Steps to validate lightning refapp on RDK 4.0

- Overview
- Prerequisites
- Build instructions
  - Create Lightning refapp dev build
- Running refapp
  - Pre-build refapp code
  - Create dist (build)
- Hosting Lighting refapp
- Lighting refapp integration
  - Pre-build rdk-generic-hybrid-image-raspberrypi image
  - Build rdk-generic-hybrid-image-raspberrypi follow below instructions
  - Flash image on the SD card
- RefApp Video Playback
  - RDK 4.0 (rdk-generic-hybrid-image) Vs LGI RefApp (rdk-generic-hybrid-refapp-image)
- Dependencies
  - Lightning RefApp on RDK 4.0
  - Lightning RefApp on Operator reference image (rdk-generic-hybrid-refapp-image)
- Reference Video

Overview

To validate lightning refapp on rdk 4.0 (rdk-generic-hybrid-image-raspberrypi) along with video playback, need to do some necessary code customizations /modifications in lightning refapp side for video player implementation provided by lightning-sdk framework and rdk library side for variable setting and url configurations.

Please follow the below instructions to validate lightning refapp on rdk-generic-hybrid-image.

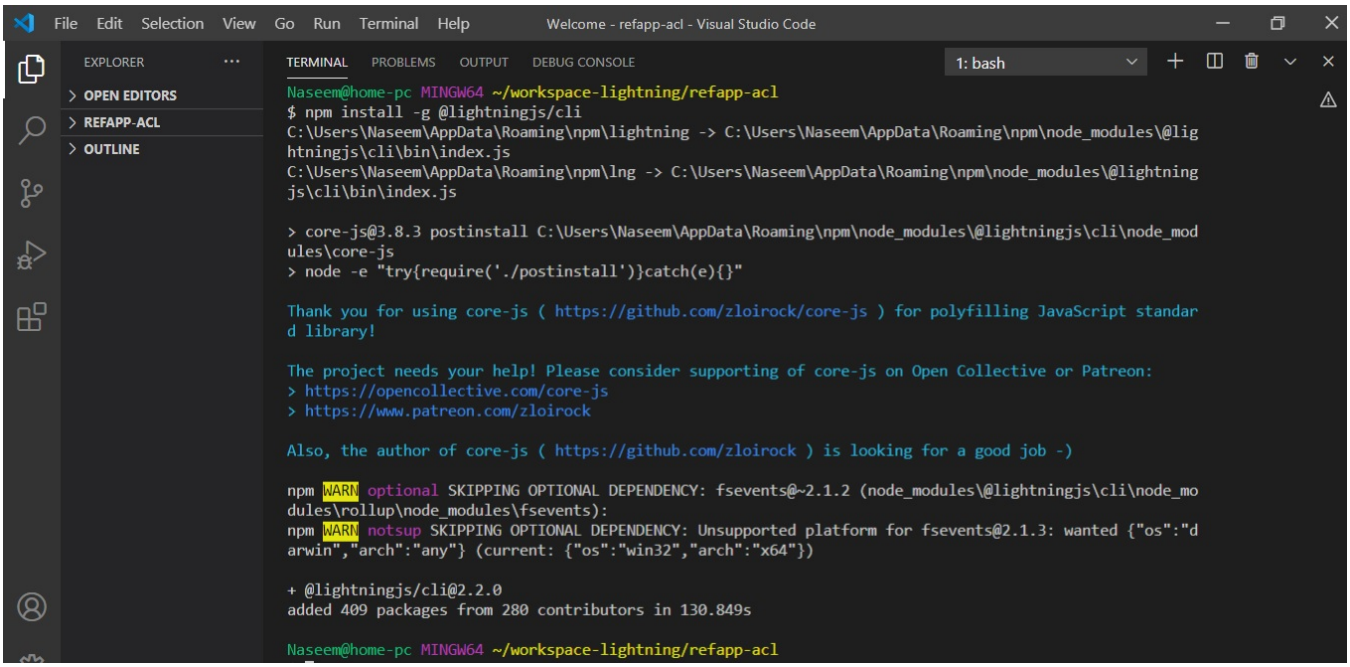
Prerequisites

Before you follow the steps below, make sure you have installed all the required prerequisites in your dev PC - Node.js , npm, Lightning-CLI

