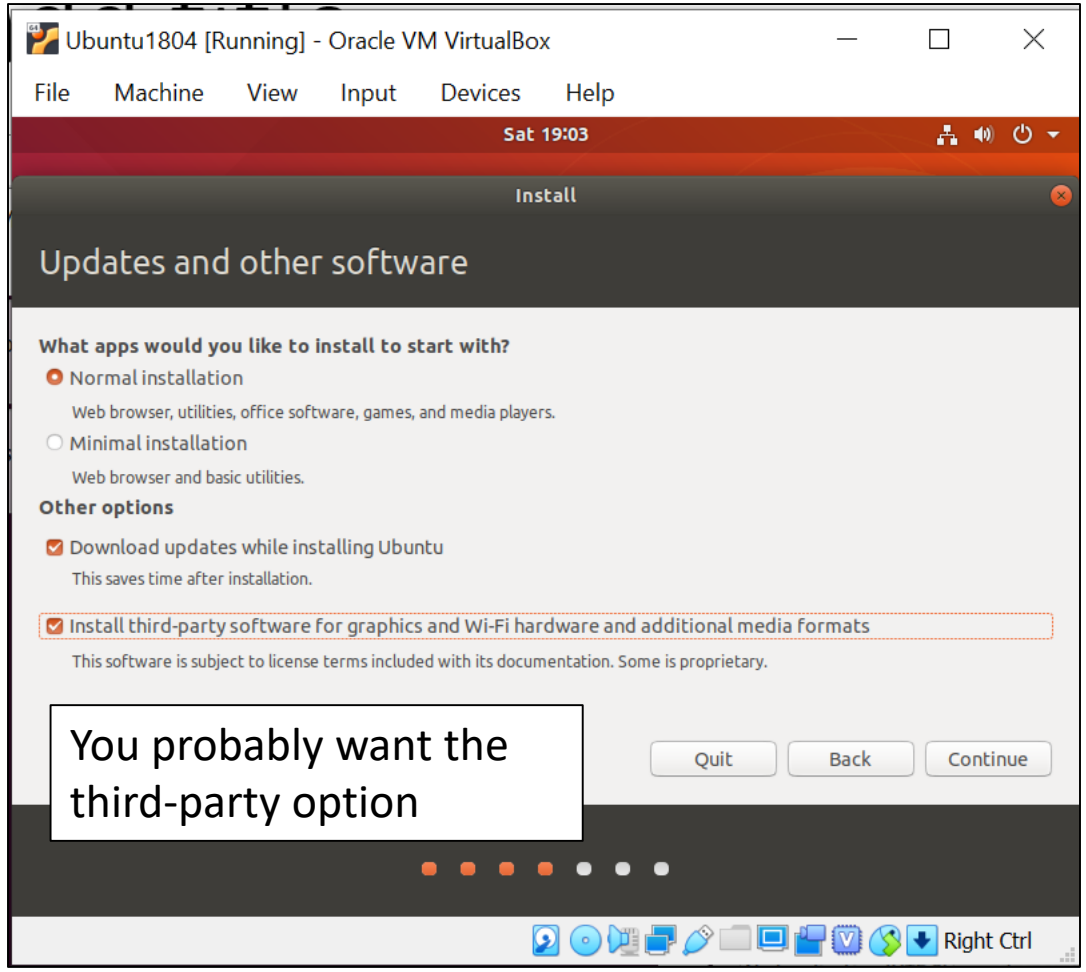


Installing Ubuntu

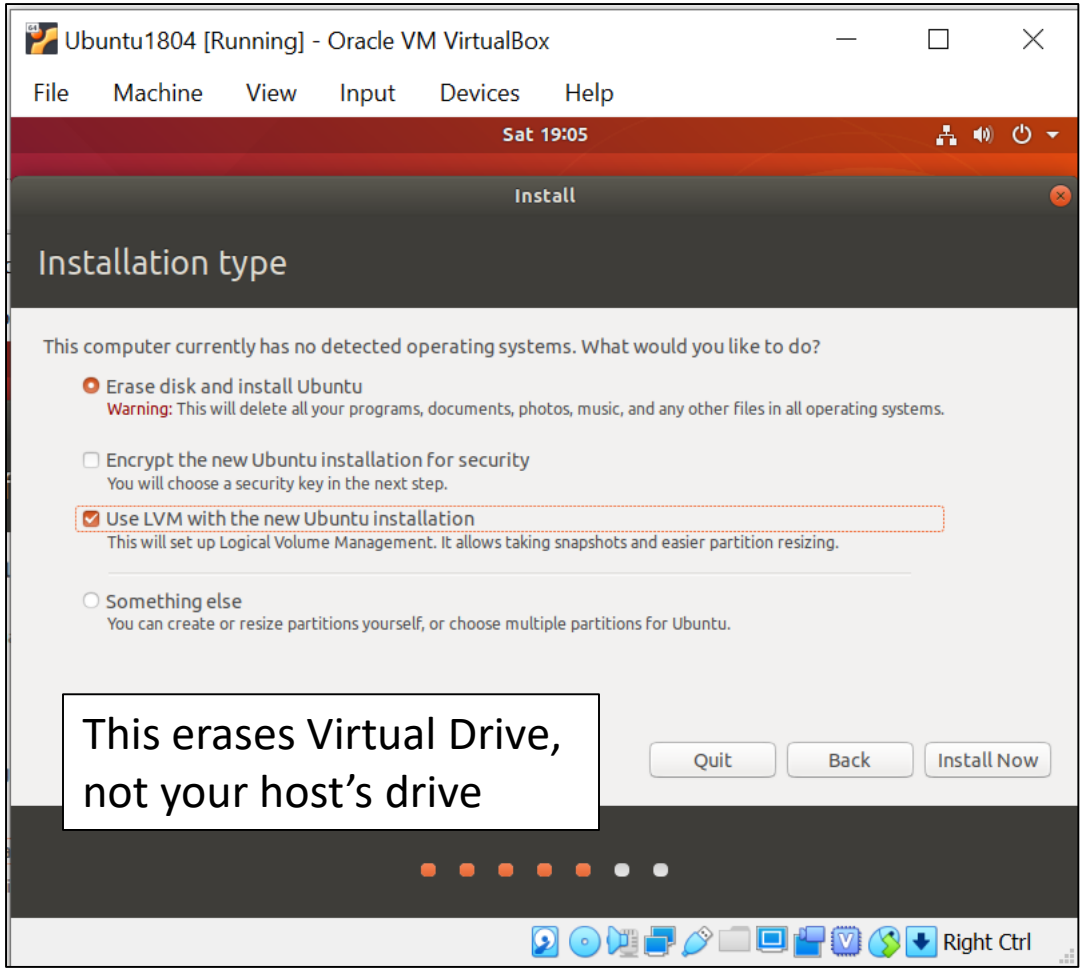




Installing Ubuntu (continued)

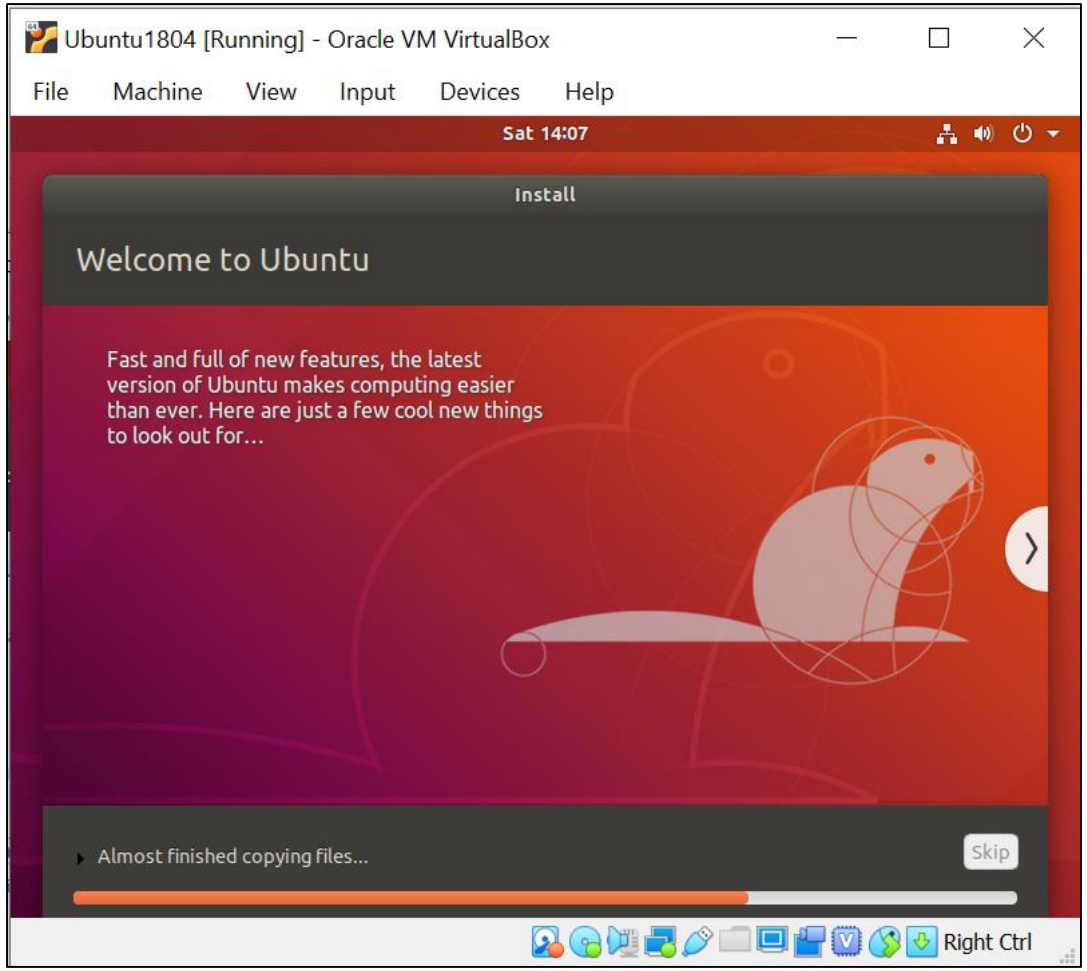


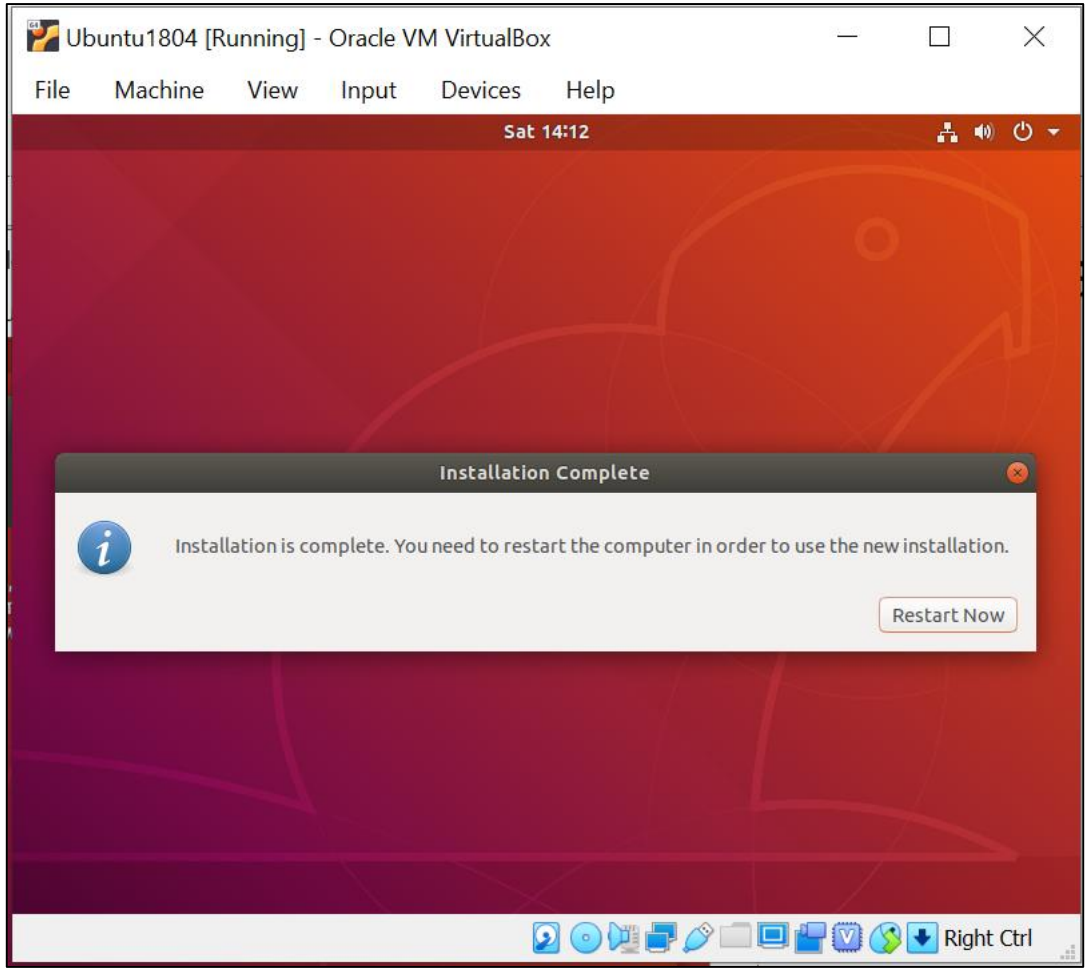
You probably want the third-party option



