

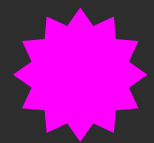

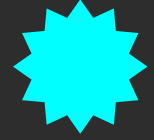
# UEFI & EDK II Training

Build Setup and Download EDK II Lab - Linux

[tianocore.org](https://tianocore.org)

Copy and Paste see [Lab Guide.md](#)

# PLATFORM BUILD LABS

-  Setup and Install Applications
-  Download the EDK II Source
-  Build BaseTools

# Install Applications for EDK II

Setup build environment

# Pre-requisites Ubuntu 20.04

## Example Ubuntu 20.04

The following need to be accessible for building Edk2, From the terminal prompt (Ctrl-Alt-T):

```
bash$ sudo apt install build-essential uuid-dev iasl git python-is-python3
```

- build-essential - Informational list of build-essential packages
- uuid-dev - Universally Unique ID library (headers and static libraries)
- iasl - Intel ASL compiler/decompiler (also provided by acpica-tools)
- git - support for git revision control system
- nasm - General-purpose x86 assembler (see below to install 2.15.05 or above)
- python-is-python3 - Ubuntu 20.04 python command is 'python3' but edk2 tools use 'python'

Now need Nasm 2.15.x <https://www.linuxfromscratch.org/blfs/view/cvs/general/nasm.html>

The Following is needed for Mounting a .VHD file to use with Simics

```
bash$ sudo apt install libguestfs-tools
```

The following will install the QEMU for Intel X86 & 64 bit

```
bash$ sudo apt install qemu-system-x86-64
```



ubuntu

**See Lab guide for Ubuntu 16.04 pre-requisites**

# Pre-requisites Clear Linux\* Project

## Example Using Clear Linux\* Project

The following need to be accessible for building Edk2, From the terminal prompt (Ctrl-Alt-T):

```
bash$ sudo swupd bundle-add devpkg-util-linux
```

Devpkg-util-linux – includes bundles for developer tools for writing “C” Applications included:  
gcc, nasm, uuid, etc.

```
bash$ sudo swupd bundle-add kvm-host
```

**Qemu – Emulation with Intel architecture with UEFI Shell**



# Setup Simics Environment

Download and Install the Simics Simulator  
(both Package Manager & Simics-6  
Packages) [Simics® Simulator Public Release](#)

