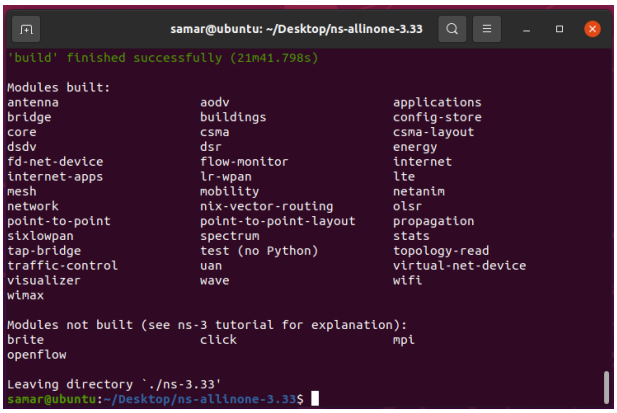
ns-3 Installation

- The ns-3 simulator is a discrete-event network simulator.
- ns-3 is designed as a set of libraries that can be combined together and also with other external software libraries. While some simulation platforms provide users with a single, integrated graphical user interface environment in which all tasks are carried out, ns-3 is more modular in this regard. Several external animators and data analysis and visualization tools can be used with ns-3. However, users should expect to work at the command line and with C++ and/or Python software development tools.
- Website and Documentation
    - The main web site is located at https://www.nsnam.org and provides access to basic information about the ns-3 system. Detailed documentation is available through the main web site at https://www.nsnam.org/documentation/.
- Installation page https://www.nsnam.org/wiki/Installation
- ***Tools***
    - VMware
    - Ubuntu (highly compatible with ns-3)
      Use the versions uploaded to this link
      https://drive.google.com/drive/folders/1Gd2iAza_oncTJZYuI7gR_WwKQrRpuuDY?usp=share_link
    - ns-3
    - Any C++/Python IDE to be used as an editor (VS Code or Eclipse)
- Ubuntu Installation into VMware, (Recommended 4GB RAM and 25GB Hard Disk)
    - https://www.youtube.com/watch?v=9rUhGWijf9U

- ***ns-3 download and building:***
  - Open a new terminal window and run the following commands (Do Not copy the $):

```
$ cd Desktop
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt install build-essential
$ sudo apt-get install gcc g++ python python-dev mercurial bzr gdb valgrind gsl-bin libgsl0-dev libgsl23
$ sudo apt-get install flex bison tcpdump sqlite sqlite3 libsqlite3-dev libxml2 libxml2-dev
$ sudo apt-get install libgtk2.0-0 libgtk2.0-dev uncrustify doxygen graphviz imagemagick texlive texlive-latex-extra texlive-xetex
$ sudo apt-get install texinfo dia texlive texlive-latex-extra texlive-extra-utils texi2html
$ sudo apt-get install gir1.2-goocanvas-2.0 python-gi python-gi-cairo python3-gi python3-gi-cairo gir1.2-gtk-3.0 python3-pygraphviz python3-pygccxml
$ sudo apt-get install openmpi-bin openmpi-common openmpi-doc libopenmpi-dev
$ sudo apt-get install dpdk dpdk-dev libdpdk-dev dpdk-igb-uio-dkms
$ sudo apt install -y python3-pip
$ pip3 install ipython
$ pip3 install kiwi
$ wget https://www.nsnam.org/release/ns-allinone-3.33.tar.bz2
$ tar xjf ns-allinone-3.33.tar.bz2
$ cd ns-allinone-3.33
$ ./build.py --enable-examples --enable-tests
```



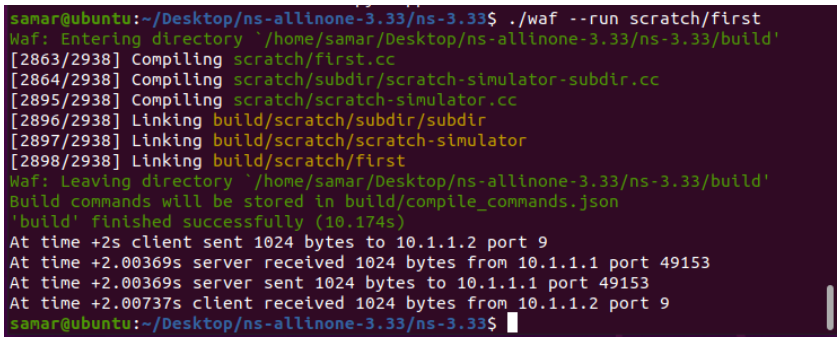
$ cd ns-3.33

$ ./test.py or ./test.py -c core

$ ./waf -d debug --enable-examples --enable-tests configure

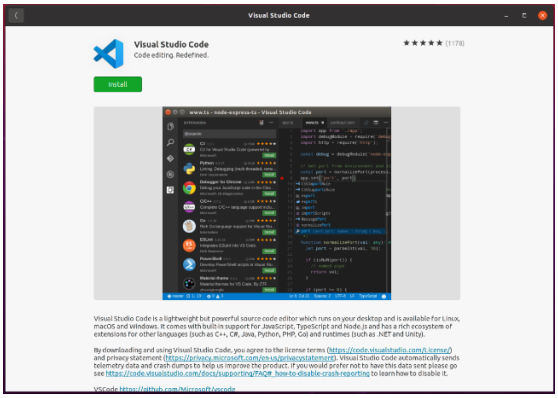
(recommended not required)

