Lab no 06 part2 –AWS IoT Rules

The purpose of this lab is to be familiar with AWS services in IoT. This lab provides how to use AWS IoT Rules.

Parts: -

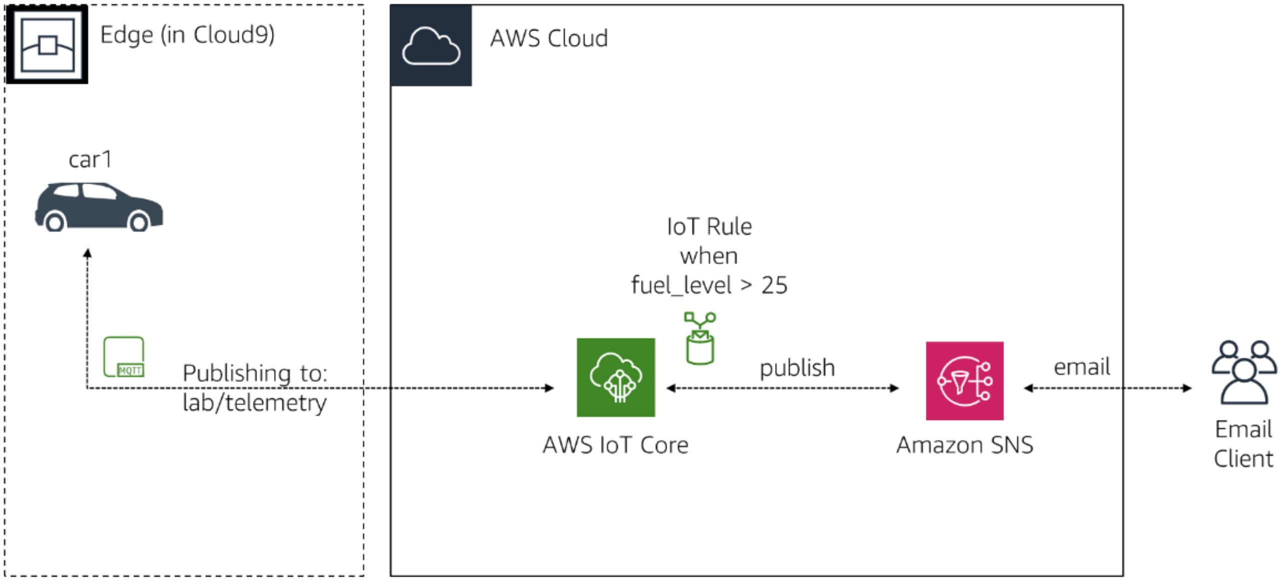
1. Create a Simple Notification Service Topic
2. Create an IAM Role.
3. Create an IoT Rule.
4. Start car1.
5. Delete the resources created in this exercise.

Required Resources

- 1 PC with Internet access.
- Account in AWS Management Console.

In this exercise, you will create a trigger that sends you an email when the fuel level attribute of a Car Thing is lower than 25%. To do this, you will create a Simple Notification Service (SNS) Topic and subscribe to it via your email. You will need to give permission to the AWS IoT service to publish a notification to SNS. Finally, you will create an IoT Rule looking for the fuel level using a SQL query publishing to your SNS Topic when it matches.

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



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 a Simple Notification Service Topic

In this section, you will create an SNS Topic, create a subscription for your email address and authorize the subscription by looking at the email.

1. In the AWS Management Console, click **Services**, and then click **Simple Notification Service** to go to the SNS dashboard.
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. On the left of the screen, click **Topics**. You may not see the side bar if you don't have any topics. To reveal it, click on the **3 lines** icon (hamburger icon) at the top left of the screen.
4. Click **Create topic**.
5. For the **Topic name** enter `labSNSFuelTopic`.
6. Click **Create topic**.
7. Click **Create subscription** to start the process of subscribing your email address to the new SNS Topic you created.
8. For **Protocol**, select **Email**.
9. For **Endpoint**, enter an email address of your choosing. You must have access to that email address to be able to confirm the subscription to the topic.
10. Click **Create subscription**.
11. Within a few minutes, you will receive an email to the address you have specified with the subject *AWS Notification - Subscription Confirmation* from the address *no-reply@sns.amazonaws.com*. **Open** that **email** and click the **Confirm subscription** link to open a web page that will confirm your subscription to the SNS Topic. You can close that web page.

You have now successfully created an SNS Topic and subscribed your email address to it.

2. Create an IAM Role

For the AWS IoT service to be able to publish a new message on the SNS Topic you created in the previous section, an IAM Role must be created. This IAM Role will need a Trust Relationship with the *iot.amazonaws.com* service principal and will require an IAM Policy to allow it to publish.

1. In the AWS Management Console, click **Services**, and then click **IAM** to go to the IAM dashboard.
2. In the left navigation menu, click **Roles**.
3. Click **Create role**.
4. Under **Select type of trusted entity**, **AWS service** should be selected.
5. Under **Or select a service to view its use cases**, select **IoT**.
6. Under **Select your use case**, select **IoT**.
7. Click **Next: Permissions**.
8. IAM Policies have already been selected for this type of IAM Role. The one we are requiring for this exercise is the *AWSIoTRuleActions* and more specifically the *sns:Publish* statement in that IAM Policy. Feel free to look at the Policies and move to the next step.
9. Click **Next: Tags**.
10. Click **Next: Review**.
11. For **Role name**, enter `labIoTRole`.
12. Click **Create role**.

