California State University, San Bernardino

CSUSB ScholarWorks

Electronic Theses, Projects, and Dissertations

Office of Graduate Studies

12-2021

Cloud Internet of Things for the Smart Environment of a Smart City

Junho Kim

Follow this and additional works at: https://scholarworks.lib.csusb.edu/etd

Part of the Management Information Systems Commons

Recommended Citation

Kim, Junho, "Cloud Internet of Things for the Smart Environment of a Smart City" (2021). *Electronic Theses, Projects, and Dissertations*. 1366.
https://scholarworks.lib.csusb.edu/etd/1366

This Project is brought to you for free and open access by the Office of Graduate Studies at CSUSB ScholarWorks. It has been accepted for inclusion in Electronic Theses, Projects, and Dissertations by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.

CLOUD INTERNET OF THINGS FOR THE SMART ENVIRONMENT OF A SMART CITY

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Information Systems and Technology

by

Junho Kim

December 2021

CLOUD INTERNET OF THINGS FOR THE SMART ENVIRONMENT OF A SMART CITY

A Project

Presented to the

Faculty of

California State University,

San Bernardino

---

by

Junho Kim

December 2021

Approved by:

Benjamin Becerra, PhD, Committee Chair

Conrad Shayo, PhD, Committee Member

Jay Varzandeh, PhD, Dept. Chair, Information & Decision Sciences

© 2021 Junho Kim

ABSTRACT

The environmental service area for smart city construction provides sustainability, economic, stage, and safe energy savings in building a smart city. The smart environment monitoring system can manage the agricultural environment, water quality, and air quality of smart cities through the Internet of Things and cloud computing technology. Smart environmental monitoring (SEM) systems to reduce environmental problems in smart cities are important service domains that can improve citizens' quality of life. The purpose of this project is to identify services through the introduction of technologies to solve environmental problems in smart cities, the use and management of the technologies introduced. The project findings are: (a) The SEM system collects environmental data through IoT devices and analyzes it through cloud computing. (b) Data from the SEM system is collected through the Internet of Things based on a wireless sensor network. The collected data is transmitted to the cloud computing platform to be analyzed and monitored. (c) For wireless Internet connections between the two technologies can connect through unique services of Message Queuing Telemetry Transport and cloud platform. (d) The SEM applications were analyzed for Smart Agricultural Monitoring, Smart Water Quality Monitoring, and Smart Air Quality Monitoring. (e) The project is analyzed the transmitted data using Amazon Web Service and Google Cloud Platform. When the analyzed environmental parameters deviate from normal values, email notification can be sent to detect abnormalities in environmental parameters. In addition, the connection between IoT devices and cloud platforms can confirm the normal connection of IoT devices to establish a security system by obtaining credibility and reliability for the connection between the two technologies. As a result, the collaboration between IoT and cloud computing technology can show how a smart environment service domain can help smart city citizens.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE:
INTRODUCTION

Introduction

The growth of the world's population due to urbanization is a key problem that can be addressed through smart cities. In the 1950s, 30% of the current population lived in cities. By 2014, the level of population growth due to urbanization had developed to 54%. By 2050, the level of population growth due to urbanization is expected to rise to 66% (Bravo, 2014).

“As population growth increases, increasing urbanization can cause problems such as soil pollution” (Food and Agriculture Organization, UN, 2018). Soil plays a primarily important role in agriculture as more people need more resources because of the growing world population. The pollution process that occurs after excessive nitrogen fertilizers are abused in the soil causes severe acidification that occurs over millions of years. Soil acidification is fatal because it can cause metal toxicity to crop. Soil contamination reduces crop yield and quality, affecting food. As a result, agricultural yields discarded in 2012 amounted to 1.3 billion tons (Food and Agriculture Organization, UN, 2018)

Environmental problems caused by an increase in the world population include not only soil problems, but also water and air pollution. The lack of clean water is one of the emerging problems due to the growth of the world's population. The lack of clean water will become a bigger problem in developing countries where water purification and management systems are lacking. Currently, 47% of the world's population suffers from water scarcity for at least one month each year. 52% of the world's population will suffer from water scarcity by 2050 if the shortage of water purification and management systems is maintained (Boretti & Rosa, 2019).

Moreover, air pollution is an environmental factor that has the greatest impact on human health. Air pollution is also one of the most unavoidable causes of disease. In 2016, indoor and outdoor air pollution killed 6.5 million people worldwide. Air pollution, particularly in developing countries, has disproportionately affected women, children and the elderly. Since the use of wood fuel and kerosene by low-income households causes indoor air pollution, the number of deaths due to air pollution is increasing. Air pollution is a global problem due to long-distance transport. Without the active intervention of each country's Ministry of Environment, the number of deaths due to air pollution is expected to increase by more than 50% by 2050 (Nations, 2021).

Smart city technologies not only improve the quality of life, but also have high potential. The background of smart cities is to intentionally apply technology and data to applications to make better decisions and provide a high quality of life while providing safety, health and living benefits to the citizens of smart cities.

Many efforts are put into solving the environmental problems mentioned above by managing environmental factors such as soil, water, and air quality. In order to ensure the benefits of a smart environment through the construction of a smart city, high-speed internet networks, IoT devices, and cloud computing service for data analysis and management are required as the basis of a three-level smart infrastructure (Johnson, 2018).

For environmental pollution and management through smart city environmental domain service, Environmental Monitoring (EM) is used based on three-level smart infrastructure. EM is a system that monitors and evaluates the environmental conditions and pollution potential of smart cities. The most common EM applications are air monitoring, water monitoring, and soil monitoring (United Nations Economic Commission for Europe, n.d.).

The Internet of Things (IoT) can be proposed for the operation of the real-time environmental monitoring system. IoT devices allow the measurement of environment and parameters extensively and continuously. Based on the Wireless Sensor Network (WSN) system, IoT devices can implement smart EM monitoring. The implemented EM collects the data and sends the data to the cloud computing service platform through a specific Internet connection protocol (Sahay et al., 2019).

