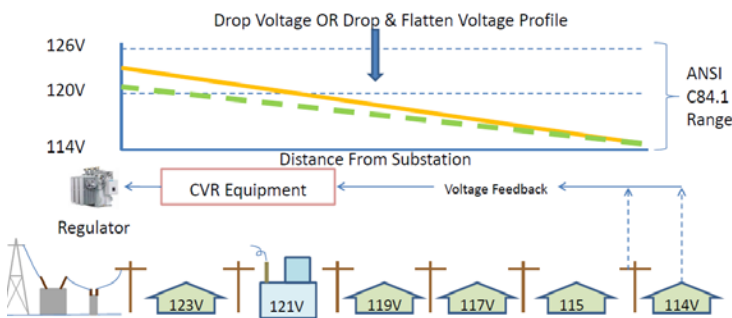


Cloud-based IOT Server for Energy Savings

Team: W205-3: Matthew Burke, Jan Forslow, Vyas Swaminathan and Xiao Wu

Project Idea and Data Acquisition



The idea is to take the particular energy savings use case of conservation voltage reduction (CVR) and show an architecture that can scale to hundreds of thousands of measuring points using AWS cloud. The goal with CVR is to optimize the voltage level on the feeder line while not expose customers at the end of the line to flickering (low voltage). This is achieved by using the SmartMeters as sensors and by sending control signals to regulators to adjust the voltage levels as shown in Figure 1.

Figure 1. Example Feeder Line with CVR activated.

There are on average 300-600 households per feeder line of which ~ 40-50 SmartMeters will need to be used as sensors (also called bellwether meters) and report back voltage levels every 5-10 minutes. For this project, we are creating a simulator to simulate the streaming of this data. The simulator is sending messages containing voltage level, detected most recent (second level) voltage change, and SmartMeter location (Lat./Long. and association to feeder line). We know from industry standard that the voltage on the feeder should not exceed 3% of 120V (123 – 117V). Therefore we are using an automatic feedback loop to the device simulator to readjust if the voltage level is below 117V or above 123V in any message. Randomization is applied in the simulator to create voltage changes leading to adjustments.

Comparing Architectures

As a group, we discussed two possible architectures:

1) AWS IoT -> Kinesis -> ElasticSearch (and stateless AWS Lambda functions)

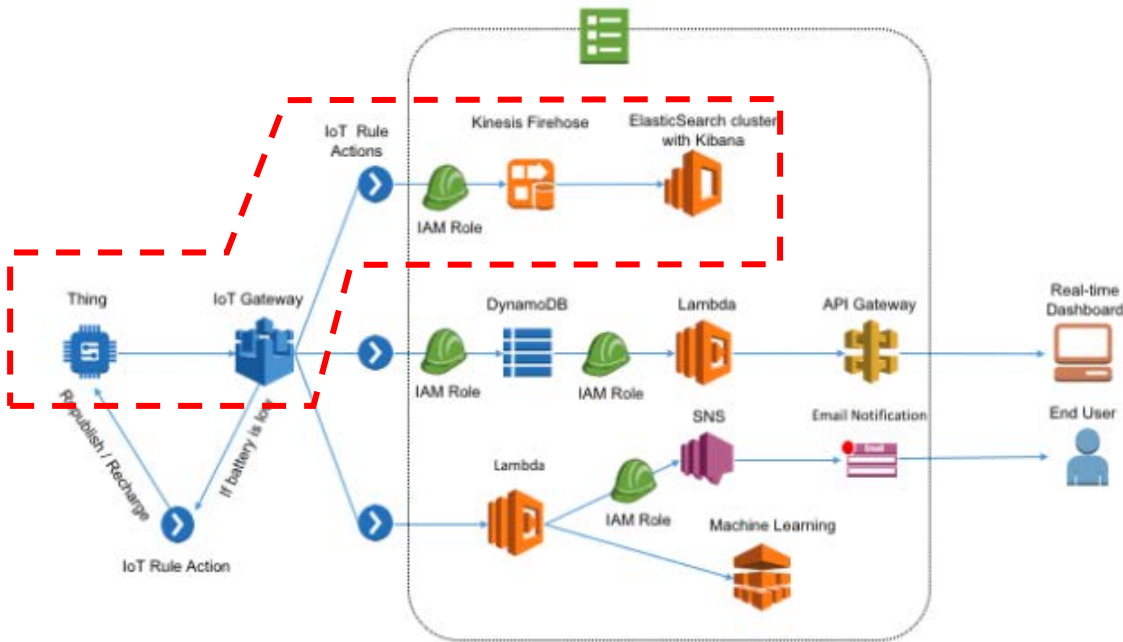


Figure 2. AWS IoT Reference Architecture.

As shown in Figure 2, the main components within the AWS IoT references architecture for near real-time analytics are:

- [AWS IoT](#) (Amazon Internet of Things) will collect and analyze data from internet-connected devices and sensors and connects that data to AWS cloud applications.
- [Amazon Kinesis Firehose](#) is a platform for streaming data on AWS, offering powerful services to make it easy to load and analyze streaming data.
- [Amazon Elasticsearch](#) is a popular open-source search and analytics engine for use cases such as data analytics, log analytics, real-time application monitoring, and clickstream analytics.

2) Kafka -> Storm -> MongoDB



Figure 3. Apache Open-source Real-time Streaming Architecture.

- [Kafka](#) is a distributed streaming platform. It allows publishing, storing, and processing streams of records.
- [Storm](#) allows for reliably processing unbounded streams of data, doing real-time processing, and integrates well with Kafka.

Selecting Architecture

For this project, we decided to use the first architecture, given the following reasons for future commercialization:

- 1) With AWS IoT, applications can keep track of and communicate with all devices, all the time, even when they aren't connected.
- 2) Kinesis was modeled after Kafka, but in addition allows for zero-administration (not managing infrastructure).
- 3) Stateless Lambda statements automatically matches capacity to request rate, which allows purchasing compute in 100ms increments.
- 4) ElasticSearch allows for flexible data visualization using in-built Kibana.