

Deploy AWS IoT Greengrass on IoT Gateway

Now that we have setup all the required resources in AWS account on cloud, we can prepare package to install AWS IoT Greengrass core software with AWS IoT fleet provisioning.

To prepare package, all the steps are part of this [script](#). You can execute this script on IoT device gateway or your computer. Please make sure that you have installed [AWS CLI](#) v2 with access to your AWS account.

For this use case, I execute on my laptop to create package in build directory. You can then copy package on your IoT gateway (e.g., Raspberry Pi). The script performs below steps.

1. Create build directory
mkdir build && cd build
2. Download AWS CA
3. Download claim certificates from AWS Secrets Manager
4. Download AWS IoT Greengrass and fleeting provisioning plugin
5. Get the endpoints and fleet provisioning template for AWS IoT Core
6. Prepare config.yml.
7. Prepare Greengrass start up command.
8. Change execution permission.

Test the Solution

As we have configured Raspberry Pi with AWS IoT Greengrass core software along with automatic fleet provision configuration, let us now run AWS IoT Greengrass service.

Test AWS IoT Greengrass device provisioning

1. Connect (ssh) to IoT Device gateway (e.g., Raspberry Pi) command line terminal and run below command to start AWS IoT Greengrass service to auto provision authenticate and establish connection to AWS IoT Core.

```
sudo build/fleet_provision.sh
```

2. On AWS IoT Core Console, expand the Greengrass section from the left panel and choose **Core Devices** option to verify the state of device. The device status should appear healthy as below.

AWS IoT ×

- Monitor
- Activity
- ▶ Connect
- ▶ Manage
- ▶ Fleet Hub
- ▼ Greengrass
 - Getting started
 - Core devices**
 - Components
 - Deployments
 - ▶ Classic (V1)
 - ▶ Wireless connectivity
 - ▶ Secure
 - ▶ Defend

Greengrass core device

A Greengrass core device runs Greengrass Core software. It is an edge device, or a controller for your edge devices. Greengrass core devices are things in AWS IoT. You can group core devices together to deploy configurations to them using AWS IoT thing groups. [Manage thing groups](#)

[Set up one core device](#)

Deploy software components to core devices

After you set up a core device or group of core devices, you can deploy software components to them. View the deployments page to create or revise a deployment and configure the software to deploy.

[View deployments](#)

Connect client devices to core devices

Set up client devices to connect to and communicate with core devices over MQTT.

[Configure discovery](#)

Greengrass core devices (1) [Set up one core device](#)

Name	Status	Status reported
DemoWasteBin	● Healthy	3 days ago

- If device does not appear as healthy, then please check Greengrass service log for any errors under `/greengrass/v2/logs` folder and follow [troubleshooting documentation](#).

Test remote application deployment

1. Under Greengrass section, choose Component option for edge application deployment and it should show private custom component as below. This custom component is already hosted in Amazon S3 repository along with CloudFormation script and made ready for deployment. However, for reference purpose, it is available on [GitHub](#). To deploy this application, please choose **monitor_wastebin_app**. Refer the procedures in the diagram below.

AWS IoT X

- Monitor
- Activity
- ▶ Connect
- ▶ Manage
- ▶ Fleet Hub
- ▼ Greengrass
 - Getting started
 - Core devices
 - Components**
 - Deployments
 - ▶ Classic (V1)
 - ▶ Wireless connectivity
 - ▶ Secure

Overview
Greengrass components are software modules that run on Greengrass core devices. Components can represent applications, runtime installers, libraries, or any other code that you run on a device.

Step 1: Develop a component
You can develop and test custom components on a local core device. Greengrass also provides public components that include common libraries, local development tools, and core features, such as machine learning. [Learn more](#)

Step 2: Create your component in the AWS Cloud
When you finish developing the software for your component, you can register it with Greengrass. Then, you can deploy and run the component on your Greengrass core devices.

