Yocto Build for Meta-Tegra Device.

Readme made by Deepak Kumar Beniya
Embedded Software Developer
deepakbeniya64@gmail.com

=> At First install all the required Packages.

```
sudo apt-get install gawk wget git git-core diffstat
unzip texinfo gcc-multilib build-essential \
chrpath socat cpio python python3 python3-pip
python3-pexpect \
python3-git python3-jinja2 libegl1-mesa pylint3
rsync bc bison \
xz-utils debianutils iputils-ping libsdl1.2-dev
xterm \
language-pack-en coreutils texi2html file
docbook-utils \
python-pysqlite2 help2man desktop-file-utils \
libgl1-mesa-dev libglu1-mesa-dev mercurial autoconf
automake \
groff curl lzop asciidoc u-boot-tools
libreoffice-writer \
sshpass ssh-askpass zip xz-utils kpartx vim screen
```

```
=> Follow The steps in your local PC to build YOCTO

=> Make sure that your system should have around
250GB of free     space.
```

1. $ git clone https://github.com/OE4T/tegra-demo-distro.git
2. $ cd tegra-demo-distro
3. $ git branch -a
4. $ git checkout remotes/origin/kirkstone
5. $ git submodule update --init
6. $ . ./setup-env --machine jetson-orin-nano-devkit --distro tegrademo build_tegra_demo
7. $ bitbake demo-image-full (it will create full images)

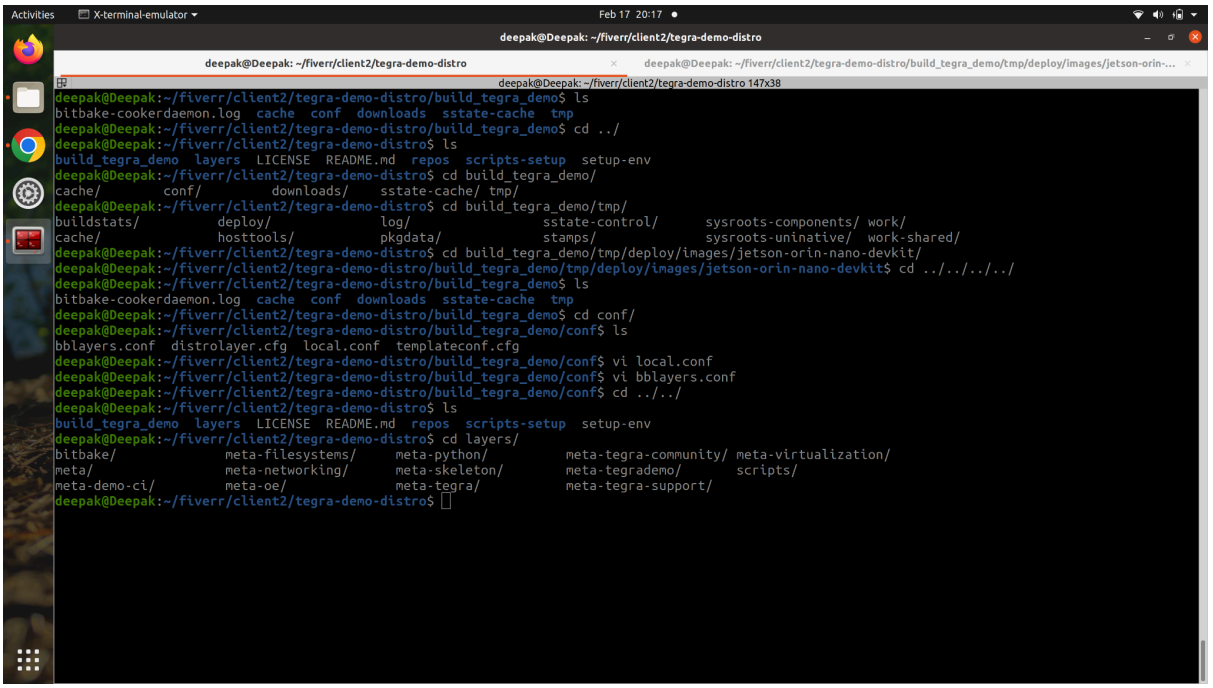
Targets for building Tegra demo images with test applications:
core-image-minimal - minimally bootable image (no demo apps, no graphics)
demo-image-base - basic image with no graphics
demo-image-egl - (basic image with DRM/EGL, no window manager)
demo-image-sato - X11 image with 'sato' UI
demo-image-weston - Wayland with Weston compositor
demo-image-full - (X11/sato UI plus docker, openCV, VPI, TensorRT and multimedia API samples)
=> After successfully compilation go to build_tegra_demo/tmp/deploy/images/jetson-orin-nano-devkit/
You will get all the images.

