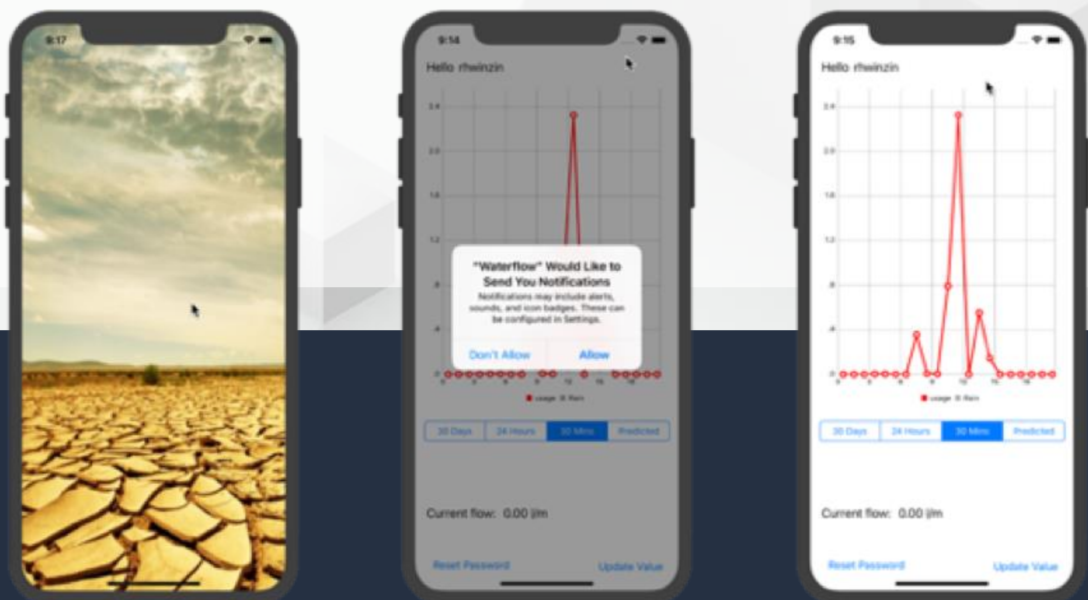
# AWS Architecture Monthly


## Internet of Things

June 2019

# Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions, or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

# Note from the Editor

The other day, I asked Alexa, “What is IoT?” and she replied: “Internet of Things is usually defined as a proposed internet-like structure connecting everyday physical objects equipped with rfid or similar tags.”

Of course she’s right (she usually is, except when asking her how to pronounce my name), but if a curious, non-technical person heard that, would they know what “RFID” means? If you’ve subscribed to this Kindle magazine, chances are good that you DO know what RFID refers to in the world of IoT. (But, in case not: Radio Frequency IDentification uses radio signals to send out information.)

Now that we know as much as Alexa, let’s move on.

Every day in the news we hear about connected tech – from our smart home speakers (the Amazon Echo, Apple HomePod, and Google Home Series) to our cars, refrigerators, alarm systems, and cameras. In other words, IoT is HOT…and it’s everywhere.

In June’s edition, we gathered architectural best practices from all over AWS, and we’ve made sure that a broad audience can appreciate it. Additionally, we’ve provided links to real-world video examples of how companies are putting AWS technologies to use.

- **Training**: The IoT Foundation Series is a big, beefy set of training modules that offers a broad experience across IoT core and its related services.
- **Quick Start**: AWS IoT Camera Connector on AWS
- **Blog post**: Using AWS IoT to Create a Smart Home Water-Monitoring Solution
- **Reference Architecture**: Processing IoT Time Series Data on AWS
- **Solution**: IoT Device Monitoring with Kinesis
- **Video**: This IS My Architecture -- Democratizing LoRaWAN and IoT with The Things Network
- **Whitepaper**: Securing Internet of Things (IoT) with AWS

We hope you find this edition of Architecture Monthly useful and we’d like your feedback. Please give us a star rating and your comments on Amazon. You can also reach out to aws-architecture-monthly@amazon.com anytime.

- Annik Stahl, Editor

# Training

## Internet of Things Foundation Series – Digital Training

(Available online at: http://bit.ly/AWS-IoT-Training )

### Description

The Internet of Things (IoT) Foundation Series comprises a set of courses that offer a broad experience across IoT Core and its associated services. Each class in the series is centered on a scenario where a fictitious customer has a business challenge. The information and activities in the class provide the knowledge and skills to overcome this challenge.

Once you complete the series, you will have gained foundational knowledge and skills within AWS IoT and hands-on experience with a broad range of topics, such as Telemetry, Predictive Maintenance, and IoT Automation. Furthermore, by completing the IoT Foundation Series, you will be better prepared to dive deeply into more technical topics, such as IoT Security, MQTT, and the AWS IoT Rules Engine.