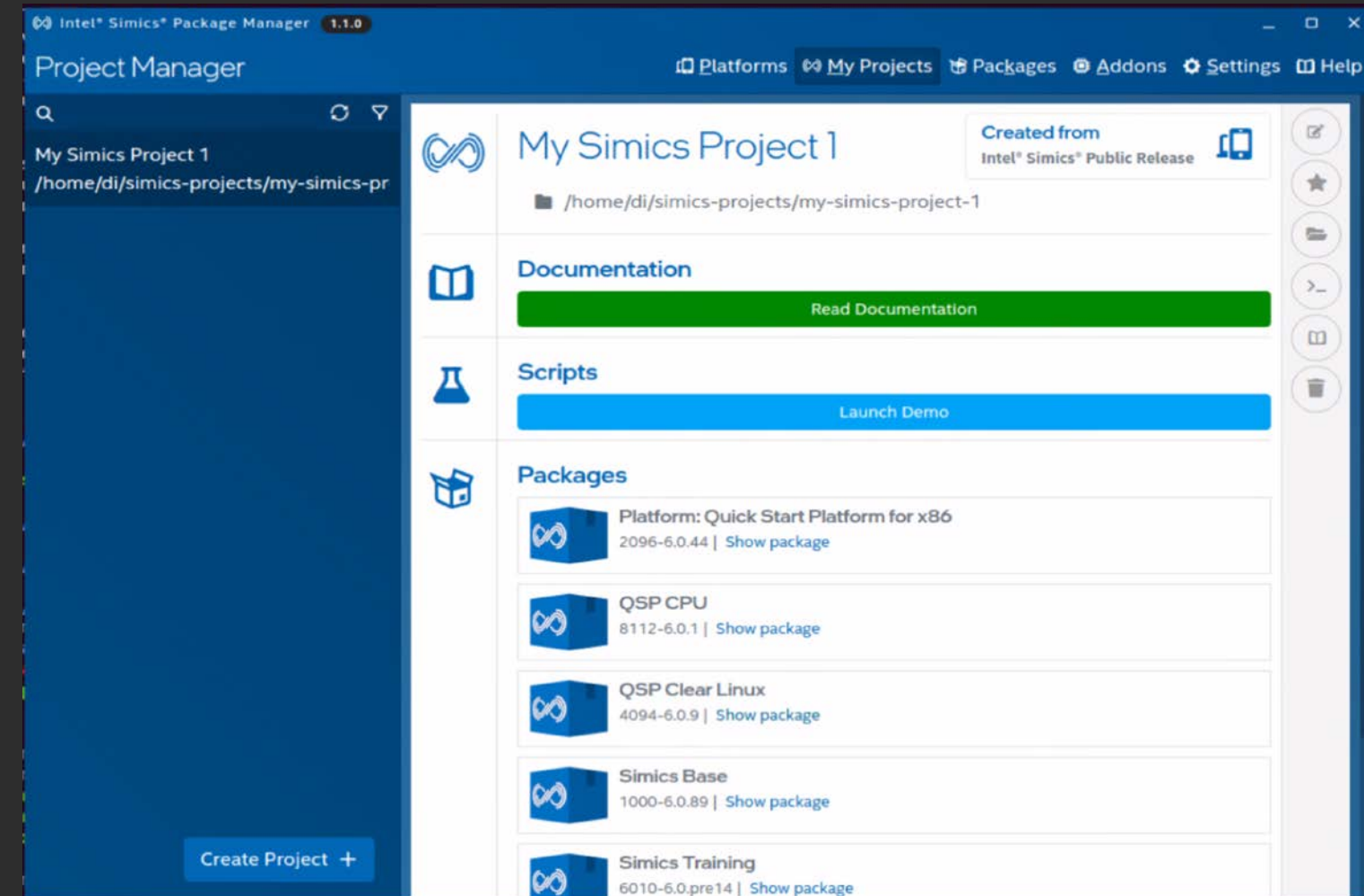
Setup the Simics Simulator

[Simics® Simulator Installation and Get Started](#)

Open the “Intel Simics Package Manager”

1. On Linux, run the ispm-gui application from inside the unpacked directory.
2. Note If the application fails to start on older Linux versions, you may have to start the application with the --no-sandbox option, as ispm-gui --no-sandbox.

Here is a snapshot of Intel Simics Package Manager with “My Simics Project 1” created





# Create QEMU Run Script

1. Create a run-ovmf directory under the home directory

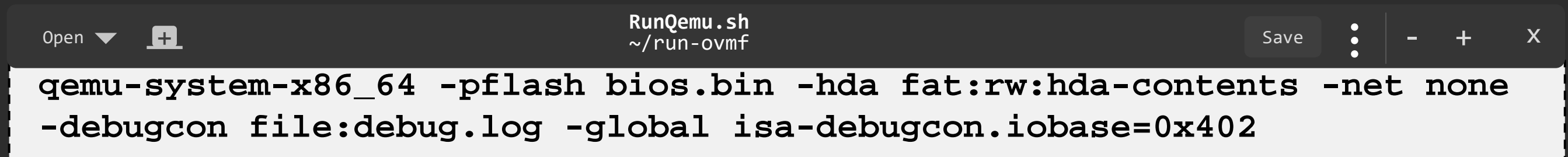
```
$ cd ~  
$ mkdir ~run-ovmf  
$ cd run-ovmf
```

