

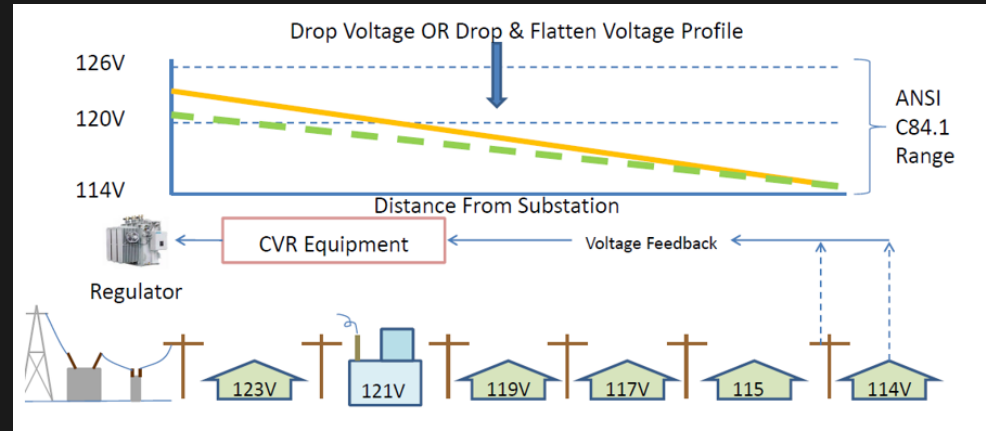
Cloud-based **IoT Server** for energy savings

W205-3 | Matthew Burke, Jan Forslow, Vyas Swaminathan, Xiao Wu

Use Case/Scope

CVR (Conservation Voltage Reduction) AWS IOT Server

Replace on-premise RDBMS systems that do not scale with frequent input data streams



- 300-600 households per feeder line of which ~ 40 SmartMeters are used as sensors (also called bellwether meters).
- These bellwether meters will report back voltage levels every 5 minutes.
- Voltage on the feeder should not exceed 3% of 120V (123 – 117V).

Leverage AWS IoT Architecture

Data Ingestion:

- IOT Gateway

Data Storage:

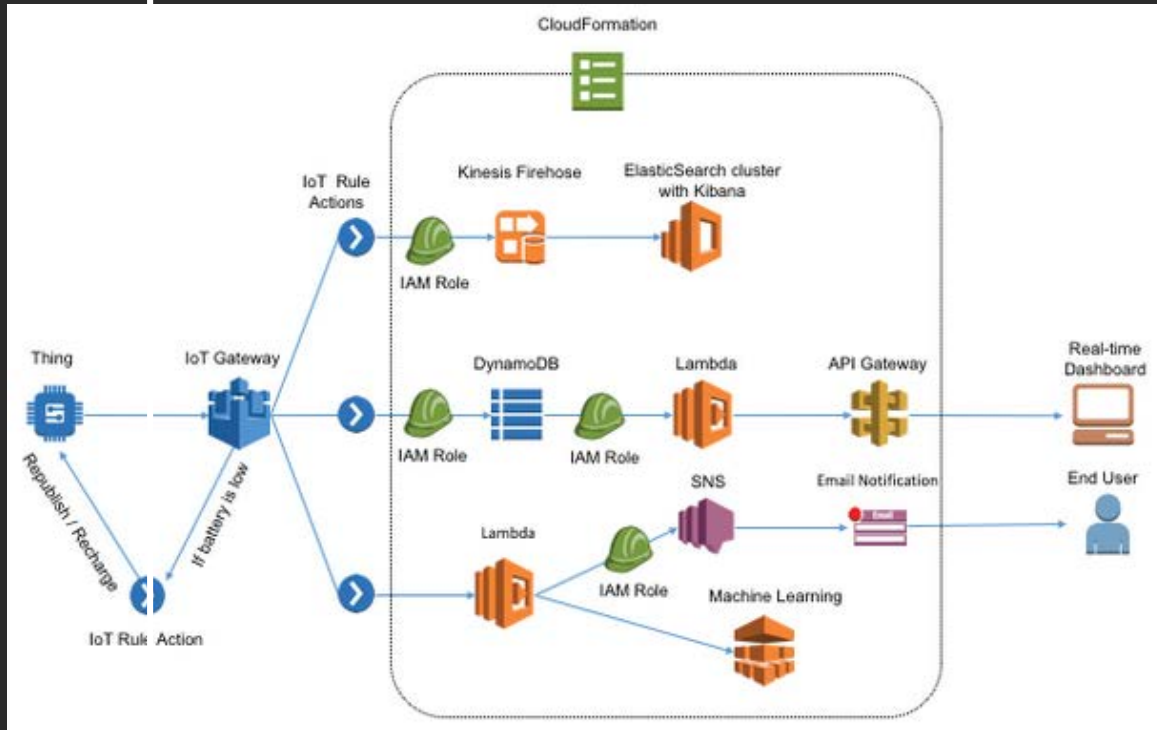
- DynamoDB

Stream Processing

- Kinesis/ElasticSearch

Data Exploration:

- Kibana and AWS ML



Repeatable System Setup

AWS CloudFormation:

- 1,000+ lines of JSON statements
- Creates a new IOT Server instance in 15-20 minutes

▼ Outputs

Key	Value	Description
IpAddressEc2DeviceSimulator	34.194.16.95	Public IP of the EC2 Device Simulator
SnsTopicArn	arn:aws:sns:us-east-1:129288622091:SmartMeter-SmartMeterSNSTopic-LY1RSPANMYEP	ARN of the SNS topic
S3BucketName	smartmeter-smartmeters3bucket-bw2opj77j5jo	Name of the S3 bucket for Smart Meter

▼ Resources

Logical ID	Physical ID	Type	Status	Status Reason
ApiGatewayLambdaInvo...	SmartMeter-ApiGatewayLambdaInvokePermission-1BD4LWV5ZCO4C	AWS::Lambda::Permission	CREATE_COMPLETE	
ApiGatewayToLambdaIn...	SmartMeter-ApiGatewayToLambdaInvokePermission-5HTZ1KZBJ9ND	AWS::Lambda::Permission	CREATE_COMPLETE	
AttachGateway	Smart-Attac-1UX3FO9T38H0K	AWS::EC2::VPCGatewayAttach...	CREATE_COMPLETE	
AwsIotRepublishRole	SmartMeter-AwsIotRepublishRole-ZE8OV6X2DQF0	AWS::IAM::Role	CREATE_COMPLETE	
AwsIotToDynamoRole	SmartMeter-AwsIotToDynamoRole-SW79EFVRT9M	AWS::IAM::Role	UPDATE_COMPLETE	
	SmartMeter-AwsIotToFirehoseDeliveryStreamRole-11MS	AWS::IAM::Role	CREATE_COMPLETE	

Data Sources/Data Generation

- External SmartMeter data was not available with necessary Voltage information
- Created Simulator in Python using MQTT for message push
- Test config had 20 Feeder Lines with 40 Bellwether meters each
- Messages sent every 500 msec (i.e. 6.67 minutes for a single meter)
- Stepdown of Voltage data per hop on Feeder Line with some randomization
- Inserting anomaly Voltage value with a possibility of 0.1%
- Test runs were done for approx. 25 minutes each

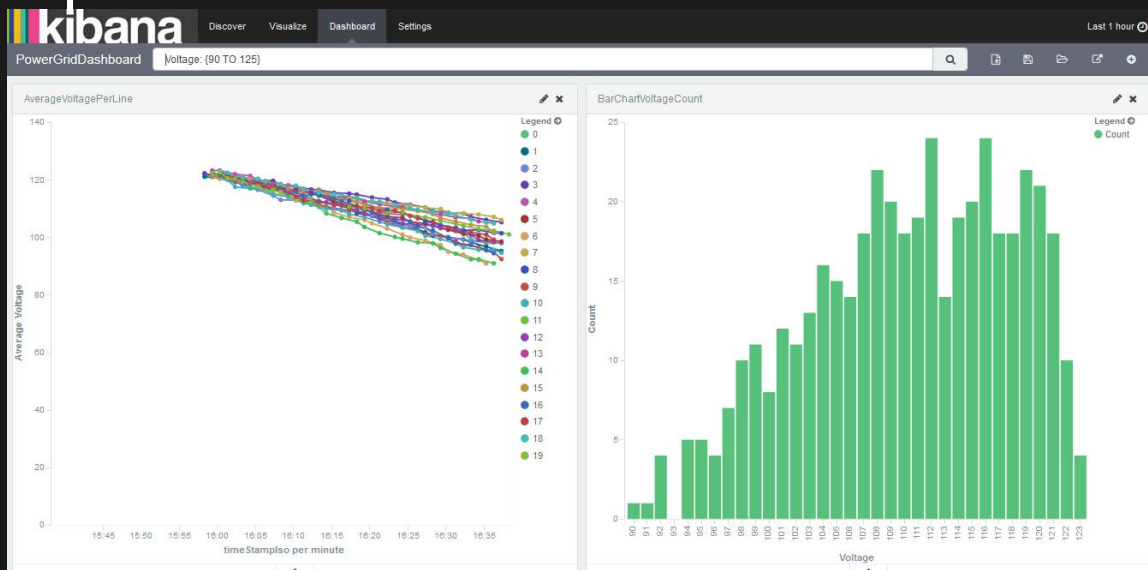
Streaming Analytics with Elasticsearch & Kibana

