

Xilinx KV260 Getting Started Guide for AWS IoT Greengrass V2

2021-07.31.0

Table of Contents

1	<i>Document Information</i>	1
2	<i>Overview</i>	1
3	<i>Hardware Description</i>	2
4	<i>Set up your hardware</i>	2
5	<i>Setup your AWS account and Permissions</i>	3
6	<i>Create Resources in AWS IoT</i>	3
7	<i>Install the AWS Command Line Interface</i>	3
8	<i>Add meta-aws to Petalinux</i>	3
9	<i>Install AWS IoT Greengrass</i>	6
10	<i>Create a Hello World component</i>	7
11	<i>Upload the Hello World component</i>	8
12	<i>Troubleshooting</i>	8

1 Document Information

Note that all instructions in this document have been written assuming a Linux host machine. Equivalent commands in Windows can be used, but have not been provided.

1.1 Revision History

07-Jun-2021 Initial Draft
31-Jul-2021 Incorporated review comments.

2 Overview

The KV260 is a fully featured evaluation kit capable of leveraging pre-built accelerated applications to quickly develop unique solutions for production deployment on the K26 SOM.

The K26 SOM features an exclusive, custom-built XCK26 SoC based on the Zynq® UltraScale+™ MPSoC architecture, that has been configured for enhanced acceleration of vision AI applications.

The KV260 comes in two Production Qualified and Certified Commercial and Industrial Grades:

- Commercial: operating range 0-85°C for smart camera, embedded vision, tracking & identification
- Industrial: operation range -40-100°C for ruggedized for extreme environments

This guide describes how to get started with AWS IoT GreengrassV2 on your board, thus enabling easier deployment and management of your devices, applications and ML models at the edge.

2.1 About AWS IoT GreengrassV2

To learn more about AWS IoT GreengrassV2, see [how it works](#) and [what's new](#).

3 Hardware Description

3.1 DataSheet

Refer to the Kria K26 SOM Product Brief and SOM Data Sheet at product documentation at the [K26 Documentation page](#).

For a view of the hardware portfolio, refer to this [Portfolio Comparison](#).

3.2 Standard Kit Contents

Details regarding the KV260 kit contents are provided [here](#).

Refer to the section **What's Inside the Box** in the online documentation [here](#).

3.3 User Provided items

Refer to the section **What You'll Need to Provide** in the online documentation [here](#).

Also see the [recommended accessories](#) the user can purchase.

3.4 3rd Party purchasable items

Not applicable.

3.5 Additional References

Design resources are available [here](#).

Refer to the all the resources related to the Kria™ KV260 Vision AI Starter Kit on the [GitHub page](#)

Read the [Kria KV260 Vision AI Starter Kit User Guide](#) and [Data sheet](#)

4 Set up your hardware

The KV260 board boots off an SD card. To create this SD card, refer to the instructions at

© Copyright 2021 Xilinx© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

[Setting up the SD card image.](#)

The instructions to set up and connect the board are available [here](#).

Follow the instructions [here](#) to boot and monitor your board.

5 Setup your AWS account and Permissions

Refer to the instructions at [Set up your AWS Account](#). Follow the steps outlined in these sections to create your account and a user and get started:

- Sign up for an AWS account and
- Create a user and grant permissions.
- Open the AWS IoT console

Pay special attention to the Notes.

6 Create Resources in AWS IoT

Refer to the instructions at [Create AWS IoT Resources](#). Follow the steps outlined in these sections to provision resources for your device:

- Create an AWS IoT Policy
- Create a thing object

Pay special attention to the Notes.

7 Install the AWS Command Line Interface

To install the AWS CLI on your host machine, refer to the instructions at [Installing the AWS CLI v2](#). Installing the CLI is needed to complete the instructions in this guide.

Once you have installed AWS CLI, configure it as per the instructions in this [online guide](#). Set the appropriate values for Access key ID, Secret access key, and AWS Region. You can set Output format to "json" if you prefer.

8 Add meta-aws to Petalinux

The meta-aws layer provides you with Java (corretto), Greengrass, AWS SDKs and other assets. Java is required for Greengrass to run. Meta-aws will have to be added to your petalinux build for these resources to be available. For more information on meta-aws, refer to [github](#).

