Example on running python3 with boto3 library in WP76 modules

Background

1. Boto3 is a software development kit (SDK) designed to improve the use of the Python programming language in Amazon Web Services.
2. According to the boto3 start guide below, since Boto3 version 1.26.161 needs to use Python 3.7 or later, we need to implement this in WP76 R17 yocto package which includes python3.8.

https://boto3.amazonaws.com/v1/documentation/api/latest/guide/quickstart.html

3. The test in this document is done on WP7605 R17.
4. The yocto image compilation in this document is done in Ubuntu 16.

Test Procedure for running python3

1. Upgrade your WP76 module to R17 firmware.
2. Download the following .tar files to the /home/root/ folder of WP76 module by SCP tool

python3.8.tar

3. Now ssh to your module and extract the python3.8.tar file by the following commands: (please note that this needs to do once only as the extract location is in userapp partition which is non-volatile memory)

```
tar -xf /home/root/python3.8.tar -C /home/root/
```

4. Make the /usr/bin, /usr/include and /usr/lib writable:

```
mkdir /tmp/tmp_usr_bin/;mkdir /tmp/tmp_usr_bin_wr;

mount -t overlay overlay /usr/bin -o
lowerdir=/usr/bin,upperdir=/tmp/tmp_usr_bin,workdir=/tmp/tmp_usr_bin_wr;

mkdir /tmp/tmp_usr_include/;mkdir /tmp/tmp_usr_include_wr;

mount -t overlay overlay /usr/include -o
lowerdir=/usr/include,upperdir=/tmp/tmp_usr_include,workdir=/tmp/tmp_usr_include_wr;

mkdir /tmp/tmp_usr_lib/;mkdir /tmp/tmp_usr_lib_wr;

mount -t overlay overlay /usr/lib -o
lowerdir=/usr/lib,upperdir=/tmp/tmp_usr_lib,workdir=/tmp/tmp_usr_lib_wr;
```

5. Type the following to put the executable and libraries to correct path:

```
cp -rf /home/root/python3.8/usr/include/* /usr/include

cp -rf /home/root/python3.8/usr/bin/* /usr/bin
```

```
ln -s /home/root/python3.8/usr/lib/libpython3.8.so.1.0 /usr/lib/
ln -s /usr/lib/libpython3.8.so.1.0 /usr/lib/libpython3.8.so

ln -s /home/root/python3.8/usr/lib/libpython3.so /usr/lib/libpython3.so
cp -rf /home/root/python3.8/usr/lib/pkgconfig/* /usr/lib/pkgconfig/

mkdir /usr/lib/python3.8
ln -s /home/root/python3.8/usr/lib/python3.8/* /usr/lib/python3.8/
```

6. Now the python3 tool should be OK to be used in your environment:

```
root@fx30:~# python3
Could not find platform dependent libraries <exec_prefix>
Consider setting $PYTHONHOME to <prefix>[:<exec_prefix>]
Python 3.8.13 (default, Mar 16 2022, 13:02:57)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> exit()
root@fx30:~#
```

Test Procedure for importing boto3 library in python3

1. Make sure you have completed the previous session to run python3 in your module
2. Download the following .tar files to the /home/root/ folder of WP76 module by SCP tool

python3.8_boto3.tar

3. Now ssh to your module and extract the python3.8_boto3.tar file by the following commands: (please note that this needs to do once only as the extract location is in userapp partition which is non-volatile memory)

```
tar -xf /home/root/python3.8_boto3.tar -C /home/root/
```

4. Type the following to put the libraries to correct path:

```
ln -s /home/root/python3.8_boto3/usr/lib/python3.8/* /usr/lib/python3.8/
```

5. Now the boto3 libraries can be imported inside python3:

```
root@fx30:~# python3
Python 3.8.13 (default, Mar 16 2022, 13:02:57)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import boto3
>>> exit()
root@fx30:~#
```

How to build python3 and boto3 libraries in yocto linux package

1. Yocto source code of R17 can be downloaded here:

https://source.sierrawireless.com/resources/airprime/software/wp76xx/wp76xx-firmware-release-17-components/#sthash.yJ1QIX6L.dpbs