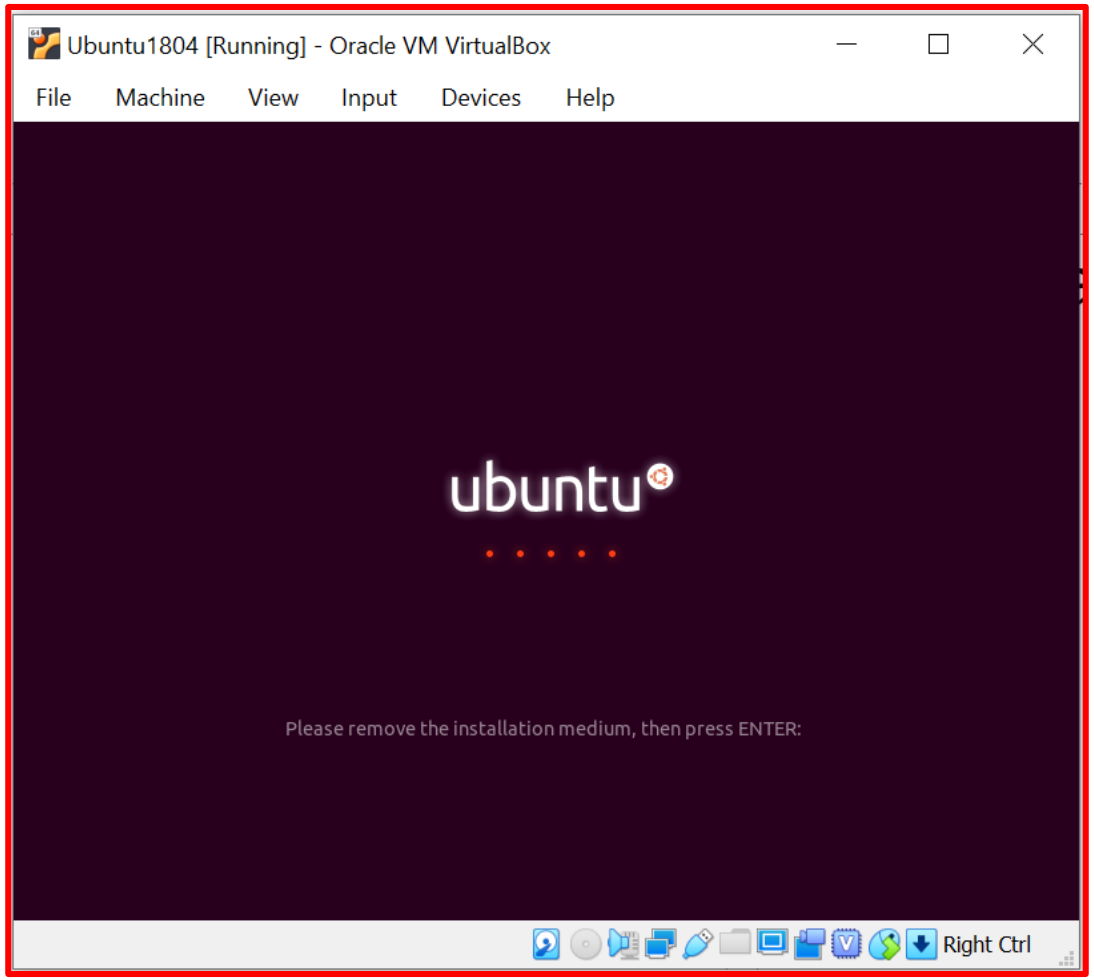
This erases Virtual Drive, not your host's drive

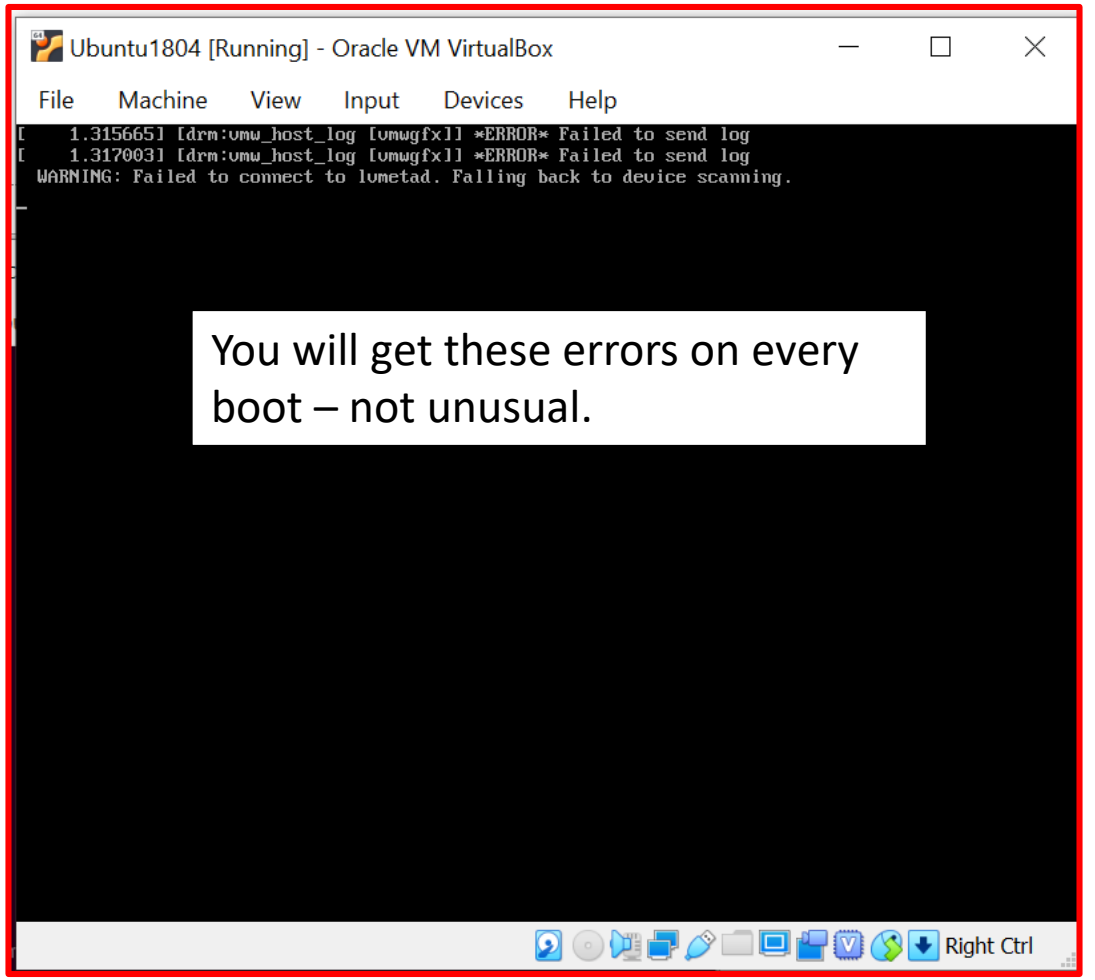
A few minutes later



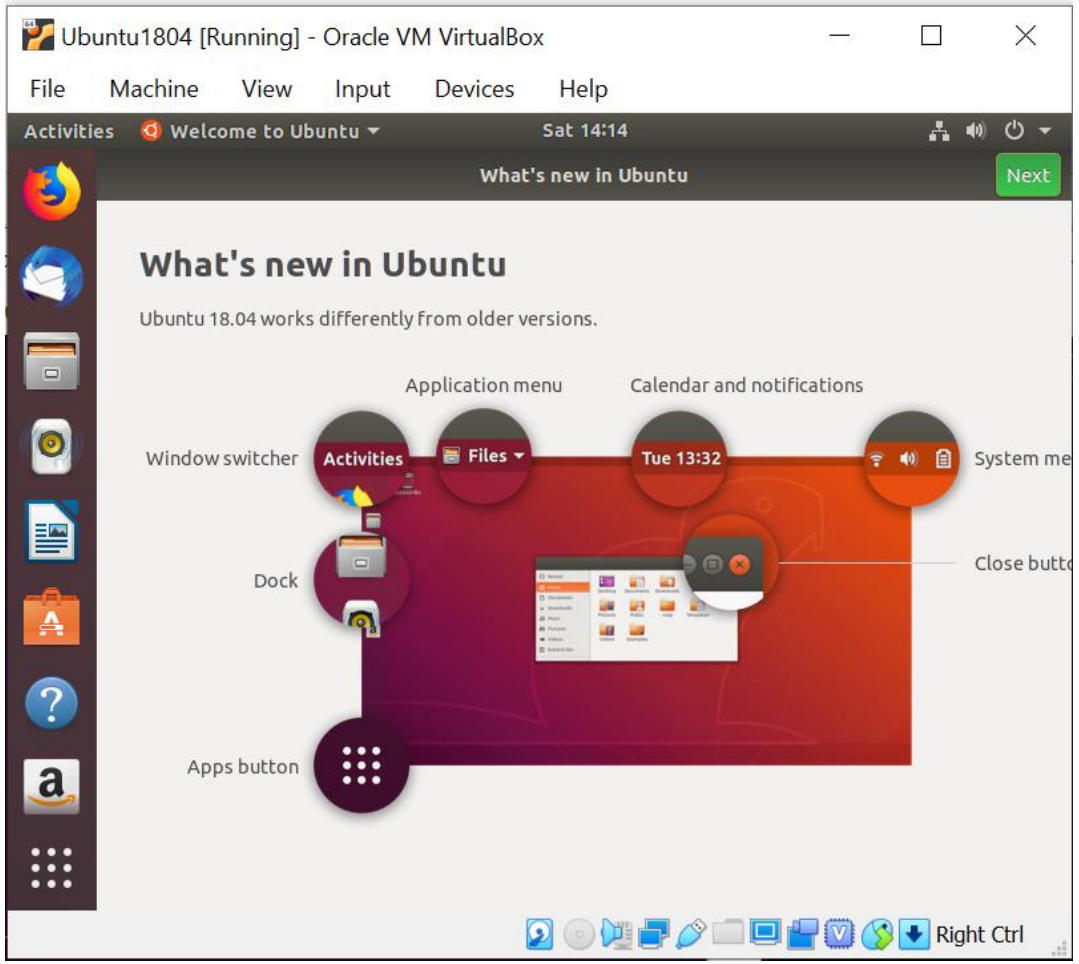


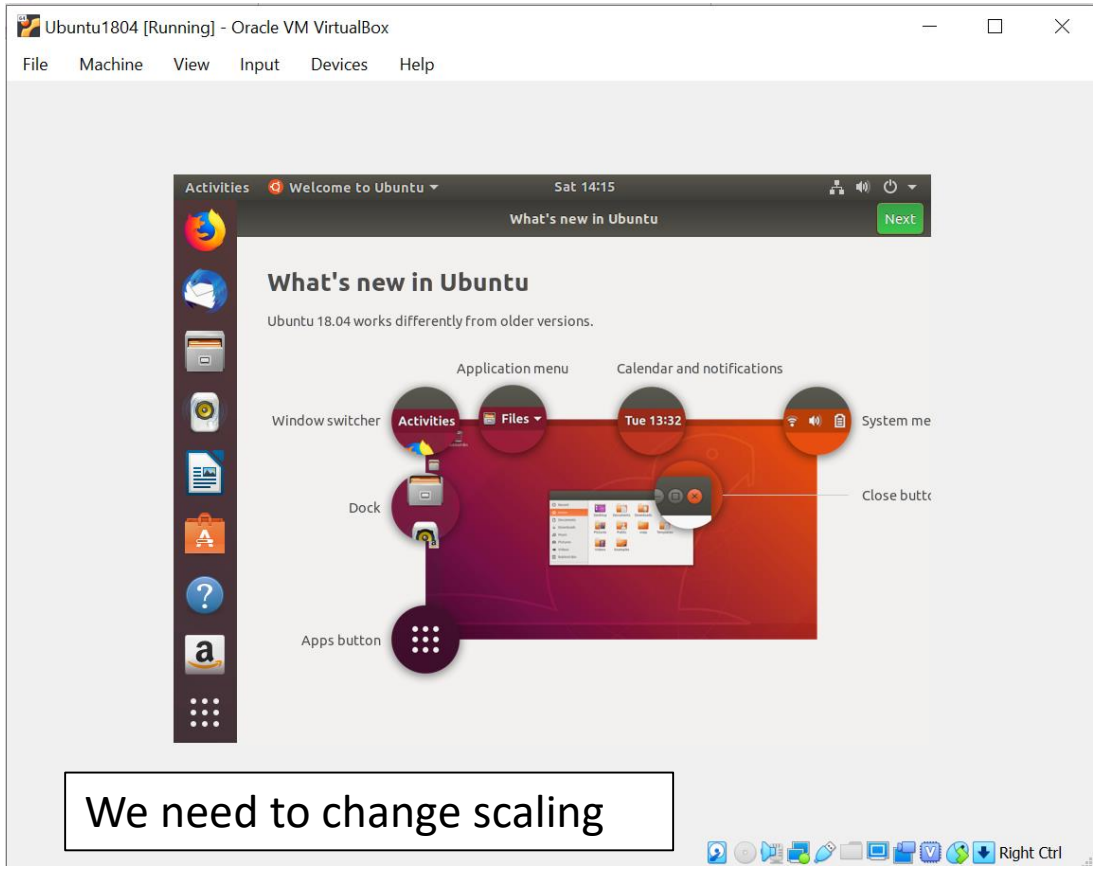
Restarting


Ubuntu1804 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
ubuntu
Please remove the installation medium, then press ENTER:
Right Ctrl