### Intended Audience

This course is intended for:

- Business decision-makers
- Data engineers
- Device engineers
- Fleet managers
- Line of business Developer
- Operational analysts
- Security architect
- Security operations engineer

### Series Prerequisites

We recommend that attendees of this course have the following prerequisites:

Introduction to AWS IoT (http://bit.ly/AWS-IoT-Intro-Training)

### Real-World Example

Curious to see how IoT works in the real world? Learn how Vizio connected millions of TVs with Amazon Alexa and AWS IoT: http://bit.ly/AWS-IoT-Video-Vizio

# Quick Start

## AWS IoT Camera Connector on AWS

**Provision cameras, and stream camera output to Amazon Kinesis Video Streams**

(Available online at: http://bit.ly/AWS-IoT-CameraConnector)

This Quick Start builds an Internet of Things (IoT) Camera Connector environment and serverless architecture on the Amazon Web Services (AWS) Cloud in about 5 minutes.

You can use this Quick Start to automate the connection—and simplify the management—of thousands of cameras through AWS IoT Core. You can also enable video streaming to Amazon Kinesis Video Streams for storage, playback, and machine learning–driven analytics.

The Quick Start includes a template and Config App to rapidly discover, provision, connect, and manage supported cameras and stream their output to Amazon Kinesis Video Streams in your AWS account.

The template creates resources in your AWS account to provision cameras as IoT things.

The Config App helps you discover supported cameras on your local network. It also provides IoT certificates and other information necessary for supported cameras to stream video content to Amazon Kinesis Video Streams.

In 5 minutes or less, you can connect up to thousands of cameras and begin implementing streaming-video analytics solutions.

### What You'll Build

Use this Quick Start to set up the following **serverless** architecture on AWS:

- An AWS IoT policy to associate with connected cameras
- An AWS Identity and Access Management (IAM) role for connected cameras to stream to Kinesis Video Streams
- An Amazon DynamoDB table to store provisioning keys. Once provisioning is complete, you should remove the keys
- AWS Lambda functions to create a provisioning key and a role alias, verify the stack, and provision cameras
- Amazon API Gateway to expose provisioning endpoints through HTTPS

- Amazon CloudWatch alarms to expose camera streaming status through an Amazon Simple Notification Service (Amazon SNS) topic, and update the associated camera's IoT thing shadow
- A separate Config App installable for provisioning cameras on the local network to stream to your AWS account

## How to Deploy

View the **Deployment Guide**: http://bit.ly/AWS-IoT-Camera-Deploy

## Real-World Example

**Introducing Amazon Kinesis Video Streams**
http://bit.ly/AWS-IoT-Video-Kinesis

Amazon Kinesis Video Streams is a fully managed video ingestion and storage service. It enables you to securely ingest, process, and store video at any scale for applications that power robots, smart cities, industrial automation, security monitoring, machine learning (ML), and more.

# Blog post

## Using AWS IoT to Create a Smart Home Water-Monitoring Solution

By Clive Charlton

(Available online at: http://bit.ly/AWS-IoT-Water-Monitor)

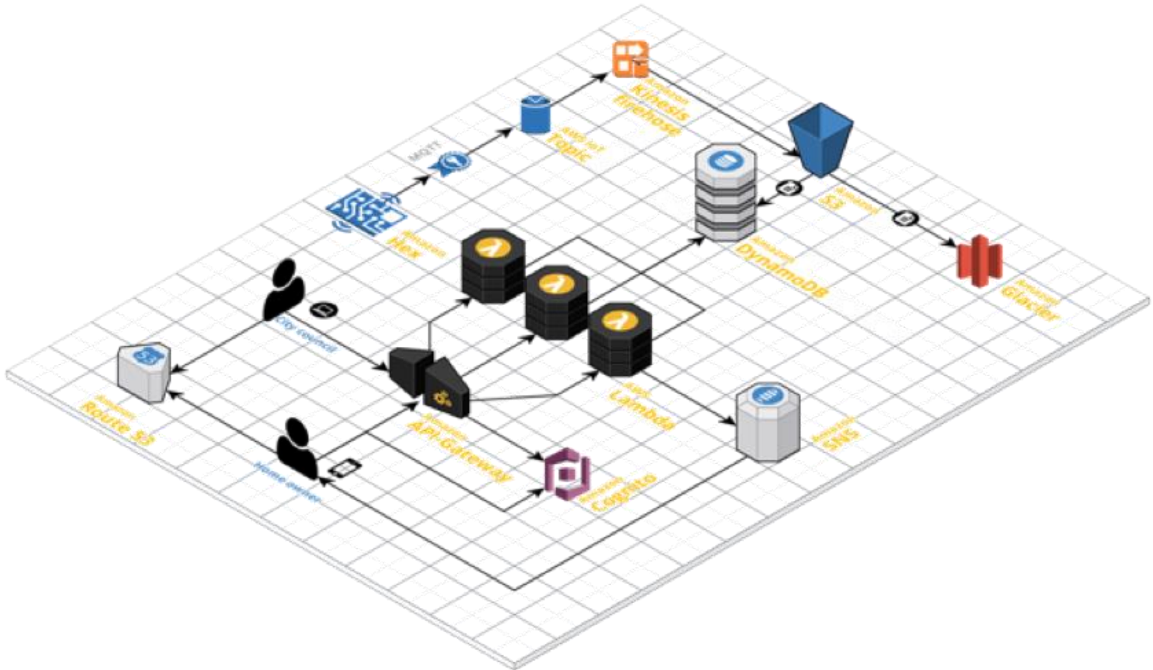
2018 saw the fourth year of drought and the worst in recorded history for the city of Cape Town, South Africa. “Day zero” was a term coined by the city for the day when they would have to turn the water off for citizens. Fortunately, “day zero” was never realized, and Cape Town didn’t go down in history as the first major city to run out of water. Water restrictions and augmentation plans alleviated the crisis until rainfall returned to the city, however, the city remains sensitive to water crises. Citizen behavior must permanently change in order to use water in a sustainable manner.

