Advantech UNO-2271G V2

Getting Started Guide for AWS IoT Greengrass

内容

1 Document Information 2

1.1 Revision History 2

2 Overview 2

2.1 About AWS IoT Greengrass 2

3 Hardware Description 2

3.1 DataSheet 2

3.2 Standard Kit Contents 2

3.3 Additional Hardware References 2

4 Set up your hardware 3

5 Setup your AWS account and Permissions 3

6 Create Resources in AWS IoT 3

7 Install the AWS Command Line Interface 4

8 Pre-requisites of Linux System 4

9 Install and Initialize AWS IoT Greengrass 4

9.1 Install with the Wrapped package 4

9.2 Configure AWS IoT Greengrass Core software wrapped by Advantech: with initial.config 5

9.3 Initialize AWS IoT Greengrass 6

9.4 Confirm Result of Initialization 7

9.5 Resolve Error Log Found with Initialization: Timeout in Docker Mode 8

10 Create a Hello World component 9

10.1 Create the component on your edge device 9

10.2 Upload the Hello World component 9

1 Document Information

1.1 Revision History

DATE	VERSION	DESCRIPTION
2022/11/8	1.0	Initial draft
2022/12/26	2.0	Revision

2 Overview

Advantech's new UNO-2000 series generation of Embedded Automation Computers are built on highly ruggedized, fanless, and modular designs. Its design features optimized I/O, offers three different configurations, and provides flexibility through Advantech's iDoor technology expansion card technology. The new UNO-2000 series delivers a variety of form factors suited for use in specific applications, including smart factory solutions for Equipment Connectivity (EC), Process Visualization (PV), Environment Management (EM), and Dispatch Management (DM).

The Advantech UNO-2271G V2 is qualified to support the Amazon SageMaker Neo DLR and TensorFlow Lite frameworks to perform ML inference at the edge with AWS IoT Greengrass on locally generated data using cloud-trained models.

2.1 About AWS IoT Greengrass

To learn more about AWS IoT Greengrass, see how it works and what's new.

3 Hardware Description

3.1 DataSheet

https://advdownload.advantech.com/productfile/PIS/UNO-2271G%20V2/file/UNO-2271G-V2_DS(112422)20221124185255.pdf

3.2 Standard Kit Contents

The standard shipping hardware package include UNO-2271G V2 hardware configuration. Advantech also offers optional accessories to choose from, including adapters and power cables, mount kits, miscellaneous, WIFI module, LTE module and antenna. Please refers to the Datasheet for more detail offerings:
https://advdownload.advantech.com/productfile/Downloadfile2/1-29045L7/UNO-2271G_V2_User_Manual_En_Ed.2.pdf

3.3 Additional Hardware References

For more information, please refers to the link below: UNO-2271G V2 - Edge IoT Gateway with 2 x GbE, 2 x USB 3.2, 1 x mPCIe, HDMI, eMMC - Advantech

4 Set up your hardware

This section provides instructions for setting up the platform's hardware. The link is to the user manual of UNO-2271G V2, https://advdownload.advantech.com/productfile/Downloadfile2/1-29045L7/UNO-2271G_V2_User_Manual_En_Ed.2.pdf , which includes the following information:

- Introduction to UNO-2 Series and Specifications for UNO-2271G V2
    - Safety precaution
    - Packing list
    - Hardware specification
    - Dimensions
- Hardware functionality
    - Internal connectors
    - External I/O connectors
    - Expansion kit's internal connects
- Initial Setup
    - Chassis grounding
    - Connecting power
    - Open and close bottom cover
    - Installing second stack extension kit
- Appendix of system settings and pin assignments

5 Setup your AWS account and Permissions

Advantech supports AWS IoT Greengrass Core software for UNO-2271G V2 by offering a wrapping package. This package helps customer to create necessary resources for AWS IoT Greengrass Core software.

