

Exercise 8.1 - AWS IoT Greengrass

In Exercise 1.1, your Car Things were connecting directly with AWS IoT. What if those Car Things were only able to communicate on a closed network? What if you didn't need to send all of that telemetry data to AWS and would prefer to do some processing locally in your network and only send pieces of the data to AWS? The service that would help with this and much more is AWS IoT Greengrass.

You will use a similar scenario as Exercise 1.1, but instead of having both Car Things send their data to AWS IoT Core, they will send their data to Greengrass Core. The devices will communicate the same way using the MQTT protocol using Certificates, but the endpoint for the connection and the Certificate Authority (CA) will be the endpoint of Greengrass Core running in your Cloud9 environment and its Certificate Authority.

Although in this case, Greengrass Core will be running in the same server as Car1 and Car2, they would normally be separate. For the devices to know how to connect to Greengrass Core within the network, you can use the Greengrass Discovery API which will return all the different connectivity options (IP and Port) of your Greengrass Core.

To do processing of the data, you will use a Lambda function that you will deploy on your Greengrass Core. That's right, Lambda will be running on your server (your Cloud9 environment). In this case, the Lambda function will simply take the data that is sent by each of the devices (on the *lab/telemetry* Topic) and re-publish it to a different IoT Topic: *lab/greengrass/telemetry*.

You will configure Subscriptions within your Greengrass Core to manage how the data is flowing. By default, there is no data that can go anywhere, even between the devices connected on your Greengrass Core. To allow that, Subscriptions are made to say what the data flow should be. In this case, you will configure that all the telemetry data sent by the Cars should be sent to your Lambda function running on Greengrass Core and that the data re-published by that Lambda function should be sent to the AWS IoT Cloud.

Finally, you will be able to subscribe to the IoT Topic that your Lambda function is publishing to (*lab/greengrass/telemetry*) using the AWS IoT MQTT Client in the AWS Management Console to validate the flow of the data.

The diagram below shows the resources and data flow that you will create in this exercise.

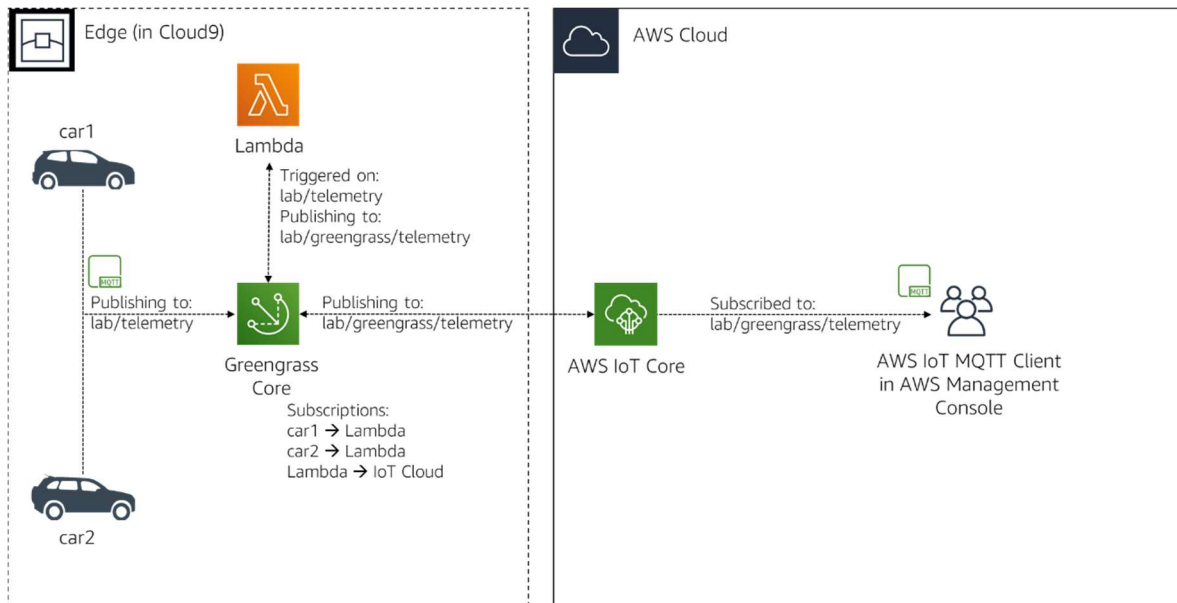


Figure 1 This exercise assumes that the resources from Exercise 1.1 haven't been deleted. If you have deleted those components, you will need to start with Exercise 1.1 again before continuing.