2. Create a directory to use as a hard disk image

```
$ mkdir hda-contents
```

3. Create a Linux shell script to run the QEMU from the run-ovmf directory

```
$ gedit RunQemu.sh
```



The screenshot shows a text editor window titled "RunQemu.sh" with the path "~/run-ovmf". The window contains the following command:

```
qemu-system-x86_64 -pflash bios.bin -hda fat:rw:hda-contents -net none  
-debugcon file:debug.log -global isa-debugcon.iobase=0x402
```

4. Save and Exit

# Download EDK II Source

Download Source and Lab Materials



# Create Workspace Directory

Open Terminal Command Prompt

Make new directory for Workspace:

```
$ cd $HOME  
$ mkdir fw  
$ cd fw  
$ mkdir edk2-ws  
$ cd edk2-ws
```

# Download EDK II Open Source

Download the open source EDK II from GitHub 

**Note if behind a firewall, set PROXYS FIRST** (example shows for Intel Corp. – Maybe different for your Corp. or GEO)

```
$ git config --global https.proxy proxy-dmz.intel.com:912  
$ git config --global http.proxy proxy-dmz.intel.com:911
```

From the command prompt use “git clone” to download the following Repos

- Edk2 – main core code

```
$ git clone -b Edk2Lab_22Q3 https://github.com/tianocore-training/edk2.git
```

- EDK II “C” Library Repo

```
$ git clone https://github.com/tianocore/edk2-libc.git
```

- EDK II Platforms Repo

```
$ git clone https://github.com/tianocore/edk2-platforms.git
```

- EDK II Non-OSI (Stand alone Binaries)

```
$ git clone https://github.com/tianocore/edk2-non-osl.git
```

- Intel FSP

```
$ git clone https://github.com/Intel/FSP.git
```

# Download the EDK II Submodules

## Download the EDK II Submodules from GitHub

### Download the Submodules and Checkout the Lab Branch


```
$ ~/fw/edk2-ws> cd edk2  
$ ~/fw/edk2-ws/edk2> git submodule update --init  
$ ~/fw/edk2-ws> cd ..
```

### Download Checkout the Sha tag for edk2-platforms repo

```
$ ~/fw/edk2-ws> cd edk2-platforms  
$ ~/fw/edk2-ws> git reset --hard c546cc01f1517b42470f3ae44d67efcb8ee257fc
```

(reset to this commit since this is used with all the labs)

# DOWNLOAD LAB MATERIAL

Download the Lab\_Material\_FW.zip from :  [github.com](https://github.com/tianocore-training/Lab_Material_FW.zip)  
[Lab\\_Material\\_FW.zip](https://github.com/tianocore-training/Lab_Material_FW.zip)

OR

Use git clone to download the Lab\_Material\_FW

```
$ cd $HOME  
$ git clone https://github.com/tianocore-training/Lab_Material_FW.git
```