Problem Statement

“Environmental pollution is becoming a big problem due to the global population growth and has a great impact on the quality of human life” (McClelland, 2016). Environmental problems in developing countries are suffering from death damage due to environmental pollution, and the death rate is rising. The environmental service domain of smart cities is a solution to the problem of environmental pollution. In the environmental service domain of the smart city, soil, water, and air quality can be analyzed through the environmental monitoring system. The environmental monitoring system can collect information about environmental pollutants through WSN based IoT devices. Data collected through IoT devices can be transmitted to IoT applications of cloud computing service platforms through fast network internet communication to monitor and evaluate environmental conditions (McClelland, 2016).

IoT devices are exposed from external physical attacks or natural disasters. In other words, the occurrence of a physical attack causes malfunction of the smart environment IoT device. A cloud computing service over an internet connection is required to solve this problem. Since the cloud computing service platform can receive and store data collected from IoT devices, it is a necessary technology for monitoring and analysis of the environmental monitoring system. In addition, the cloud computing service platform can update and analyze the device's status based on the transmitted data to ensure normal data processing.

Data of smart environment IoT devices may change due to external attacks or changes in the environment. Changed data is very important for monitoring and evaluation of the environment. Critical environmental pollution data through monitoring and analysis can be distributed via email or text message notification through the application of the cloud computing service platform. Research on various cloud platforms is also required to show suitable use cases for IoT devices through cloud computing services. (Mohamed Sadeeq et al., 2021).

According to Gartner's Magic Quadrant for Cloud Infrastructure and Platform Service, Amazon Web Service (AWS) is a leader in the cloud industry. AWS focuses on becoming a wide range of IT service providers. It is also expanding its supply chain that provides cloud services. Not only does the engineering supply chain provide better pricing and performance, but it also utilizes engineering capabilities to provide innovative supply chains. AWS is also an innovation leader that provides a wider range of services than other providers to suit customer types (Bala et al., 2021).

Google Cloud Platform (GCP) is also one of the leaders in the cloud industry. As GCP has so many cases of use, it is compatible enough to be used in a variety of businesses Google can operate through geographic advantages through a wide service supply chain worldwide. Furthermore, GCP has Kubernetes, the most fully functional provider in the cloud industry (Bala et al., 2021). Google Kubernetes Engine (GKE) is a management system that provides automatic deployment, expansion, and scaling of containerized applications. Also, GKE is very convenient for users to visualize or analyze data (Google Kubernetes Engine (GKE), n.d.)

According to the above two reasons, AWS and GCP were selected as platforms for analyzing Environmental monitoring in the project. Furthermore, A Magic Quadrant for Cloud Infrastructure and Platform Service and objective evaluation approaches of AWS and GCP were suggested to show more reason in detail. In addition, MQTT, an Internet connection protocol between IoT device and cloud platform, was explained.

<u>Research Questions</u>

1. What technologies are needed for smart environmental services?
2. How is the smart environmental monitoring data collected, analyzed and monitored?
3. Which components enable the connection of IoT devices with cloud service platforms?
4. What applications of Smart Environment are there and how were they used?
5. How does smart environment application through IoT devices connect to IoT service of cloud platforms?

Organization of Project

Chapter 2 explained the background knowledge of the technology needed to build a smart environment in a smart city and the statistics and impact of the technologies to build a current smart environment through the Literature Review. Chapter 3 researched and explained the analysis methodology for Smart Environmental Monitoring (SEM) systems based on IoT devices and cloud platforms for building smart environments. Chapter 4 proposed SEM with IoT and Cloud Computing. The project proved SEM's applications by researching actual cases that were used in the real world. Chapter 5 analyzed SEM systems through AWS IoT Core and GCP IoT Core. This analysis study presented data collection, transmission, defense, security, and management between IoT devices and cloud platforms for SEM applications.

CHAPTER TWO:

LITERATURE REVIEW

Literature Review

Smart City and Environment

Smart Cities are spaces for human activities. In addition, cities are places where economic, environmental, and social needs expand. Urbanization causes many important economic, social, and environmental changes (Abu-Lughod & Hay, 2007).

The National Institute of Standards and Technology (NIST) reported that thousands of cities and communities around the world are currently developing smart cities. Smart technologies benefit citizens through advances in various fields such as transportation, energy, industry, environment, and public safety (Nguyen, 2018).

A smart environment is a smart solution and domain for the smart city environment. It consists of an environmental monitoring system for managing essential elements such as water, air, soil, a smart system for weather observation, and resource supply. The goal of the smart environment is to improve the efficiency and quality of the environment of the city and energy management (Autor: T M Vinod Kumar, 2020).

The Smart Environment Monitoring (SEM) is a service that can improve the quality of smart cities. The SEM is a system that can provide very efficient services to citizens of smart cities by monitoring building intensity, waste management, air pollution quality management, weather monitoring, traffic management, parking management, and energy consumption (Shah & Mishra, 2016).

The SEM is monitoring air quality management, water waste management, and weather monitoring for future agricultural action through parameters such as soil moisture, temperature, humidity, and carbon dioxide (Folea & Mois, 2015).

The global environmental monitoring market is estimated to grow 4.1% annually, reaching $17.1 billion by 2025, from $14 billion in 2020. Increasing pollution in smart cities, increasing installation of SEM infrastructure, developing eco-friendly industries, increasing citizens' awareness, and expanding pollution monitoring infrastructure are major growth drivers (Markets & Markets, 2020).

In Uruguay, a smart environment establishes a soil management plan through smart agriculture production. Since 2014, 2,946,000 hectares have been introduced into climate-smart agriculture to improve energy efficiency. Also, a soil management system is launched by climate-smart agriculture sensors to 5,139 farms (The World Bank, 2017).

The Internet of Things

The term Internet of Things was first named and documented in 1999 by British pioneer Kevin Ashton (Lueth, 2019). Kevin Ashton defined his IoT theory as "A system connected to a physical device through a ubiquitous data sensor network through the Internet." (Corcoran, 2016). Of course, the use of this term has grown somewhat beyond its original intent.

IoT devices are rapidly evolving due to Radio Frequency Identification (RFID) and WSN technologies. RFID enables device tagging and labeling recognition to act as a basic identification mechanism for the IoT. In addition, WSN technology has enabled wireless identification of devices. IoT applications allow communication between the physical, cyber, and digital worlds. IoT applications are applied to smart homes and cities, intelligent transportation, and healthcare (Mahmoud et al., 2015).