Anomalies Dashboard:

- Anomalies Count per Feeder
- DrillDown by Timestamp

PowerGrid Dashboard:

- Voltage per Line over Time
- Voltage Measurement Count
- Voltage Statistics Table



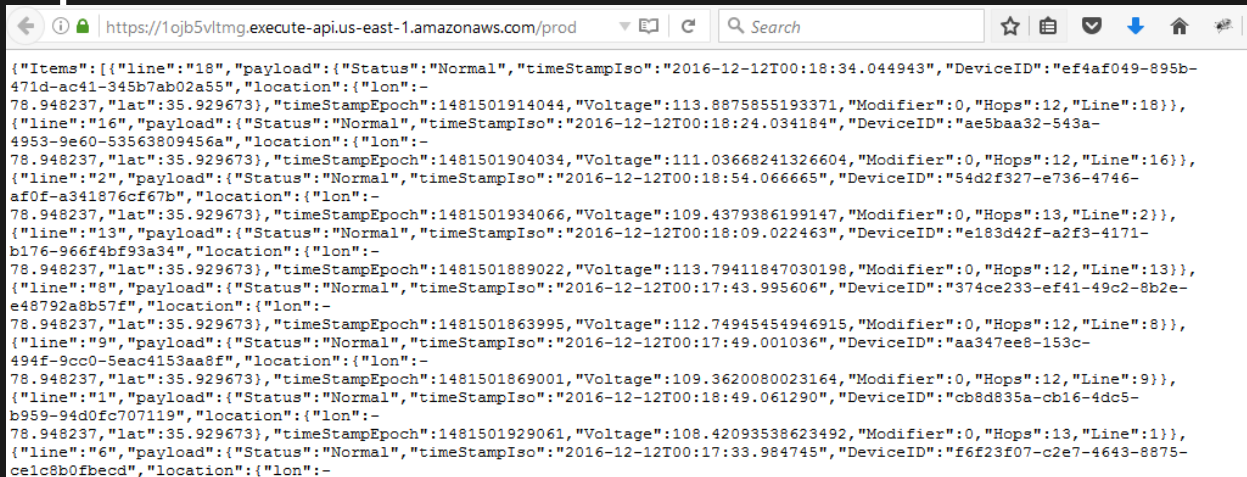
Storing Data and Making it Available through API

DynamoDB:

- TimeSeriesTable
- DeviceStatusTable

API Gateway:

- Device data in JSON format
- Future Web Dashboard



The screenshot shows a web browser window with the address bar displaying the URL: `https://1ojb5vltmg.execute-api.us-east-1.amazonaws.com/prod`. The page content is a JSON array of items, each representing a device status record. The JSON structure is as follows:

```
{
  "Items": [
    {
      "line": "18",
      "payload": {
        "Status": "Normal",
        "timeStampIso": "2016-12-12T00:18:34.044943",
        "DeviceID": "ef4af049-895b-471d-ac41-345b7ab02a55",
        "location": {
          "lon": -78.948237,
          "lat": 35.929673
        },
        "timeStampEpoch": 1481501914044,
        "Voltage": 113.8875855193371,
        "Modifier": 0,
        "Hops": 12,
        "Line": "18"
      }
    },
    {
      "line": "16",
      "payload": {
        "Status": "Normal",
        "timeStampIso": "2016-12-12T00:18:24.034184",
        "DeviceID": "ae5baa32-543a-4953-9e60-53563809456a",
        "location": {
          "lon": -78.948237,
          "lat": 35.929673
        },
        "timeStampEpoch": 1481501904034,
        "Voltage": 111.03668241326604,
        "Modifier": 0,
        "Hops": 12,
        "Line": "16"
      }
    },
    {
      "line": "2",
      "payload": {
        "Status": "Normal",
        "timeStampIso": "2016-12-12T00:18:54.066665",
        "DeviceID": "54d2f327-e736-4746-af0f-a341876cf67b",
        "location": {
          "lon": -78.948237,
          "lat": 35.929673
        },
        "timeStampEpoch": 1481501934066,
        "Voltage": 109.4379386199147,
        "Modifier": 0,
        "Hops": 13,
        "Line": "2"
      }
    },
    {
      "line": "13",
      "payload": {
        "Status": "Normal",
        "timeStampIso": "2016-12-12T00:18:09.022463",
        "DeviceID": "e183d42f-a2f3-4171-b176-966f4bf93a34",
        "location": {
          "lon": -78.948237,
          "lat": 35.929673
        },
        "timeStampEpoch": 1481501889022,
        "Voltage": 113.79411847030198,
        "Modifier": 0,
        "Hops": 12,
        "Line": "13"
      }
    },
    {
      "line": "8",
      "payload": {
        "Status": "Normal",
        "timeStampIso": "2016-12-12T00:17:43.995606",
        "DeviceID": "374ce233-ef41-49c2-8b2e-e48792a8b57f",
        "location": {
          "lon": -78.948237,
          "lat": 35.929673
        },
        "timeStampEpoch": 1481501863995,
        "Voltage": 112.74945454946915,
        "Modifier": 0,
        "Hops": 12,
        "Line": "8"
      }
    },
    {
      "line": "9",
      "payload": {
        "Status": "Normal",
        "timeStampIso": "2016-12-12T00:17:49.001036",
        "DeviceID": "aa347ee8-153c-494f-9cc0-5eac4153aa8f",
        "location": {
          "lon": -78.948237,
          "lat": 35.929673
        },
        "timeStampEpoch": 1481501869001,
        "Voltage": 109.3620080023164,
        "Modifier": 0,
        "Hops": 12,
        "Line": "9"
      }
    },
    {
      "line": "1",
      "payload": {
        "Status": "Normal",
        "timeStampIso": "2016-12-12T00:18:49.061290",
        "DeviceID": "cb8d835a-cb16-4dc5-b959-94d0fc707119",
        "location": {
          "lon": -78.948237,
          "lat": 35.929673
        },
        "timeStampEpoch": 1481501929061,
        "Voltage": 108.42093538623492,
        "Modifier": 0,
        "Hops": 13,
        "Line": "1"
      }
    },
    {
      "line": "6",
      "payload": {
        "Status": "Normal",
        "timeStampIso": "2016-12-12T00:17:33.984745",
        "DeviceID": "f6f23f07-c2e7-4643-8875-ce1c8b0fbecd",
        "location": {
          "lon": -78.948237,
          "lat": 35.929673
        },
        "timeStampEpoch": 1481501893475,
        "Voltage": 108.42093538623492,
        "Modifier": 0,
        "Hops": 13,
        "Line": "6"
      }
    }
  ]
}
```