Create a AWS IAM user and attach AWS managed policy, named "IAMFullAccess" (ARN: "arn:aws:iam::aws:policy/IAMFullAccess). Create access key and download it. It's a .csv file which contains access key ID and secret access key of your IAM user. Back up it before using it with Advantech wrapping package.

Refer to the online AWS documentation at Set up your AWS Account. Following steps outlined in the sections below will show how 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

6 Create Resources in AWS IoT

Advantech wrapping package helps customer to create resources in AWS IoT. Initialization script can assist user to create resource by IAM user access key file.

Refer to the online AWS documentation at Create AWS IoT Resources. Following steps outlined in these sections will show how 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

Advantech wrapping package installs AWS command line interface (v2.9.13) before initialize AWS IoT Greengrass Core software. User can reference following reference for more information.

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 Pre-requisites of Linux System

To facilitate installation of AWS IoT Greengrass Core software for user with UNO-2271G V2, wrapping package for AWS IoT Greengrass Core software is supported by Advantech. These scripts inside the wrapping package require Ubuntu-based distribution.

9 Install and Initialize AWS IoT Greengrass

Instructions in this section explains how to use package wrapped by Advantech to install and initialize AWS IoT Greengrass Core software. Within files downloaded from Advantech, there is a file named “greengrassv2_linux-amd64.tar.gz”. Follow instructions below to install and initialize AWS IoT Greengrass in UNO-2271G V2.
For the package details and download, please refer to Contact Us | Request a Demo for UNO-2271G V2 - Advantech

9.1 Install with the Wrapped package

STEP 1. Extract the wrapping package first

```
$ tar xf greengrassv2_linux-amd64.tar.gz
```

STEP 2. Enter the extracted folder and run script with root privilege. (This script installs necessary programs and setup environment for AWS Greengrass.)

```
$ sudo ./installer.sh
```

NOTE: verify availability of Java here with command:
$ java –version

NOTE: The installer.sh supported by Advantech installs software package for customer, e.g., openjdk-8-jre, so the version verification command must be run after installer.sh

9.2 Configure AWS IoT Greengrass Core software wrapped by Advantech: with initial.config

Edit the config file and input preferred configuration. Reference table below about definition of each configuration variable (path: /opt/greengrassv2/config/initial.config). Because this file requires root privilege to access, launching editor with sudo command is recommended.

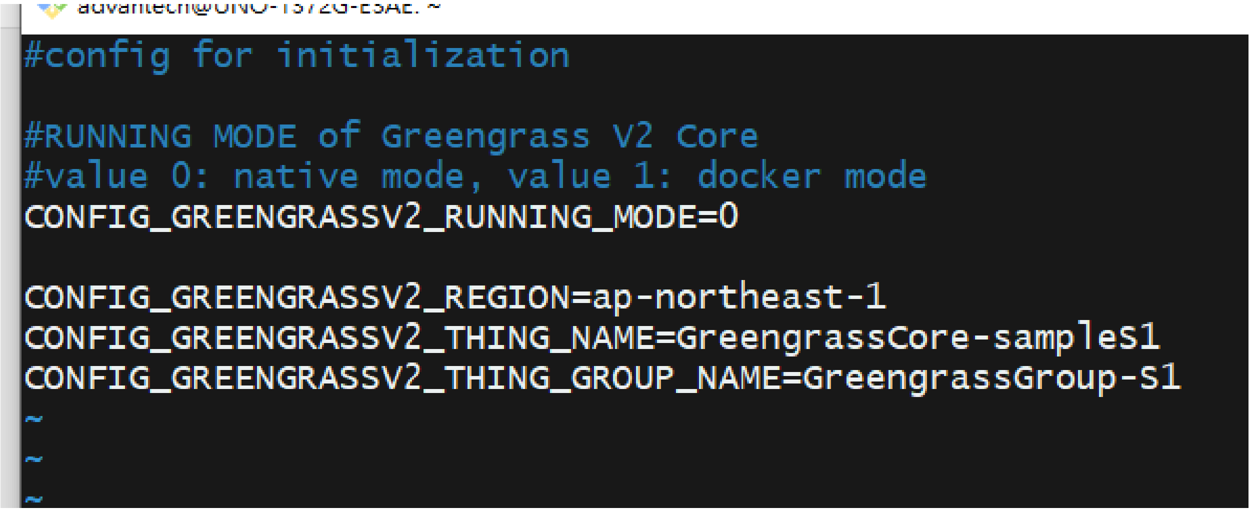
STEP 3. Edit the config file

```
$ sudo vim /opt/greengrassv2/config/initial.config
```