AWS presented this session at the AWS Summit in Cape Town where I met the CTO of a company called Apex Innovation who are now registered AWS partners. Apex Innovation has commercialized the entire architecture above. They spent time resolving the problem of using a hall-effect sensor in a commercial use case. Hall effect sensors are susceptible to a phenomenon called pulse drift which produces inaccurate pulse counts over time, rendering the water flow measurement inaccurate. Apex Innovation has many installations of this solution in commercial buildings across the continent and is working with city municipalities for large scale installations of more than 50,000 units to assist municipalities with accurate water consumption data.

In this blog post, I share an AWS IoT water solution that provides real-time water consumption information to home owners. The information can be used to encourage a change in consumption behavior to help the city conserve water. This solution uses AWS IoT core, Amazon Kinesis Data Firehose, Amazon S3, AWS Lambda, Amazon DynamoDB and an iOS application to create a real-time water consumption application.

### Architecture

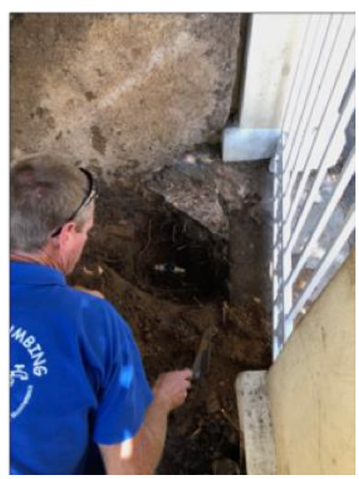
The diagram illustrates how simple it is to design a real-time water consumption application. I will delve into more detail of each component in the diagram in the sections below.



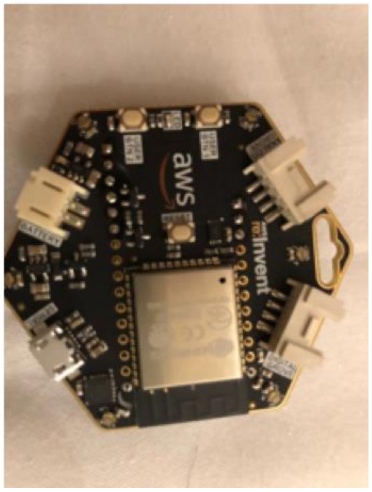

## The physical install

I installed a water flow sensor available on Amazon [(http://bit.ly/AWS-IoT-flow-sensor)](http://bit.ly/AWS-IoT-flow-sensor) for $11.19 on the water mains supply line into my house. As the water flows through the sensor, it turns a cog. As the cog turns, it passes over a coil. Each pass produces an electrical pulse. By counting the pulses in a given time period and factoring the diameter of the pipe, it's possible calculate the volume of water flowing through the sensor.

These photos show the installation of the water flow sensor on the main water supply.

Each pulse is sent to a micro-controller board, which is similar to the ESP32 series of low-cost, low-power system on a chip microcontrollers. Pulses are counted over a period of one minute. The volume is calculated and sent to AWS IoT Core for processing. Before any communication can take place between the AWS Hex board and AWS IoT Core, you must register the AWS Hex board as a thing.



A JavaScript method running on the micro-controller board:

- Runs the timer and the code to execute the flow calculation.
- Sends an MQTT message to the AWS topic.

