Pipewire use case validation with gst-launch

- Introduction
- Build and Flash Procedure
- AWS Account creation steps
- CVR Validation Procedure with gst-launch
- Limitations

Introduction

Pipewire is a server and user space API to deal with multimedia pipelines. This page dedicated to bringing up and validation of Continuous video recording functionality with pipewire in RPI-3 B+ target using gst-launch.

CVR uses Kinesis Video Streams and supporting 24/7 video recording support. Allows the feature of storing the recorded content in AWS server.

Build and Flash Procedure

RDK-C rdk-next Yocto 3.1 dunfell build for Raspberrypi

AWS Account creation steps

STEP 1:

Can't able to validate without AWS account. so,Refer below link to create AWS account.

Procedure To Create AWS Account

Video Stream creation step

After AWS account creation ,Refer same link to create Access key ID ,Secret Access Key and Region Name.

STEP 2:

Follow below steps to create kinese video stream or refer below link.

https://docs.aws.amazon.com/kinesisvideostreams/latest/dg/gs-createstream.html

1.Sign in to the AWS Management Console and open the Kinesis console at https://eu-west-1.console.aws.amazon.com /kinesisvideo/home?region=eu-west-1#/. 2.On the Video streams page, choose Create video stream. 3.On the Create a new video stream page, type ExampleStream for the stream name. Leave the Default configuration radio button selected. 4.Choose Create video stream. 5.After Kinesis Video Streams creates the stream, review the details on the ExampleStream page.

CVR Validation Procedure with gst-launch

STEP 1:

Add require SSID and PSK in /etc/wpa_supplicant.conf file in below format

network={

ssid="username"

```
psk="password"
```

}

Console output

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
update_config=1
```

network={
ssid="RDK"
psk="Comcast1"
}

STEP 2:

Reboot the Target

After Reboot don't do step 1 and 2.

Note : Step 1 & 2 is only applicable for fresh target boot-up with new image.

STEP 3:

WiFi connection is must needed for CVR-AWS validation.

Check WiFi connection by using below command.

ifconfig

Console output

```
root@raspberrypi3-rdk-camera:~# ifconfig
         Link encap:Local Loopback
10
         inet addr:127.0.0.1 Mask:255.0.0.0
         inet6 addr: ::1/128 Scope:Host
         UP LOOPBACK RUNNING MTU:65536 Metric:1
         RX packets:87 errors:0 dropped:0 overruns:0 frame:0
         TX packets:87 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:4552 (4.4 KiB) TX bytes:4552 (4.4 KiB)
         Link encap:Ethernet HWaddr B8:27:EB:2E:72:2B
wlan0
         inet addr:192.168.43.246 Bcast:192.168.43.255 Mask:255.255.255.0
         UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
         RX packets:23 errors:0 dropped:0 overruns:0 frame:0
         TX packets:44 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:2893 (2.8 KiB) TX bytes:5887 (5.7 KiB)
```

STEP 4:

Before validate cvr use case we should stop rms binary and mediastreamer binary with below command.

systemctl stop rms-launcher

systemctl stop mst-launcher

STEP 5:

Before start the CVR binary please check the current date and time in RPI with "date" command, If you get wrong date and time please manually set current date and time with the below command.

date -s "Mon Nov 30 03:42:44 UTC 2020"

Console output

root@raspberrypi3-rdk-camera:~# date -s "Mon Nov 30 03:42:44 UTC 2020" Mon Nov 30 03:42:44 UTC 2020 root@raspberrypi3-rdk-camera:~# date Mon Nov 30 03:43:55 UTC 2020

STEP 6:

Run the pipewire binary with below command

pipewire &

Console output

root@raspberrypi3-rdk-camera:~# pipewire &

<u>STEP 7:</u>

Run the pipewire media session binary with below command

pipewire-media-session &

Console output

root@raspberrypi3-rdk-camera:~# pipewire-media-session &

STEP 8:

Run the below gst-launch command to stream the live data into AWS server.

Need to give proper credential in this pipeline

gst-launch-1.0 pipewiresrc ! videoconvert ! video/x-raw,width=640,height=480 ! omxh264enc ! h264parse ! kvssink stream-name="XXXX" storagesize=100 access-key="YYYYY" secret-key="ZZZZZ" aws-region="ap-south-1"

STEP 9:

Enter into AWS account and select the corresponding video stream service which was we created for streaming.

Press "Media Playback" option on the video stream player.

Streaming video content should decode properly on AWS.



Limitations

• Used USB webcamera in RPI-3 target to validate CVR use case with gst-launch using pipewire multimedia framework.