

Week 2 Notes and Resources

KEY TOPICS

Rules, Device Shadows, and the SDKs

This week, we focused on using Rules and Device Shadows to interact with your IoT devices. In the hands-on exercises, you created rules to notify you when the fuel level in your "car" fell below a specific value. You completed an a second exercise where you used the Device Shadow to indirectly interact with your "car" as being connected to an IoT device at all times is not assured or optimal.

Finally, we looked at using the AWS IoT APIs and SDK to allow your two "cars" to interact, sending messages between them, using your own code.

Rules

The Rules Engine makes it possible to build IoT applications that gather, process, analyze and act on data generated by connected devices at global scale without having to manage any infrastructure. The Rules Engine evaluates inbound messages published into AWS IoT Core and transforms and delivers them to another device or a cloud service, based on business rules you define. A rule can apply to data from one or many devices, and it can take one or many actions in parallel.

The Rules Engine can also route messages to AWS endpoints including [AWS Lambda](#), [Amazon Kinesis](#), [Amazon S3](#), [Amazon SageMaker](#), [Amazon DynamoDB](#), [Amazon CloudWatch](#), [Amazon Simple Notification Service \(SNS\)](#), [Amazon Simple Queue Service \(SQS\)](#), [AWS IoT Analytics](#), [Amazon Elasticsearch Service](#) with built-in [Kibana](#) integration, and [AWS Step Functions](#). External endpoints can be reached using AWS Lambda, Amazon Kinesis, Amazon SNS, and AWS Step Functions.

You can author rules within the management console or write rules using a SQL-like syntax. Rules can be authored to behave differently depending upon the content of the message. For example, if a temperature reading exceeds a certain threshold it could trigger a rule to transmit data to AWS Lambda. Rules can also be authored to take into account other data in the cloud, such as data from other devices. For example you could say take an action if this temperature is more than 15% higher than the average of 5 other devices.

The Rules Engine provides dozens of available functions that can be used to transform your data, and it's possible to create infinitely more via AWS Lambda. For example, if you're dealing with a wide range of values you could take the average of incoming numbers. Rules can also trigger the execution of your Java, Node.js or Python code in AWS Lambda, giving you maximum flexibility and power to process device data.

To learn more read the [Rules Engine section](#) of the AWS IoT Core user guide.

Device Shadows

With AWS IoT Core, you can create a persistent, virtual version, or Device Shadow, of each device that includes the device's latest state so that applications or other devices can read messages and interact with the device. The Device Shadow persists the last reported state and desired future state of each device even when the device is offline. You can retrieve the last reported state of a device or set a desired future state through the API or using the rules engine.

The Device Shadow makes it easier to build applications that interact with your devices by providing always available REST APIs. In addition, applications can set the desired future state of a device without accounting for the device's current state. AWS IoT Core will compare the difference between the desired and last reported state, and command the device to make up the difference.

The AWS IoT Device SDK makes it easy for your device to synchronize its state with its Device Shadow, and to respond to desired future states set via the Device Shadow.

The Device Shadow lets you store the state of your devices for up to a year for free. The Device Shadow persists forever if you update them at least once per year, otherwise they expire.

To learn more, see the [Device Shadow section](#) of the AWS IoT Core user guide.

AWS IoT APIs and SDKs

The [AWS IoT Developer guide](#) provides the latest detailed instructions on how to work with the AWS IoT services, including tutorials, how to use the SDKs and CLI with AWS IoT.

As detailed in the videos and exercise, the Github repository for the [Javascript/node.js SDK](#) can be used to connect to AWS IoT from a device. Alternatively, you could use Python. The Github repository for the [Python SDK] (<https://github.com/aws/aws-iot-device-sdk-python>) can also be used to connect to AWS IoT from a device.

WHAT YOU ACCOMPLISHED THIS WEEK

- You created an IoT Rule to process data from your "car" device.
- You used the AWS IoT Thing Shadow to interact with your "car".
- You wrote some code and had two "car" Things, interact with each other.