The timer method stores the number of pulses received in the last minute. Every minute, the flow is calculated by factoring the flow rate for 1L per minute, which for this size sensor is a factor of 4.5. Divide that by 60 to calculate the liters per minute, as follows:

(pulse count / 4.5) / 60

Here is the source code for the timer function and helper functions (available at http://bit.ly/AWS-IoT-Water-Monitor):

```
PYTHON

// Load  Mongoose OS API
load('api_timer.js');
load('api_uart.js');
load('api_sys.js');
load('api_mqtt.js');
load('api_config.js');

// Uart  number used for this example
let uartNo = 1;
```

```
//  Accumulated Rx data, will be echoed back to Tx
let rxAcc = "";

let value = false;

// Set the  global pulse counter
let pulseCounter = 0;

// Set the  conversion factor for the hall effect calculation
let HEFactor = 4.5;

/*
 * Configure UART at 115200 baud
 * Configure the water flow sensor pin: 22
 */
UART.setConfig(uartNo, {
    baudRate: 115200,
    esp32: {
        gpio: {
            rx: 22,
            tx: 23,
        },
    },
});

/*
 * Set dispatcher callback, it will be called  whenver new Rx
data or space in
 * the Tx buffer becomes available.
 * The number of messages received per minute  x 4.5 is the
L/min
 * Count the number of messages per minute.
 */
UART.setDispatcher(uartNo, function (uartNo, ud) {
    let ra = UART.readAvail(uartNo);
    if (ra > 0) {
        // Received new data:  print it immediately to the
console, and also
        // accumulate in the  "rxAcc" variable which will be
echoed back to UART later
        let data = UART.read(uartNo);
        pulseCounter++;
        print("Received  UART data:", data);
        rxAcc += data;
    }
}, null);
```

```
// Enable  Rx
UART.setRxEnabled(uartNo, true);

/**
 * Create a timer to run every minute.
 * This timer method will store the number of  pulses received
in the last minute.
 * Once the timer is done, send data to AWS  IoT
 * Reset the pulse counter.
 * Rinse & repeat.
 *
 * Calculating the flow rate per hour.
 * For every litre of liquid passing through  the sensor per
minute,
 * it pulses 4.5 times. Divide the pulseCount  by 4.5 to get the
litres per minute.
 * Divide that by 60 to get litres per hour.
 *
 * (pulseCount / 4.5) / 60
 *
 */
Timer.set(60000 /*  milliseconds */, true /* repeat  */,
function () {
    // Send the MQTT message
    // Hard-coded device ID was  866191037731759
    let topic = '/things/waterflowcontrollers/' +
Cfg.get('device.id');
    // Set the date and time
    let Timestamp = Timer.now();
    // Date format day/month/year  hour:minute:second
    let datetime = Timer.fmt("%d/%m/%Y  %H:%M:%S", Timestamp);
    let flowRate = (pulseCounter / HEFactor) / 60;

    let res = MQTT.pub(topic, JSON.stringify(
        {
            "SerialNumber": Cfg.get('device.id'),
            "BatteryVoltage": "2080mV",
            "QCCID": "89445385230124710069",
            "GSN": Cfg.get('device.id'),
            "FlowRate": flowRate,
            "FlowCalibration": "4.5",
            "Date": datetime
        }
    ), 1);
    print('Published  flow rate: ', res ? flowRate : 'no');

    // Reset the pulse counter
```

```
    pulseCounter = 0;
}, null);
```

Because you configure most of this solution in AWS IoT Core, here is some background about the registry, AWS IoT policies, and X.509 certificates.

**Registering a device in the registry**

The registry allows you to keep a record of all of the devices that are registered to your AWS IoT account. It tracks metadata, such as device attributes and capabilities. In the case of the example used in this post, the registry supports metadata that describes whether a sensor reports temperature and if the data is Fahrenheit or Celsius. The registry assigns a unique identity to each device that is consistently formatted, regardless of the device type or the way the device connects.

For more information, see Register a Device in the AWS IoT Core Developer Guide.

**AWS IoT Policy**

X.509 certificates are used to authenticate your device with AWS IoT Core. AWS IoT Core policies are used to authorize your device to perform operations, such as subscribing or publishing to MQTT topics. Your device presents its certificate when it sends messages to AWS IoT Core. To allow your device to perform operations, you must create an AWS IoT policy and attach it to your device certificate.

For more information, see Create an AWS IoT Policy in the AWS IoT Core Developer Guide.

**Attach a certificate to your device**

A device must have a X.509 certificate, private key, and root CA certificate to authenticate with AWS IoT Core. AWS recommends that you also attach the device certificate to the thing that represents your device in AWS IoT Core. This way you can create AWS IoT Core policies that grant permissions based on certificates attached to your things.

For more information, see Attach a Certificate to a Thing in the AWS IoT Core Developer Guide.