1. Create Greengrass Group

In this section, you will create a Greengrass Group. When doing so, you will use a wizard that will create all of the different resources required to make that Greengrass Group work: Core, Certificate and Key Pairs and Policy.

1.1 Create Greengrass Group, Core, Certificate and Policy

1. In the AWS Management Console, click **Services**, and then click **IoT Greengrass** to go to the IoT Greengrass console. It is in the same console as IoT Core.
2. Make sure you are in the same **Region** as the one you used in Exercise 1.1. It should be **Frankfurt, Ireland, N. Virginia, Ohio, Oregon or Tokyo**. You can validate that by going to the Cloud9 service and looking for the IoTOnAWS environment. If you don't see it, you aren't in the right region.
3. Click **Create a Group**.
4. If this is your first time using Greengrass, you will receive the message *Greengrass needs your permission to access other services*. AWS IoT Greengrass works with other AWS services, such as AWS IoT and AWS Lambda. Greengrass needs your permission to access these services and read and write data on your behalf. Click **Grant permission**.

5. Click **Use default creation** to automatically create a Greengrass Group, an IAM Role for Greengrass to access Lambda and other services, a Greengrass Core and a Certificate and key pair for the Core. Any of the two buttons named *Use default creation* works.
6. In the **Group Name** field, enter `labIoTGGGroup`.
7. Click **Next**.
8. Leave the **Name** field as is `labIoTGGGroup_Core`.
9. Click **Next**.
10. In the next screen, you will see the list of actions that this *default creation* wizard will create. It first creates the Group. The Greengrass Group requires a Core. This Core is similar to an IoT Thing. In fact, it will be listed in the Things list. Similar to a Thing, the Core requires a Certificate, a Private Key and a Public Key. For the Authorization piece, the Certificate will require a Policy. All of those are created for you.
11. Click **Create Group and Core**.
12. Click **Download these resources as a tar.gz** and save the file to your computer.
13. Click **Finish**.
14. There is one feature that we don't need for this Greengrass group which was enabled by default: stream manager. This is a feature that makes it easier and more reliable to transfer high-volume IoT data to the AWS Cloud. It was added automatically by the default creation and adds an extra requirement to have Java installed. To make things easier, you will remove this feature.
15. Click on the Greengrass group **labIoTGGGroup**.
16. Click **Settings**.
17. Next to **Stream manager**, click **Edit**.
18. Click **Disable**.
19. Click **Save**.
20. Click **Cores** within the Greengrass group. You can see that the `labIoTGGGroup_Core` Core is associated to this Group.
21. Expand **Manage** and click **Things**. You can see that a new Thing has been created called `labIoTGGGroup_Core`.
22. Click **labIoTGGGroup_Core**.

23. Click **Security**. You can see that a Certificate is associated to this Thing.
24. Click on the **certificate**.
25. Click **Policies**. You can see that an IoT Policy is associated to this Certificate.
26. Click on the **labIoTGGGroup_Core-...** policy. You can see the IoT Policy that was automatically created.

As you can see, a Greengrass Group Core is like a normal IoT Thing similar to car1. It has a certificate and a policy. You now have an Greengrass Group and a Core ready to be connected to.

2. Setup Greengrass Core on Cloud9 and start it

In this section, you will configure a few settings on the Cloud9 environment to be ready to install Greengrass. Those settings are a user and a group under which Lambda will run, hardlink and softlink protection and mounting Linux control groups.

You will then install Greengrass by downloading it from source and extracting it in the root (/). You will also extract all of the files from the archive that you downloaded in the previous section.

Finally, you will start Greengrass.

2.1 Start Cloud9

Your Cloud9 environment may have shut down at this point as it's supposed to automatically shutdown after 30 minutes. To restart it, follow these steps:

1. In the AWS Management Console, click **Services**, and then click **Cloud9** to go to the Cloud9 console.
2. You should see a list of *environments*. If you don't, click on the hamburger menu icon (the three parallel lines) near the top left of the screen and click on **Your environments**.
3. Click the **Open IDE** button in the **IoTOnAWS** card.
4. It may take a minute for your environment to start.

