Robert Morris University Data Science Capstone
AWS Cloud Connected Smart Space

by Patrick Burke | April 2023 | Amazon IoT Core, Amazon SageMaker, AWS Lambda, AWS S3, Apache Superset, Docker

Internet-of-Things (IoT) devices are becoming a ubiquitous part of life in America. From Amazon Alexa-enabled devices powering smart home solutions like Amazon Smart Thermostats, to Google's Nest Audio providing intelligent audio at home, IoT devices project to only continue to grow in the near future. Per Statista.com, by 2030 IoT devices 'are projected to almost triple from 9.7 billion in 2020 to more than 29 billion [..., with t]he highest number of IoT devices [...] found in China with around 5 billion consumer devices.'[1] As the number of internet-connected devices grow, the need for efficient pipelines to capture, analyze, and act on the data each device produces will likewise grow.

As a demonstration of how IoT device data can be effectively captured, analyzed, and acted upon, we will create a Smart Space device utilizing cloud offerings powered by AWS. From this Smart Space project, we will acquire physical data from sensors located within the M5Stack Core2 for AWS device, synchronize states between the edge and the cloud, build a serverless application within AWS to control the edge device, and transform room occupancy and temperature data into actionable insights with machine learning.

Overview:

In this report, we will be utilizing the M5Stack Core2 for AWS IoT Kit to develop our Smart Space. This device will be used to demonstrate how physical edge devices can be connected and controlled via the AWS platform. To guide our architecture, we will be keeping a few major points in mind:

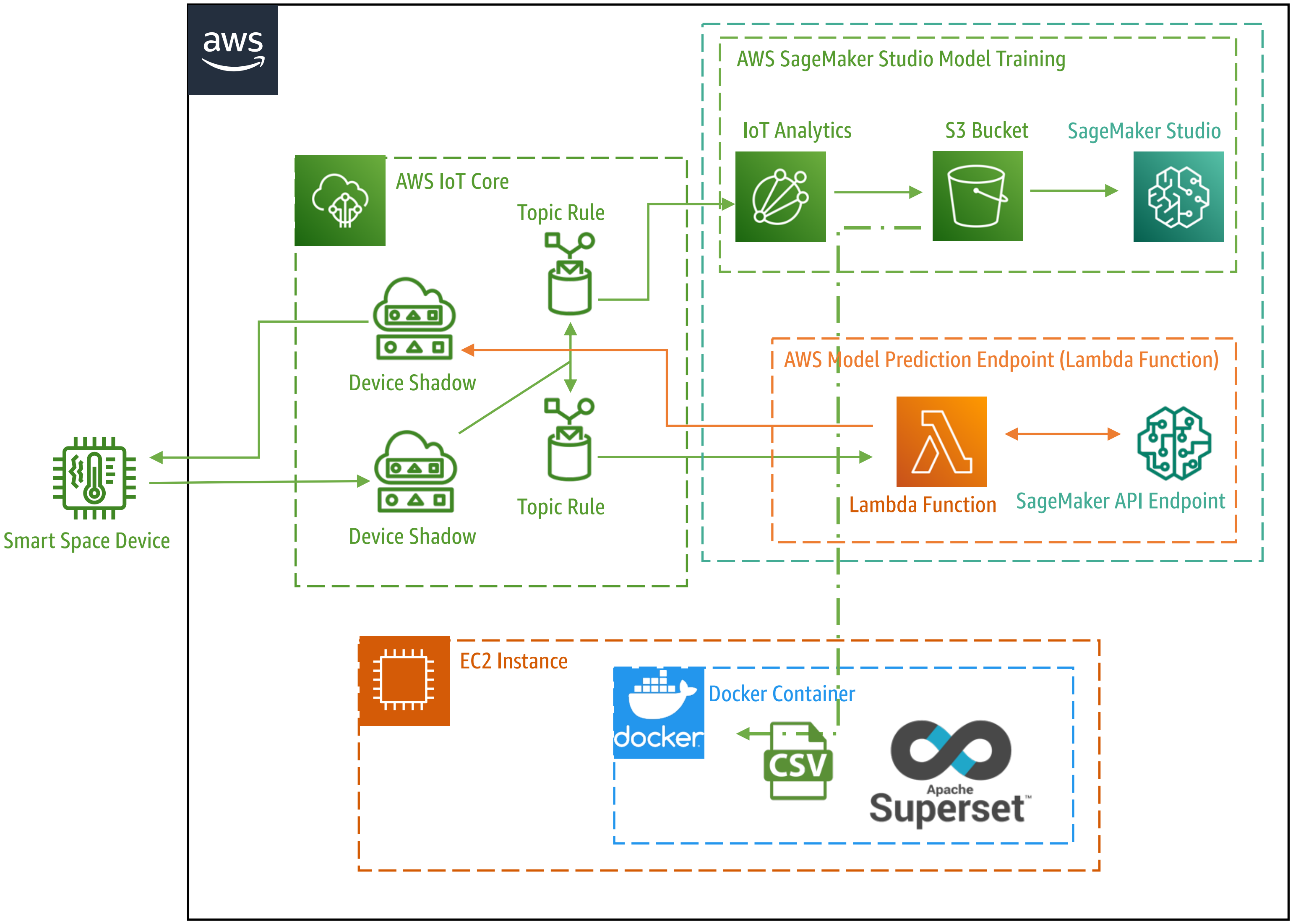
- **Serverless architecture:** our solution will not rely on local servers to power any aspect of our pipeline or analytics, helping minimize any potential horizontal scaling concerns.
- **Low latency:** our device should be able to send and receive information at a near real time rate, with minimal time for processing.
- **Cost efficient:** our cloud solution should minimize costs, with components carefully considered.
- **Understandable results:** our solution should also have an understandable result dataset, to view potential issues with predictions.