Ubuntu1804 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
[1.315665] [drm:vmw_host_log [vmwgfx]] *ERROR* Failed to send log
[1.317003] [drm:vmw_host_log [vmwgfx]] *ERROR* Failed to send log
WARNING: Failed to connect to lvmetad. Falling back to device scanning.
You will get these errors on every boot – not unusual.
Right Ctrl

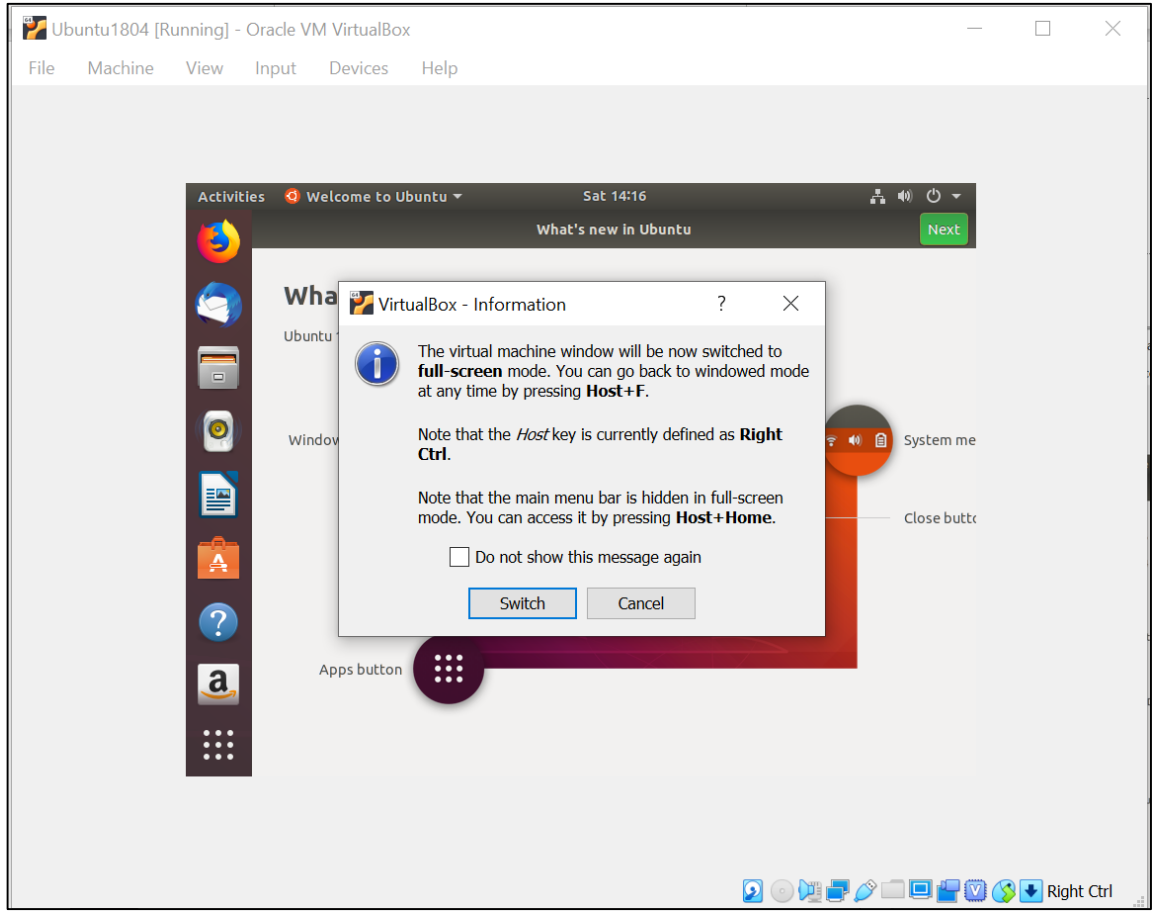
Now it’s time to configure

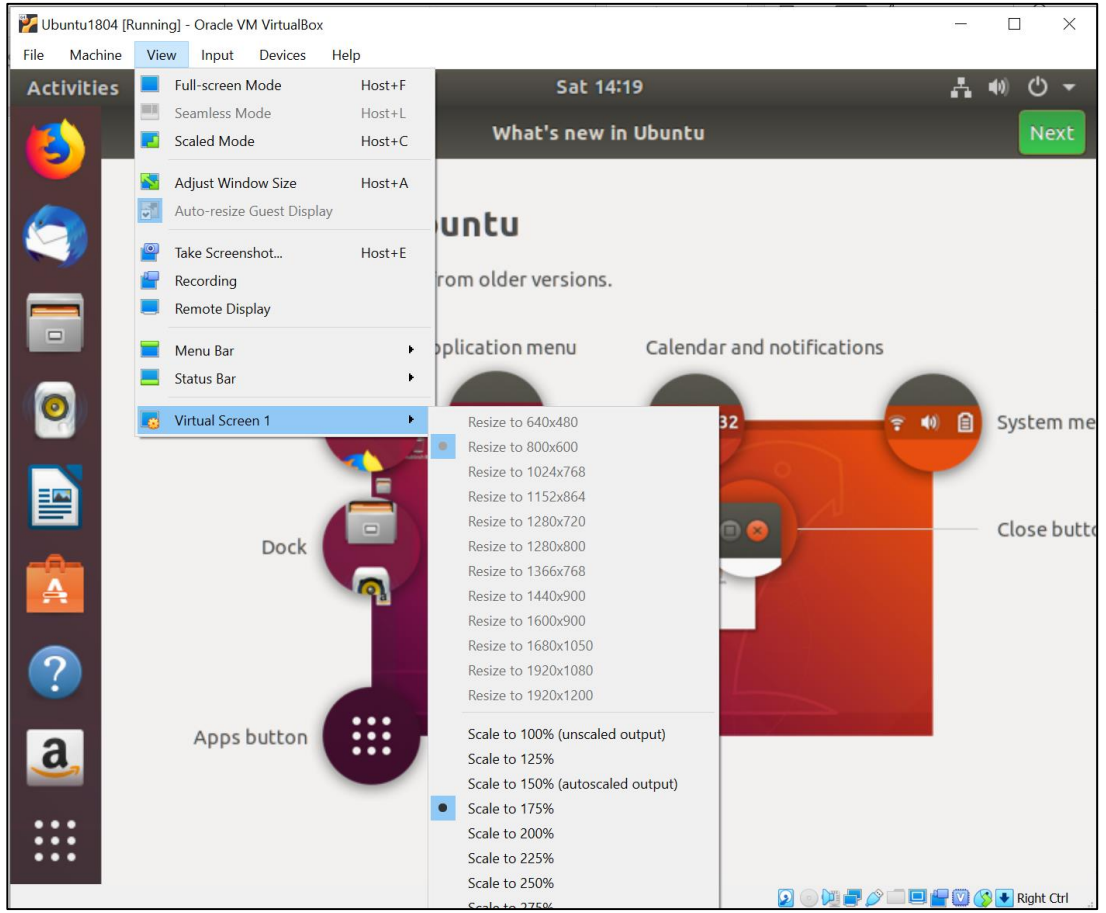