The Internet of Things has become one of the most important and essential technologies of the 21st century in recent years. Internet access is possible with built-in devices that can be used in fields such as smart home, smart city, and healthcare, enabling active communication between processes and objects. In addition, low-cost cloud computing services, big data analysis, and mobile technology can collect the data with less manpower. (Oracle IoT n.d).

Despite the lack of chip supply due to the lack of semiconductor production and the increasing impact on supply chain problems caused by COVID-19, the IoT market is growing. In 2021, the number of IoT devices connected worldwide is expected to reach 12.3 billion, up 9% from 2020. By 2025, more than 27 billion IoT devices are expected to be connected (Sinha, 2021).

Cloud Computing

Over the years, computer technology has steadily developed computer technology through the Internet. In 1961, Professor John McCarthy of MIT created the concept of cloud computing. McCarthy foresaw and mentioned the concept of pay-as-you-go for today's cloud platforms, "Computing can one day be organized into public facilities. Each subscriber only has to pay for the capacity they use but has access to all the programming languages on the super-large system" (Zehner et al., 2016).

Cloud computing is a ubiquitous and convenient network access technology for providing services and sharing resources with little management and effort. Cloud computing characterizes the important role that cloud services play and provides standards for how and by which they are used. In addition, cloud computing provides services by simply classifying them without restricting operation methods and regulations through various characteristics, service models, and deployment models. The characteristics of cloud computing include on demand self-service, connecting multiple devices, resource pooling, fastest Resiliency, and measurement service (Mell & Grance, 2011).

Cloud computing services are basically classified into three deployment models to provide services. The private cloud is managed and operated by a single organization or group. A private cloud is called an internal cloud or an in-house cloud. This service provides consumers and users with limited access to resources and services. In other words, a private cloud is a cloud service designed to maintain administration and control at a level related to security, personal information, and governance (Mell & Grance, 2011).

The public cloud is the most popular cloud service available to end-users. This service is provided by large infrastructures such as public facilities. In other words, the public cloud can be used by large industry groups, and the provision of cloud services is owned by organizations such as public organizations (Mell & Grance, 2011).

The hybrid cloud configuration consists of two or more unique clouds: Private Cloud and Community Cloud or Public Cloud. Even if a combination of two cloud configurations, hybrid cloud is combined with standardized technology of data and applications. Hybrid cloud allows companies to manage cloud services in a stable state by incorporating a stable state in the private cloud into the Public or Community cloud (Mell & Grance, 2011).

Also, deployment models are provided based on the service model so that the service can be provided according to the user's purpose. SaaS is a complete software solution service that provides users with cloud applications, basic IT infrastructure, and platforms. This service model runs on a web browser by default. The most representative example is Gmail. Although Gmail is charged according to usage and has a multi-tenant environment (Mell & Grance, 2011).

IaaS is an application infrastructure service and software service for building infrastructures such as server resources, IP, network, storage, and power to operate servers. The IaaS can avoid cost and complexity by purchasing and managing the infrastructure of physical servers and data centers. (Mell & Grance, 2011).

PaaS is a complete development and deployment environment provided in the cloud with resources that can deliver everything from simple cloud-based apps to sophisticated cloud-enabled enterprise applications. (Mell & Grance, 2011).

Cloud Platform

According to Gartner, companies are increasingly trying to connect various IoT endpoints to incorporate them into their businesses and manage them more effectively. The cloud computing service platforms collaborate without restrictions on location and time using the characteristics of the cloud platform is a service as a software suite or cloud service. In other words, various types of IoT devices can be monitored, managed, controlled, and secured through applications built on the platform. In addition, cloud platforms generally provide web service infrastructure functions to support solutions and operations of IoT(Gartner IoT Platform n.d).

Magic Quadrant for Cloud Infrastructure and Platform Services 2021 suggested, more than 90% of the cloud infrastructure and platform services market is focused on AWS, Microsoft Azure, GCP, and Alibaba Cloud providers. The report described leaders, challengers, visionaries, and niche players to describe quadrant description. Cloud platform leaders AWS and GCP provide strategic and appropriate services. Both cloud platforms also have many customers with high market share and reference (Bala et al., 2021).

The IDC MarketScape: Worldwide IoT Applications Platforms for Smart Cities 2019–2020 Vendor Assessment evaluates suppliers of IoT applications of cloud platform in smart cities. Data collection, analysis, prediction, and control of IoT devices to digitally transform smart city functions enhance the performance and capabilities of cities. In addition, according to IDC MarketScape's evaluation, AWS is ranked as a leader in global application platforms for smart cities (Yesner, 2020).

The City of Pittsburgh, Pennsylvania, announced a four-year contract to migrate the city's IT infrastructure to GCP. Google has announced that it will reduce the barriers to entry for IT technologies and enable the transition to a cloud-centric approach. The announcement that not only urban environmental issues, but also applications in areas such as transportation, infrastructure, and public safety will be created to form a smart city (Swinhoe, 2021).

The global IoT cloud platform market is expected to grow from $6.4 billion in 2020 to $11.5 billion in 2025, with an average annual growth rate of 12.6% during the forecast period (Markets & Markets, 2020). Cloud service platforms for IoT have been widely studied, but certain functions of IoT are still not making much progress. In addition, objective comparison of IoT cloud service providers is difficult because suppliers all provide slightly different services (Pflanzner & Kertesz, 2016).

In summary, Chapter 2 is a documentary reference to the technologies needed to build the environment of a smart city. The environment of smart cities can be digitized based on IoT and cloud computing technologies to create a more efficient and convenient smart environment without using much manpower. In chapter 3, methodology is suggested for SEM systems with IoT and cloud computing service.