1. Add the user `ggc_user` and `ggc_group` to the Cloud9 instance under which the Lambda functions running in Greengrass will run under. In the Cloud9 **terminal** enter the following commands:

```
sudo adduser --system ggc_user
```

```
sudo groupadd --system ggc_group
```

2. Greengrass requires that the security on the instance be improved by enabling hardlink and softlink protection on the operating system. In the Cloud9 **terminal** enter the following commands:

```
echo 'fs.protected_hardlinks = 1' | sudo tee -a /etc/sysctl.d/00-defaults.conf
```

```
echo 'fs.protected_symlinks = 1' | sudo tee -a /etc/sysctl.d/00-defaults.conf
```

```
sudo sysctl --system
```

3. Extract and run the following script to mount [Linux control groups](#) (cgroups). This is an AWS IoT Greengrass dependency. In the Cloud9 **terminal** enter the following commands:

```
cd /tmp
```

```
curl https://raw.githubusercontent.com/tianon/cgroupfs-mount/951c38ee8d802330454bdede20d85ec1c0f8d312/cgroupfs-mount > cgroupfs-mount.sh
```

```
chmod +x cgroupfs-mount.sh
```

```
sudo bash ./cgroupfs-mount.sh
```

2.3 Install Greengrass

Download Greengrass and extract it. In the Cloud9 **terminal** enter the following commands:

```
cd /tmp
```

```
wget https://d1onfpft10uf5o.cloudfront.net/greengrass-core/downloads/1.10.2/greengrass-linux-x86-64-1.10.2.tar.gz

sudo tar -xzf greengrass-linux-x86-64-1.10.2.tar.gz -C /
```

2.4 Upload and extract your Greengrass Group configuration

Upload the `...-setup.tar.gz` file that you downloaded in section 1.

1. Click on the folder **IoTOnAWS** in the directory tree on the left in Cloud9.
2. Click **File > Upload Local Files...**
3. Click **Select files**.
4. Select the `...-setup.tar.gz` file in your file system that you downloaded in section 1.1.
5. Click on the **x** icon to close the *Upload Files* window.
6. Extract the `...-setup.tar.gz` archive. This will place the *Certificate* and *Private Key* of your Core in the `/greengrass/certs` folder. It will also place the `config.json` file containing all the information to connect to the AWS IoT Thing Core you created earlier in the `/greengrass/config` folder. In the Cloud9 **terminal** enter the following commands:

```
cd /tmp

mv ~/environment/*-setup.tar.gz setup.tar.gz

sudo tar -xzf setup.tar.gz -C /greengrass
```

7. Place the AWS IoT Root Certificate Authority in the `/greengrass/certs` folder. In the Cloud9 **terminal** enter the following commands:

```
cd /greengrass/certs/

sudo wget -O root.ca.pem
https://www.amazontrust.com/repository/AmazonRootCA1.pem
```

2.5 Start Greengrass

1. Start Greengrass by executing the following commands in the Cloud9 **terminal**:

```
cd /greengrass/ggc/core/
```

```
sudo ./greengrassd start
```

You should see the following output:

```
Setting up greengrass daemon
```

```
Validating hardlink/softlink protection
```

```
Waiting for up to 40s for Daemon to start
```

```
Greengrass successfully started with PID: .....
```

You have successfully setup Greengrass on your Cloud9 instance and connected it to AWS IoT Greengrass.

3. Create the Greengrass Lambda function

In this section, you will create a Lambda function. When creating a Lambda function, the Lambda service requires a Role to be passed. However, as this function will run on the Greengrass Core, this Role will never be used so you will use the default Role creation from Lambda. Greengrass can only use an alias or a publish version of your Lambda function (immutable versions). As the last step, you will publish your Lambda function.

This function has been developed in Python 2.7 especially because that's the version installed on Cloud9. It could have been developed in any of the supported language of Greengrass. At the time of writing this class, these were: Python 3.7, Java 8, Node.js 6.10, Node.js 8.10, Node.js 12.x and C/C++.

1. In the AWS Management Console, click **Services**, and then click **Lambda** to go to the Lambda console.
2. Click **Create a function**.
3. For **Function name**, enter `labIoTGreengrassLambda`.