Directory Lab\_Material\_FW will be created

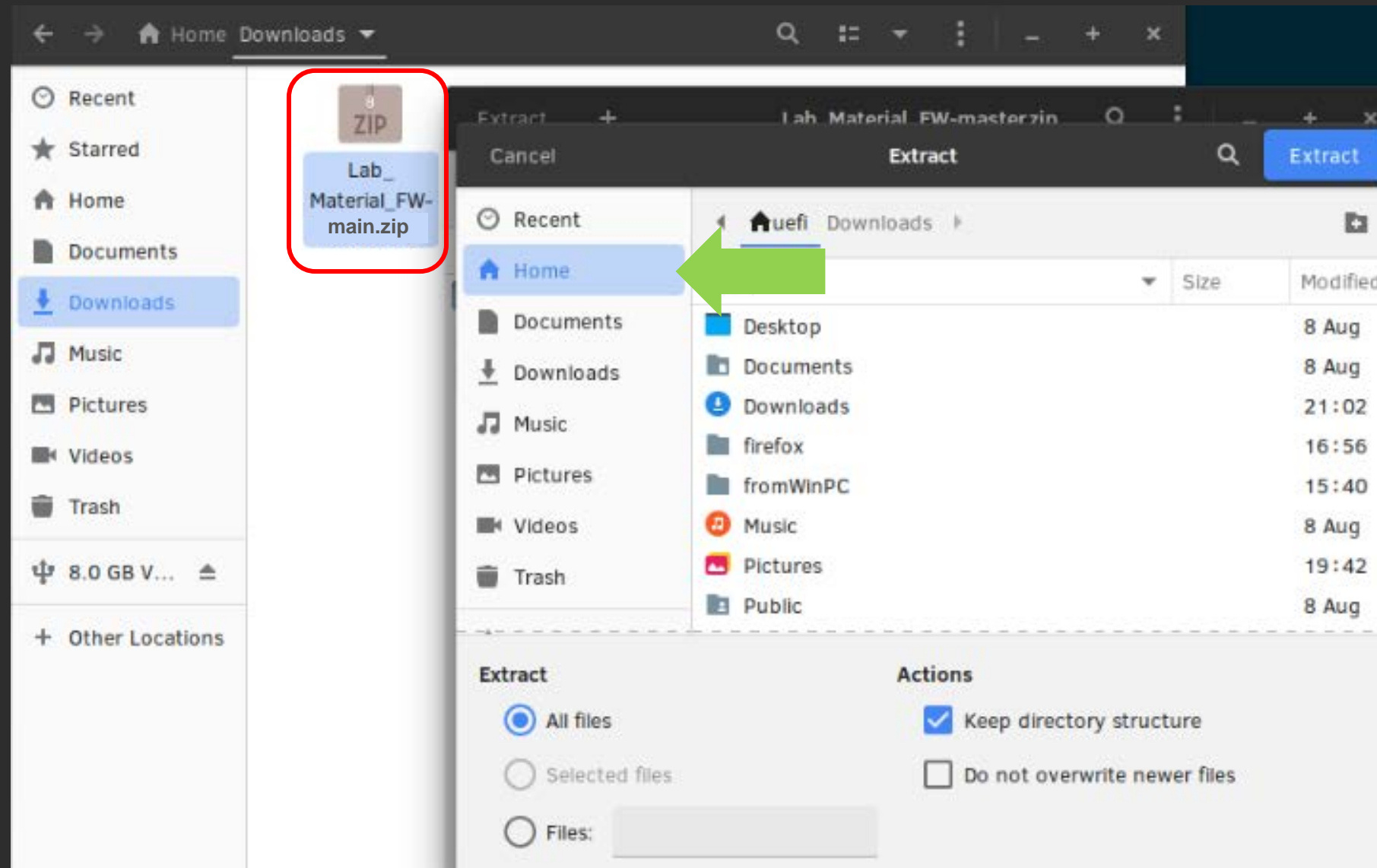
FW

- Documentation
- DriverWizard
- edk2-ws
- edk2Linux
- LabSampleCode

# BUILD EDK II OVMF

## -Extract the Source

1. Extract the Downloaded Lab\_Material\_FW-main.zip to