2. Make sure you can build the yocto cwe image in your Ubuntu PC.
3. download and extract the following zip files

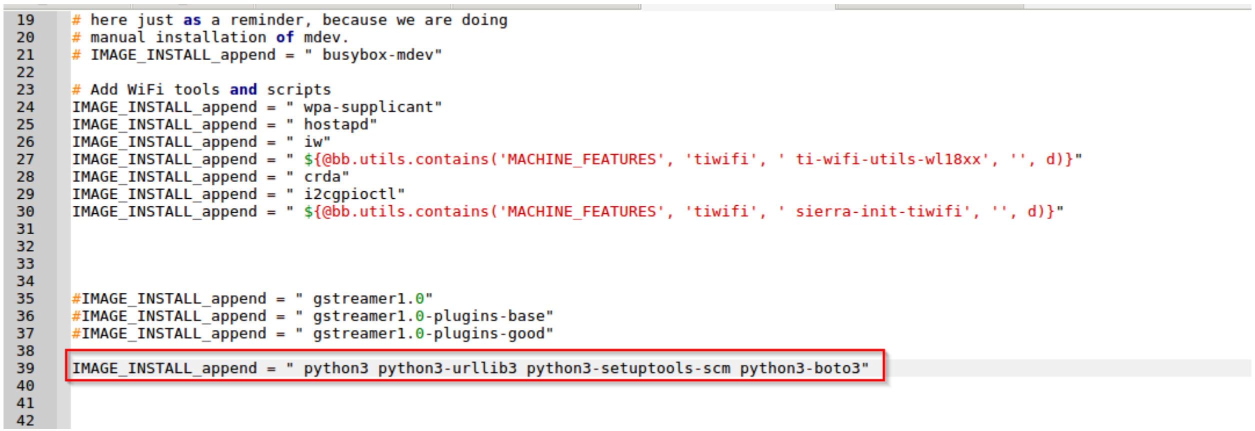
bb_files.zip

4. put the following .bb files in R17_SWI9X07Y_03.01.07.00/yocto/meta-openembedded/meta-python/recipes-devtools/python

python3-boto3_1.26.160.bb
python3-botocore_1.29.160.bb
python3-jmespath_1.0.1.bb
python3-python-dateutil_2.8.2.bb
python3-s3transfer_0.6.1.bb
python3-six_1.16.0.bb

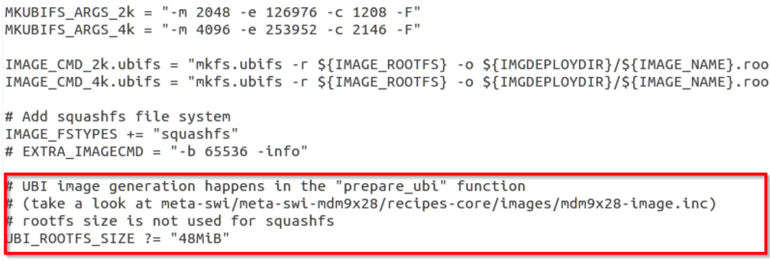
5. in R17_SWI9X07Y_03.01.07.00/yocto/meta-swi/meta-swi-mdm9x28/recipes-core/images/mdm9x28-image.inc, add the following:

IMAGE_INSTALL_append = " python3 python3-urllib3 python3-setuptools-scm python3-boto3"



```
# here just as a reminder, because we are doing
# manual installation of mdev.
# IMAGE_INSTALL_append = " busybox-mdev"

# Add WiFi tools and scripts
IMAGE_INSTALL_append = " wpa-supplicant"
IMAGE_INSTALL_append = " hostapd"
IMAGE_INSTALL_append = " iw"
IMAGE_INSTALL_append = " ${@bb.utils.contains('MACHINE_FEATURES', 'tiwifi', ' ti-wifi-utils-wl18xx', '', d)}"
IMAGE_INSTALL_append = " crda"
IMAGE_INSTALL_append = " i2cgpioctl"
IMAGE_INSTALL_append = " ${@bb.utils.contains('MACHINE_FEATURES', 'tiwifi', ' sierra-init-tiwifi', '', d)}"




#IMAGE_INSTALL_append = " gstreamer1.0"
#IMAGE_INSTALL_append = " gstreamer1.0-plugins-base"
#IMAGE_INSTALL_append = " gstreamer1.0-plugins-good"

IMAGE_INSTALL_append = " python3 python3-urllib3 python3-setuptools-scm python3-boto3"
```