CHAPTER THREE:

TECHNOLOGIES FOR MONITORING SYSTEM

Wireless Sensor Network (WSN) with the IoT

The most beneficial technology for environmental monitoring is the IoT. The WSN based IoT device not only identifies the presence of substances in air, water, and soil, but also collects status and condition information to generate data. Through smart environmental monitoring, smart cities' environmental problems can maintain a safer and cleaner environment (Verma, 2021). The SEM system is an efficient, inexpensive and ready-to-use environmental monitoring system based on a WSN to collect data using sensors from IoT devices. The SEM system uses serial communication through Arduino, one of the IoT devices, to transmit the collected data to a cloud platform called a base station. SEM systems can potentially reduce environmental risks because users can make specific modeling that can be used for analysis and forecasting (Jiang & Huacón, 2017). The collected data from the SEM systems uses wireless sensors that connect via a routable wireless link to the cloud computing service platform. The cloud computing service platform can analyze and monitor the status and condition of the environment in real time through the transmitted data. Sensors in IoT devices can measure parameters in the environment without direct human action (Corbellini et al., 2018)

WSN has evolved in the area classified as an SEM application. The advantages of WSN are low cost, low power, low consumption, and it is being developed into a flexible system. WSN develops and deploys wireless monitoring systems in the environment to provide data collection services through sensors. The sensor's network hardware consists of a microcontroller with a digital port via an analog sensor output type along with an onboard sensor. In addition, software consists of sensor driver, networking protocol and algorithm processing device, and application-specific detection sensor driver (Deshmukh & Shinde, 2016).

Cloud Computing with the IoT

Cloud computing is software designed to store, analyze, and process data sent from IoT devices in environmental monitoring to get the most out of your IoT infrastructure. The SEM IoT solution through cloud computing needs to transmit, store, process, and connect and communicate the collected data between processes. Before connecting cloud computing and IoT devices, two completely different technologies need to be understood for a complete solution. Cloud computing and IoT come from two completely different worlds, but their characteristics are complementary. Two completely different techniques can help each other. In other words, IoT can help the cloud while gaining some of the benefits of cloud computing (Gomes et al., 2014).

The huge resources provided by cloud computing services are the best alternative technology to complement the resource constraints of IoT. At the same time, IoT can extend cloud computing services in a more decentralized way, resulting in new scalable services (Alhakbani et al., 2014). Cloud computing services can store and analyze large amounts of data that are normally generated through the supply of data collected from SEM IoT devices. Therefore, the data storage space of cloud computing services is larger than that of SEM IoT devices, so data can be stored and used flexibly (Botta et al., 2016).

On the other hand, general SEM IoT devices cannot directly monitor or analyze complex data because there is a possibility of numerous errors in the process of data collection. However, if SEM IoT devices use cloud computing resources, they enable monitoring and analysis through limited data processing (Ray, 2016). For this reason, the integration of IoT with the cloud is more innovative and motivates users and developers (Botta et al., 2016).

Connection Components

Message Queuing Telemetry Transport (MQTT)

The MQTT is a topic-based publish/subscribe protocol that supports hierarchical topics using strings (Hunkeler et al., 2008). The MQTT is an alternative to client-server architecture that communicates directly with endpoints to maximize bandwidth. In contrast, clients sending messages in the publish/subscribe model are separated from the client or subscriber receiving the message because they do not contact each other directly, the MQTT broker, the link between them (IoT Agenda, 2019).

Publish/subscribe allows IoT devices to post messages about topics. In other words, send a message to the broker. The device can receive messages from the broker of the subscribed item. The MQTT broker acts as an intermediary between the client sending the message and the subscriber receiving the message. In other words, brokers are like post offices. All messages are delivered through the broker. The topic can be thought of as a filter for identifying messages. Clients register a topic filter to identify messages that they want the subscription. Brokers use topics and topics filters to route messages from publishing customers (IoT Agenda, 2019).

Quality of Service (QoS)

The MQTT supports basic end-to-end QoS. Depending on how reliable the message is delivered to the recipient, the QoS is divided into three categories: level 0 to level 2. QoS level 0 is the simplest one: QoS level 0 offers a best-effort delivery service, in which messages are delivered either once or not at all to their destination. QoS level 0 is not defined as retransmission or approval. QoS level 1 provides more stable transmission than zero. QoS level 1 will be retransmitted until the recipient's approval is confirmed. As a result, the stability of message arrival is specific, but the retransmission rate of unconfirmed messages is very high. The highest level, QoS level 2, not only receives messages but also ensures delivery to the receiving entity. However, QoS level 2 is not recommended and used due to increased latency and fluency of service (Hunkeler et al., 2008).

Application Programming Interface (API)

The API allows companies to open application data and functions to external third-party developers, business partners, and internal departments. In other words, services and products can communicate and utilize each other's data and functions through a documented interface. Use the interface to communicate with other products and services. Open-APIs, Partner-APIs, Internal-APIs, and Composite-APIs, and information exchange through Simple Object Access Protocol (SOAP), XML, JSON, and Present State Transfer (REST) protocols (IBM Cloud Education, 2020).

Software Development Kit (SDK)

The SDK helps software developers create applications for specific platforms and systems or programming languages. The SDK varies from manufacturer to manufacturer because each requires a development kit for a different cloud platform. In general, the default SDK includes a compiler, debugger, and API. The (Redhat SDKs) The AWS IoT SDK consists of an open-source library, a developer guide with samples, and a porting guide, allowing users to build innovative IoT products or solutions on the chosen hardware platform. The central computing languages used are C++, Python, JavaScript, and Java (AWS IoT Device SDK, n.d).

Cloud Computing Platforms

Cloud computing service platforms are needed to analyze and monitor the EM system’s collected data from WSN based IoT devices. The meaning of communication with the cloud computing platform encourages data processing for the transmission and monitoring of the collected data. In order to transmit and process the collected data to the cloud platform, the IoT-based cloud computing service platform area should be researched, and the cloud computing service platform should be selected according to the research content on the appropriate service.

There is no cloud platform for the best IoT service. In other words, each business has different specific requirements. For example, The AWS is the platform which is used the most and established as a high-level technology in the cloud platform field. However, there are various choices for each specific business requirement because the AWS platform is less efficient than other platforms in terms of cost (Palmer, 2020). However, according to Gartner's Magic Quadrant for Cloud Infrastructure and Platform Service, AWS, Microsoft Soft Azure, and GCP are sequentially listed in Figure 1 (Bala et al., 2021).