8. $ cd build_tegra_demo/tmp/deploy/images/jetson-orin-nano-devkit/
9. $ mkdir tegra_flash
10. $ tar -xvf demo-image-full-jetson-orin-nano-devkit.tegraflash.tar.gz -C tegra_flash/
11. $ cd tegra_flash

=> Now, you can flash the images through doflash.sh script
=> check this link for more information.

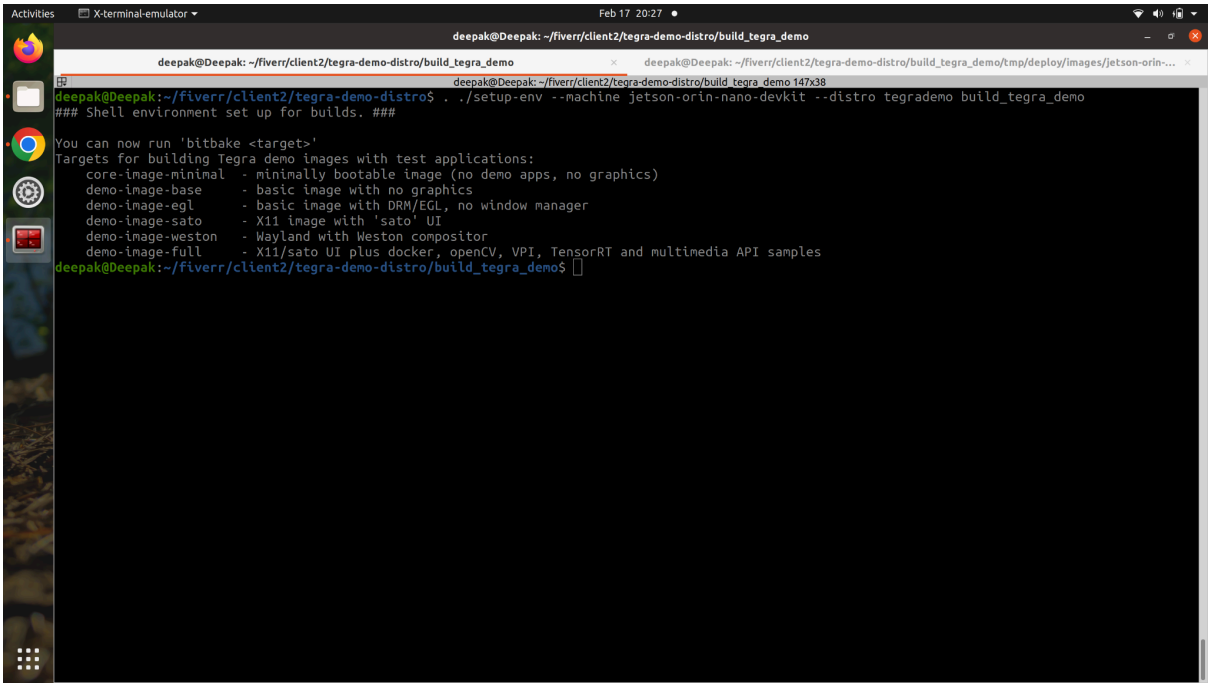
https://github.com/OE4T/meta-tegra/wiki/Flashing-the-Jetson-Dev-Kit

=> use different **bitbake demo-image-base - basic image with no graphics**
=> bitbake demo-image-egl - basic image with DRM/EGL, no window manager
=> bitbake demo-image-sato - X11 image with 'sato' UI
=> bitbake demo-image-weston - Wayland with Weston compositor
=> bitbake demo-image-full - X11/sato UI plus docker, openCV, VPI, TensorRT