We need to change scaling

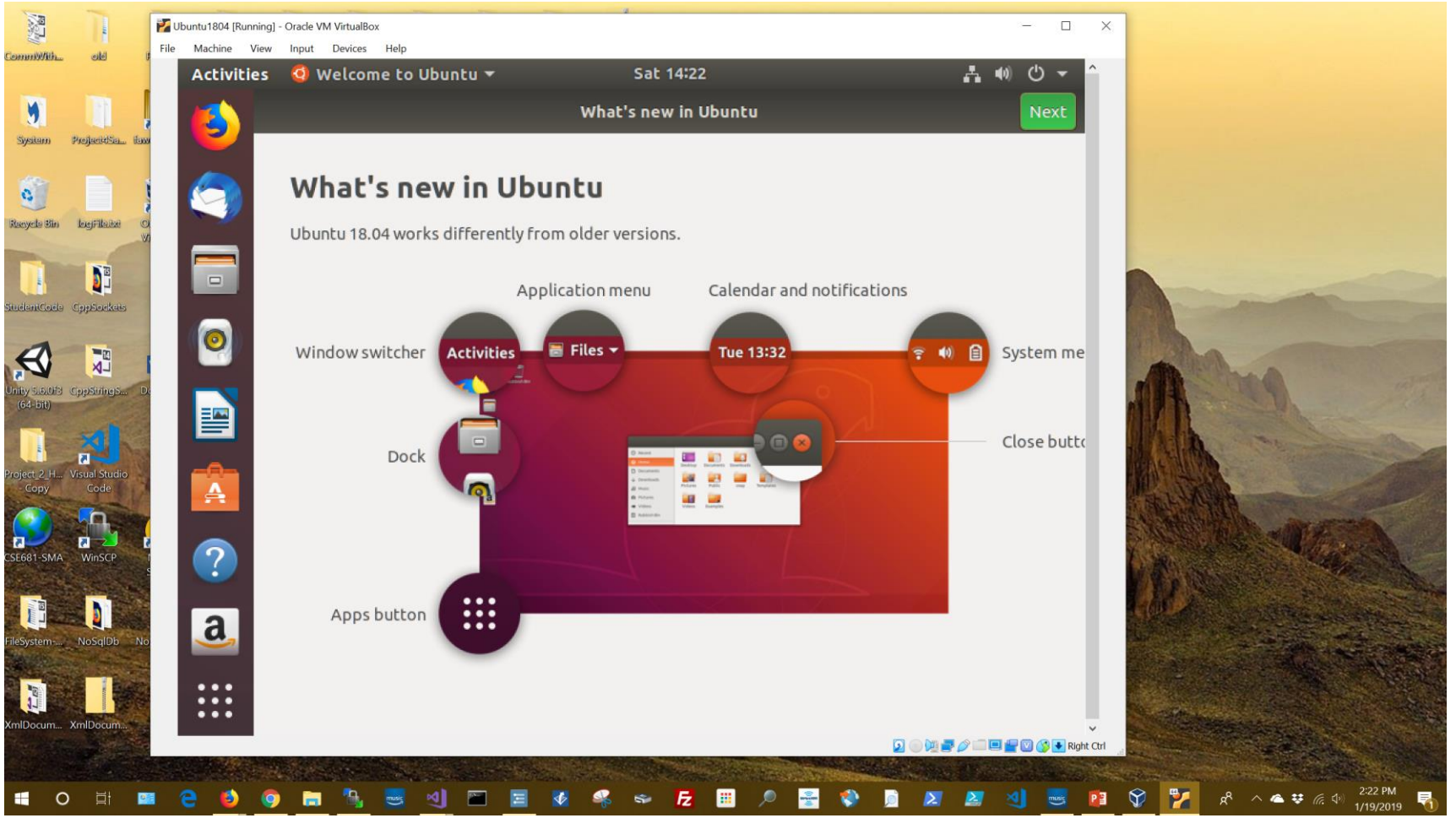
Setting Scaling



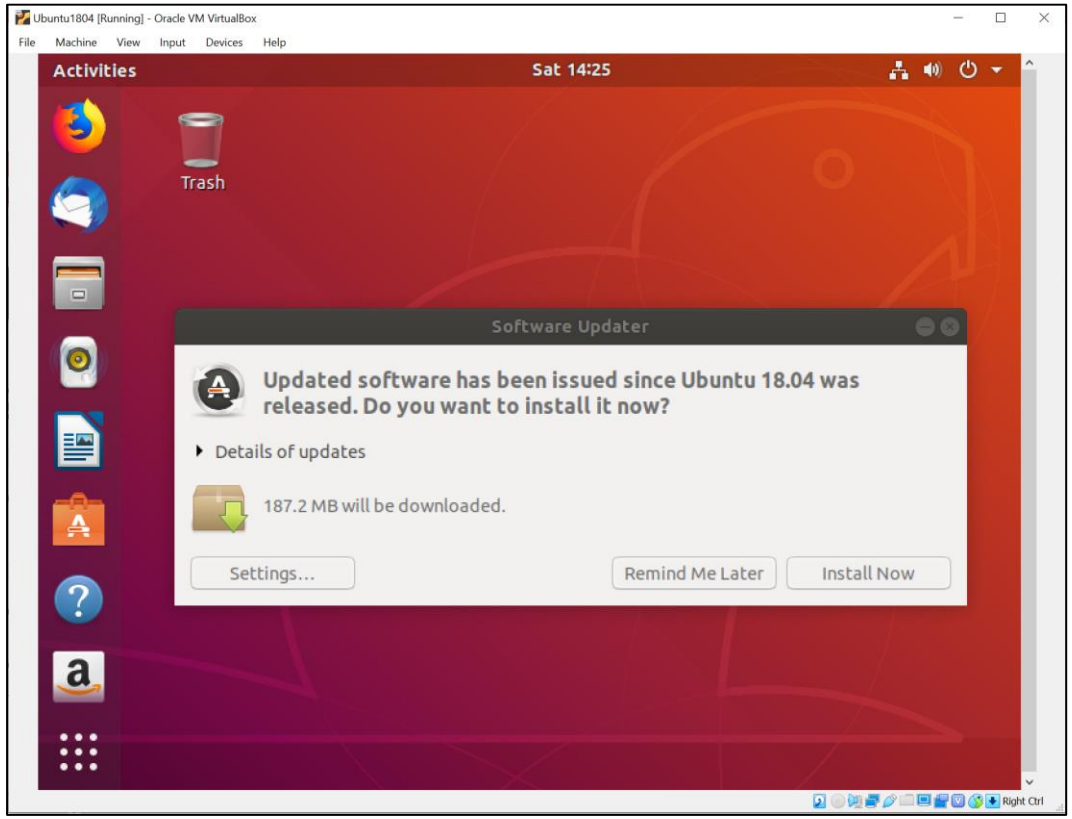


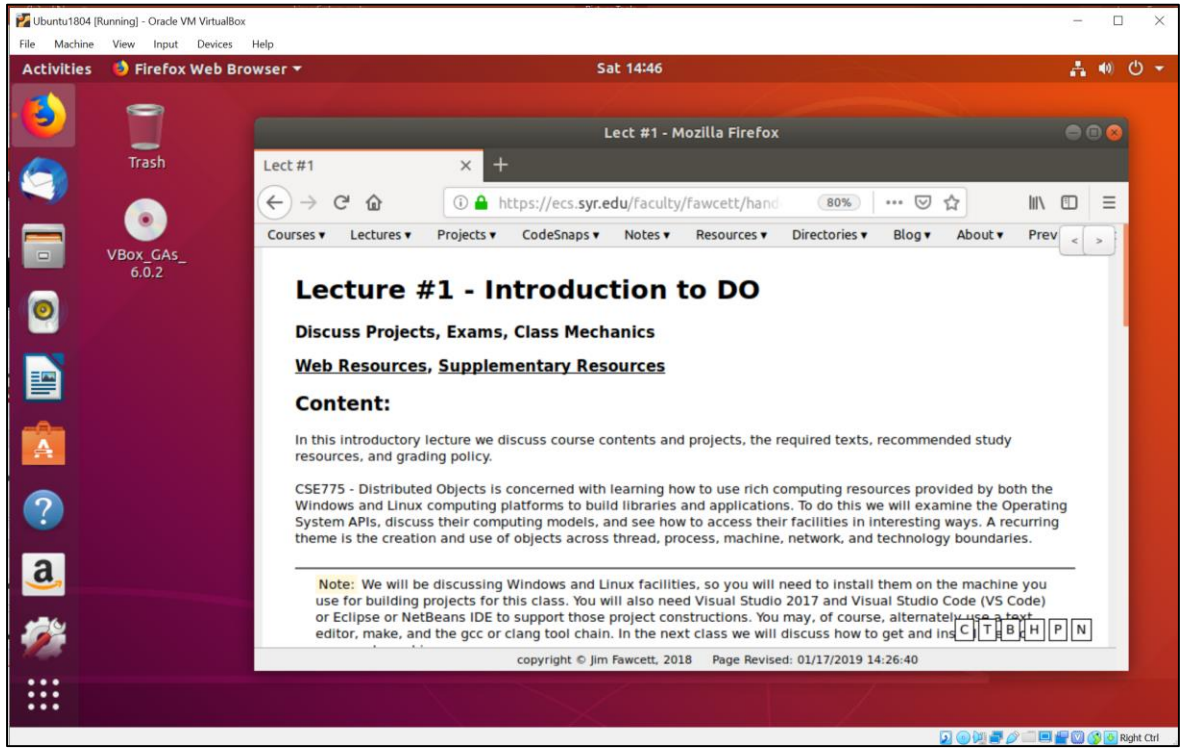
Here's Scaled View

- The view is configured with the VM host, VirtualBox.
- Set menu bar icon sizes with settings > dock in Ubuntu

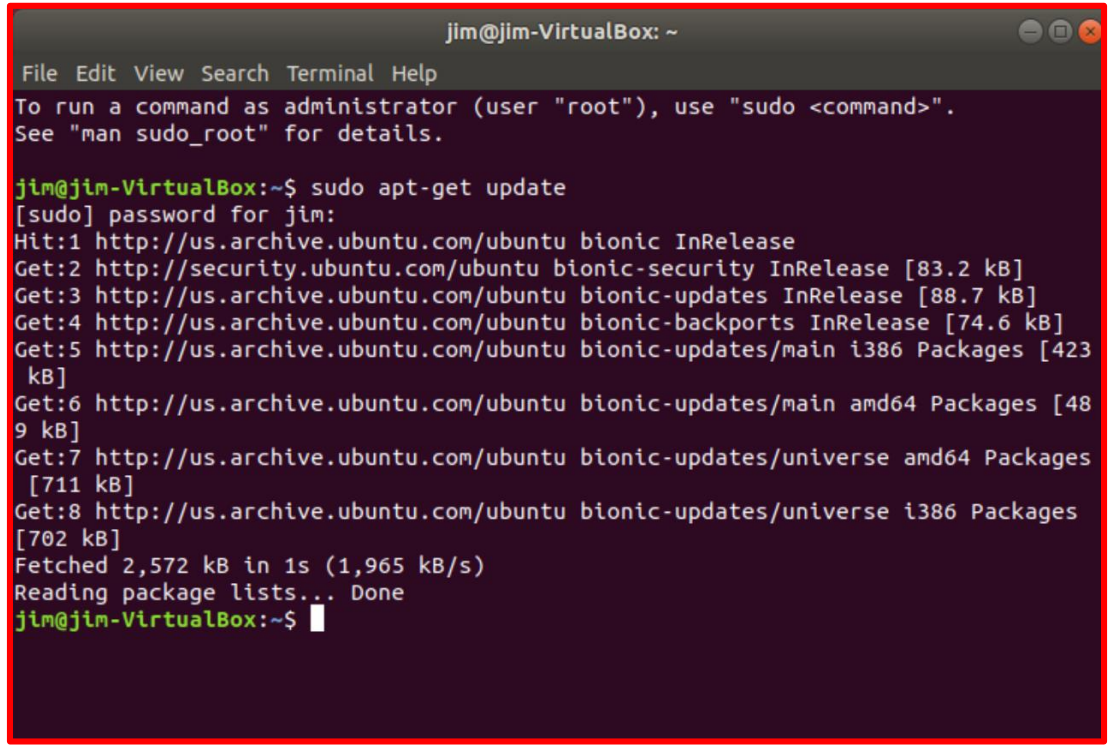


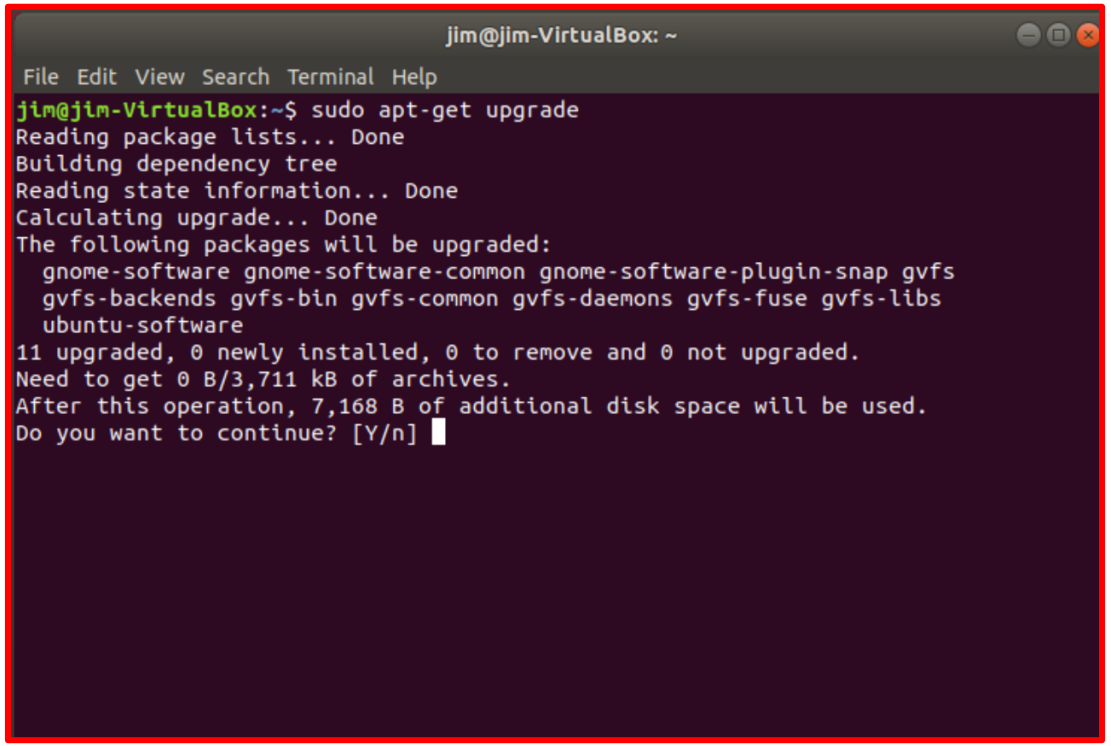
What do you know - It Works!