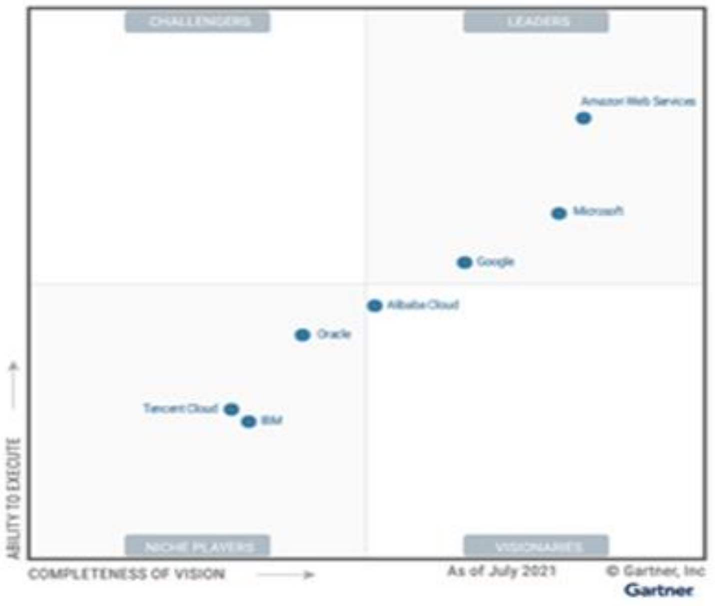


Figure 1. Gartner's magic quadrant for cloud platforms

Evaluating AWS and GCP Platform

The analyzed AWS and GCP platforms may not be perfect cloud platforms, but they can be fully used for connection with IoT devices. AWS and Google Cloud platforms have chosen two excellent cloud platforms that can connect with IoT devices. The analysis of both cloud platforms is for storing and analyzing data collected and transmitted from IoT devices. Table 1 and 2 is design and implementation objective approach format based on A Framework for Evaluating IoT Platforms: Application Provider Viewpoint (Mazhelis & Tyrvainen, 2014).

Table 1 analyzes the service design and implementation of publicly available platforms of AWS and GCP. The user interface can interact with IoT devices based on a dedicated app or web interface on the cloud platform. Also, the service provision form of each cloud platform and the form of additional services are compared through the service and deployment model. Table 2 is the objective approach format for AWS and GCP to connect to IoT devices. This is a comparison of additional services other than IoT Core required for the two cloud platforms to connect with IoT devices. In addition, Table 2 evaluates the unique SDK, message analysis, and device gateway of each cloud platform to recognize the different usage of the two cloud platforms.

Table 1. Design and implementation evaluating AWS vs. GCP

Design/Implementation	AWS	GCP
User Interface	A collection of components that help create intuitive and accessible user experiences for web applications. The AWS user interface is available under the terms of the Apache 2.0 open-source license. (The graphical user interface of Porting Assistant for .NET is now open source, n.d.)	A web-based graphical user interface that can be used to manage projects and resources. Create a new project in the Cloud Console or select an existing project and use the resources in that project. (Google Cloud overview | Overview, n.d.)

Service Model	AWS is basically providing an IaaS solution, but many other services exist and are provided as a mixture of IaaS, SaaS, and PaaS. (Types of cloud computing, 2019)	Google Compute Engine is an IaaS that allows clients to run workloads on Google's physical hardware. However, Google currently use internally its own products such as Google search, Gmail, and YouTube. So, Google Cloud provides IaaS, PaaS, and SaaS. (Bigelow, 2019)
Deployment Model	AWS is a comprehensive and evolving offering from Amazon, including IaaS, PaaS, and SaaS. (Types of cloud computing, 2019)	GCP is started as a Public Cloud. Hybrid and multi cloud deployment are being allowed to manage various workloads. (Bigelow, 2019)

Table 2. Operation evaluating AWS vs. GCP

Operation	AWS	GCP
Service Domain	IoT Core, EC2, SNS, SDK	IoT Core, PowerShell, gcloud, SDK

SDKs/API Language	AWS SDK – C++, Java, JavaScript, .NET, Node.js, PHP, Python, and Ruby (SDKs and Programming Toolkits for AWS, n.d.).	Cloud SDK - Java, Perl, PHP, Python, and C#, but any language that can interface with SOAP (Simple Object Access Protocol) can be used (Cloud Client Libraries, 2021)
Message Analysis	Publish/Subscribe, Simple Amazon Service (SNS), Amazon Simple Queueing Service (SQS) (Google, n.d.)	Publish/Subscribe, Publish/Subscribe Lite (Google, n.d.)
Device Gateway (Connection)	MQTT, HTTPS, WebSockets, LoRaWAN (AWS IoT Core Features - Amazon Web Services, n.d.).	MQTT and HTTP. (Using gateways | Cloud IoT Core Documentation, n.d.)
Pricing	AWS IoT Core – Free Tier	GCP – IoT Core – Free Tier

Free Tier (IoT Core)	2,250,000 minutes of connection. 500,000 messages. 225,000 Registry or Device Shadow operations. 250,000 rules triggered and 250,000 actions executed (AWS IoT Core Pricing - Amazon Web Services, n.d.).	Cost is estimated according to data volume. Up to 250 MB (Pricing | Cloud IoT Core, n.d.).

Amazon Web Service (AWS) IoT Core

The AWS IoT Core is an IoT platform provided by AWS's cloud service. The AWS IoT Core can connect IoT devices to the AWS cloud without providing and managing servers. The AWS IoT Core supports billions of devices and trillions of messages and enables routing to IoT devices through endpoint rules connected to the AWS for stable message processing. In addition, the AWS IoT Core allows tracking and communicating with all devices in the IoT application through the Device Shadow function even if the IoT devices are not connected to the Internet (AWS IoT Core Overview - AWSs, n.d.).

Advantages of the AWS IoT Core

Serverless Management and Connection of the Device.

The AWS IoT Core makes it easy to connect any number of devices to the cloud and other devices without the need to provision or manage servers. In other words, it helps to expand stably.

Preferred Protocol

The AWS IoT Core provides connecting and managing devices by selecting the best communication protocol for different use cases. MQTT(Message Queuing and Telemetry Transport), HTTPS(Hypertext Transfer Protocol - Secure), MQTT over WSS (WebSockets Secure), and LoRaWAN (low-power long-range wide-area network) are mainly supported by AWS IoT Core.

Secure Data and Device Connection

The AWS IoT Core provides end-to-end encryption at all connection points with automated configuration and authentication, so data exchange is not allowed without proven credentials, as well as fine-grained policies to protect access to devices.