- Download and install VSCode IDE (code editor) - https://code.visualstudio.com/download
- Download and install Node.js and npm - https://nodejs.org/en/download/
- Install Lightning-CLI globally in your dev PC

```
$ npm install -g @lightningjs/cli
```

reference image for Lightning-CLI install



Build instructions

- **Create Lightning refapp dev build**
  1. pull code from source repo using below command

     ```
     $ git clone https://github.com/naseemshekh20/refapp.git
     ```

  2. goto refapp directory

     ```
     $ cd refapp
     ```

  3. checkout to branch refapp-vid-playback-validation-on-rdk-4.0

     ```
     $ git checkout refapp-vid-playback-validation-on-rdk-4.0
     ```

  4. install the NPM dependencies by running below command

     ```
     $ npm install
     ```

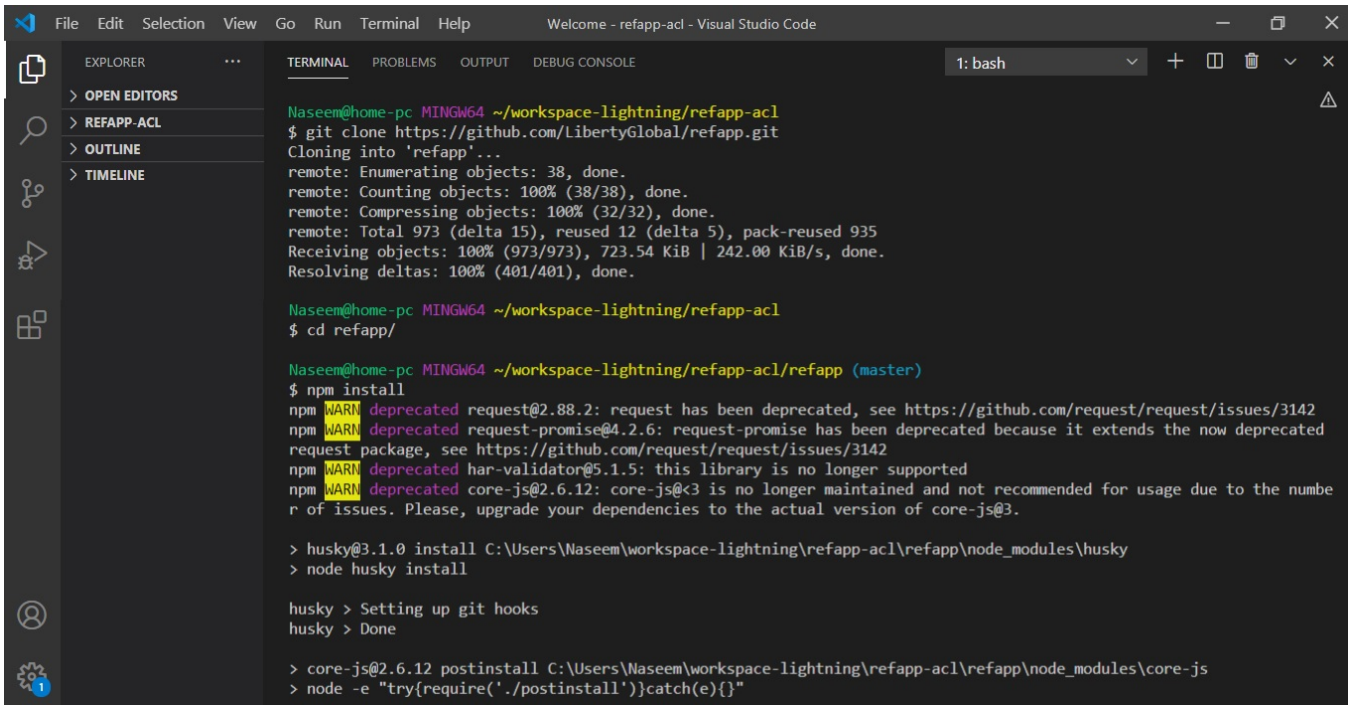
  5. run copyAssets.sh command to create static folder and copy images, json files etc