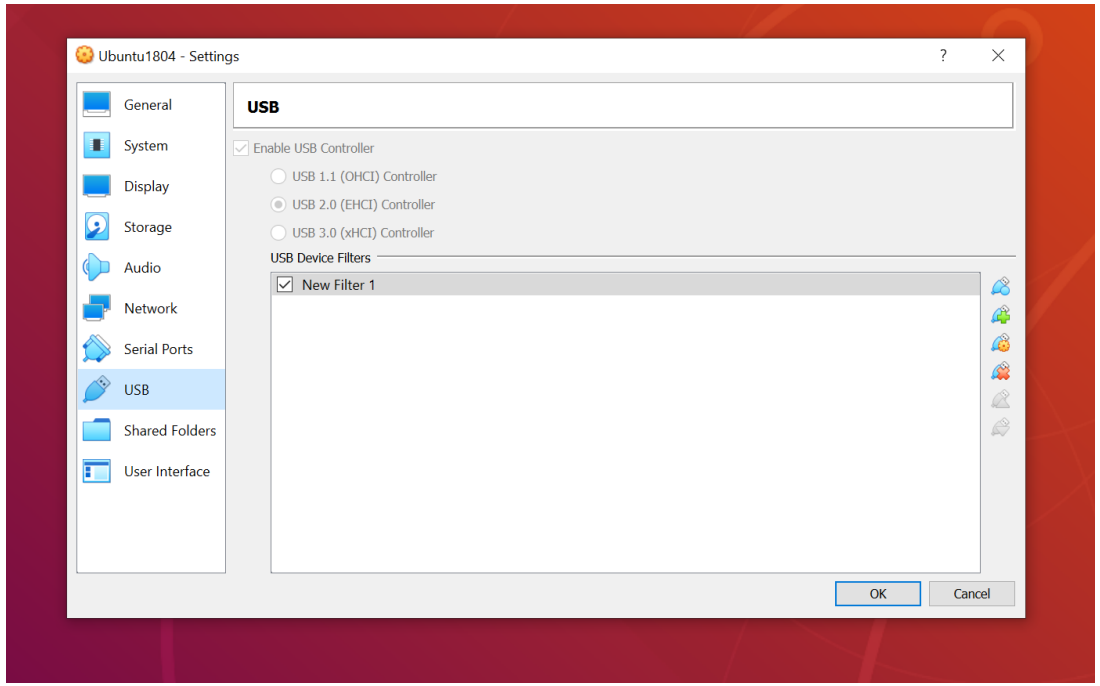


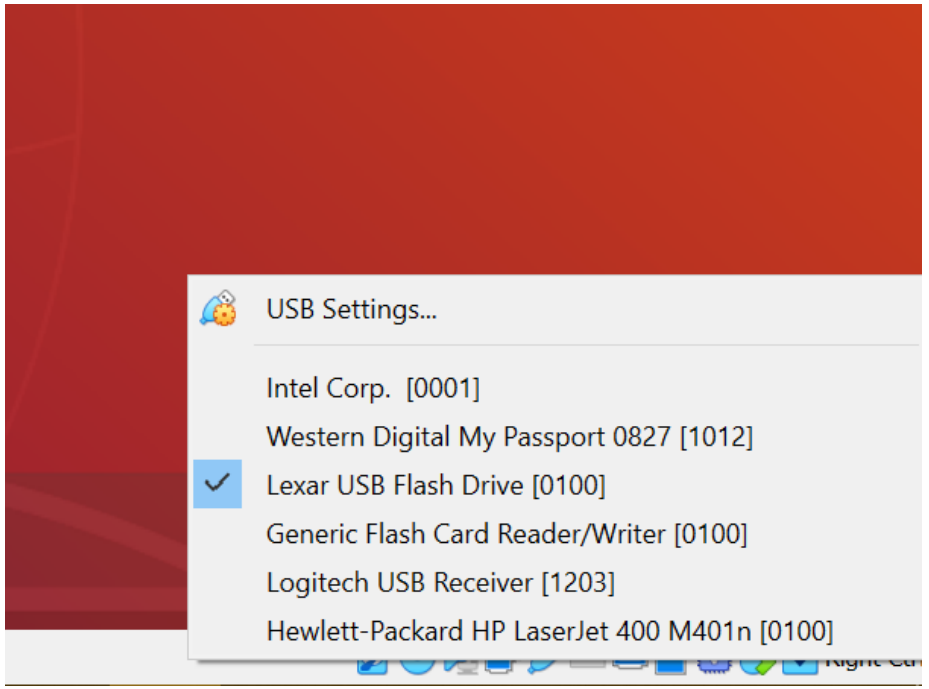
Update Ubuntu Installation using apt pkgmgr





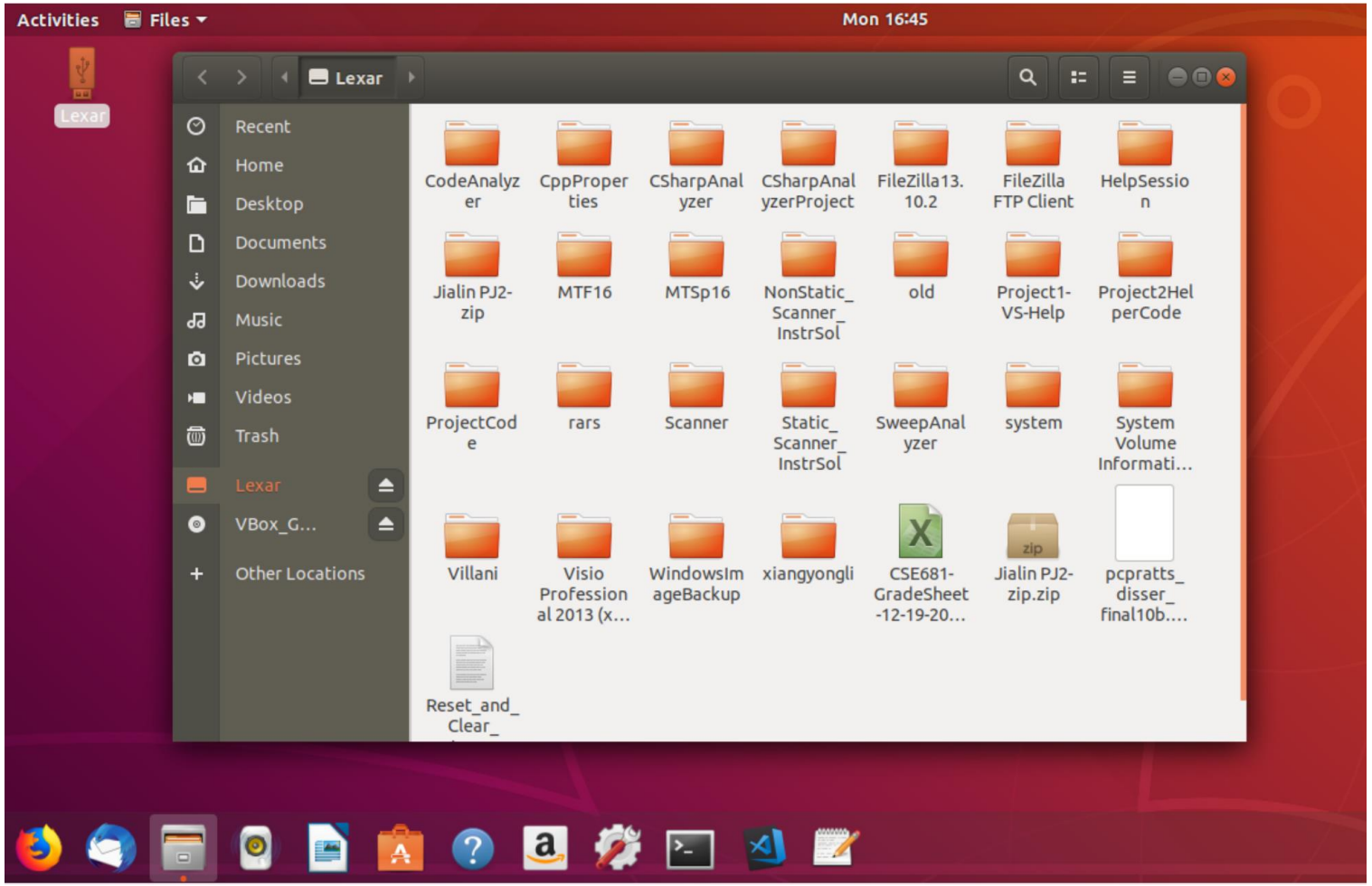
Accessing usb Devices





Using usb devices

- A bug in Virtualbox causes other devices, e.g., Bluetooth mouse, to fail.
- Restarting the VM resolves that problem.
- Large drives will probably fail to connect.