## Data

Incoming messages from the thing are sent to an AWS IoT topic configured for the thing. Amazon Kinesis Data Firehose subscribes to the topic to send every message to Amazon S3 for long-term durable storage. When the message is received in Amazon S3, an S3 event triggers an AWS Lambda function that writes a copy of the message into Amazon DynamoDB, where analysis and visualization is performed. After the raw message has been copied to Amazon DynamoDB, the message is archived in [Amazon Glacier](#) to reduce storage costs.

Data from the thing is sent in JSON format for ease of use with applications. The timestamp is important for the downstream applications to collate the data into comprehensible time series data blocks. The flow rate is the calculated volume of water flowing through the water flow sensor per minute. A serial number uniquely identifies the thing and the longitude and latitude are the GPS coordinates of the installed device.

Here is a sample JSON message:

```
Js

{
    "Date": "21/02/2019 16:14:10",
    "FlowRate": 28,
    "SerialNumber": "esp32_39B3F8",
    "latitude": -33.9399299,
    "longitude": 18.4751000
}
```

## Visualization

Visualizing the data from the water flow sensors in a format that is easily interpreted is crucial to the success of the application. A mobile application and a web application make it easy for the user to access and read the data. Different graphs are important for interpreting different trends in water usage.

An hourly graph is useful for leak detection, where a sudden spike in water usage over the last hour can alert the user to a potential leak. The user is notified of the high water usage with an SMS using Amazon SNS.

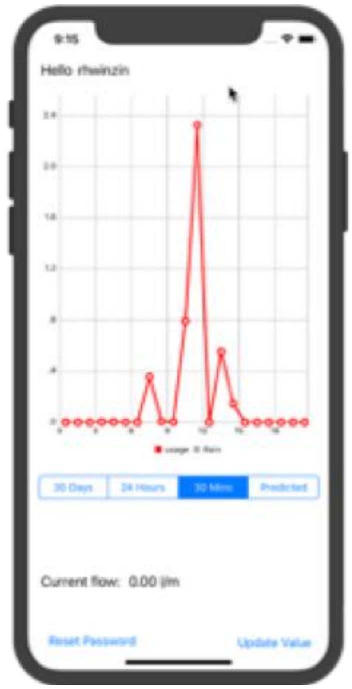
A weekly and a monthly graph is useful for the user to quickly identify trends over the last few days or the last month. Additional visual space in the web dashboard allows us to provide comparative graphs for water consumption over years so the user can compare his or her water consumption at the same time in previous years. This is useful for seasonal changes in water consumption.

## Mobile application

In this example, an iOS application was developed in XCode to visualize the data in graph format by connecting to a gateway in Amazon API Gateway which exposes AWS Lambda functions to fetch the data required to render each graph.

Here are some photos of the mobile application screens.