     ```
     $./copyAssets.sh
     ```

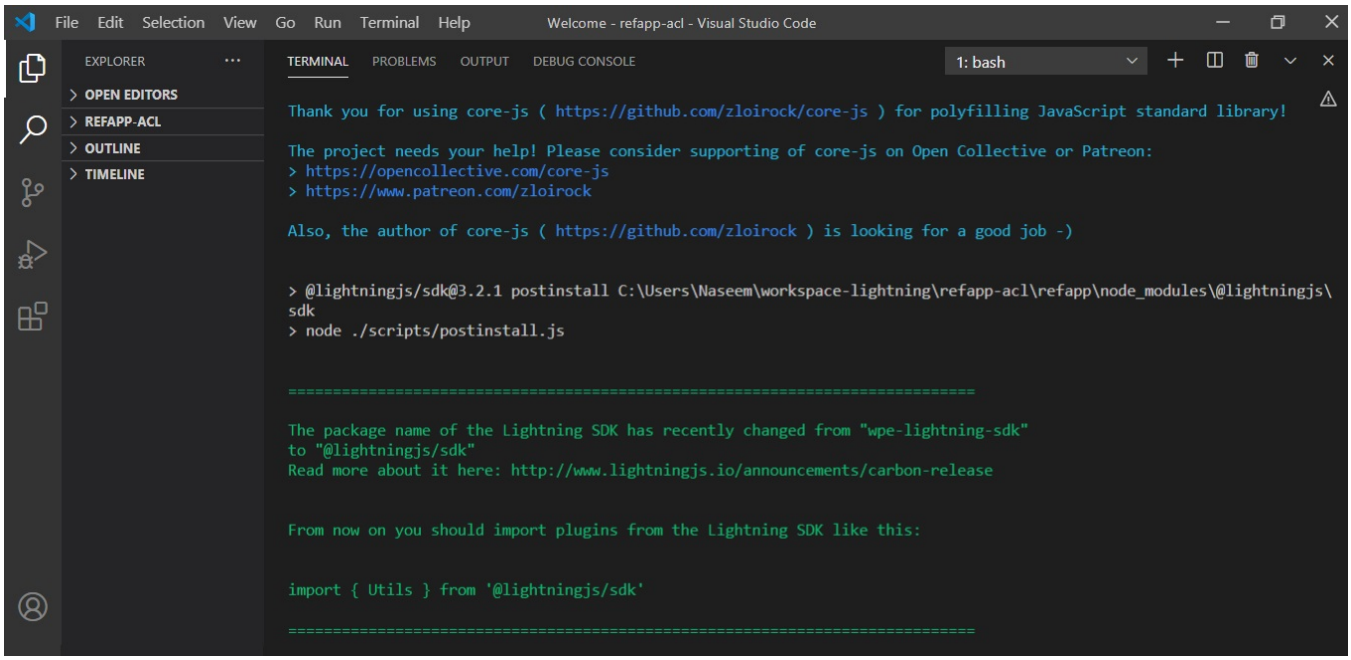
  6. run below command for testing in browser

     ```
     $ lng dev
     ```

refer below screenshots for reference



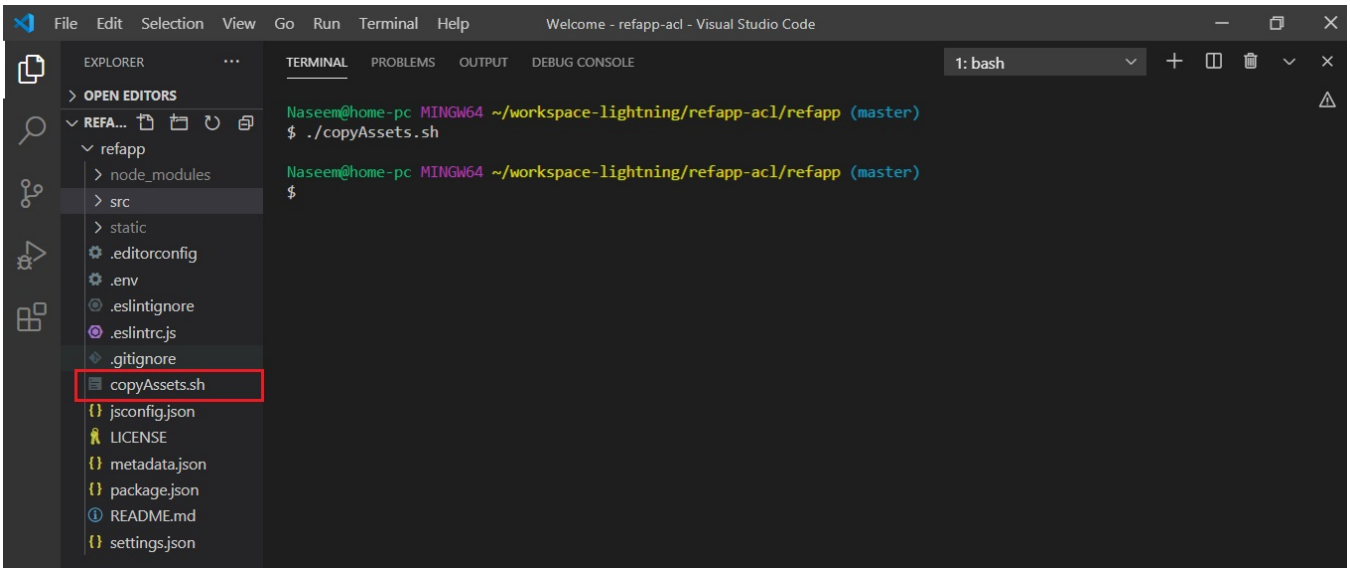
reference images for build instructions-1