NOTE: Because AWS IoT Greengrass can be configured to run in a Docker container, the wrapping package defines configurable running mode of AWS IoT Greengrass Core software: native mode and Docker mode. User need to select one of them in config file of wrapping package before initialization.

Reference links below for detail information about AWS Greengrass V2 Core. These links can help you to select running mode of your device and help you know more about how to setup a Greengrass Core device well.

- AWS Greengrass V2 developer guide : Walks through how to set up AWS IoT Greengrass V2 and integrate it with other services.
- Run AWS IoT Greengrass Core software in a Docker container : This link is also part of developer guide. If you want to know detail information of initialization and launch Greengrass V2 inside Docker container, reference this link.



```
advantech@UNO-1372G-E3AE: ~
#config for initialization

#RUNNING MODE of Greengrass V2 Core
#value 0: native mode, value 1: docker mode
CONFIG_GREENGRASSV2_RUNNING_MODE=0

CONFIG_GREENGRASSV2_REGION=ap-northeast-1
CONFIG_GREENGRASSV2_THING_NAME=GreengrassCore-sampleS1
CONFIG_GREENGRASSV2_THING_GROUP_NAME=GreengrassGroup-S1
~
~
~
```

Variable	Value
CONFIG_GREENGRASSV2_RUNNING_MODE	0 for native mode, 1 for Docker mode
CONFIG_GREENGRASSV2_REGION	Region configuration for core device

CONFIG_GREENGRASSV2_THING_NAME	Thing name of core device
CONFIG_GREENGRASSV2_THING_GROUP_NAME	Thing group of core device

9.3 Initialize AWS IoT Greengrass

Make sure user secure credential file(.csv) is ready before next instruction. (Please refer to Section 5: Set up Your AWS account and permissions).Backup the secure credentials file before execution of following command because the initialization script remove .csv file after successful initialization.