You have now created the IAM Role that will be used in the next section.

3. Create an IoT Rule

In this section, you will create an IoT Rule that will use a SQL statement to watch for a *fuel_level* lower than 25%. If this happens, it will publish to the SNS Topic you have created in a previous section.

1. In the AWS Management Console, click **Services**, and then click **IoT Core** to go to the IoT console.
2. In the left navigation menu, expand **Act** and click **Rules**. This is where you configure rules in IoT Core.
3. Click **Create a rule**.
4. For **Name**, enter `labFuelRule`.
5. Under **Set one or more actions**, click **Add action**.
6. Select **Send a message as an SNS push notification**.
7. Click **Configure action**.
8. Under **SNS target**, click **Select**.
9. Next to **labSNSFuelTopic**, click **Select**.
10. Under **Message format**, select **RAW**.
11. Under **Choose or create a role to grant AWS IoT access to perform this action**, click **Select**.
12. Next to **labIoTRole**, click **Select**.
13. Click **Add action**.
14. In the **Rule query statement** box, replace everything with the following:

```
SELECT
   'The fuel level for ' + device + ' is currently at ' + round(fuel_level) + '%. The car is at '
   + longitude + ' of longitude and ' + latitude + ' of latitude.' AS message
FROM 'lab/telemetry'
WHERE
   fuel_level < 25
```

This rule is using the SQL language to look for the *fuel_level* attribute in the *lab/telemetry* IoT Topic (remember that this is where the Car Things send their telemetry data). If the *fuel_level* is below 25, it creates a statement providing the name of the Car Thing using the *device* attribute and using the *round* function to round the *fuel_level* decimal to the nearest integer. It also provides the coordinate for the car so it can be found to be refueled.

15. Click **Create rule**.

You have now created an IoT Rule that is waiting for the fuel_level to be under 25 to send a notification.

4. Start car1

In this section, you will connect to the Cloud9 environment and start the car1 Thing again so telemetry data can be sent.

4.1 Start Cloud9

Your Cloud9 environment has probably 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.

4.2 Start car1

1. In the Cloud9 terminal, start car1 by executing the following commands.

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

```
node exercise-1.1.js
```

You should see the following:

```
Connected to AWS IoT

Sending car telemetry data to AWS IoT for car1

Sending car telemetry data to AWS IoT for car1

...
```

The car1 Thing has now been started and sends telemetry data every 5 seconds. The *fuel_level* attribute is set to a random decimal number between 0 and 100. Since the IoT Rule created is looking for the *fuel_level* to be lower than 25, there is only 1 chance on 4 for the IoT Rule to be triggered. This is on purpose so that you don't receive too many emails. Within a minute or so, you should receive an email from *no-reply@sns.amazonaws.com* with the subject *AWS Notification Message* and with a message like the following: *{"message":"The fuel level for car1 is currently at 17%. The car is at -77.133578 of longitude and 39.122229 of latitude."}*. You will probably start receiving many more, so continue to the next step to stop these emails.

5. Delete the resources created in this exercise

In this section, you will remove all the different resources created as part of this exercise that won't be required for the other exercises. The resources from Exercise 1.1 will still be there and should remain in place. If you would like to remove the resources from Exercise 1.1, refer to that exercise.

5.1 Stop car1

1. **Press Ctrl-c** in the Cloud9 **terminal** to stop car1 from interacting with AWS IoT.

5.2 Delete IoT Rule

1. In the AWS Management Console, click **Services**, and then click **IoT Core** to go to the IoT console.
2. Expand **Act** and click **Rules** in the left menu.
3. Click on the **3 dots** next to the **labFuelRule**.
4. Select **Delete**.
5. Click **Yes, continue with delete**.

5.3 Delete SNS Topic

1. In the AWS Management Console, click **Services**, and then click **Simple Notification Service** to go to the SNS console.
2. Click **Topics** from the left menu.
3. Click the radio button next to **labSNSFuelTopic**.
4. Click **Delete**.
5. In the input box, enter `delete me`.
6. Click **Delete**.
7. Click **Subscriptions**.
8. In the **Search** field, enter `labSNSFuelTopic`.
9. Click the radio button of that subscription.
10. Click **Delete** and **Delete** again.

5.4 Delete the IAM Role

1. In the AWS Management Console, click **Services**, and then click **IAM** to go to the IAM console.
2. Click **Roles** from the left menu.
3. In the **Search** field, enter `labIoTRole`.
4. Click the **checkbox** next to **labIoTRole**.

5. Click **Delete role**.
6. Click **Yes, delete**.

5.5 Stop the Cloud9 environment

The Cloud9 environment will automatically shut down after 30 minutes of inactivity. For your Cloud9 environment to be considered inactive, you need to close the browser tab. All the settings will be saved.

1. Close the **browser tab** where your environment was running.

As the operating system is Amazon Linux, you are billed by the second during those 30 minutes of inactivity. If you are under the free tier, this would be covered. If you are no longer under the free tier, you can force a stop of the EC2 instance that runs your Cloud9 environment. This will have no effect on the future exercises.

1. In the AWS Management Console, click **Services**, and then click **EC2** to open the EC2 console.
2. Click **Instances** in the left menu.
3. Select the EC2 Instance that has a name that starts with **aws-cloud9**.
4. Click **Actions > Instance State > Stop instance**

Note:
Labs are from Course: AWS IoT: Developing and Deploying an Internet of Things
https://www.edx.org/course/aws-iot-developing-and-deploying-an-internet-of-th