Various Service

AWS can create more robust IoT applications by using AWS services such as AWS Lambda, Amazon Kinesis, Amazon S3, Amazon DynamoDB, Amazon CloudWatch, and Amazon Elasticsearch Service on the AWS Cloud Platform. (AWS IoT Core n.d)

AWS IoT Core MQTT Test

The AWS IoT Core publishes/subscribes messages from devices and clients using the MQTT v3.11 and MQTT over WSS protocols. It enables devices and clients using the HTTPS protocol to publish messages. (AWS IoT Core MQTT) Figure 2 is the operation order of AWS IoT Core MQTT. (MQTT - AWS IoT, n.d.)

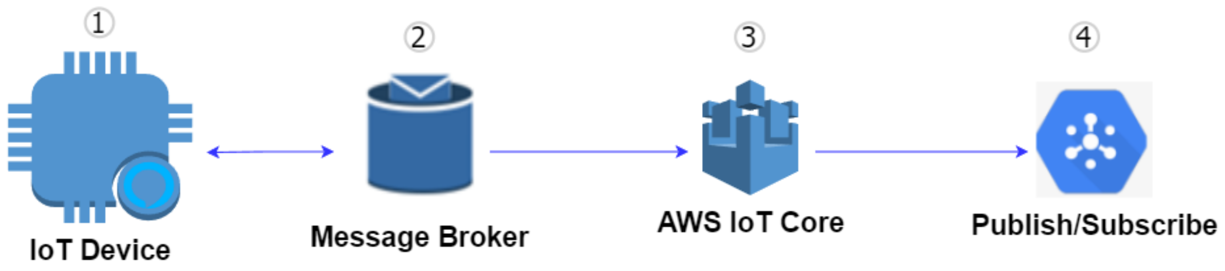


Figure 2. AWS IoT Core MQTT framework

1. IoT Device - Billions of devices create the messages that are collected from the sensor.
2. Message broker - Message brokers transmit the data to AWS IoT Core by using the MQTT communication protocol.

3. AWS IoT Core – AWS IoT Core receives the transmitted data from message brokers to analyze and monitor data.
4. Analyzed and monitored data is sent to publish/subscribe message analysis to notify events of data.

Device Shadow

The device shadow allows checking the status of the device in apps and other services, regardless of whether the device is connected to the AWS IoT Core. In other words, the device shadow proposes to store the best condition of the device so that the device can be checked, set, and managed at any time. In addition, setting the initial state to be implemented when the device is reconnected (AWS IoT Device Shadow service - AWS IoT, n.d.)

Google Cloud Platform (GCP) IoT Core

The GCP IoT Core is a complete management service that easily and safely connects, manages, collects, and secures hundreds of millions of messages from millions of globally distributed IoT devices. The GCP IoT Core combines with various services provided by GCP to present a complete solution to collect, process, analyze, and visualize IoT data in real-time and provide efficient operation to increase cost-saving efficiency (*Cloud IoT Core*, n.d.).

Main components of the GCP IoT Core are device management and protocol bridges. The device management for registering devices in IoT cloud services can monitor and configure through connected IoT devices' collected data. Two protocol bridges can be used when the IoT device is connected to the GCP through HTTP and MQTT communication protocol. The device's collected data is forwarded to a cloud's Publish/Subscribe topic, which can then trigger cloud functions through two main components of GCP. The GCP IoT Core also supports the data for communication and management between IoT devices and the GCP through HTTP and MQTT communication protocol. The GCP IoT Core uses Transport Layer Security (TLS) transport over TCP port 8883 that is to support open-source clients. Also, TLS ensures all traffic between devices with credentials and message broker with encryption. The MQTT Bridge provides QoS levels 0 and 1 (*Publishing over the MQTT bridge* | *Cloud IoT Core Documentation*, n.d.).

The GCP IoT Core has their own SDK. The GCP IoT Core's SDK manages authentication, local configuration, developer workflow, and interactions with API through gcloud Command Line. The GCP SDK comes with C#, Java, Node.js, GO, PHP, Python, and Ruby. In addition, authentication is made by unique rules, type of supported language, and optimizing workflows through the GCP. (*Getting started* | *Cloud IoT Core Documentation*, n.d.).

GCP IoT Core MQTT Test

According to the GCP IoT Core MQTT test framework in Figure 3, the connection between GCP IoT Core and IoT device was performed based on the MQTT device gateway and publish/subscribe message analysis mentioned in Table 2.

Device Gateway attempts to connect between the IoT device and GCP IoT Core. If a communication device such as Bluetooth or ZigBee cannot be directly connected by itself, an IoT device and GCP IoT Core can be connected using a specific communication protocol through an Internet connection. Device gateways can publish their own telemetry data and receive data by reporting status. As mentioned in Table2, GCP IoT Core supports gateway connections and communication via MQTT and HTTP. (Using gateways | Cloud IoT Core Documentation, n.d.)

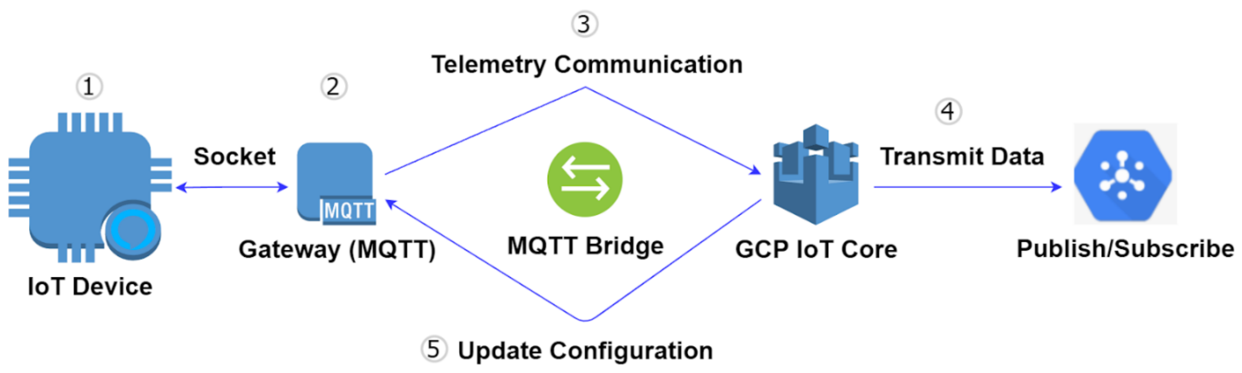


Figure 3. GCP IoT Core MQTT framework

1. Authenticate the IoT device to connect GCP IoT Core through the gateway.
2. Configure the MQTT as a gateway and send the collected data to the GCP IoT Core
3. Use the MQTT bridge to relay the messages from IoT device.
4. Transmit the collected data from MQTT bridge to publish/subscribe message analysis.
5. GCP IoT Core can send updated data that is analyzed in GCP IoT core to IoT device.