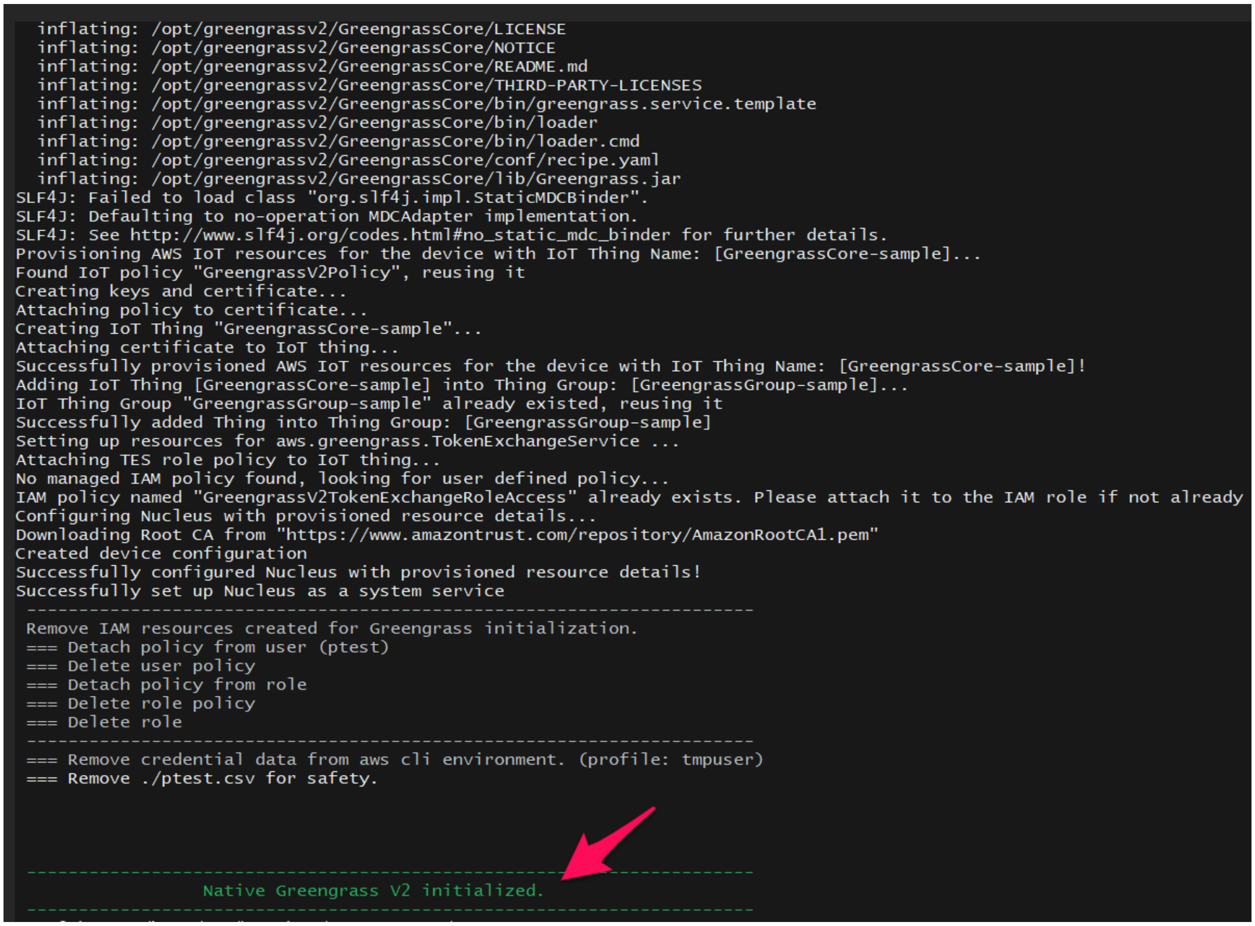
STEP 4. Run initialization script prepared in the wrapping package

```
$ sudo /opt/greengrassv2/tools/greengrassv2_initializer.sh <path of user secure
credentials>
```

After execution of this initialization script, AWS IoT Greengrass Core software will be launched with configuration and secure credentials offered by user.

9.4 Confirm Result of Initialization

To confirm initialization result, read log printed by greengrassv2_initializer.sh. **In native mode**, last line printed by greengrassv2_initializer.sh is “Native Greengrass V2 initialized”.