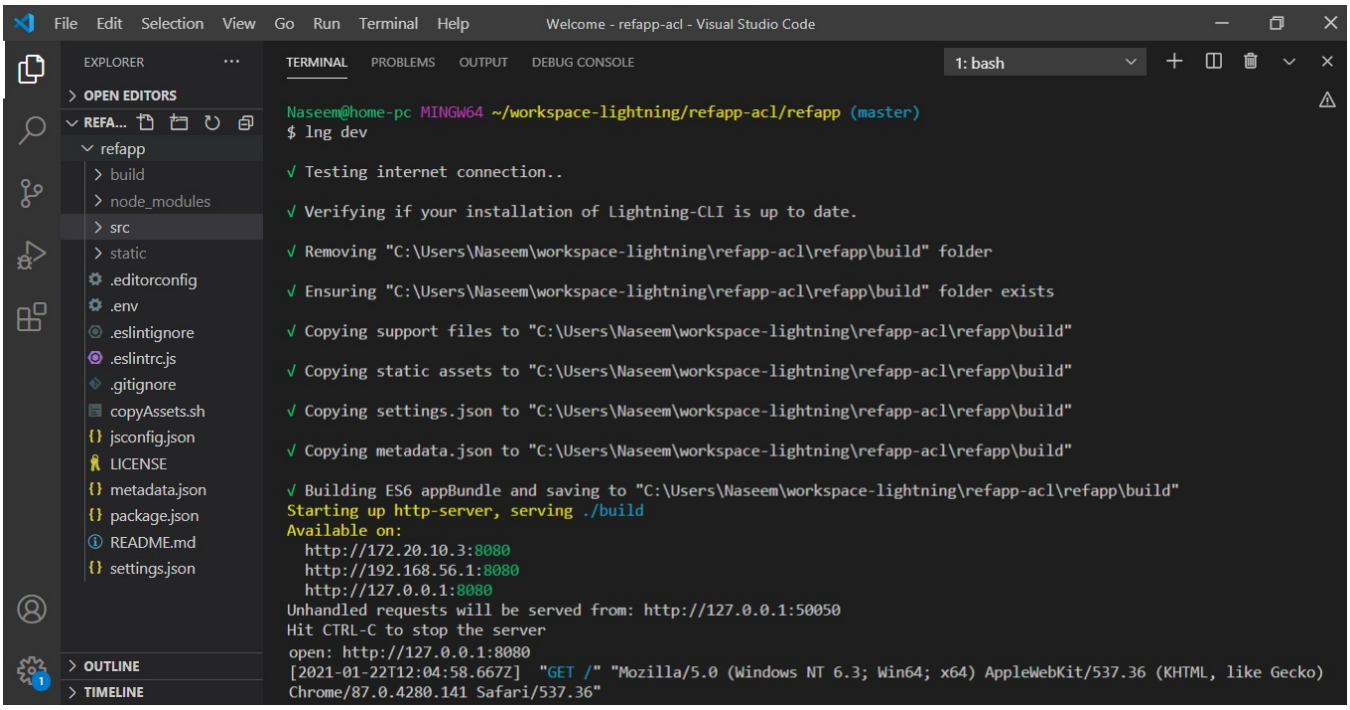
reference images for build instructions-2

once npm install complete, you will get message at the end "Do you want us to automatically check for old Lightning-SDK imports and update them in your project files? y/n

press n and hit enter button because on pressing y it gets hanged or stuck, in this case press ctrl+c



reference images for build instructions-3



reference images for build instructions-4

Running refapp

by default vscode launch generated url on chrome browser in dev pc.