Chapter 3 proposed a methodology to run a SEM system in a smart environment. Methodology proposed collaboration between WSN-based IoT devices and cloud computing, not basic IoT devices and cloud computing. AWS and GCP cloud computing platforms' objective evaluation approaches have been proposed. In addition, components that can connect IoT devices and cloud computing platforms have been suggested. Based on the suggested components, AWS and GCP could operate the MQTT framework as an example of SEM system communication.

CHAPTER FOUR:
SMART ENVIRONMENT MONITORING SYSTEM

Smart Environment Monitoring (SEM)

The smart city environment of IoT devices consists of sensors and actuators. IoT devices store the data that can be used to analyze them. IoT devices are connected via a Wireless Network Sensor (WSN) to collect data from the environment and infer the device's health based on itself. A WSN can be implemented through the Software Development Kit (SDK), a language kit that can execute data, and cloud platforms provide services to help it (Rahmati et al., 2015).

The WSN is a computer network composed of sensor nodes. It consists of one network to monitor the surrounding environment using sensor nodes. The WSN is a network without a fixed infrastructure between the sensor node and the end device (Ghazal et al., 2021). The WSN collects data while SEM is monitoring the required data. The WSN works through a densely distributed sensor node to reliably cover the monitoring area (Pinto et al., 2014).

The monitoring and controlling method of an artificial intelligence (AI) based smart environment via the WSN consists of the latest sensors. WSN's AI-based IoT devices are effectively used for air pollution, water waste management, and soil management. In other words, a smart environment is monitored and managed through IoT, AI, and WSN. Moreover, SEM data not only collected from IoT, AI, and the WSN but also SEM data can be stored, executed, and secured through a cloud computing service platform (Bhoomika et al., 2016).

Smart Environment Monitoring (SEM) Applications

Smart Agriculture Monitoring (SAM)

Much research is being conducted in the field of smart environment agriculture. Most smart agriculture projects use the WSN of IoT devices to collect data from sensors in various nodes and transmit the collected data through wireless communication protocols to cloud computing service platforms. The collected data provides information on SEM devices due to various environmental factors. Promoting productivity growth of crops through monitoring of environmental factors is not a complete solution. Monitoring environmental factors that affect productivity is the more primary purpose of monitoring (Suma et al., 2017).

With the IoT technology, smart agriculture can perform tasks such as weaving, spraying, and soil humidity and temperature monitoring through smart GPS-based remote-control robots (Gondchawar & Kawitkar, 2016).

A cloud computing system is used that observes the collected data from the WSN of the IoT device and precisely feeds the data along with the location via GPS coordinates to the storage. In other words, the SEM system communicates wirelessly by connecting smart agriculture’s collected data from WSN of IoT devices to cloud computing services (Gayatri et al., 2015).

Norwegian telecommunications group, DTAC launched the SAM)IoT solution in partnership with Thailand's Department of Agricultural Extension (DOAE) and the National Electronics and Computer Technology Center (NECTEC) in 2017. Through the new smart agricultural solution, telecommunication companies and government agencies have provided necessary technologies to farms in Thailand, such as climate change, plant diseases, and soil moisture monitoring. The proposed solution was carried out through a wireless Internet connection to connect cloud computing services. NECTEC developed and studied the WSN of the IoT devices to find standard parameter indicators of smart agriculture to collect correct data information. As a result, crop production increased, agricultural quality improved, and helped reduce production costs. Also, the proposed efficient solution encourages Thailand farmers and reduces the inconvenience and inequality of farmers’ labor (Telenor Group, 2017).

Smart Water Quality Monitoring (SWQM)

The Smart Water Quality Monitoring (SWQM) is an efficient system designed to monitor the water quality of drinking water through IoT technology in the smart city environment. The SWQM is a system that not only reduces cost and workforce but also increases reliability. The SWQM is a technology that measures collected parameter data through the water's pH value sensor, the water level present sensor in the tank, and the water turbidity sensor and temperature sensor capable of wireless connection of IoT devices. The collected parameter data is sent to the PC through the Micro Controller Unit (MCU) to measure the water quality, and then the measured data is transmitted to the IoT-based cloud platform application through the wireless communication protocol. (Pasika & Gandla, 2020).

Fiji Islands in the Pacific Ocean are companies that provide clean drinking water. Fiji Island water requires regular collection and analysis of collected data to upload data from the SWQM to the server. The IoT device and the WSN technology have been used to monitor water quality (Prasad et al., 2015). Fiji Island used the WSN of IoT devices for the SWQM to connect accurate and consistent data to cloud servers and effectively maintained water quality by accessing the necessary information in real time (Mamun et al., 2019).

Smart Air Quality Monitoring (SAQM)

Before the SAQM was studied and used, existing air quality monitoring systems had the disadvantages of being large, heavy, and costly. For this reason, the monitoring station is increasing along with the research and development of the monitoring system. The air pollution situation in the smart city environment has worse air quality than average due to human activity and location dependence (Yi et al., 2015).

The SAQM is a system and facility that continuously measures wind speed, direction, weather variables, and air pollutant concentrations such as SO2, NOx, CO, O3, THC (Arroyo et al., 2019).

The measured and collected data is IoT-based and monitors air quality through a web server using the Internet to issue an alarm or notification when the measured data is above the standard level of air quality (Yi et al., 2015). The SAQM is reported primarily based on the Air Pollution Index (API) calculated from five reference parameters of the atmosphere: PM10, PM 2.5, carbon monoxide, nitrogen dioxide, ozone, and temperature (Murad & Pereira, 2011).