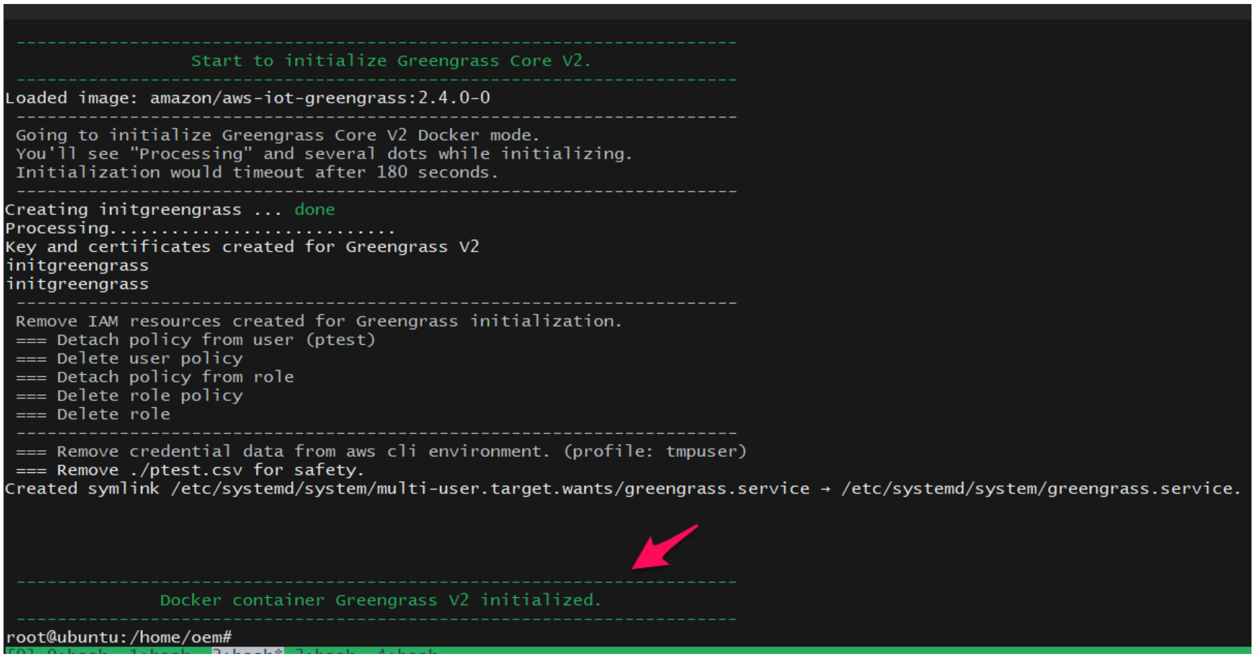


```
   inflating: /opt/greengrassv2/GreengrassCore/LICENSE
   inflating: /opt/greengrassv2/GreengrassCore/NOTICE
   inflating: /opt/greengrassv2/GreengrassCore/README.md
   inflating: /opt/greengrassv2/GreengrassCore/THIRD-PARTY-LICENSES
   inflating: /opt/greengrassv2/GreengrassCore/bin/greengrass.service.template
   inflating: /opt/greengrassv2/GreengrassCore/bin/loader
   inflating: /opt/greengrassv2/GreengrassCore/bin/loader.cmd
   inflating: /opt/greengrassv2/GreengrassCore/conf/recipe.yaml
   inflating: /opt/greengrassv2/GreengrassCore/lib/Greengrass.jar
 SLF4J: Failed to load class "org.slf4j.impl.StaticMDCBinder".
 SLF4J: Defaulting to no-operation MDCAdapter implementation.
 SLF4J: See http://www.slf4j.org/codes.html#no_static_mdc_binder for further details.
 Provisioning AWS IoT resources for the device with IoT Thing Name: [GreengrassCore-sample]...
 Found IoT policy "GreengrassV2Policy", reusing it
 Creating keys and certificate...
 Attaching policy to certificate...
 Creating IoT Thing "GreengrassCore-sample"...
 Attaching certificate to IoT thing...
 Successfully provisioned AWS IoT resources for the device with IoT Thing Name: [GreengrassCore-sample]!
 Adding IoT Thing [GreengrassCore-sample] into Thing Group: [GreengrassGroup-sample]...
 IoT Thing Group "GreengrassGroup-sample" already existed, reusing it
 Successfully added Thing into Thing Group: [GreengrassGroup-sample]
 Setting up resources for aws.greengrass.TokenExchangeService ...
 Attaching TES role policy to IoT thing...
 No managed IAM policy found, looking for user defined policy...
 IAM policy named "GreengrassV2TokenExchangeRoleAccess" already exists. Please attach it to the IAM role if not already
 Configuring Nucleus with provisioned resource details...
 Downloading Root CA from "https://www.amazontrust.com/repository/AmazonRootCA1.pem"
 Created device configuration
 Successfully configured Nucleus with provisioned resource details!
 Successfully set up Nucleus as a system service
  ----------------------------------------------------------------------
  Remove IAM resources created for Greengrass initialization.
  === Detach policy from user (ptest)
  === Delete user policy
  === Detach policy from role
  === Delete role policy
  === Delete role
  ----------------------------------------------------------------------
  === Remove credential data from aws cli environment. (profile: tmpuser)
  === Remove ./ptest.csv for safety.



  ----------------------------------------------------------------------
                  Native Greengrass V2 initialized.
  ----------------------------------------------------------------------
```