reference image for running app on browser

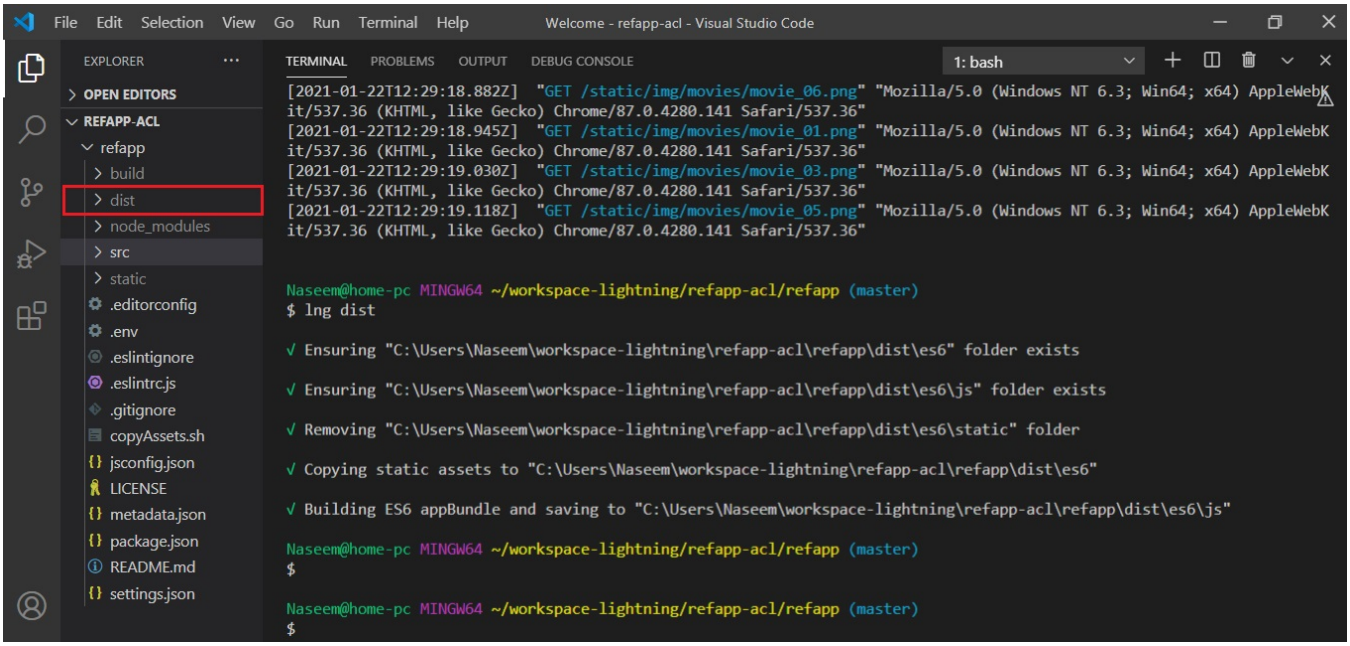
- **Pre-build refapp code**

  Download customized pre-build refapp code which has lightning sdk video player integrated: build.zip

  (or)

- **Create dist (build)**
  - run below command to make build.

```
$ lng dist
```



reference image for dist creation

Hosting Lighting refapp

1. copy the lighting build app from dist folder.
2. host build folder into any http server.
3. test the hosted application on browser

Lighting refapp integration

- **Pre-build rdk-generic-hybrid-image-raspberrypi image**

  Download pre-build rdk-generic-hybrid-image-raspberrypi image from here: rdk-generic-hybrid-image-raspberrypi-rdk-hybrid.zip

  (or)