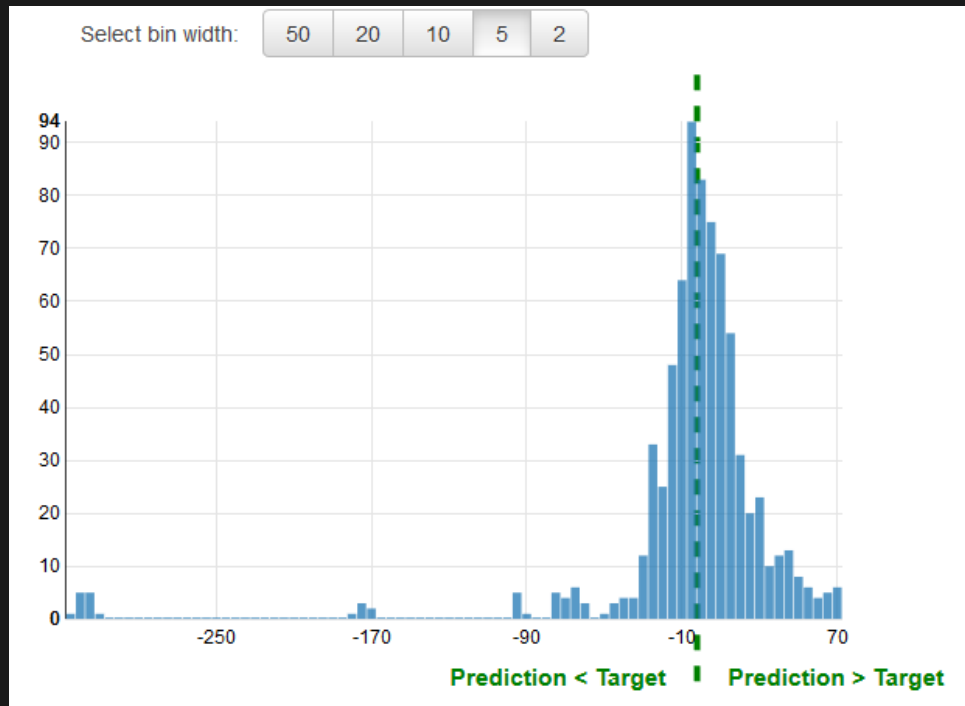
Anomaly Detection using Machine Learning

AWS ML:

- 25 min and 2,500 training data
- Numerical Regression

SNS Notification Service:

- Email alert based on deviation to deviation to prediction



Data Processing and Cost Assessment

EC2 t2 Medium:

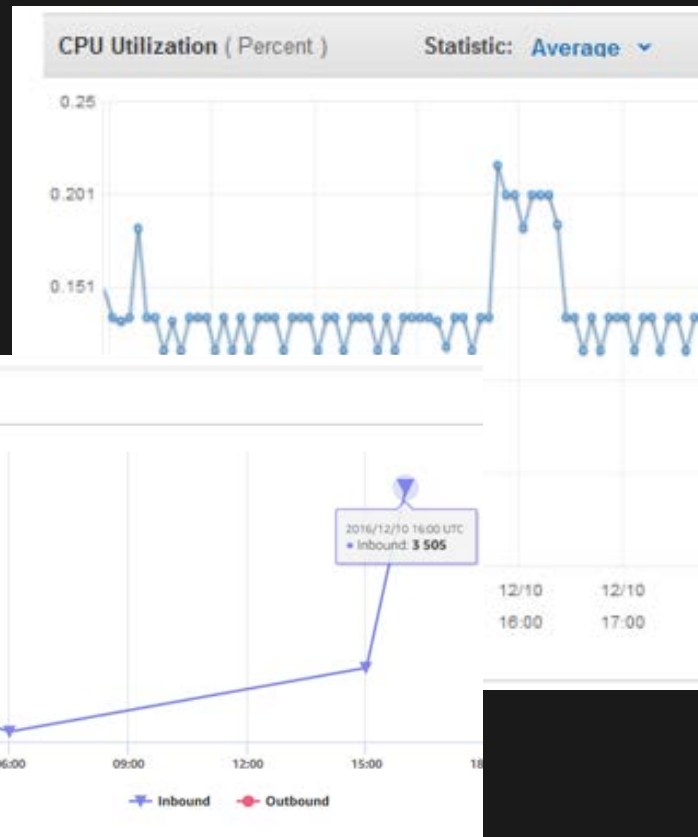
- Only DynamoDB, S3
- Less than 0.2% CPU utilization

AWS IOT Gateway:

- \$300/year (6 million messages)

Lambda functions:

- Server-less; on any AWS resource when executed



Future Extensions

Data Acquisition:

- Finetuning the Simulator to have less Voltage swings
- Add location information
- Enable automatic control loop from IOT Server
- Validate with real SmartMeters and Voltage Regulators

IOT Server:

- Enable DR and Elastic Instance Scaling using AWS Autoscaling Groups
- Run system with 100,000 of bellwether meters
- More advanced control algorithms
- Complete a web dashboard

References

1. [Tennessee Valley Authority Voltage Optimization Program](#)
2. [AWS IoT Pilot Light Hack Day](#)