Create component

My components | Public components | Community components

My components (1)
Your components are private components that only you can see and deploy to core devices. [Learn more](#)

Find by name, operating system, or architecture

Name	Version	Operating systems	Architectures	Version created
monitor_wastebin_app	2.0.0	linux	All	3 days ago

2. Verify details version 2.0.0 and choose deploy button.

AWS IoT X

AWS IoT > Greengrass > Components > monitor_wastebin_app:2.0.0

monitor_wastebin_app Version: 2.0.0 Create version Delete version View recipe **Deploy**

Overview

Description
Uses stream manager to upload a file to an S3 bucket.

Version 2.0.0	Version created 3 days ago	Status Deployable	Component scope Private
Type aws.greengrass.generic	ARN arn:aws:greengrass:eu-west-1:592092194010:components:monitor_wastebin_app:versions:2.0.0	Publisher sbmane@amazon.com	

3. On deployment stage, select create new deployment option

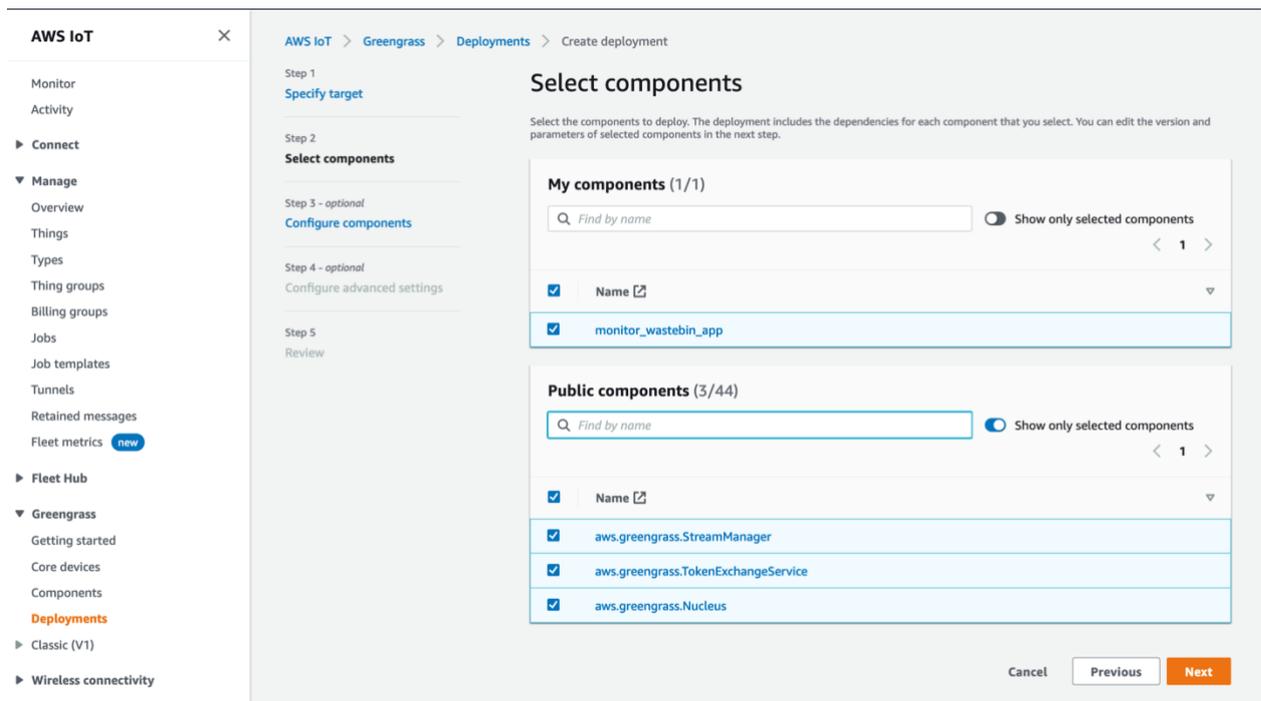
Add to deployment X

Deployment

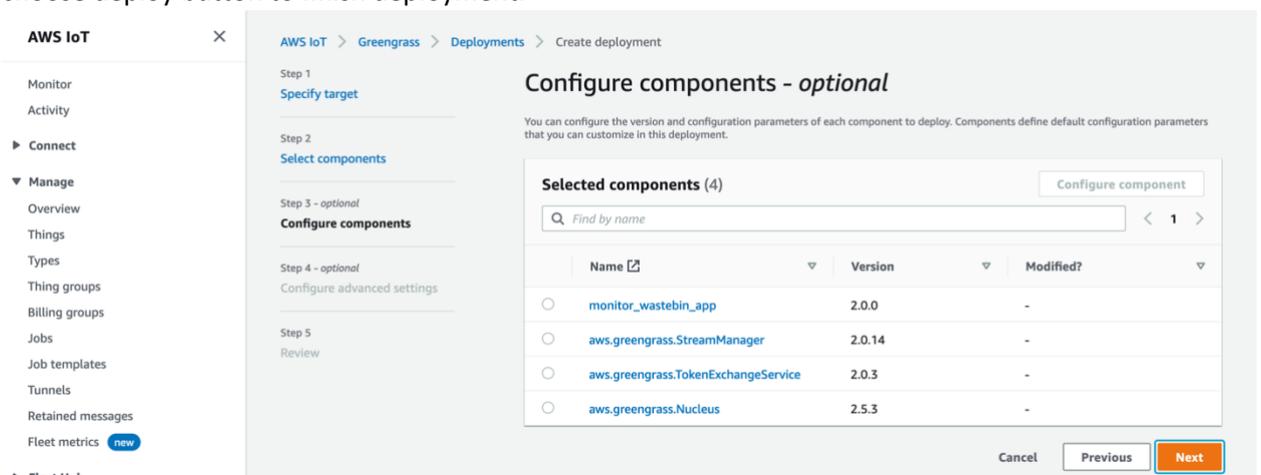
Add to existing deployment Create new deployment

Cancel Next

4. On specify target page, select core device as target and enter the name of core device from step 2 in section Test Greengrass device provisioning. For the rest of fields, follow the instruction on page
5. On select component page, please select below components (My components and public components) as shown screen shot



- Finally check component configuration and select **Next** button. Then on **Configure advanced settings section**, only choose Next button and move to Review stage. On Review stage, choose deploy button to finish deployment.



- Please note that if you are redeploying same component, then please select modified component and select **“Configure component”** button in the top right corner. Then in the **“Configuration to merge”** section as shown in screen shot, please enter some text e.g., **“deployment7”**

Previous configuration