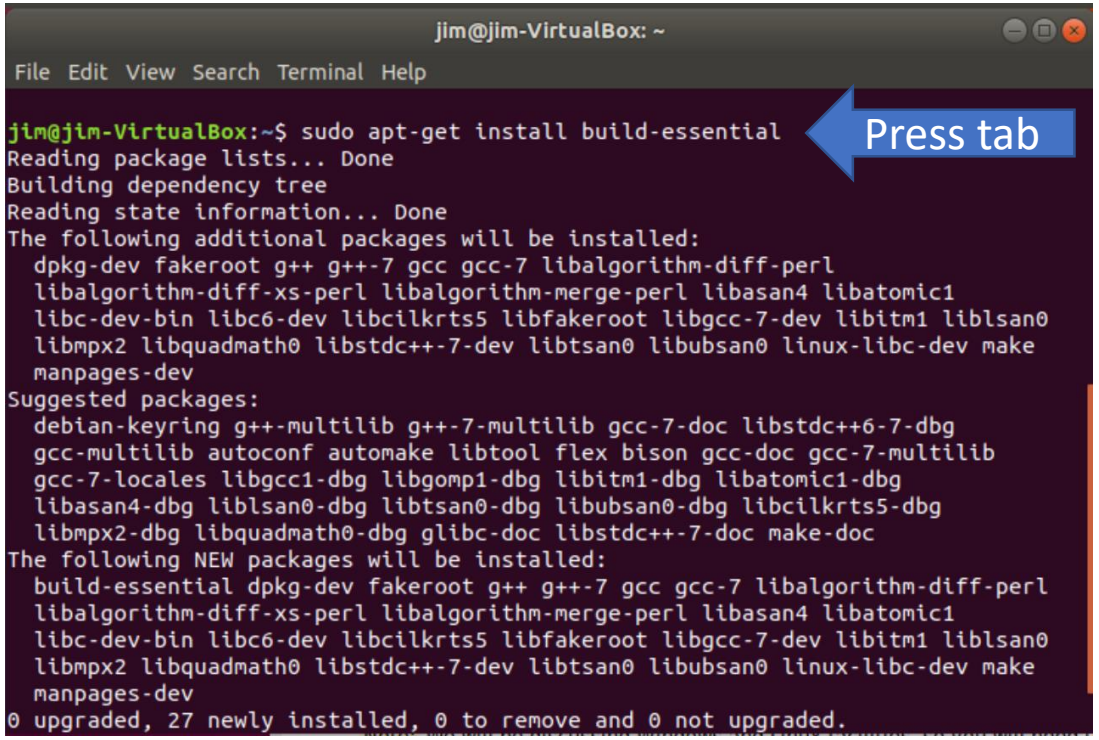


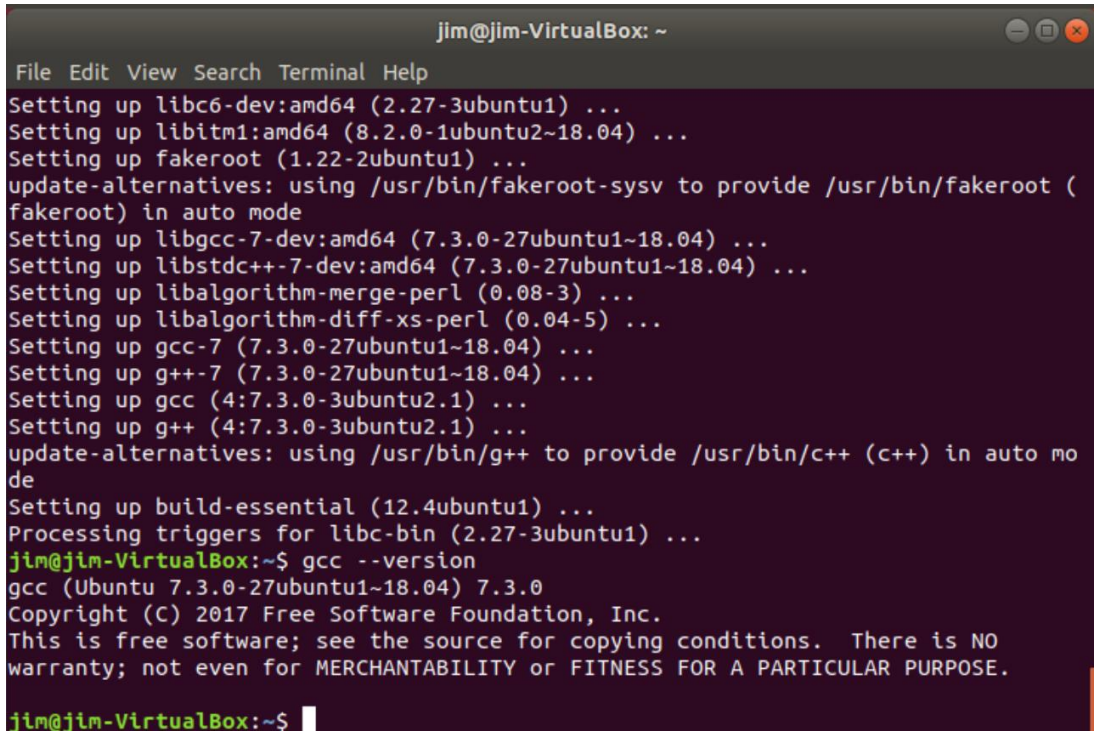
Task #3 – Install build-essential

gcc toolchain configured by Ubuntu team to build Ubuntu

- gcc (c/c++ compiler and linker)
- Make
- Many other tools

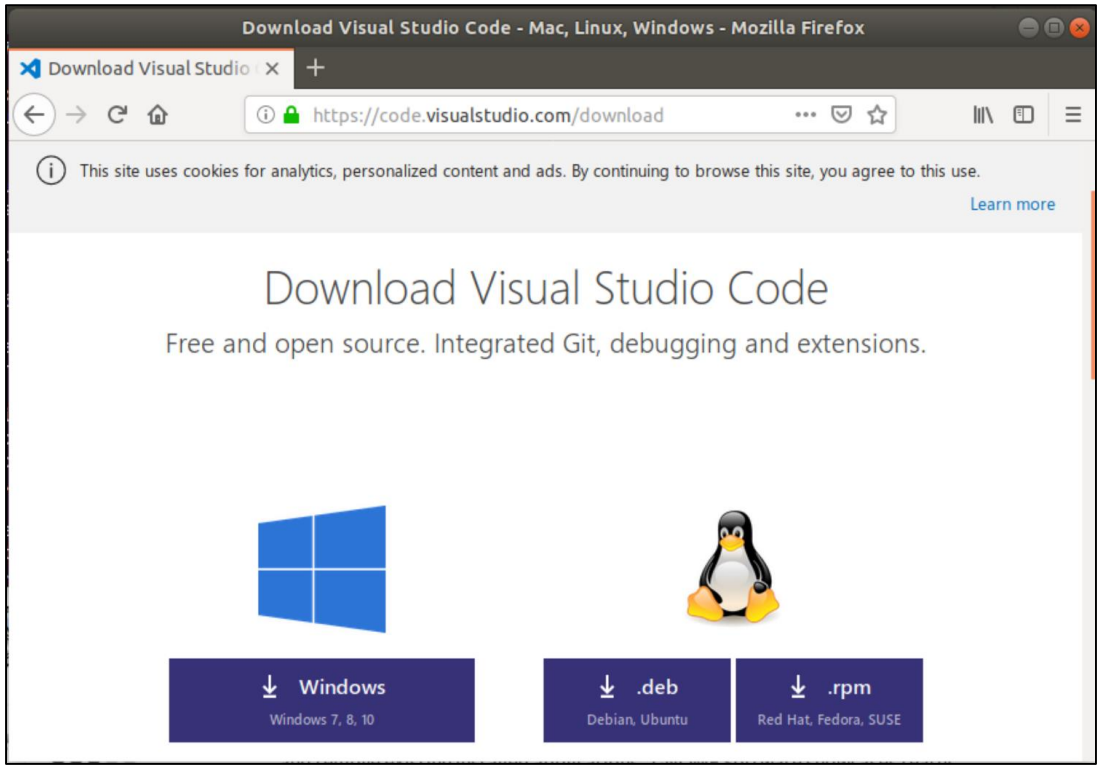
Install “build essential” gcc tool chain

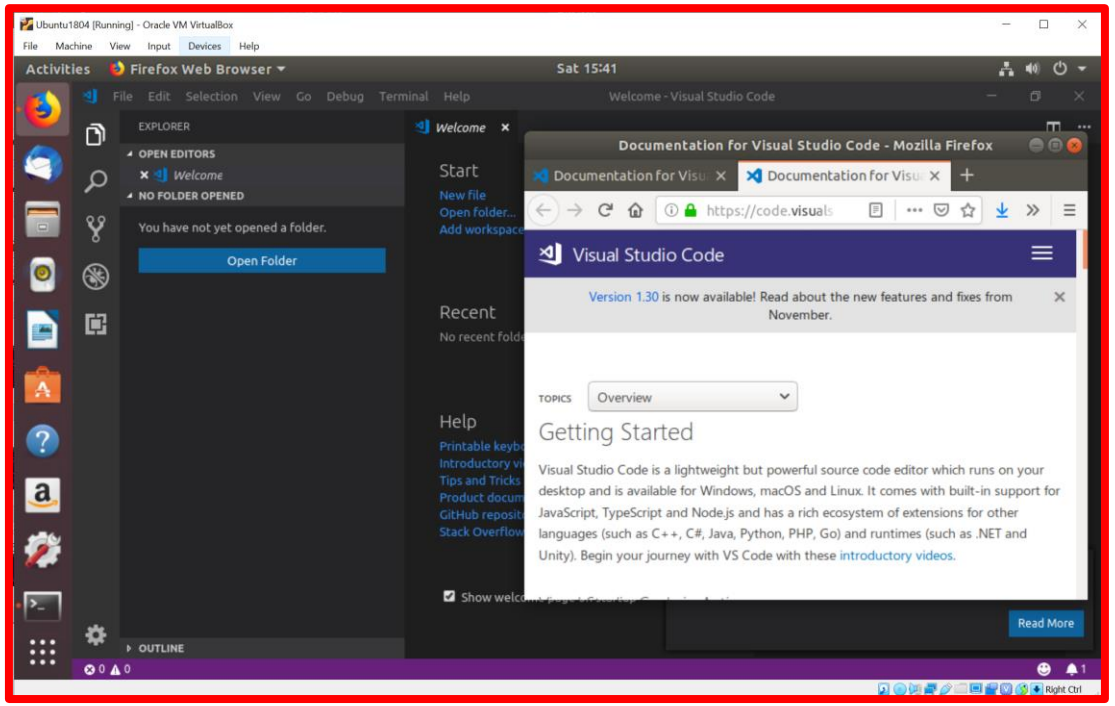




Task #4 – Install VS Code

Download and Install Visual Studio Code

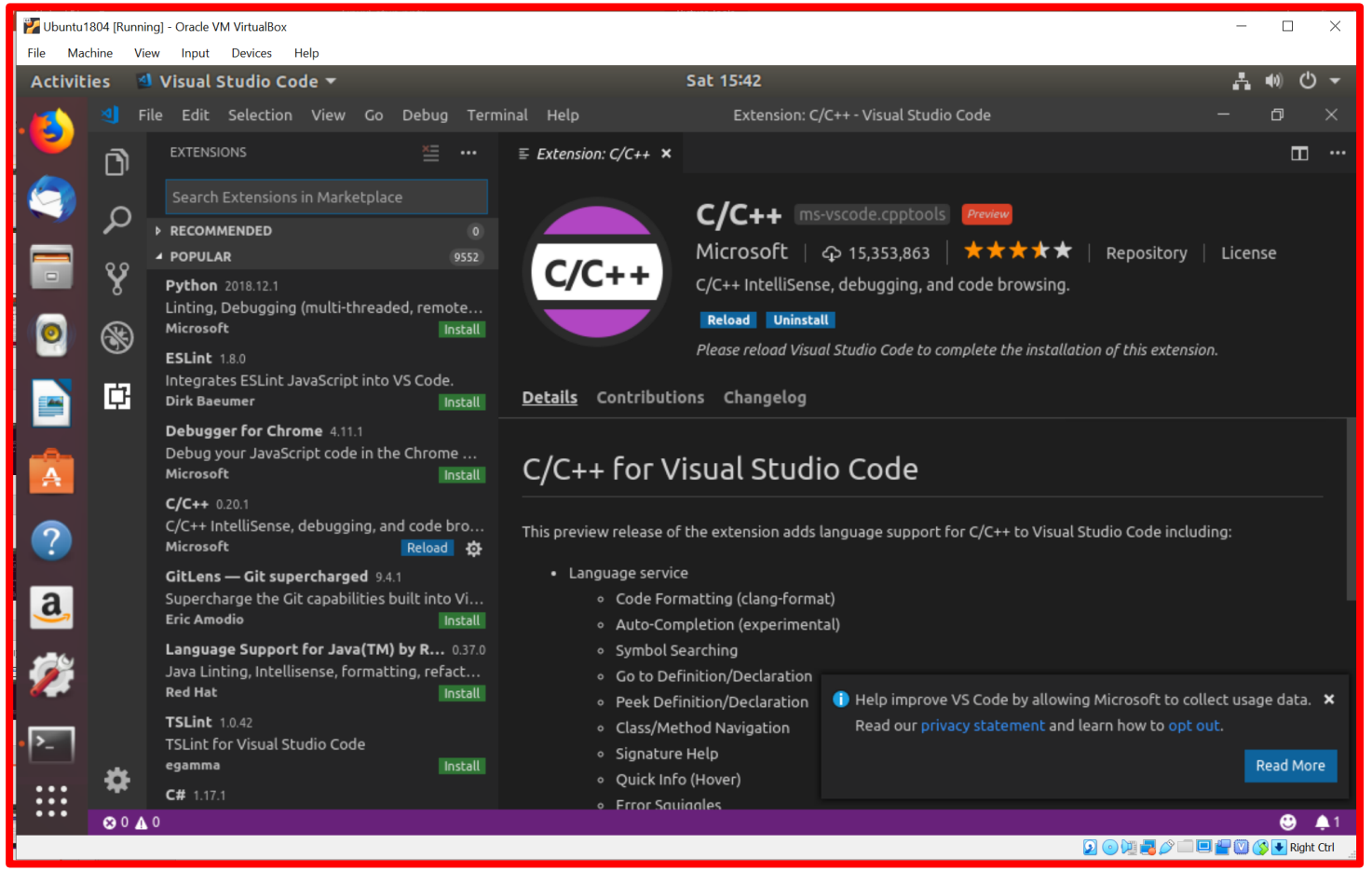




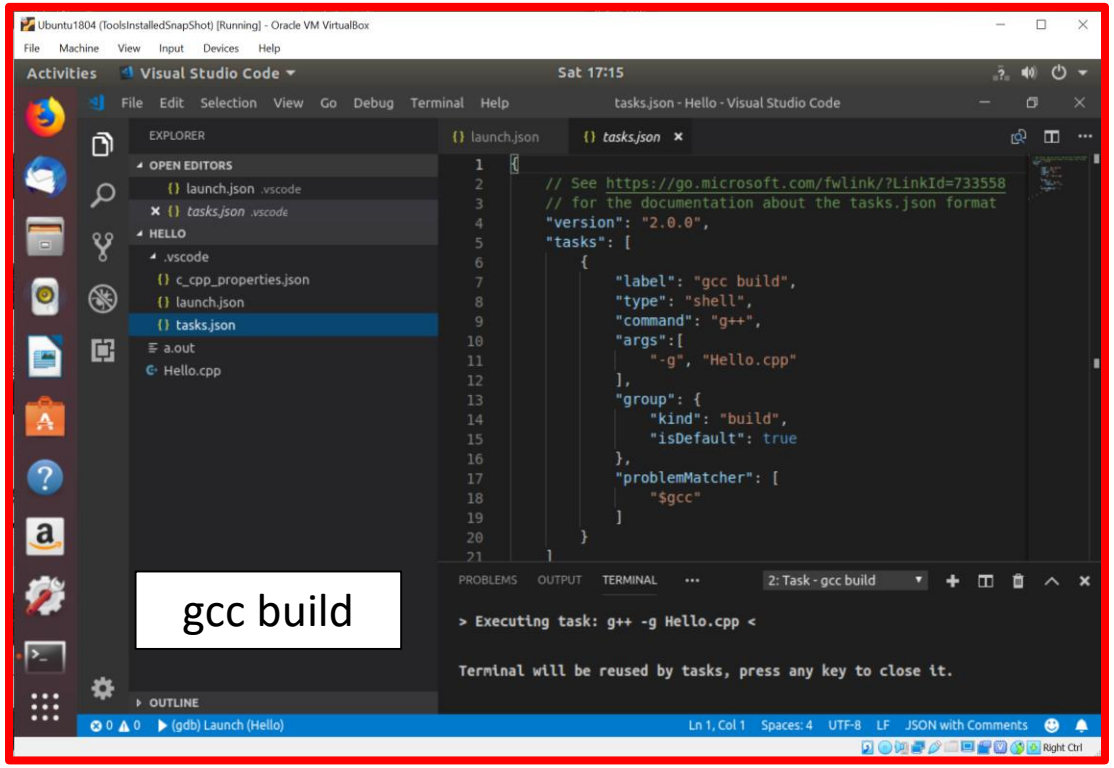
Follow directions on this page.
You can't just apt-get install.