- ***Run first script:***
  - Open a new terminal window.
  - Go to the tutorial folder → $ cd Desktop/ns-allinone-3.33/ns-3.33/examples/tutorial
  - Copy the code file in scratch folder → $ cp first.cc ../../scratch/
  - $ cd ../..
  - ./waf --run scratch/first
  - The output should look like this:

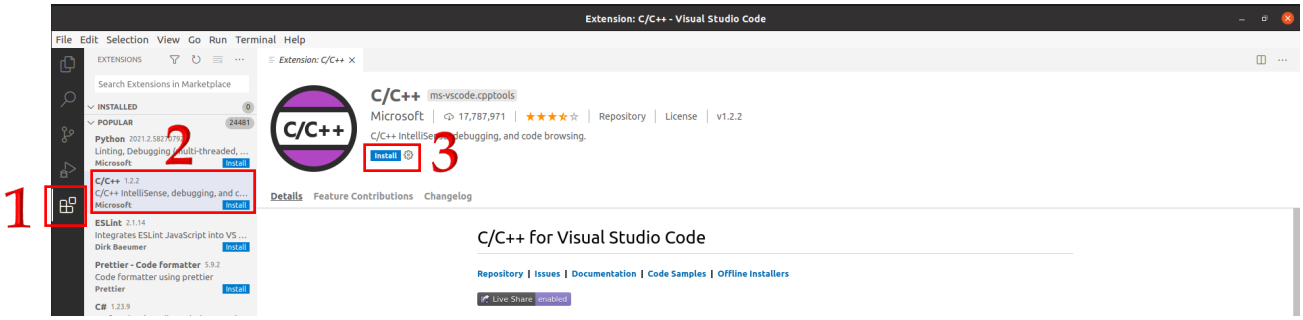


```
samar@ubuntu:~/Desktop/ns-allinone-3.33/ns-3.33$ ./waf --run scratch/first
Waf: Entering directory `/home/samar/Desktop/ns-allinone-3.33/ns-3.33/build'
[2863/2938] Compiling scratch/first.cc
[2864/2938] Compiling scratch/subdir/scratch-simulator-subdir.cc
[2895/2938] Compiling scratch/scratch-simulator.cc
[2896/2938] Linking build/scratch/subdir/subdir
[2897/2938] Linking build/scratch/scratch-simulator
[2898/2938] Linking build/scratch/first
Waf: Leaving directory `/home/samar/Desktop/ns-allinone-3.33/ns-3.33/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (10.174s)
At time +2s client sent 1024 bytes to 10.1.1.2 port 9
At time +2.00369s server received 1024 bytes from 10.1.1.1 port 49153
At time +2.00369s server sent 1024 bytes to 10.1.1.1 port 49153
At time +2.00737s client received 1024 bytes from 10.1.1.2 port 9
samar@ubuntu:~/Desktop/ns-allinone-3.33/ns-3.33$
```

- ***netanim Installation***
  - Open a new terminal window.
  - $ sudo apt-get install qt5-default mercurial
  - $ cd Desktop/ns-allinone-3.33/netanim-3.108
  - $ make clean (make: *** No rule to make target 'clean'. Stop. → this output is not an error it means that the make is already clean)
  - $ qmake NetAnim.pro
  - $ make
  - Open the NetAnim application → $ ./NetAnim
- ***VS Code Installation***
  - Open Ubuntu Software → Development → Visual Studio Code → Install.



- Open VS code and install C/C++ Microsoft Extension.



- Open a new terminal window.
- $ cd Desktop/ns-allinone-3.33/ns-3.33
- $ code .
- Ctrl+Shift+P → Edit C/C++: Edit Configuration (UI)
- Include the following paths:

Include path

An include path is a folder that contains header files (such as `#include "myHeaderFile.h"`) that are included in a source file. Specify a list of paths for the IntelliSense engine to use while searching for included header files. Searching on these paths is not recursive. Specify `**` to indicate recursive search. For example, `${workspaceFolder}/**` will search through all subdirectories while `${workspaceFolder}` will not. If on Windows with Visual Studio installed, or if a compiler is specified in the `compilerPath` setting, it is not necessary to list the system include paths in this list.

One include path per line.

```
${workspaceFolder}/build/ns3/**
/home/samar/Desktop/ns-allinone-3.33/ns-3.33/build
```