Revision or default configuration

Revision: 8

Configuration update

```
{
  "reset": [],
  "merge": {
    "Message": "deployment7"
  }
}
```

Configuration update

Reset paths
A list of JSON pointers that define which configuration value, the deployment removes that value from the configuration, merges the values in the configuration to merge. Specify a list of default values. [Learn more](#)

[]

Configuration to merge
The configuration to merge with the configuration on each core device after it resets the values that you specify in the list of reset paths

```
1 {
2   "Message": "deployment7"
3 }
```

8. Now that application component deployment is completed, check the Greengrass logs on Raspberry Pi to verify the deployed application is not terminated because of any errors like permission issue etc.
 - a. **Troubleshoot** : If you observe greengrass service failed to download the custom component artifacts from the Amazon S3 bucket because device name, certificate path cannot be empty, then stop and start greengrass service as below and repeat 1 to 6 steps again to redeploy application.

```
sudo systemctl stop greengrass.service
sudo systemctl start greengrass.service
```

9. On AWS IoT Greengrass console, deployment should appear as completed

The screenshot shows the AWS IoT Greengrass console interface. On the left is a navigation menu with options like Monitor, Activity, Connect, Manage, Fleet Hub, Greengrass, Getting started, Core devices, Components, Deployments (highlighted), Classic (V1), Wireless connectivity, and Secure. The main content area has three cards: 'Create a continuous deployment to your core devices', 'Revise and copy deployments', and 'Cancel a deployment'. Below these is a 'Deployments (2)' section with a search bar and a table of deployments.

Deployment	Target name	Target type	Status	Deployment created
Deployment for smart bin	MyGreengrassCoreGroup	Thing group	Active	2 months ago
Deployment for smart waste bin	DemoWasteBin	Core device	Completed	3 days ago