6. go to R17_SWI9X07Y_03.01.07.00/yocto/meta-swi/meta-swi-mdm9x28/conf/machine/swi-mdm9x28.conf, change the UBI_ROOTFS_SIZE to be 48MiB



```
MKUBIFS_ARGS_2k = "-m 2048 -e 126976 -c 1208 -F"
MKUBIFS_ARGS_4k = "-m 4096 -e 253952 -c 2146 -F"

IMAGE_CMD_2k.ubifs = "mkfs.ubifs -r ${IMAGE_ROOTFS} -o ${IMGDEPLOYDIR}/${IMAGE_NAME}.rootfs.2k.ubifs ${MKUBIFS_ARGS_2k}"
IMAGE_CMD_4k.ubifs = "mkfs.ubifs -r ${IMAGE_ROOTFS} -o ${IMGDEPLOYDIR}/${IMAGE_NAME}.rootfs.4k.ubifs ${MKUBIFS_ARGS_4k}"

# Add squashfs file system
IMAGE_FSTYPES += "squashfs"
# EXTRA_IMAGECMD = "-b 65536 -info"

# UBI image generation happens in the "prepare_ubi" function
# (take a look at meta-swi/meta-swi-mdm9x28/recipes-core/images/mdm9x28-image.inc)
# rootfs size is not used for squashfs
UBI_ROOTFS_SIZE ?= "48MiB"
```

7. Type “make” to build the yocto image, there will be error to make the yocto cwe image in the final stage as the image is too big, but we can still build the binaries of python3/boto3 libraries in the following output paths:
   - R17_SWI9X07Y_03.01.07.00/yocto/build_bin/tmp/work/armv7a-neon-poky-linux-gnueabi/python3/3.8.13-r0/image/usr
   - R17_SWI9X07Y_03.01.07.00/yocto/build_bin/tmp/work/armv7a-neon-poky-linux-gnueabi/python3/3.8.13-r0/image/usr/lib/python3.8
   - R17_SWI9X07Y_03.01.07.00/yocto/build_bin/tmp/work/armv7a-neon-poky-linux-gnueabi/python3-boto3/1.26.160-r0/image/usr/lib/python3.8/site-packages/
   - R17_SWI9X07Y_03.01.07.00/yocto/build_bin/tmp/work/armv7a-neon-poky-linux-gnueabi/python3-botocore/1.29.160-r0/image/usr/lib/python3.8/site-packages/
   - R17_SWI9X07Y_03.01.07.00/yocto/build_bin/tmp/work/armv7a-neon-poky-linux-gnueabi/python3-jmespath/1.0.1-r0/image/usr/lib/python3.8/site-packages
   - R17_SWI9X07Y_03.01.07.00/yocto/build_bin/tmp/work/armv7a-neon-poky-linux-gnueabi/python3-python-dateutil/2.8.2-r0/image/usr/lib/python3.8/site-packages
   - R17_SWI9X07Y_03.01.07.00/yocto/build_bin/tmp/work/armv7a-neon-poky-linux-gnueabi/python3-six/1.16.0-r0/image/usr/lib/python3.8/site-packages
   - R17_SWI9X07Y_03.01.07.00/yocto/build_bin/tmp/work/armv7a-neon-poky-linux-gnueabi/python3-urllib3/1.25.7-r0/image/usr/lib/python3.8/site-packages

Reference

1. https://boto3.amazonaws.com/v1/documentation/api/latest/guide/quickstart.html
2. https://stackoverflow.com/questions/68097671/how-to-install-botocore-in-yocto