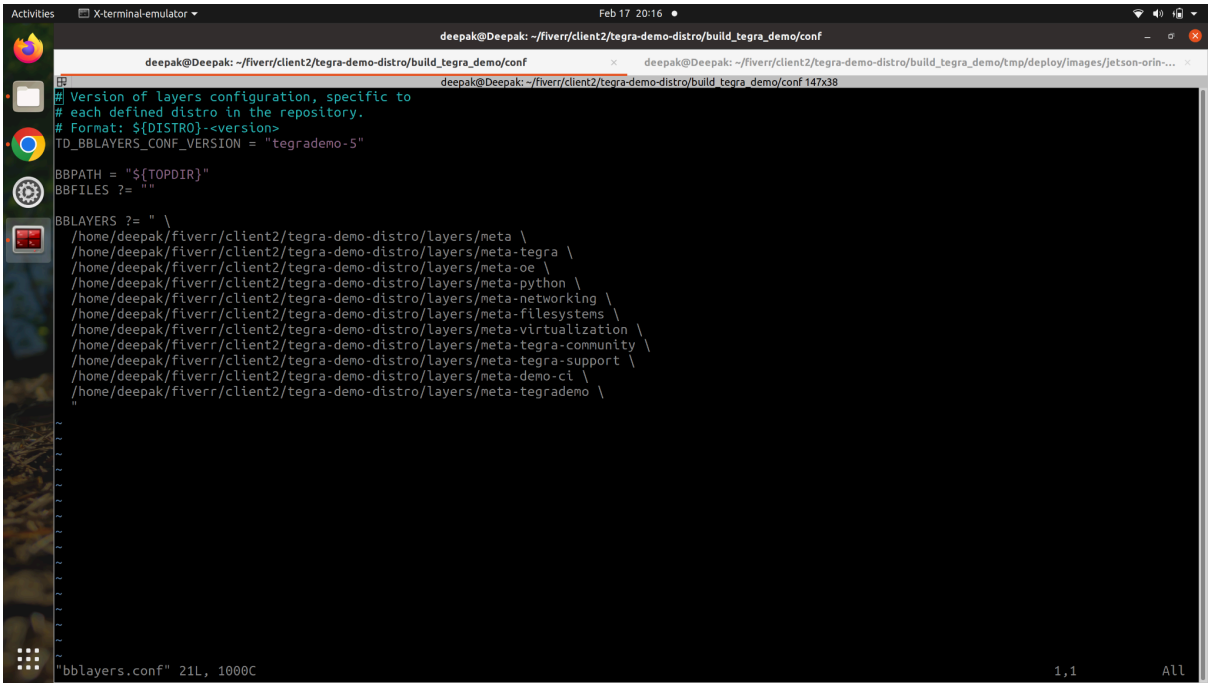


=> When ever you need to compile then this command should be executed at first so that the terminal will be ready . It’s basically called enableing toolchain.

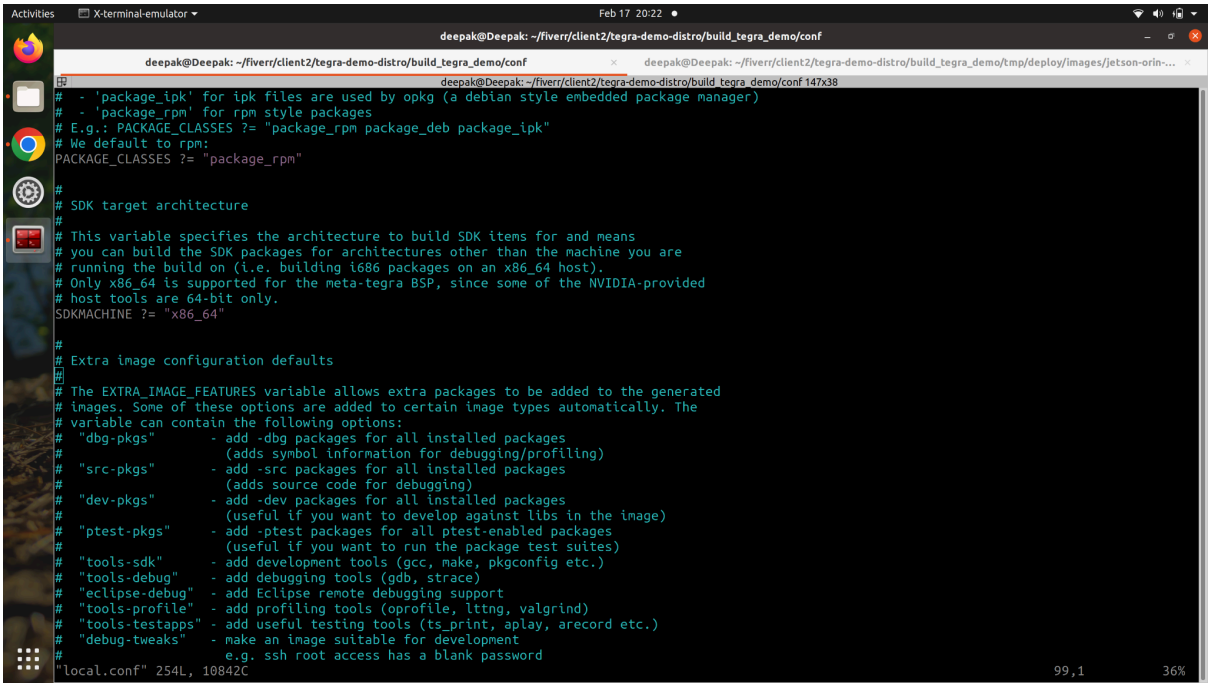
. ./setup-env --machine jetson-orin-nano-devkit --distro tegrademo build_tegra_demo



=> These are meta layers that add into this bblayers.conf file



=> If you want to add some packages like (vim, mqtt, tensor flow) then it can be added in this file bblayers.conf



=> for kernel modification you can modify here.