In Docker mode, greengrassv2_initializer.sh prints “Docker container Greengrass V2 initialized.” to report success of initialization. While initializing, the script prints several dots to tell user how many seconds spent on waiting initialization. If it’s timeout after 180 seconds, error message is reported with red logs. Following picture is captured with success initialization of Docker mode.

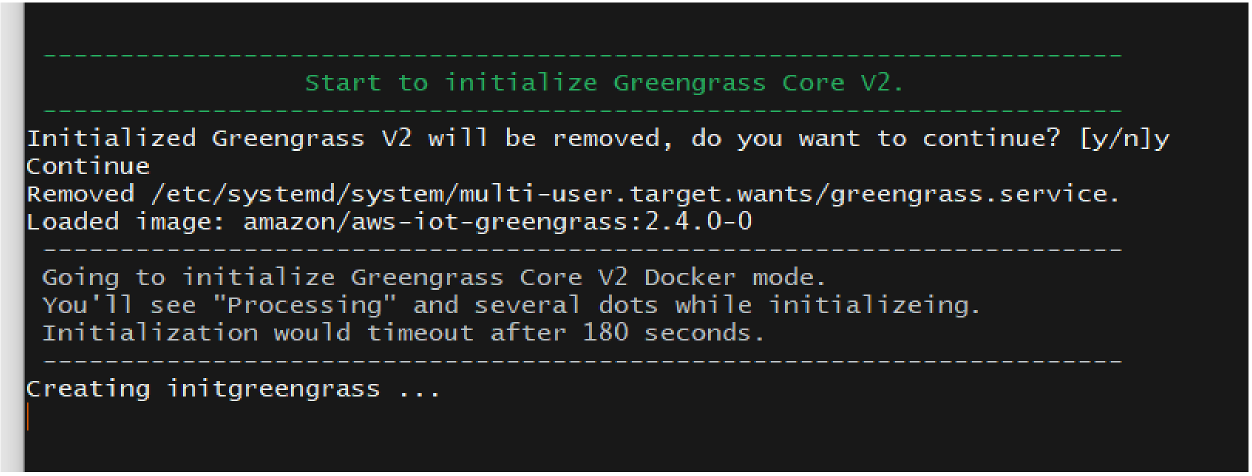


```
 ----------------------------------------------------------------------
                  Start to initialize Greengrass Core V2.
 ----------------------------------------------------------------------
Loaded image: amazon/aws-iot-greengrass:2.4.0-0
 ----------------------------------------------------------------------
 Going to initialize Greengrass Core V2 Docker mode.
 You'll see "Processing" and several dots while initializing.
 Initialization would timeout after 180 seconds.
 ----------------------------------------------------------------------
Creating initgreengrass ... done
Processing............................
Key and certificates created for Greengrass V2
initgreengrass
initgreengrass
 ----------------------------------------------------------------------
 Remove IAM resources created for Greengrass initialization.
 === Detach policy from user (ptest)
 === Delete user policy
 === Detach policy from role
 === Delete role policy
 === Delete role
 ----------------------------------------------------------------------
 === Remove credential data from aws cli environment. (profile: tmpuser)
 === Remove ./ptest.csv for safety.
Created symlink /etc/systemd/system/multi-user.target.wants/greengrass.service → /etc/systemd/system/greengrass.service.



 ----------------------------------------------------------------------
               Docker container Greengrass V2 initialized.
 ----------------------------------------------------------------------
root@ubuntu:/home/oem#
```

The initialize script can detect initialized Greengrass V2 and confirm for overwrite if new initialization is launched. Answer "n" to stop initialize process to backup anything you need, and run the initialize script again with answer "y" to start a new initialization.