- **Build rdk-generic-hybrid-image-raspberrypi follow below instructions**

```
$ repo init -u https://code.rdkcentral.com/r/manifests -b yocto-dunfell-upgrade -m rdkv-nosrc.xml
$ repo sync -j4 --no-clone-bundle
$ MACHINE=raspberrypi-rdk-hybrid source meta-cmf-raspberrypi/setup-environment
$ bitbake rdk-generic-hybrid-image
```

- **Flash image on the SD card**
  - Execute the following command to flash the image on the SD card

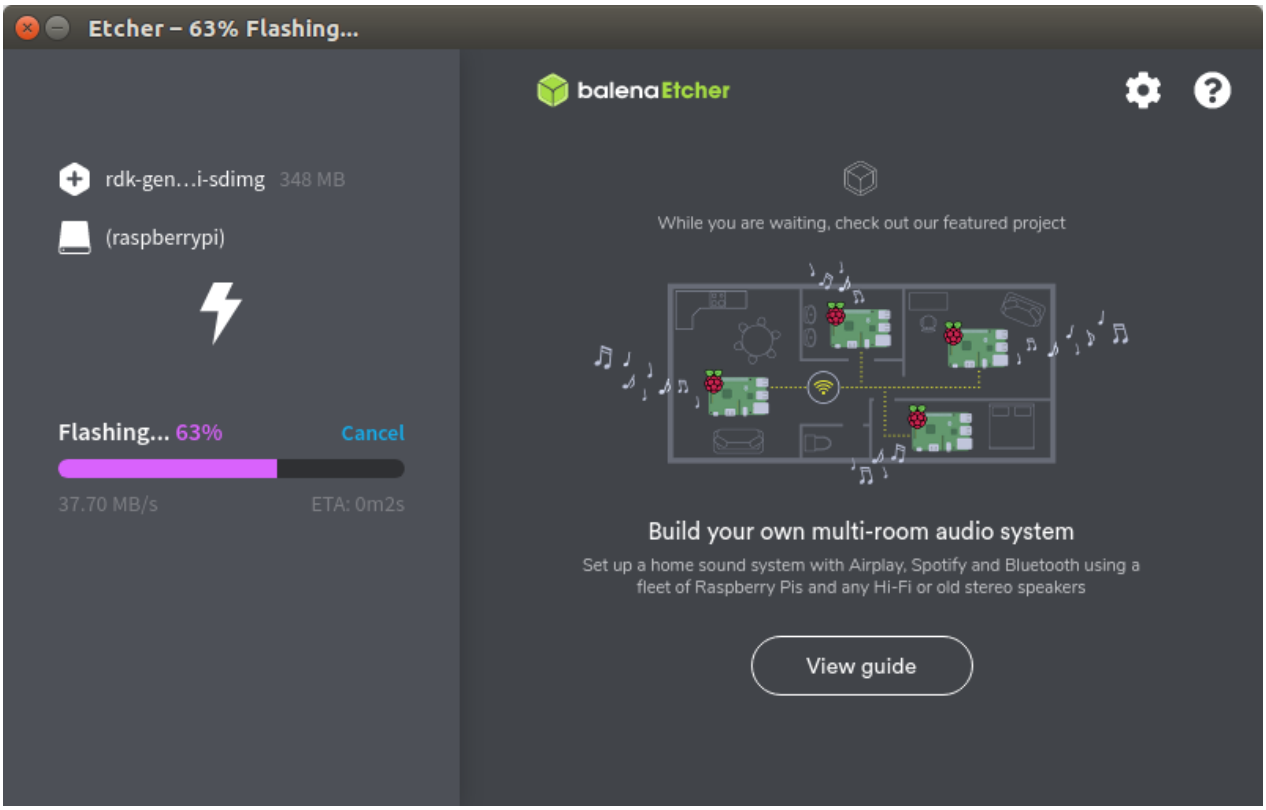
```
# sudo dd if=<r-pi sdimg> of=<SD card device> bs=1M

for ex: # sudo dd if=rdk-mc-rpi.sdimg of=/dev/sdb bs=1M
```

(or)

Windows user can download and install Balena Etcher to write the Raspberry Pi SD card: https://www.balena.io/etcher/

- Download the pre-build image file from rdk-generic-hybrid-image-raspberrypi-rdk-hybrid.zip link to your PC and extract
- Flash it to the Raspberry Pi's SD card using Etcher