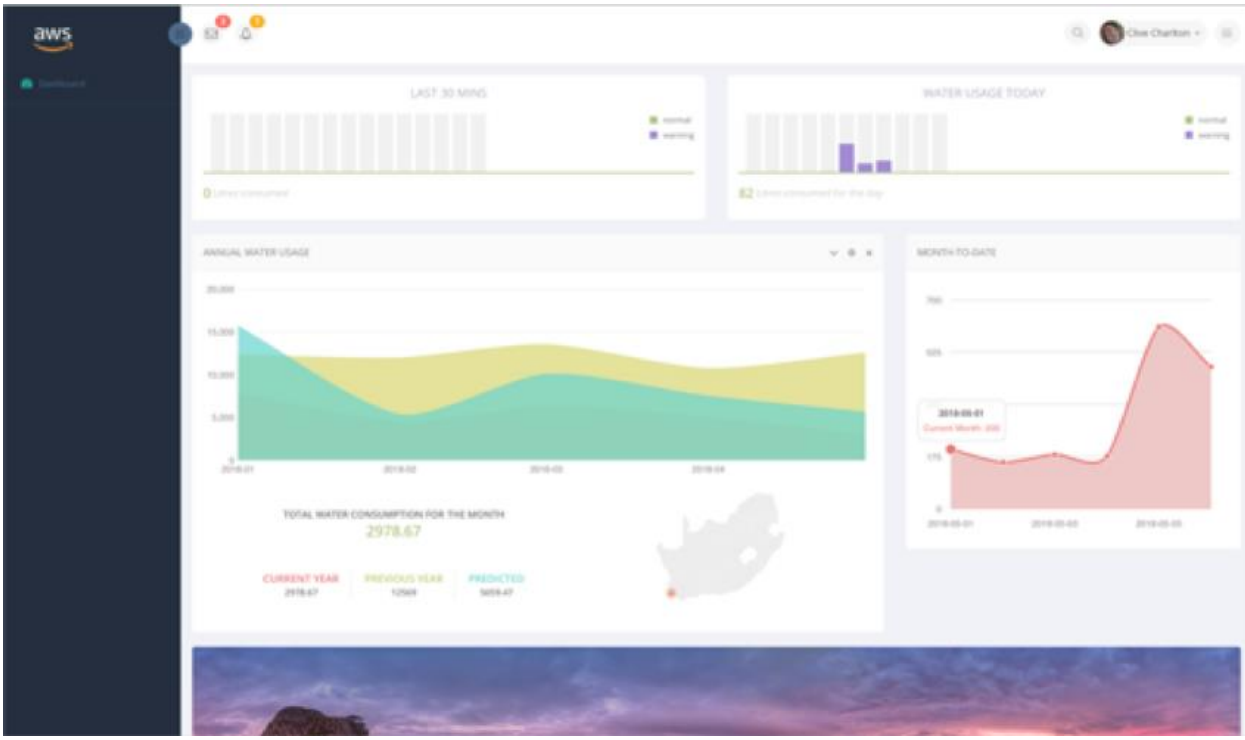


The mobile app and the website both need security credentials in order to make programmatic requests to AWS. Amazon Cognito is ideal for this scenario. You can use this service with the AWS Mobile SDK for iOS and the AWS Mobile SDK for Android and Fire OS to create unique identities for users and authenticate them for secure access to your AWS resources.

## Web application

The same AWS Lambda functions used by the iOS mobile application are accessed through Amazon API Gateway for the web application dashboard to render the graphs on the website. The website, hosted in Amazon S3, uses Node.js to access the gateway in Amazon API Gateway and render the graphs.

Here is a dashboard in the web application:




## Next steps

Being able to accurately measure the water consumption in our house has changed my family's consumption of this most precious resource. We now actively reduce our water consumption. Now that you know how easy it is to measure important environmental data, like water usage using AWS IoT services, I recommend you find opportunities in your everyday life to leverage IoT technologies. Whether it's temperature, smart home utilities, or metering. The ability for you to build IoT applications with simple devices unlocks a world of data and insights into your environment.

(Again, you can read this blog post online at http://bit.ly/AWS-IoT-Water-Monitor)

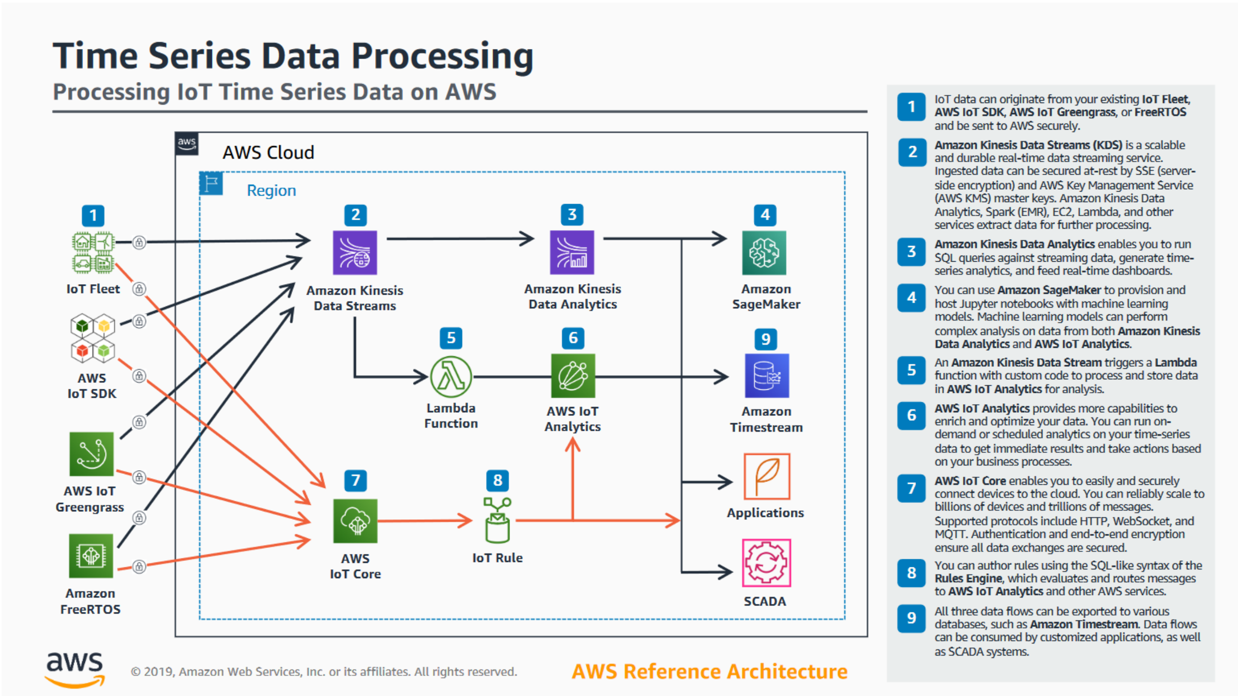
## Real-World Example

**mPrest & Netafim: Smarter Irrigation with AWS IoT Core**
http://bit.ly/AWS-IoT-video-smart-irrigation

mPrest shows how they helped Netafim build a multi-tenant SaaS solution for irrigation systems that uses AWS IoT Core and other AWS services.