Home (this will create a directory ~/FW )



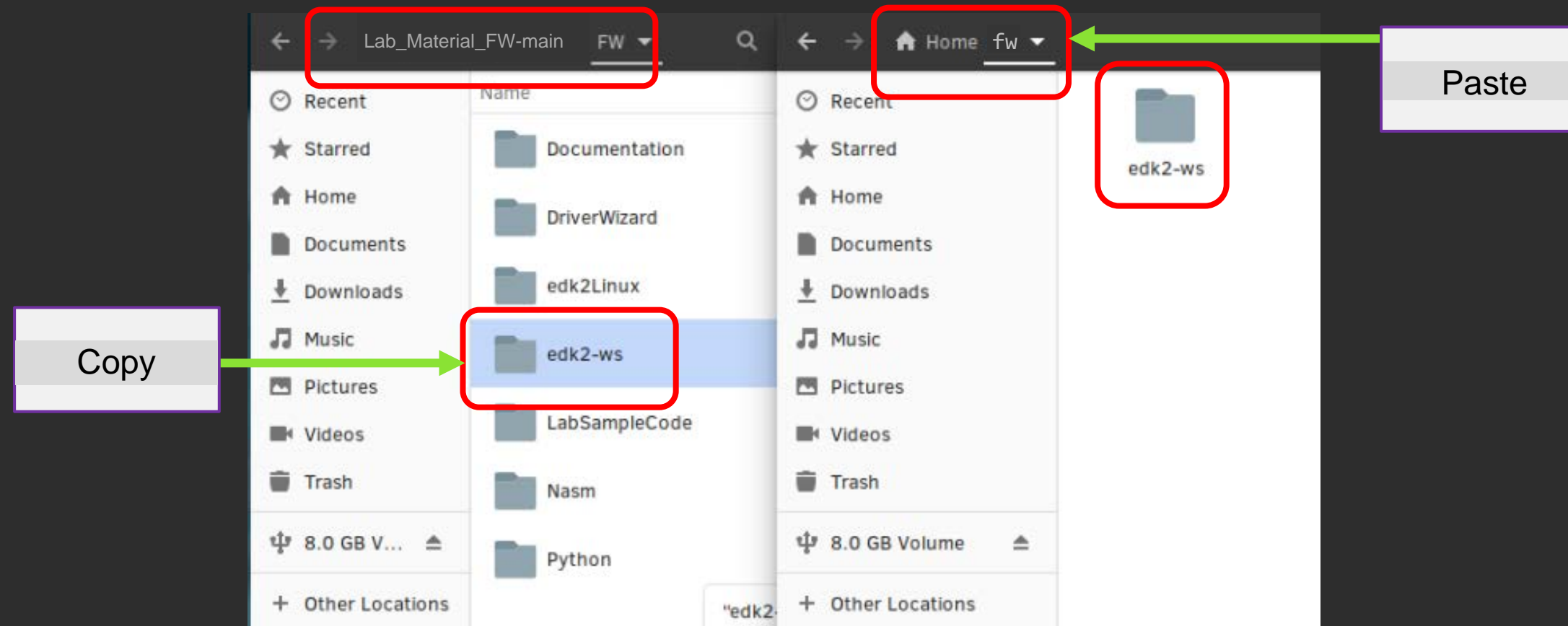
# BUILD EDK II OVMF

## - Copy the Source

2. Open a terminal prompt (Alt-Ctrl-T)
3. Create a working space source directory under the home directory

```
bash$ cd ~fw
```