4. For **Runtime**, select **Python 2.7**.
5. Expand **Choose or create an execution role** under **Permissions**.
6. Select the radio button next to **Create a new role from AWS policy templates**.
7. For **Role name**, enter `labIoTGGLambdaRole`. Leave the *Policy templates* empty as you are creating a Role with no specific permissions other than logging.
8. Click **Create function**.
9. Download the code of the Lambda function to your computer by clicking [here](https://aws-tc-largeobjects.s3.amazonaws.com/OTP-AWS_D5-2019/v1.0/code/exercise-3.2-greengrassLambda-python.zip).

https://aws-tc-largeobjects.s3.amazonaws.com/OTP-AWS_D5-2019/v1.0/code/exercise-3.2-greengrassLambda-python.zip

10. In the **Function code** section, click **Actions** and select **Upload a .zip file**.
11. Click **Upload**, select the **exercise-3.2-greengrassLambda-python.zip** you just downloaded and click **Save**.
12. The code has been be uploaded and Lambda will show the code in the browser. Feel free to review the code. In summary, it ensures that the data is coming from car1 or car2. It then publishes using the Greengrass IoT client the same data that was received on the *lab/greengrass/telemetry* IoT Topic.
13. At the top of the page, click **Actions > Publish new version**.
14. Click **Publish**.
15. At the top of the page, click **Actions > Create alias**.
16. Under name, enter `prod`.
17. Click **Save**.

You now have a Lambda function that's published to version 1 and an alias called *prod* that points to it. Greengrass can only point to published version of Lambda as well as aliases. However, the best practice is to use the alias as it's much easier to change the version the alias in Lambda points to than having to redo all subscriptions after they are created when they point to a specific Lambda version.

4. Add Cars, Lambda and Subscriptions to Greengrass and Deploy

In this section, you will add the 2 Car Things from Exercise 1.1 to your Greengrass Group. You will also associate the Lambda function you created to your Greengrass Group so it can be deployed to the Greengrass Core in Cloud9.

You will then create 3 Subscriptions to allow the data sent from the cars to be sent to Lambda. As well as for data sent by Lambda to be sent to AWS IoT Cloud.

- car1 data sent on the *lab/telemetry* IoT Topic is sent to the Lambda function
- car2 data sent on the *lab/telemetry* IoT Topic is sent to the Lambda function
- Lambda data sent on the *lab/greengrass/telemetry* IoT Topic is sent to AWS IoT Cloud

Finally, you will deploy the Greengrass configuration to your Greengrass Core running in Cloud9.

4.1 Cars

1. In the AWS Management Console, click **Services**, and then click **IoT Greengrass** to go to the IoT Greengrass console.
2. Click **Groups** in the left menu.
3. Click on the **labIoTGGGroup** link.
4. Click **Devices**.
5. Click **Add Device**.
6. Click **Select an IoT Thing**.
7. Click the **radio button** next to **car1**.
8. Click **Finish**. The car1 Thing has now been added.
9. Click **Add Device**.
10. Click **Select an IoT Thing**.
11. Click the **radio button** next to **car2**.
12. Click **Finish**. The car2 Thing has now been added.

4.2 Lambda

1. Click **Lambdas**.
2. Click **Add Lambda**.
3. Click **Use existing Lambda**.
4. Click the **radio button** next to **labIoTGreengrassLambda**.
5. Click **Next**.
6. Click the **radio button** next to **Alias: prod**.
7. Click **Finish**.

4.3 Subscriptions

In this section, you will configure the subscriptions for the data flow between car1, car2, Lambda and AWS IoT Cloud.

4.3.1 car1 to Lambda

You will add the Subscription from car1 to Lambda for the *lab/telemetry* Topic.

1. Click **Subscriptions**.
2. Click **Add Subscription**.
3. Under **Select a source**, click **Select**.
4. Click **Devices**.
5. Click **car1**.
6. Under **Select a target**, click **Select**.
7. Click **Lambdas**.
8. Click **labIoTGreengrassLambda**.
9. Click **Next**.
10. For **Topic filter**, enter `lab/telemetry`.
11. Click **Next**.
12. Click **Finish**.

4.3.2 car2 to Lambda

You will add the Subscription from car2 to Lambda for the *lab/telemetry* Topic.

1. Click **Add Subscription**.
2. Under **Select a source**, click **Select**.
3. Click **Devices**.
4. Click **car2**.
5. Under **Select a target**, click **Select**.
6. Click **Lambdas**.
7. Click **labIoTGreengrassLambda**.
8. Click **Next**.
9. For **Topic filter**, enter `lab/telemetry`.
10. Click **Next**.
11. Click **Finish**.

4.3.3 Lambda to AWS IoT Cloud

You will add the Subscription from Lambda to AWS IoT for the *lab/greengrass/telemetry* Topic.