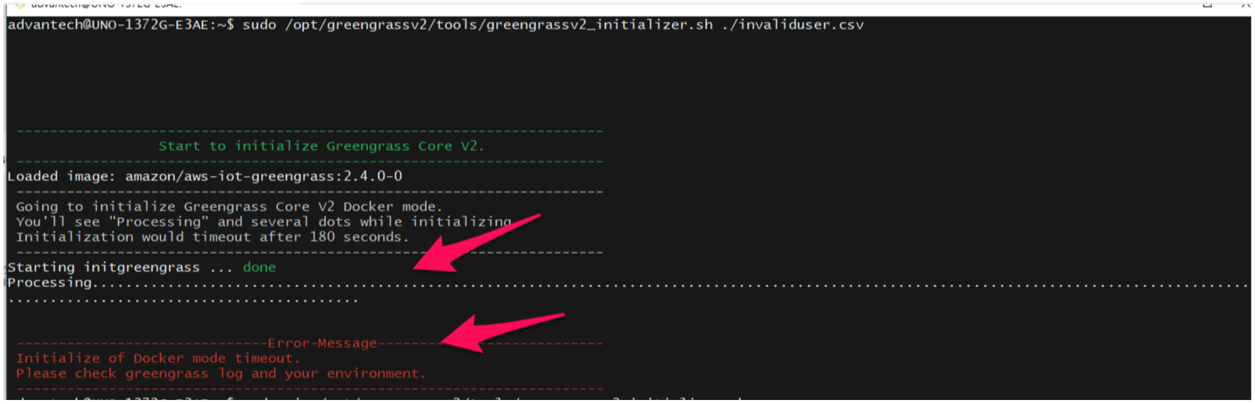


```
 ----------------------------------------------------------------------
                  Start to initialize Greengrass Core V2.
 ----------------------------------------------------------------------
Initialized Greengrass V2 will be removed, do you want to continue? [y/n]y
Continue
Removed /etc/systemd/system/multi-user.target.wants/greengrass.service.
Loaded image: amazon/aws-iot-greengrass:2.4.0-0
 ----------------------------------------------------------------------
 Going to initialize Greengrass Core V2 Docker mode.
 You'll see "Processing" and several dots while initializeing.
 Initialization would timeout after 180 seconds.
 ----------------------------------------------------------------------
Creating initgreengrass ...
```

9.5 Resolve Error Log Found with Initialization: Timeout in Docker Mode

In Docker mode, the initialize script spends 180 seconds on waiting for initialization progress, including device certification and device configuration. The initialization progress would fail with any invalid environment, e.g., an invalid IAM access key file.

As the picture below, the initialize script prints "Processing" and dots while spending 180 seconds on waiting. And the red text in picture tells that it's timeout and user have to check Greengrass log to get detail information of initialization failure. ("/greengrass/v2/logs/greengrass.log")



```
advantech@UNO-1372G-E3AE:~$ sudo /opt/greengrassv2/tools/greengrassv2_initializer.sh ./invaliduser.csv




 ----------------------------------------------------------------------------
                  Start to initialize Greengrass Core V2.
 ----------------------------------------------------------------------------
Loaded image: amazon/aws-iot-greengrass:2.4.0-0
 ----------------------------------------------------------------------------
 Going to initialize Greengrass Core V2 Docker mode.
 You'll see "Processing" and several dots while initializing
 Initialization would timeout after 180 seconds.
 ----------------------------------------------------------------------------
Starting initgreengrass ... done
Processing.........................................................................................................................
..............................................


 --------------------------------Error-Message-------------------------------
 Initialize of Docker mode timeout.
 Please check greengrass log and your environment.
 ----------------------------------------------------------------------------
```

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.

10.1 Create the component on your edge device

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.

10.2 Upload the Hello World component

Follow the instructions online at Upload your component to upload your component to the cloud, where it can be deployed to other devices as needed.