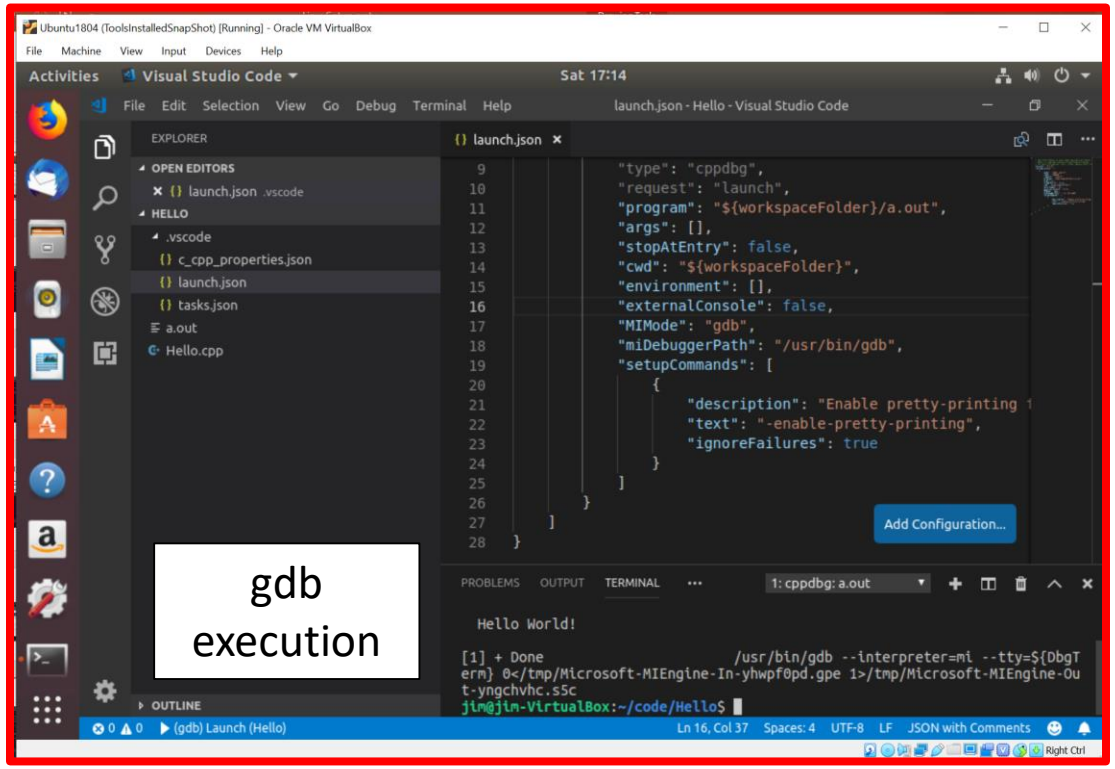
Adding C++ Plugin

- Click on plugin icon
  - Bottom of left pane
- Select C/C++
- This configures json setting files (partially)
- You must already have a C++ tool chain installed



VS Code – Building C++





Tasks.json and Launch.json

Build Task

```
{
    // See https://go.microsoft.com/fwlink/?LinkId=733558
    // for the documentation about the tasks.json format
    "version": "2.0.0",
    "tasks": [
        {
            "label": "gcc build",
            "type": "shell",
            "command": "g++",
            "args":[
                "-g", "Hello.cpp"
            ],
            "group": {
                "kind": "build",
                "isDefault": true
            },
            "problemMatcher": [
                "$gcc"
            ]
        }
    ]
}
```

Debugger Launch

```
{
    // Use IntelliSense to learn about possible attributes.
    // Hover to view descriptions of existing attributes.
    // For more information, visit: https://go.microsoft.com/fwlir
    "version": "0.2.0",
    "configurations": [
        {
            "name": "(gdb) Launch",
            "type": "cppdbg",
            "request": "launch",
            "program": "${workspaceFolder}/a.out",
            "args": [],
            "stopAtEntry": false,
            "cwd": "${workspaceFolder}",
            "environment": [],
            "externalConsole": false,
            "MIMode": "gdb",
            "miDebuggerPath": "/usr/bin/gdb",
            "setupCommands": [
                {
                    "description": "Enable pretty-printing for gdb
                    "text": "-enable-pretty-printing",
                    "ignoreFailures": true
                }
            ]
        }
    ]
}
```

Add Configuration...

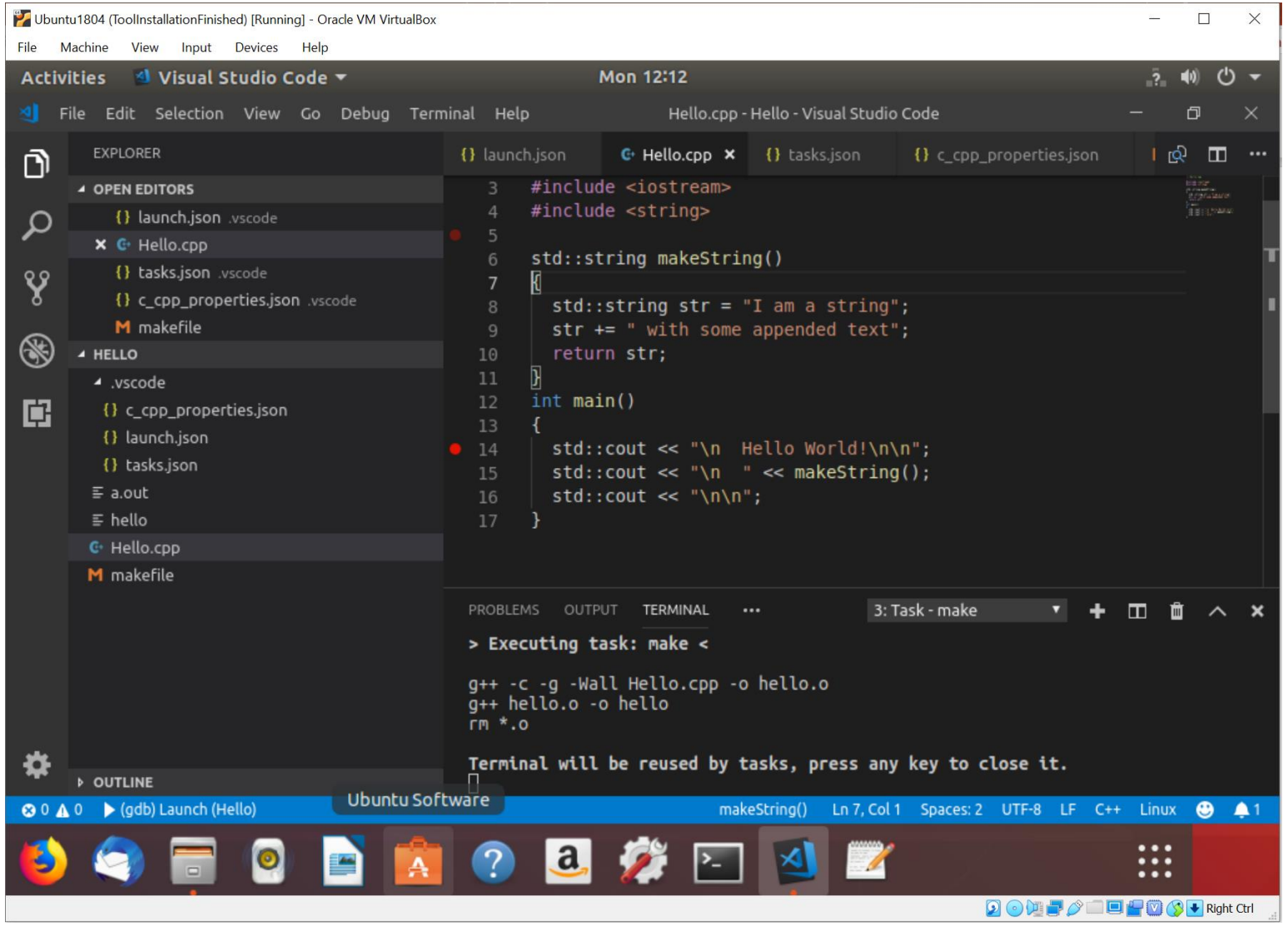
Makefile Example

```
// Hello.cpp

#include <iostream>
#include <string>

std::string makeString()
{
  std::string str = "I am a string";
  str += " with some appended text";
  return str;
}
int main()
{
  std::cout << "\n  Hello World!\n\n";
  std::cout << "\n  " << makeString();
  std::cout << "\n\n";
}
```

```
#-------------------------------------------------------
# makefile - demo for Project #1
#
# Jim Fawcett, CSE775 - Distributed Objects, Spring 2019
#-------------------------------------------------------
# Notes:
#   - Indentations must be a single Tab (not spaces)
#   - Dependencies are not indented
#   - commands are indented with single tab
#-------------------------------------------------------

all: hello clean

# link hello.o to create executable hello
# you may add additional object files as needed

hello: hello.o
        g++ hello.o -o hello

# compile Hello.cpp to create object file hello.o
# you may add additional source files as needed

hello.o: Hello.cpp
        g++ -c -g -Wall Hello.cpp -o hello.o

# remove object files
# only called if cited in all: directive

clean:
        rm *.o
```

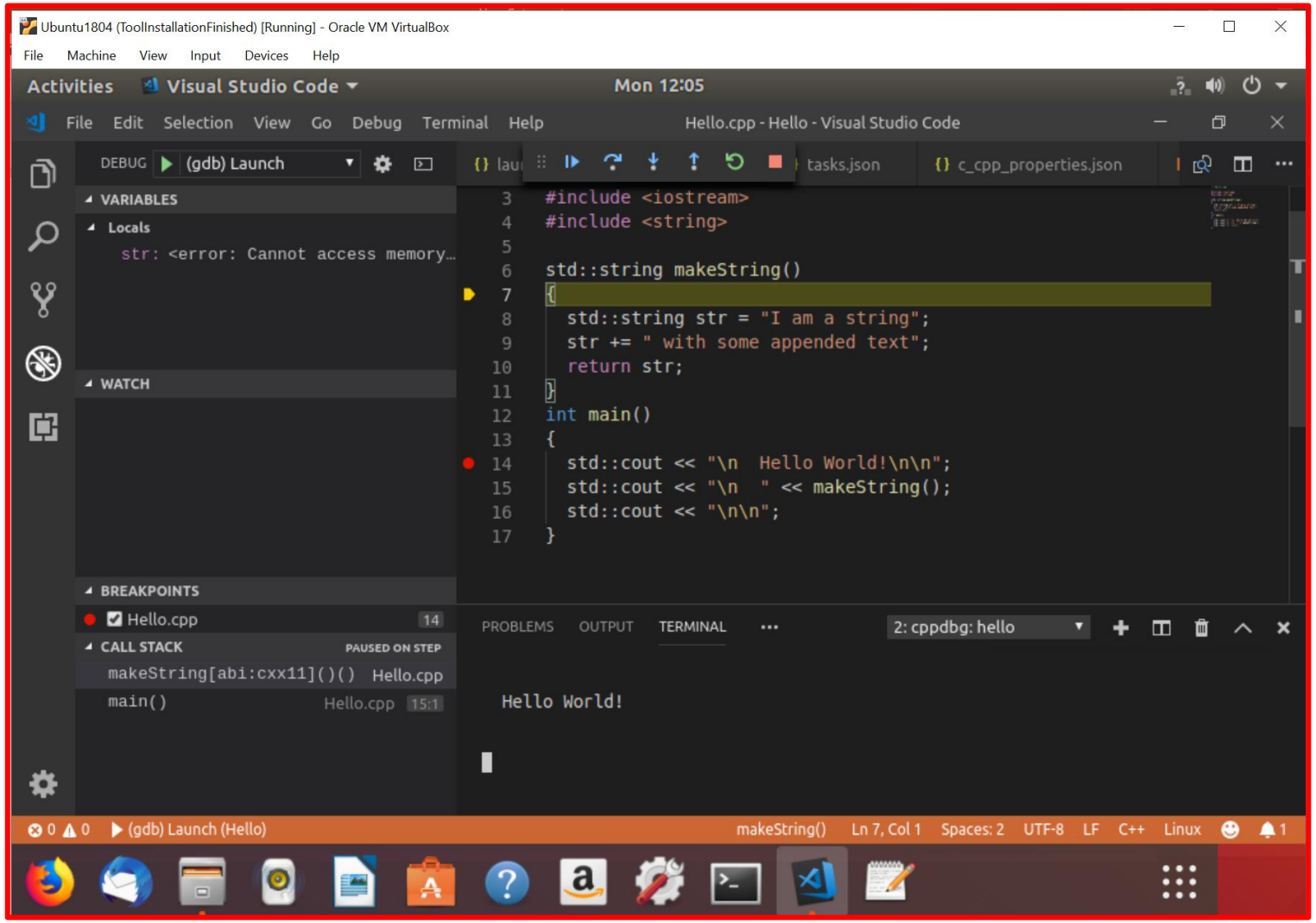
Running make

- Terminal > Run Task > make
- Builds application as specified by makefile



Debugging hello: Debug > Start Debugging

- Note breakpoint
- F10 => single step
- F5 => go to next breakpoint
- Note call stack
- Note Terminal