The instructions in this document are based on petalinux 2020.2.2

8.1 Build Petalinux for KV260

Confirm that you are using a supported version of Linux on your host machine for the Petalinux build (per [UG1144](#)) -e.g. Ubuntu 18.04. Also see [Build Petalinux](#).

Ensure that bash is being used:

```
sudo dpkg-reconfigure dash
```

Select No.

Install pre-requisites:

```
sudo apt-get install -y gcc git make net-tools libncurses5-dev tftpd zlib1g-dev  
libssl-dev flex bison libselinux1 gnupg wget diffstat chrpath socat xterm  
autoconf libtool tar unzip texinfo zlib1g-dev gcc-multilib build-essential  
libsdl1.2-dev libglib2.0-dev zlib1g:i386 screen pax gzip
```

```
sudo dpkg --add-architecture i386
```

```
sudo apt-get install -y zlib1g-dev zlib1g:i386
```

```
sudo apt update
```

```
sudo apt-get install zlib1g:i386
```

Download the petalinux installer from [here](#).

Enable exec permissions

```
chmod +x petalinux-v2020.2.2-final-installer.run
```

and run

```
./petalinux-v2020.2.2-final-installer.run --dir /opt/pkg/petalinux/2020.2.2 --  
platform aarch64
```

Download PetaLinux BSP installer from [KV260BSP](#) as referenced in [Petalinux BSP wiki](#)

Set up the environment

```
source /opt/pkg/petalinux/2020.2.2/settings.sh
```

Create the project

```
petalinux-create -t project -s ~/xilinx-k26-starterkit-v2020.2.2-final.bsp
```

```
cd xilinx-k26-starterkit-2020.2.2
```

```
echo 'BOARD_VARIANT = "kv"' >> project-spec/meta-user/conf/petalinuxbsp.conf
```

Add meta-aws using the instructions at [Add meta-aws](#).

You may want to do a clean make at this point. See [Doing a clean build](#)

Follow the instructions at [Build the image](#) and then [Package the image](#).

8.2 Add meta-aws

meta-aws has to be added not as a layer in bblayers, but as a project in project-spec.

```
cd ~/xilinx-k26-starterkit-2020.2.2/project-spec
```

Clone the meta-aws repo

```
git clone https://github.com/aws/meta-aws.git
```

```
cd meta-aws
```

```
git checkout zeus // to match xilinx yocto release
```

run

```
petalinux-config -> select Yocto Settings ->
```

```
select User layers -> and enter ${PROOT}/project-spec/meta-aws
```

Save the configuration

Review bblayers.conf to ensure that the change was made appropriately - the line below should be present in the file:

```
<basedir>/xilinx-k26-starterkit-2020.2.2/project-spec/meta-aws
```

Add greengrass-bin to the config:

```
edit <basedir>/project-spec/meta-user/conf/user_rootfsconfig to add the line  
CONFIG_greengrass-bin
```

Also add the greengrass-bin_2.3.0.bb file to <basedir>/project-spec/meta-aws/recipes-iot/aws-iot-greengrass (if not available from the zeus branch of meta-aws, please copy this file from the master-next branch of <https://github.com/aws/meta-aws>).

Run `petalinux-config -c rootfs`

Select **user packages** and enable greengrass-bin. To enable, hit Y (the * should appear next to greengrass-bin).

Save and Exit

8.3 Doing a clean build

In general, run the clean commands before a major build (recipe change etc.)

```
petalinux-build -x mrproper  
petalinux-build -x cleanall
```

8.4 Build the image

```
petalinux-config --silentconfig  
petalinux-build
```

Once the build completes, verify that greengrass-bin_2.3.0 has been included in the image. Using vi, open <base dir>/images/linux/rootfs.cpio and search for greengrass-bin_2.3.0.