4. From the FW folder, copy and paste folder “~Lab\_Material\_FW/edk2-ws” to ~/fw





# BUILD EDK II OVMF

## - Building BaseTools

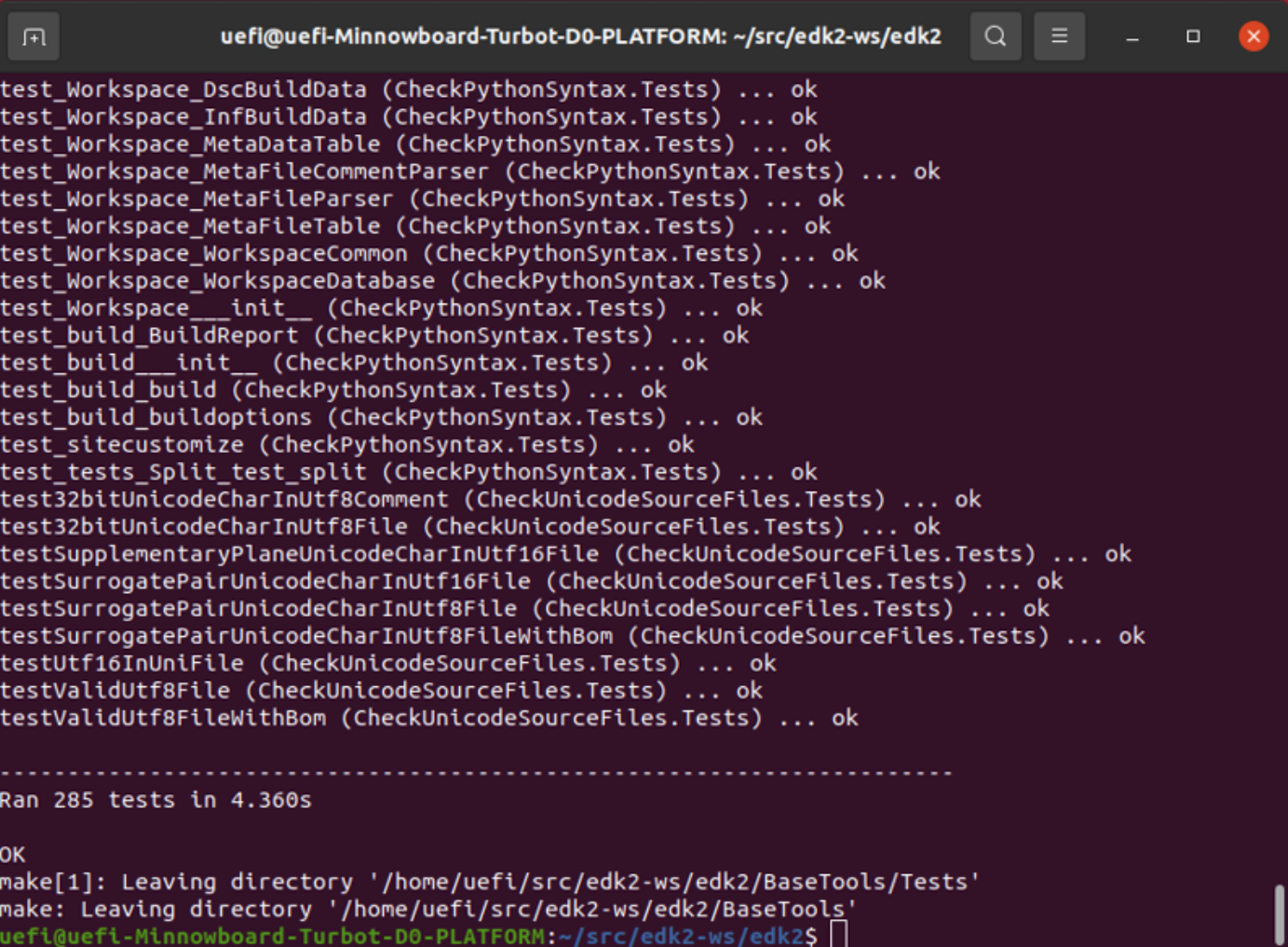
5. Export work space & platform path

```
$ cd ~fw/edk2-ws  
$ . setenv.sh  
$ export WORKSPACE=$PWD  
$ export PACKAGES_PATH=$WORKSPACE/edk2:$WORKSPACE/edk2-libc
```