# Reference Architecture

(Available online at: http://bit.ly/AWS-IoT-TimeSeries)




## Real-World Example

**Geezeo: Automating Data Processing on AWS**
http://bit.ly/AWS-IoT-video-Geezeo

Geezeo shows us how they automate data ingestion and processing. Learn how they use Amazon Kinesis Streams to collect and process these large streams of data and then persist data into Amazon DynamoDB.

# Solution

## Real-Time IoT Device Monitoring with Kinesis Data Analytics

(Available online at: http://bit.ly/AWS-IoT-Kinesis)

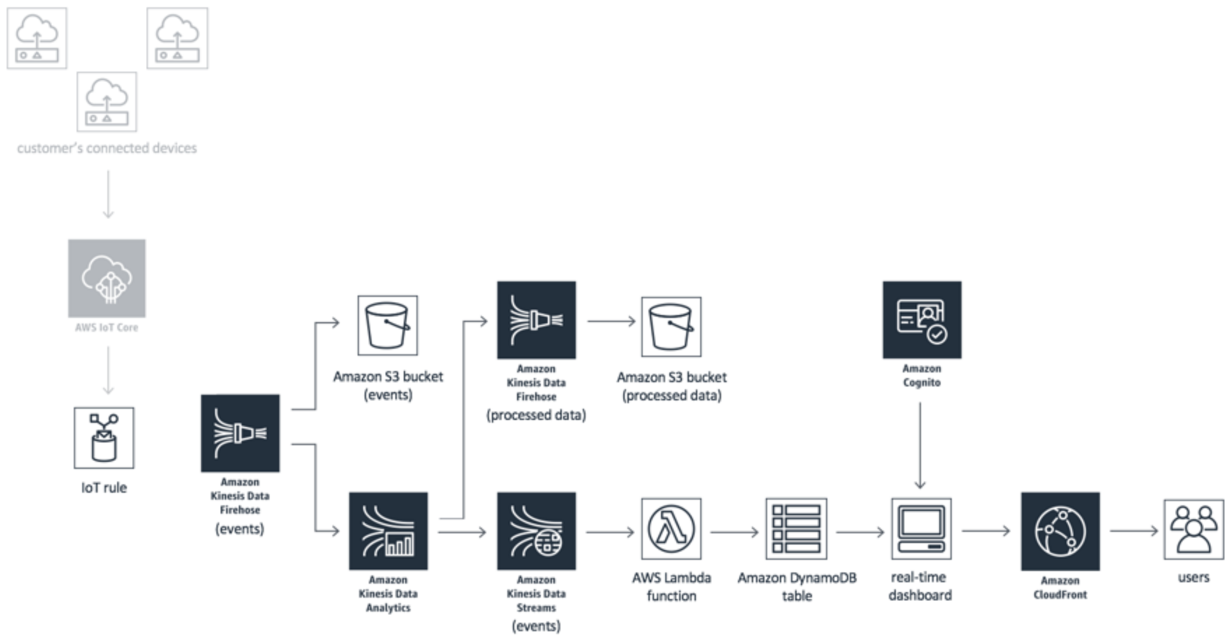
### What does this AWS Solution do?

Monitoring IoT devices in real-time can provide valuable insight that can help you maintain the reliability, availability, and performance of your IoT devices. You can track time series data on device connectivity and activity. This insight can help you react quickly to changing conditions and emerging situations.

To help customers more easily leverage Kinesis Data Analytics, Amazon Web Services (AWS) offers the Real-Time IoT Device Monitoring with Kinesis Data Analytics solution, a reference implementation that automatically provisions the services necessary to collect, process, analyze and visualize IoT device connectivity and activity data in real-time. This solution is designed to provide a framework for analyzing and visualizing metrics, allowing you to focus on adding new metrics rather than managing the underlying infrastructure.

### AWS Solution Overview

AWS offers a solution that uses AWS IoT to ingest device data, Amazon Kinesis Data Firehose to archive the data, Kinesis Data Analytics to compute metrics in real-time, and Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB to durably store metric data. The solution features a dashboard that visualizes your device connectivity metrics in real-time.




*Real-Time IoT Device Monitoring with Kinesis Data Analytics solution architecture*

When AWS IoT ingests data from your connected devices, an AWS IoT rule sends the data to a Kinesis data delivery stream. The delivery stream archives the events in an Amazon S3 bucket and sends the data to a Kinesis Data Analytics application for processing.