- **Lighting refapp integration into rdk-generic-hybrid-image-raspberrypi**

    - **Method 1:**

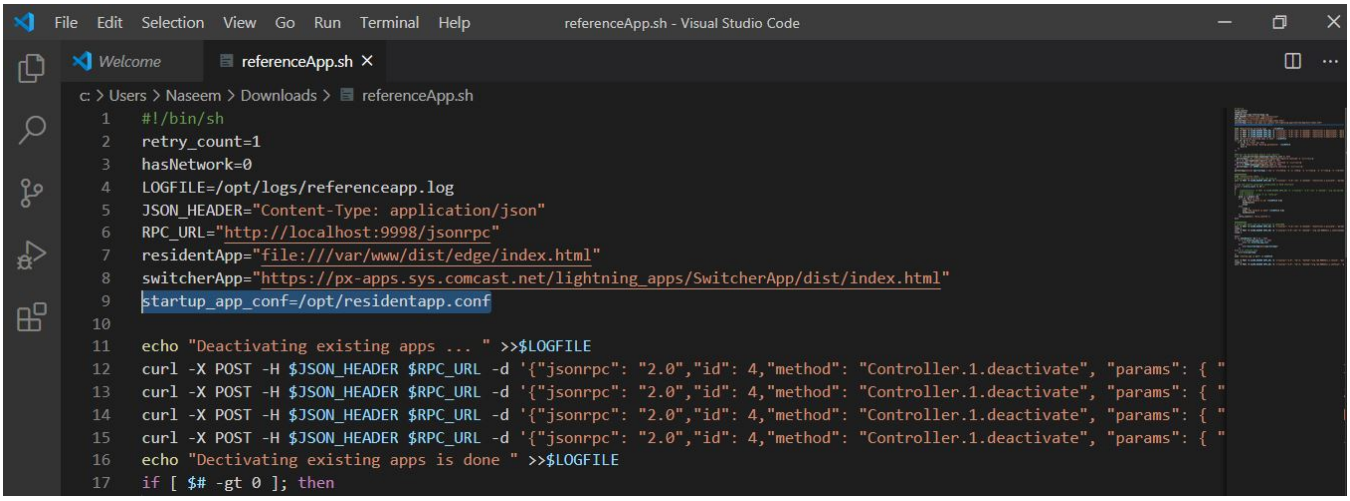
1. login into raspberrypi and change directory:

```
$ ssh root@<<IP address>>

# after login into box
$ cd /lib/rdk/
```

2. add startup_app_conf variable into referenceApp.sh config file, if referenceApp.sh file is not available in /lib/rdk then refer **method 2** below to validate refapp.

```
startup_app_conf=/opt/residentapp.conf
```

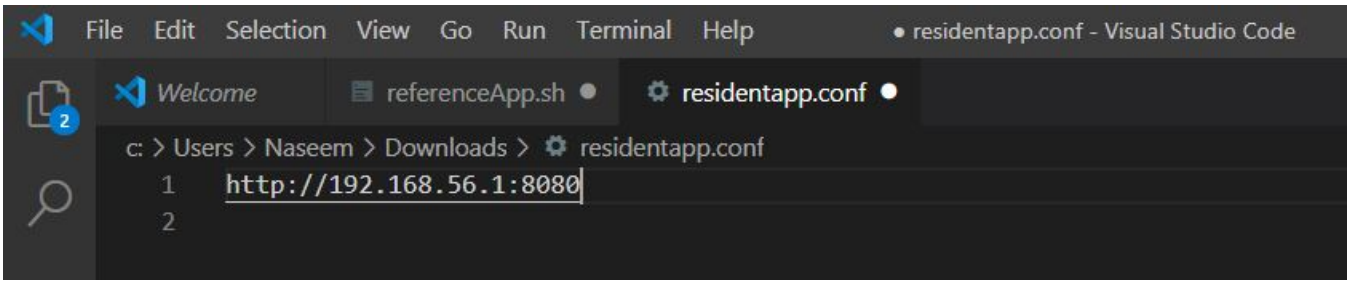


3. add below line to avoid loading default app

```
url="http://192.168.0.106:8080/"
echo $url
```