6. Run Make

```
$ cd edk2  
$ make -C BaseTools/
```

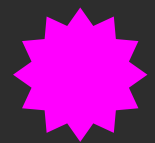

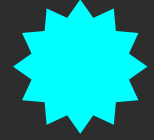
7. Make sure the tests pass OK



```
uefi@uefi-Minnowboard-Turbot-D0-PLATFORM: ~/src/edk2-ws/edk2  
test_Workspace_DscBuildData (CheckPythonSyntax.Tests) ... ok  
test_Workspace_InfBuildData (CheckPythonSyntax.Tests) ... ok  
test_Workspace_MetaDataTable (CheckPythonSyntax.Tests) ... ok  
test_Workspace_MetaFileCommentParser (CheckPythonSyntax.Tests) ... ok  
test_Workspace_MetaFileParser (CheckPythonSyntax.Tests) ... ok  
test_Workspace_MetaDataTable (CheckPythonSyntax.Tests) ... ok  
test_Workspace_WorkspaceCommon (CheckPythonSyntax.Tests) ... ok  
test_Workspace_WorkspaceDatabase (CheckPythonSyntax.Tests) ... ok  
test_Workspace__init__ (CheckPythonSyntax.Tests) ... ok  
test_build_BuildReport (CheckPythonSyntax.Tests) ... ok  
test_build__init__ (CheckPythonSyntax.Tests) ... ok  
test_build_build (CheckPythonSyntax.Tests) ... ok  
test_build_buildoptions (CheckPythonSyntax.Tests) ... ok  
test_sitecustomize (CheckPythonSyntax.Tests) ... ok  
test_tests_Split_test_split (CheckPythonSyntax.Tests) ... ok  
test32bitUnicodeCharInUtf8Comment (CheckUnicodeSourceFiles.Tests) ... ok  
test32bitUnicodeCharInUtf8File (CheckUnicodeSourceFiles.Tests) ... ok  
testSupplementaryPlaneUnicodeCharInUtf16File (CheckUnicodeSourceFiles.Tests) ... ok  
testSurrogatePairUnicodeCharInUtf16File (CheckUnicodeSourceFiles.Tests) ... ok  
testSurrogatePairUnicodeCharInUtf8File (CheckUnicodeSourceFiles.Tests) ... ok  
testSurrogatePairUnicodeCharInUtf8FileWithBom (CheckUnicodeSourceFiles.Tests) ... ok  
testUtf16InUniFile (CheckUnicodeSourceFiles.Tests) ... ok  
testValidUtf8File (CheckUnicodeSourceFiles.Tests) ... ok  
testValidUtf8FileWithBom (CheckUnicodeSourceFiles.Tests) ... ok  
  
-----  
Ran 285 tests in 4.360s  
  
OK  
make[1]: Leaving directory '/home/uefi/src/edk2-ws/edk2/BaseTools/Tests'  
make: Leaving directory '/home/uefi/src/edk2-ws/edk2/BaseTools'  
uefi@uefi-Minnowboard-Turbot-D0-PLATFORM:~/src/edk2-ws/edk2$
```



# SUMMARY

-  Setup / Install Applications
-  Download EDK II Source
-  Build BaseTools

# Questions?



# Return to Main Training Page



Return to Training Table of contents for next presentation [link](#)





# ACKNOWLEDGEMENTS

Redistribution and use in source (original document form) and 'compiled' forms (converted to PDF, epub, HTML and other formats) with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code (original document form) must retain the above copyright notice, this list of conditions and the following disclaimer as the first lines of this file unmodified.

Redistributions in compiled form (transformed to other DTDs, converted to PDF, epub, HTML and other formats) must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS DOCUMENTATION IS PROVIDED BY TIANOCORE PROJECT "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL TIANOCORE PROJECT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2021-2022, Intel Corporation. All rights reserved.





# Pre-requisites Ubuntu 16.04

Instructions from: [tianocore wiki Ubuntu\\_1610](#)

Example Ubuntu 16.04

The following need to be accessible for building Edk2, From the terminal prompt (Cnt-Alt-T):

```
bash$ sudo apt-get install build-essential uuid-dev iasl git gcc-5 python3-distutils
```

build-essential - Informational list of build-essential packages

uuid-dev - Universally Unique ID library (headers and static libraries)

iasl - Intel ASL compiler/decompiler (also provided by acpica-tools)

git - support for git revision control system

gcc-5 - GNU C compiler (v5.4.0 as of Ubuntu 16.04 LTS)

nasm - General-purpose x86 assembler (see below to install 2.15.05 or above)

python3 - distutils - distutils module from the Python standard library

Now need Nasm 2.15.x <https://www.linuxfromscratch.org/blfs/view/cvs/general/nasm.html>

The Following is needed for Mounting a .VHD file to use with Simics

```
bash$ sudo apt install libguestfs-tools
```

The following will install the QEMU for Intel X86 & 64 bit

```
bash$ sudo apt-get install qemu
```



ubuntu