1. Click **Add Subscription**.
2. Under **Select a source**, click **Select**.
3. Click **Lambdas**.
4. Click **labIoTGreengrassLambda**.
5. Under **Select a target**, click **Select**.
6. Click **IoT Cloud**.
7. Click **Next**.
8. For **Topic filter**, enter `lab/greengrass/telemetry`.
9. Click **Next**.
10. Click **Finish**.

4.4 Deploy the configuration to Greengrass Core

1. Click **Actions > Deploy**.
2. Click **Automatic detection**.
3. Click **Deployments**.
4. Under the name of the group *labIoTGGGroup*, you will see the status of the deployment. After a few seconds, you should see a green light with the message *Successfully completed*.

Your Greengrass Core on Cloud9 has now been updated with all the configuration settings you have set. It's time to use it.

5. Download car code, Start the cars and Subscribe to lab/greengrass/telemetry topic

In this section, you will download the code that connects to Greengrass. The code isn't that different from Exercise 1.1 other than it needs to query the Greengrass Discovery API to know how to connect to Greengrass and loop through all the connectivity options returned and try to connect to each of them until one works. Then, it's the same as Exercise 1.1 where the code uses MQTT to send telemetry data on the *lab/telemetry* IoT Topic.

You will then start both cars which will use the same Certificate and Key used in Exercise 1.1 to communicate to Greengrass. The Certificate Authority certificate for connecting to Greengrass Core will be downloaded by using the Greengrass Discovery API.

Finally, you will use the AWS IoT MQTT Client in the AWS Management Console to subscribe to the *lab/greengrass/telemetry* to validate that data is flowing from the cars, to Lambda and into AWS IoT Cloud.

5.1 Download the car code and install code dependencies

1. Download the code by running the following commands in the Cloud9 **terminal**.

```
cd ~/environment
```

```
wget https://aws-tc-largeobjects.s3.amazonaws.com/OTP-AWS_D5-2019/v1.0/code/exercise-3.2.js
```

```
cp exercise-3.2.js car1/
```

```
cp exercise-3.2.js car2/
```

```
rm exercise-3.2.js
```

2. The code uses the *request* library that needs to be installed. Run the following command in the Cloud9 **terminal** to install it.

```
npm install request
```

You can ignore any warnings produced.

5.2 Start the cars

1. If you **don't** have 2 Cloud9 **terminal**, click the **circled +** icon that is next to your current terminal and select **New Terminal**. You now have 2 different terminals.
2. In the left terminal, execute the following commands to start the code for car1.

```
cd ~/environment/car1
```

```
node exercise-3.2.js
```

You should see the following:

```
Data received from Greengrass Discovery API:
```

```
{
```

```
...
```

```
}  
  
Trying to connect to Greengrass Core with the address 127.0.0.1 and port 8883  
  
Connected to Greengrass Core  
  
Sending car telemetry data to AWS IoT for car1  
  
Sending car telemetry data to AWS IoT for car1  
  
...
```

Look at the JSON payload returned by the Greengrass Discovery API. You will see all the different Connectivity options (HostAddress and PortNumber) that the Greengrass Core running in Cloud9 can be reached at. You can also see the Certificate Authority certificate for your Greengrass Core.

3. In the right terminal, execute the following commands to start the code for car2.

```
cd ~/environment/car2  
  
node exercise-3.2.js
```

You should see the following:

```
Data received from Greengrass Discovery API:  
  
{  
  ...  
}  
  
Trying to connect to Greengrass Core with the address 127.0.0.1 and port 8883  
  
Connected to Greengrass Core
```

```
Sending car telemetry data to AWS IoT for car2
```

```
Sending car telemetry data to AWS IoT for car2
```

```
...
```

5.3 Subscribe to lab/greengrass/telemetry topic

1. In the AWS Management Console, click **Services**, and then click **IoT Core** to go to the IoT Core console.
2. Click **Test** in the left menu to start the AWS IoT MQTT Client.
3. In the **Subscribe topic** field, enter `lab/greengrass/telemetry`.
4. Click **Subscribe to topic**.

You should see the telemetry data sent by each car every 5 seconds. Car1 and Car2 are connecting to Greengrass which runs a Lambda function that parses the data and sends it back to AWS IoT Cloud as you can see it here.

top Greengrass

1. In the Cloud9 **terminal**, enter the following commands:

```
cd /greengrass/ggc/core/
```

```
sudo ./greengrassd stop
```