4. replace exting url https://px-apps.sys.comcast.net/lightning_apps/SwitcherApp/dist/index.html in /opt/residentapp.conf with lighting app url which is hosted on server

for example your server providing url http://192.168.56.1:8080



5. reboot the box

- **Method 2:**

Validated on rdk-generic-hybrid-wpe-image_rdk-next image, download image link: https://drive.google.com/file/d/1_VgFDk_lRWLoMwv0JKQrZ4HUifN-F6Qa/view?usp=sharing

step 1: login to box $ ssh root@<your box ip>

step 2: copy refapp build into /opt/www

step3: $ cd /lib/rdk

step 4: $ vi residentApp.sh

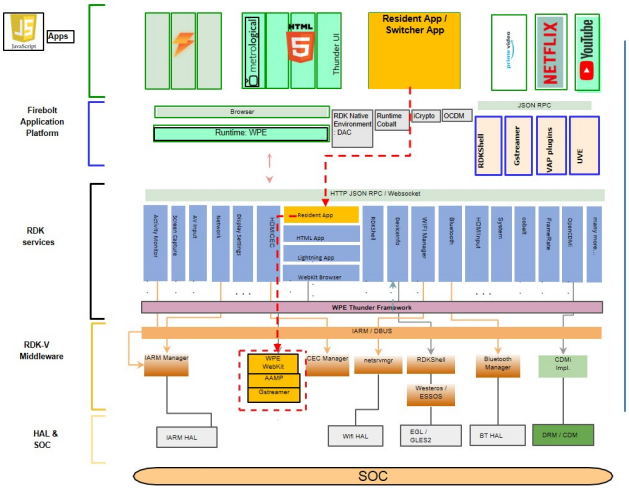
step 5: configure refapp url into residentApp.sh > find in residentApp.sh line "offlineApp="http://127.0.0.1:50050/lxresui/index.html" and replace with refapp url, Example: "offlineApp="http://127.0.0.1:50050/refapp/index.html"

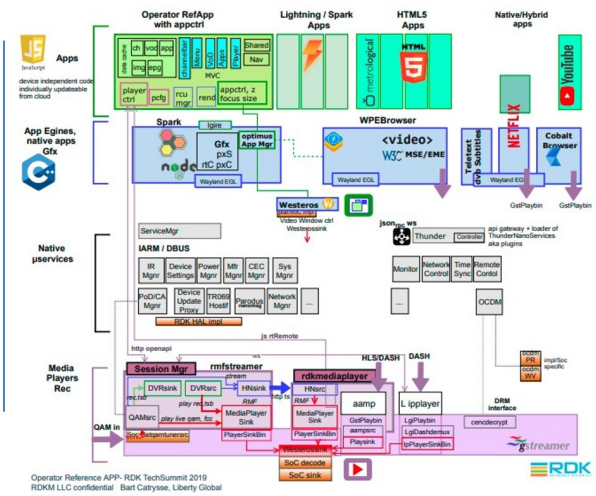
Note: if you are downloading prebuild refapp from download link: build.zip after extract you will see build/build/<files>, you can copy build folder as it is and in this case url will be "offlineApp="http://127.0.0.1:50050/build/build/index.html".

step 6: After reboot app come up with a splash screen. Lighting application will launch instead of switcher app. Main menu screen is displayed which allows users to launch apps, vod, and play video content.

RefApp Video Playback

- **RDK 4.0 (rdk-generic-hybrid-image) Vs LGI RefApp (rdk-generic-hybrid-refapp-image)**





Dependencies

- **Lightning RefApp on RDK 4.0**
  - Uses lighting sdk media player to play video, verified video playback using mp4 and HLS streams.
  - Tested mp4 source: http://clips-media-aka.warnermediacdn.com/cnn/clips/2021-01/79895-97f21462b4db4236abcbd68a581[…]remarks-vpx-primary-58581-79895-1920x1080_8000k.mp4
  - Tested hls source: //cdn.metrological.com/hls/greenland720.m3u8
  - Lightning (a javascript TV app development framework) refapp is a web based app, which runs on browser (supports WebGL). To run this wpe framwork or rdkbrowser2 is required.
  - sessionmanager, rmfstreamer, spark environment, nodejs and LGI based IP player not required.
- **Lightning RefApp on Operator reference image (rdk-generic-hybrid-refapp-image)**
  - uses various native RDK components such as spark, rmfstreamer with sessionmanager, IP players (aamp and Liberty IPplayer) to play QAM source, IP video streams (hls/dash).

Reference Video

Your browser does not support the HTML5 video element