The application sends the data to an AWS Lambda function that sends it in real-time to a DynamoDB table to be stored. The application also sends processed data to a second Kinesis data delivery stream which archives it in an Amazon S3 bucket.

The solution also creates an Amazon Cognito user pool, an Amazon S3 bucket, an Amazon CloudFront distribution, and a real-time dashboard to securely read and display the account activity stored in the DynamoDB table.

## Real-World Example

**Philips Healthcare: Connected Medical Devices in the Cloud**
http://bit.ly/AWS-IoT-video-Philips

Philips Healthcare walks us through the architecture for the Philips HealthSuite Digital Platform that provides cloud-based services for connected medical devices.

# Video

## This Is My Architecture:

### Democratizing LoRaWAN and IoT with The Things Network

(Available online at: http://bit.ly/AWS-TMA-IoT)

LoRaWAN is a Low Power Wide Area networking protocol designed to wirelessly connect batter-operated-things: the internet in regional, national, or global networks, and targets key IoT requirements.

The Things Industries, a LoRaWAN Network Service Provider, provides an integrated chain of products and services for you to start working on the IoT.

On this special long-format episode of This Is My Architecture, learn how The Things Network (https://www.thethingsnetwork.org) and its commercial wing, The Things Industries (https://www.thethingsindustries.com), are helping both hobbyists and businesses connect low-power, long-range devices to the cloud.

The Things Network is a community project in over 84 countries that is building a global Internet of Things data network. The Things Network devices connect to community-maintained gateways. They can communicate over very long distances and last on a single alkaline battery for up to 5-10 years thanks to the LoRaWAN protocol.

The Things Industries has built a platform using AWS IoT that allows device data to be collected and processed in the cloud using multiple AWS services. You'll learn about The Things Industries' architecture, how Netcetera is leveraging The Things Network for air quality monitoring in Skopje, how Decentlab builds high-quality, long-lasting LoRaWAN devices that work with The Things Network to track environmental conditions, and get a feel for the community at a local meetup.

# Whitepaper

## Securing Internet of Things (IoT) with AWS

(Available online at: http://bit.ly/AWS-IoT-Security)

### Purpose

This whitepaper is a detailed look at the security-enabling Internet of Things (IoT) services that customers can harness in the AWS Cloud. This paper is intended for senior-level program owners, decision makers, and security practitioners considering secure enterprise adoption of IoT solutions.

### Background

IoT technology enables organizations to optimize processes, enhance product offerings, and transform customer experiences in a variety of ways. While business leaders are excited about the way in which their businesses can benefit from this technology, security, risk, and privacy concerns remain. This is, in part, due to a struggle with disparate, incompatible, and sometimes immature security offerings that fail to properly secure deployments, leading to an increased risk for customer or business owner data.

Organizations are eager to deliver smart services that can drastically improve the quality of life for populations, business operations and intelligence, quality of care from service providers, smart city resilience, environmental sustainability, and a host of scenarios yet to be imagined. Most recently, AWS has seen an increase in IoT adoption from the healthcare sector and municipalities, with other industries expected to follow in the near term. Many municipalities are early adopters and are taking the lead when it comes to integrating modern technologies, like IoT. For example:

- Kansas City, Missouri: Kansas City created a unified smart city platform to manage new systems operating along its KC streetcar corridor. Video sensors, pavement sensors, connected street lights, a public WiFi network, and parking and traffic management have supported a 40% reduction in energy costs, $1.7 billion in new downtown development, and 3,247 new residential units.
- City of Chicago, Illinois: Chicago is installing sensors and cameras in intersections to detect pollen count and air quality for its citizens.
- City of Catania, Italy: Catania developed an application to let commuters know where the closest open parking spot is on the way to their destination.

- City of Recife, Brazil: Recife uses tracking devices placed on each waste collection truck and cleaning trolley. The city was able to reduce cleaning costs by $250,000 per month, while improving service reliability and operational efficiency.
- City of Newport in Wales, UK: Newport deployed smart city IoT solutions to improve air quality, flood control, and waste management in just a few months.
- Jakarta, Indonesia: As a city of 28 million residents that often deals with flooding, Jakarta is harnessing IoT to detect water levels in canals and lowlands, and is using social media to track citizen sentiment. Jakarta is also able to provide early warning and evacuation to targeted neighborhoods so that the government and first responders know which areas are most in need and can coordinate the evacuation process.

## Real-World Example

**Eseye: The AWS IoT Circle of Life**
http://bit.ly/AWS-IoT-video-eseye

You want secure communication to a device in the field, but how do you make that first connection without touching the device? Ian Marsden, CTO of Eseye (an AWS IoT Competency partner) shares how they built AnyNet Secure for AWS IoT to solve this problem.