Build and Launch JSON

```
{
    // See https://go.microsoft.com/fwlink/?LinkId=733558
    // for the documentation about the tasks.json format
    "version": "2.0.0",
    "tasks": [
        {
            "label": "gcc build",
            "type": "shell",
            "command": "g++",
            "args":[
                "-g", "Hello.cpp"
            ],
            "group": {
                "kind": "build",
                "isDefault": true
            },
            "problemMatcher": [
                "$gcc"
            ]
        },
        {
            "label": "make",
            "type": "shell",
            "command": "make",
            "args":[

            ],
            "group": {
                "kind": "build",
                "isDefault": true
            },
            "problemMatcher": [
                "$gcc"
            ]
        }
    ]
}
```

gcc build

make build

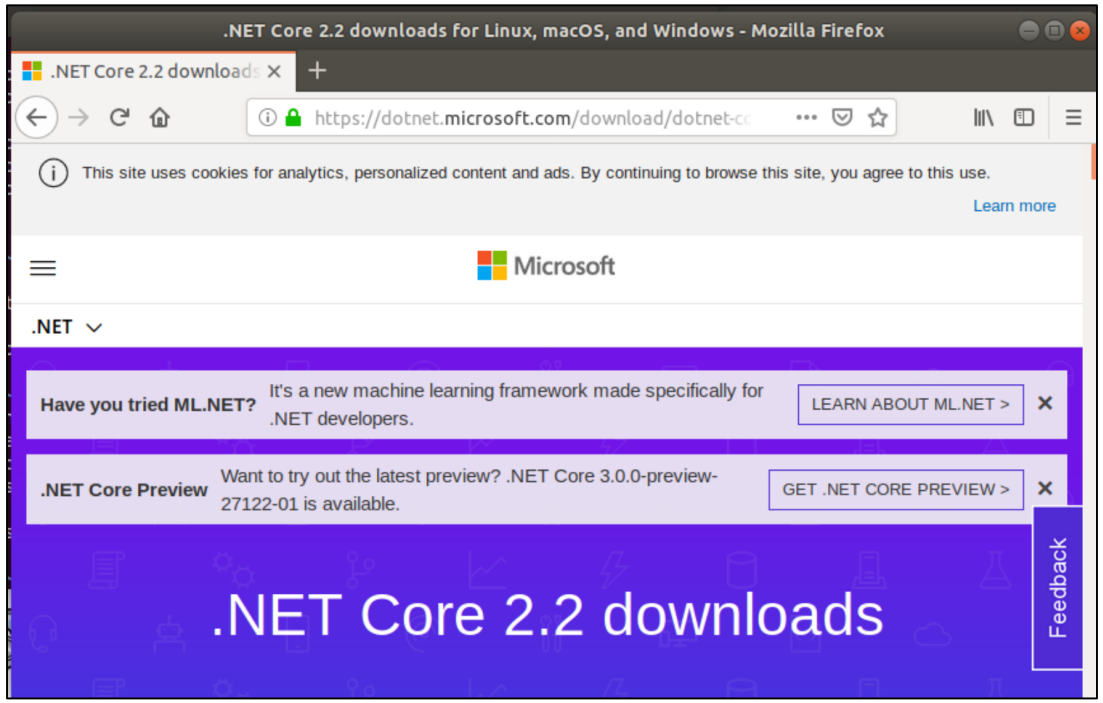
```
{
    // Use IntelliSense to learn about possible attributes.
    // Hover to view descriptions of existing attributes.
    // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
    "version": "0.2.0",
    "configurations": [
        {
            "name": "(gdb) Launch",
            "type": "cppdbg",
            "request": "launch",
            "program": "${workspaceFolder}/hello",
            "args": [],
            "stopAtEntry": false,
            "cwd": "${workspaceFolder}",
            "environment": [],
            "externalConsole": false,
            "MIMode": "gdb",
            "miDebuggerPath": "/usr/bin/gdb",
            "setupCommands": [
                {
                    "outfiles": true,
                    "description": "Enable pretty-printing for gdb",
                    "text": "-enable-pretty-printing",
                    "ignoreFailures": true
                }
            ]
        }
    ]
}
```

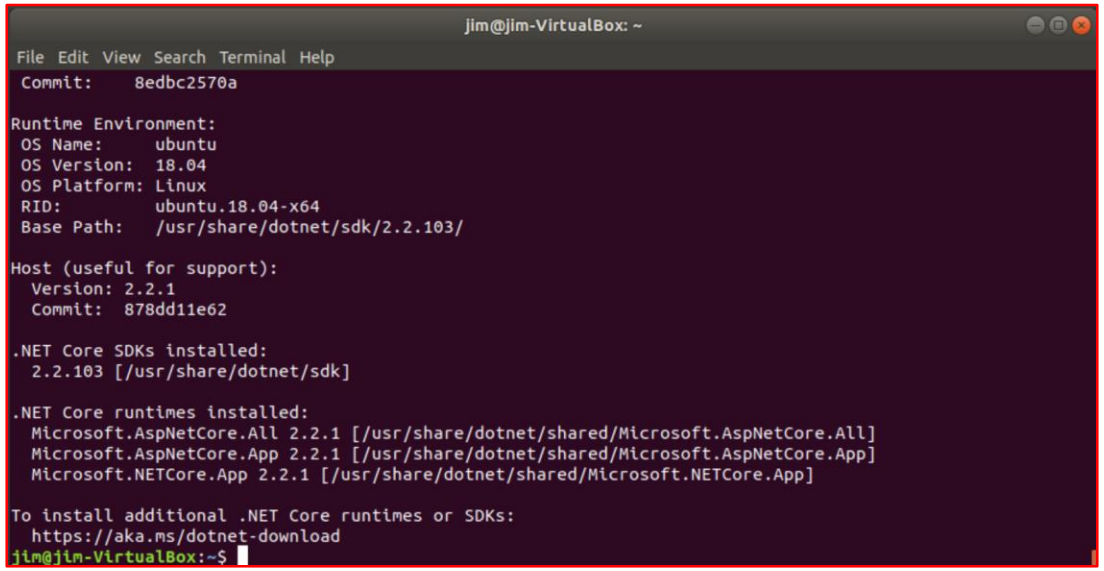
Launch gdb degugger

Task #5 – Optional Installs

Download Asp.Net Core 2.2 (optional)

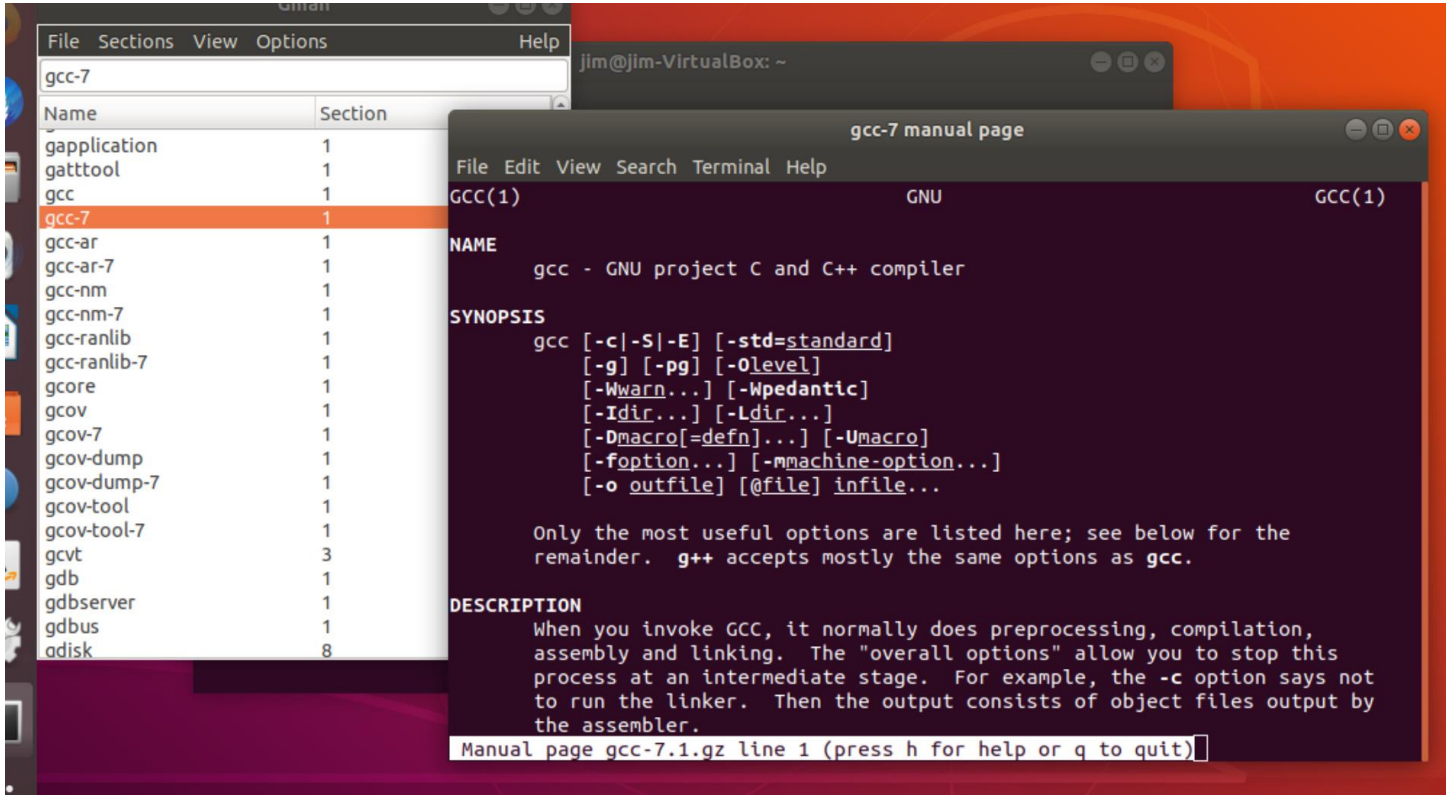
https://dotnet.microsoft.com/download/linux-package-manager/ubuntu18-04/sdk-2.2.102
Several detailed command line invocations are needed.



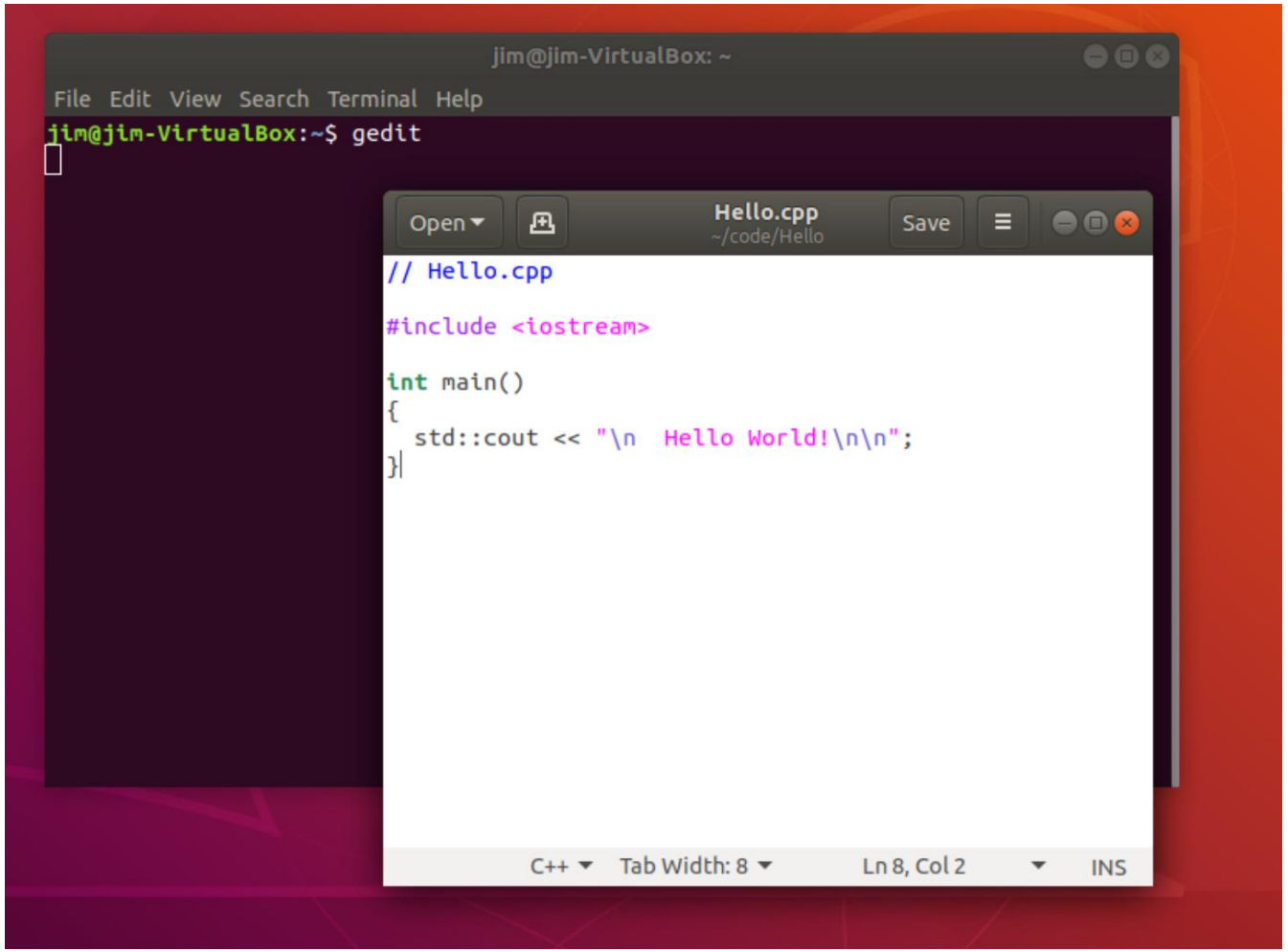


Gman – man page helper

- Install gman
  - Sudo apt install gman



gedit – installed with Ubuntu



That’s All Folks!