8.5 Package the image

Run

```
petalinux-package --wic --bootfiles "ramdisk.cpio.gz.u-boot boot.scr Image system.dtb"
```

The wic file petalinux-sdimage.wic will be available in <basedir>/images/linux/petalinux-sdimage.wic

If needed:

```
copy the petalinux-sdimage.wic file to s3
aws s3 cp <basedir>/images/linux/petalinux-sdimage.wic <s3 bucket URI>
```

```
download the .wic file from s3 to your local machine
aws s3 cp <s3 bucket URI>/petalinux-sdimage.wic .
```

Flash the SD card using [Balena etcher](#) or equivalent.

8.6 Verify that Java is available

Insert the SD card into the KV260

Power on

Once the system has booted, verify that java is available using the command:

```
java --version
```

9 Install AWS IoT Greengrass

9.1 Download the AWS IoT Greengrass Core software

Download the latest greengrass core software as follows:

```
wget https://d2s8p88vqu9w66.cloudfront.net/releases/greengrass-nucleus-latest.zip
```

9.2 Install the AWS IoT Greengrass Core software

Unzip the AWS IoT Greengrass Core software to a folder on your device. Replace **GGCoreInstall** with the folder that you want to use

```
unzip greengrass-nucleus-latest.zip -d GGCoreInstall
rm greengrass-nucleus-latest.zip
```

Verify the version of the AWS IoT Greengrass Core software:

```
java -jar ./GGCoreInstall/lib/Greengrass.jar --version
```

You will see the Greengrass version displayed - similar to:

```
AWS Greengrass v2.3.0
```

9.2.1 Provide your credentials

Run the following commands to provide the credentials to the AWS IoT Greengrass Core software.

```
export AWS_ACCESS_KEY_ID=<the access key id for your account>
export AWS_SECRET_ACCESS_KEY=<the secret access key for your account>
```

9.2.2 Run the installer

Run the installer as shown below. Modify the values as per your region, install directory and thing name.

Use the `--provision true` option to have the installer set up the thing and required policies for you. If you prefer to configure Greengrass manually, see the [online guide](#).

Since petalinux uses sysvinit, don't specify the `--setup-system-service true` option.

```
sudo -E java -Droot="/greengrass/v2" -Dlog.store=FILE \  
-jar ./GGCoreInstall/lib/Greengrass.jar \  
--aws-region us-west-2 \  
--thing-name KV260GGCore \  
--tes-role-name GreengrassV2TokenExchangeRole \  
--tes-role-alias-name GreengrassCoreTokenExchangeRoleAlias \  
--component-default-user ggc_user:ggc_group \  
--provision true \  
--deploy-dev-tools true
```

If all goes well, you will see the following output on the device console:

```
Successfully configured Nucleus with provisioned resource details!  
Configured Nucleus to deploy aws.greengrass.Cli component
```

The local development tools (specified by the `--deploy-dev-tools` option) take some time to deploy. The following command can be used to check the status of this deployment:

```
aws greengrassv2 list-effective-deployments --core-device-thing-name KV260GGCore
```

When the status is SUCCEEDED, run the following command to verify that the Greengrass CLI is installed and runs on your device. Replace `/greengrass/v2` with the path to the base folder on your device as needed.

```
/greengrass/v2/bin/greengrass-cli help
```

Since Greengrass core was not installed as a system service, it must be manually run. The following command can be used:

```
sudo /greengrass/v2/alts/current/distro/bin/loader &
```

The software prints the following message if it launches successfully.

```
Launched Nucleus successfully.
```

If you don't run it as a background process, the terminal from which the loader is run must be kept open for the Greengrass Core software to continue running.

10 Create a Hello World component

In Greengrass v2, components can be created on the edge device and uploaded to the cloud, or vice versa.

Follow the instructions online under the section [To create a Hello World component](#) to create, deploy, test, update and manage a simple component on your device.

11 Upload the Hello World component

Follow the instructions online under the section [Upload your component](#) to upload your component to the cloud, where it can be deployed to other devices as needed.

12 Troubleshooting

Refer to the instructions at [Troubleshooting Greengrass v2](#) for information on:

- How to View AWS IoT Greengrass Core software logs
- How to View component logs
- AWS IoT Greengrass Core software issues
- AWS IoT Greengrass cloud issues
- Core device deployment issues
- Core device component issues

You can also refer to [Logging and Monitoring](#) to learn how to log API calls, gather system health telemetry data, and check core device status.