

These steps are for Ubuntu18.04 LTS Desktop 64 bits. Follow the same steps for Ubuntu20.04, but just check if some dependencies are different from <https://github.com/intel/linux-sgx>.

Step1: SGX SDK Driver: <https://github.com/intel/linux-sgx-driver>

IMPORTANT: ENABLE “SGX” & DISABLE “SECURE BOOT” from BIOS

Check SGX status: <https://github.com/intel/sgx-software-enable>

Output: Intel SGX is already enabled on this system

```
sudo su
cd /opt/intel/    # create this directory if does not exist
```

Install SGX-driver ([intel/linux-sgx-driver: Intel SGX Linux* Driver](https://github.com/intel/linux-sgx-driver)) - Follow the README steps.

TURN OFF SECURE BOOT IN CASE OF ERROR “Operation Not permitted”.

If there are no errors in the above steps, proceed further.

Step2: SGX SDK: <https://github.com/intel/linux-sgx>

Build and install SGX-SDK & SGX-SDK-installer using the steps below.

```
sudo apt-get install build-essential ocaml ocamlbuild automake autoconf libtool
wget python libssl-dev git cmake perl
```

Note: To build Intel(R) SGX SDK, gcc version is required to be 7.3 or above and glibc version is required to be 2.27 or above.

```
gcc --version
gcc (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0
```

```
sudo apt-get install libssl-dev libcurl4-openssl-dev protobuf-compiler
libprotobuf-dev debhelper cmake reprepro unzip
```

```
git clone https://github.com/intel/linux-sgx.git
```

```
cd linux-sgx && make preparation
```

Final output:

```
/opt/intel/linux-sgx/external/dcap_source/QuoteGeneration /opt/intel/linux-sgx  
prebuilt_dcap_1.10.tar.gz: OK  
/opt/intel/linux-sgx
```

```
sudo cp external/toolset/ubuntu18.04{as,ld,ld.gold,objdump} /usr/local/bin
```

```
# change ubuntu18.04 to ubuntu20.04 or something else
```

```
which as ld ld.gold objdump # should print the paths
```

```
make sdk
```

```
make sdk_install_pkg
```

You can find the generated Intel(R) SGX SDK installer

sgx_linux_x64_sdk_\${version}.bin located under linux/installer/bin/, where
\${version} refers to the version number.

Generated sdk installer: ./linux/installer/bin/sgx_linux_x64_sdk_2.13.100.4.bin

```
sudo apt-get install build-essential python
```

```
cd linux/installer/bin
```

```
./sgx_linux_x64_sdk_2.13.100.4.bin # when prompted for directory, select NO and  
install in /opt/intel
```

```
source /opt/intel/sgxsdk/environment
```

```
# change the path above as per the terminal output in your case
```

Installation done.

```
cd SampleCode/LocalAttestation # You can also test SampleCode/SampleEnclave
```

```
make SGX_MODE=SIM
```

```
cd bin
```

```
./app
```

Output:

succeed to load enclaves.
succeed to establish secure channel.
Succeed to exchange secure message...
Succeed to close Session...

Testing of SIMULATION MODE done.

Step3: SGX PSW: Same as SDK

```
cd /opt/intel/linux-sgx # main repo directory
```

```
make psw
```

Installation can be done in 2 ways:

3.1 : Using Installer

```
make psw_install_pkg
```

```
cd linux/installer/bin
```

```
./sgx_linux_x64_psw_2.13.100.4.bin
```

Installation done. Go to testing part!

3. 2 : Using Local Repository

```
make deb_psw_pkg
```

```
make deb_local_repo
```

Output:

Local repository is successfully generated at

```
/opt/intel/linux-sgx/linux/installer/deb/local_repo_tool/../../sgx_debian_local_repo.
```

Please follow the instructions in README to use this repository.

Append

```
deb [trusted=yes arch=amd64]
```

```
file:/opt/intel/linux-sgx/linux/installer/deb/local_repo_tool/../../sgx_debian_lo  
cal_repo bionic main
```

at the end of `/etc/apt/sources.list`

`# change the above path to the terminal output in your specific case`

```
sudo apt update
sudo apt-get install libssl-dev libcurl4-openssl-dev libprotobuf-dev
sudo apt-get install libsgx-launch libsgx-urts
sudo apt-get install libsgx-epid libsgx-urts
sudo apt-get install libsgx-quote-ex libsgx-urts
sudo apt-get install libsgx-dcap-ql
```

Installation done. Test now!

```
cd SampleCode/SampleEnclave
```

```
make
```

```
./app
```

```
Checksum(0x0x7ffdac22c320, 100) = 0xfffd4143
```

```
Info: executing thread synchronization, please wait...
```

```
Info: SampleEnclave successfully returned.
```

```
Enter a character before exit ...
```

Testing of HARDWARE MODE done! *Take a break :)*

Additional note:

You will need to source the environment path when you **make** in SW mode.

```
source /opt/intel/sgx/sdk/environment
```