The diagram below shows a high-level architecture of our resulting solution:

[1] Sujay Vailshery, Lionel. "IoT connected devices worldwide 2019-2030." Statista. Nov 22, 2022. https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

Robert Morris University Data Science Capstone
AWS Cloud Connected Smart Space

by Patrick Burke | April 2023 | Amazon IoT Core, Amazon SageMaker, AWS Lambda, AWS S3, Apache Superset, Docker



Walkthrough:

The primary datastore for this solution is an Amazon S3 bucket, due to the scalability and high availability S3 provides. The three main components of our architecture are the IoT Core data ingest and distribution block, the SageMaker Studio training and prediction blocks, and the data visualization block, an EC2 instance powering our Apache Superset dashboard, running within a Docker container. Let's look at each section in more detail:

- **AWS IoT Core (Data Ingest and Synchronization):** Our first block is how we communicate with the IoT edge device. Our device is connected to AWS via shadow reports. The device publishes these reports to **AWS IoT Core**, which has a shadow report service that can trigger different events on a specific topic. These topics can accomplish things like forward the raw, unprocessed data to **AWS IoT Analytics** for training a model, or an AWS Lambda function to use the data for occupancy prediction. AWS IoT Analytics cleans our data into a useable format for machine learning model development and prediction generation.

- **Amazon SageMaker Studio (Model Creation & API Endpoint):** Amazon SageMaker Studio is the primary platform for creating a new machine learning model from the acquired device telemetry data. Within Amazon SageMaker Studio, device telemetry is consumed from an S3 bucket and used as a training set, then deployed as a model with an API endpoint. Once trained, the IoT Core rules engine will invoke an **AWS Lambda** function that uses this API endpoint to forward classification results. AWS Lambda is an event-driven platform that runs programs in response to events, like receiving device shadow reports from AWS IoT Analytics.

- **Amazon EC2 (Data Visualization with Apache Superset):** To complete our architecture, we will also create a way to visualize our training data stored in the S3 Bucket. To accomplish this, we first provision a new **AWS EC2** instance, for virtual cloud computing. This instance will be used to run Docker, a lightweight container engine that can help running application containers. Finally, we can run an **Apache Superset** container. Apache Superset is an open-source data visualization application. Our S3 data can be exported to a .csv file, which can then be displayed in Apache Superset to explore if there are any potential outliers or incorrect datapoints, and what an occupied room's data appears like as opposed to an unoccupied one.

Results:

Our resulting device will detect and predict changes in room occupancy, and control a (non-existant) HVAC system to change room temperature for when a room is occupied or empty.

Final Thoughts:

To summarize, our architecture is a simple example of a solution to how best capture and act on IoT device telemetry data. With our architecture, the Core2 device sends data to the cloud, which receives the data via a device shadow. This data is then relayed to either IoT Anlaytics to clean and use for model creation with Amazon SageMaker Studio, or a Lambda Function to transform the data into an actionable classification and confidence level with the developed SageMaker model API. To view our training data, we can export the data into a .csv file and visualize with Apache Superset.

From this project, a few lessons to take away are the simplicity of developing an intelligent device using available cloud offerings, the number of resources available to learn, and the usefullness and useability of open-source software. Using the offerings provided by AWS, a full IoT pipeline can be developed with relative ease.