Based on IoT's WSN, the SAQM system operates through a system that measures the quality of the atmosphere consisting of a distributed sensor network connected to the cloud system. The WSN of SAQM transmits field-measured data to the cloud in real-time through a cloud gateway. The SAQM provides a solution to detect and optimize pollutants in the atmosphere by applying AI technology that stores, monitors, and analyzes the collected data. The SAQM not only can multiple nodes be deployed, but also can share information related to air quality distribution in different regions through cloud services, which can improve efficiency in measuring and comparing air quality. In addition, low power consumption and small scale can measure atmospheric quality at a low cost. (Arroyo et al., 2019).

The two main causes of air pollution in Korea are automobile exhaust gas and PM2.5 pollutants caused by strong winds from China. Pollutants caused by low airflow travel all over China during the cold winter. In spring, strong winds blow by partial winds along with high airflow, causing air pollution. The rapidly developing Chinese economy burns about 4 billion tons of coal every year. For this reason, air pollution in Korea has emerged as a big problem (IQ Air, n.d.).

Korea Telecommunication (KT) invested 10 billion won in 2017 to provide more accurate standby data and installed an air quality monitoring solution at base stations across Korea. KT has launched a large air map Korea project that can collect air quality data information by installing surveillance infrastructure through 4.5 million power poles, 330,000 mobile base stations, and 4,000 central offices. KT said that fine dust, volatile organic compounds, noise, and humidity can be measured in a short time through the WSN-based monitoring system of the IoT devices. In addition, big data analysis is used to select the area where air quality can be measured most effectively. Moreover, the collected data is provided to the government by selecting an open-oriented IoT cloud platform (Tomás, 2017).

CHAPTER FIVE:

ENVIRONMENT MONITORING ANALYSIS

Analysis by Using AWS IoT Core

Table 3. Smart environment monitoring application analysis

App	Purpose	Contributory factor	Parameter	Analysis Method	Condition
Smart Agriculture Monitoring (SAM)	Soil monitoring for farming	Deforestations, discharge of industrial waste, Use of animal and human waste in agriculture, disposal of farm and poultry waste	Soil moisture level, Salinity – PPM Temperature, pH level (Arroyo et al., 2019).	A WSN based IoT device will collect and send the data to AWS IoT Core by using MQTT communication protocol.	Soil Moisture > 50%

Smart Water Quality Monitoring (SWQM)	Water quality monitoring for citizen	Poor monitoring of water, Leakage of pipe, Contaminated raw water source	pH level, Mounted water level, Water turbidity , Water temperature. (Pasika & Gandla, 2020).		pH level > 8.6
Smart Air Quality Monitoring (SAQM)	Air quality monitoring system for citizen	Open burning, Industrial emissions, Vehicle’s emissions, Bushfire	AQI level, PM10, PM 2.5, Temperature (Murad & Pereira, 2011).		AQI level > 150

The purpose of the analysis is to manage and secure nature-friendly infrastructure through SAM, SWQM, and SAQM, which are environmental monitoring systems for smart environments, simultaneously with the establishment of smart cities.

As mentioned in Table 3, SAM's soil monitoring system was developed to measure and monitor the moisture level, temperature, and pH values of the soil. Soil monitoring systems can have a significant impact on agricultural management of smart city environments by collecting soil conditions and status by quickly updating information through soil condition monitoring (Athani et al., 2017).

SWQM is a water quality system to provide clean water to citizens by establishing a water quality monitoring system to build a smart city environment. The proposed water quality monitoring system measures the pH level, turbidity, temperature, and mounted water level of water (Pasika & Gandla, 2020).

The SAQM system is the most important system for measuring air quality in a smart city environment. It is a monitoring system developed to monitor the minimum harmful air pollutants that can adversely affect health through the establishment of a smart environment for citizens of smart cities. Through the monitoring system, the Air Quality Index (AQI) level can be measured to measure API levels that are fatal to humans (Murad & Pereira, 2011).

Data collected from IoT devices for the above environmental monitoring system is transmitted to the cloud platform IoT service domain to analyze and monitor environmental conditions and status. By analyzing and monitoring the transmitted data, the monitoring system can detect abnormalities through data analysis.

Table 3 mentioned the method of this analysis. The mentioned MQTT test transmits and analyzes collected data by environmental monitoring IoT devices to the cloud platform through the MQTT communication protocol. The data analyzed on the cloud platform monitors environmental parameters that can have a fatal impact on citizens. Analysis of this project can set and generate notifications that can be received via email, SMS, or the unique application of each monitoring when environmental parameters are figures that can affect citizens.

Analysis proposes environmental management through environmental monitoring systems using MQTT communication protocols between WSN-based IoT devices and cloud platforms. WSN-based IoT devices can be remotely monitored, and data controlled. In addition, the analysis has the advantage of being able to obtain the current status of the collected data in real time whenever necessary.

The actual analysis generated data by creating virtual WSN-based IoT devices in the IoT service domain of the cloud platform without using WSN-based IoT devices such as Raspberry Pi or Arduino devices that include sensors that can collect data. Analysis tests virtualized WSN-based IoT devices on laptop computers using SDKs on cloud platforms such as JSON, Python, Node.js, and Structured Query Language (SQL) needed to run services for proper IoT device creation on cloud platforms.

Information of the Project Analysis:

Model: Asus Rog Zephyrus G15

Specification: AMD Ryzen 9 4900HS CPU, 3.0GHz, 24GB RAM

Cloud Platform: AWS and GCP

The analysis through AWS IoT is carried out based on objective approaches, table 2 and 3. The AWS IoT Core analysis proves and analyzes the communication connection between the IoT device and the AWS Cloud of the SAM, SWQM, and SAQM applications mentioned in Table3. The main purpose of this analysis is not only to analyze the communication connection method and communication connection protocol between the two technologies, but also to build a monitoring system by storing and analyzing the data transmitted after connection. The built monitoring system can detect the most basic function of dangerous environment parameter values and build a function to send notifications. Additionally, AWS IoT Core's unique defense and security can provide credibility and reliability between the two technologies.

Below are the steps which are the sequence of Soil monitoring system configuration by using the AWS IoT Core.

1. Register the IoT devices in AWS IoT Core.
2. Create device shadow and MQTT test.
3. Create Rule and Action
4. Sending a data Device to AWS IoT Core

5. Email Notification

6. Defend

7. Security

The step of the analysis

1. Register the Thing in AWS IoT Core.

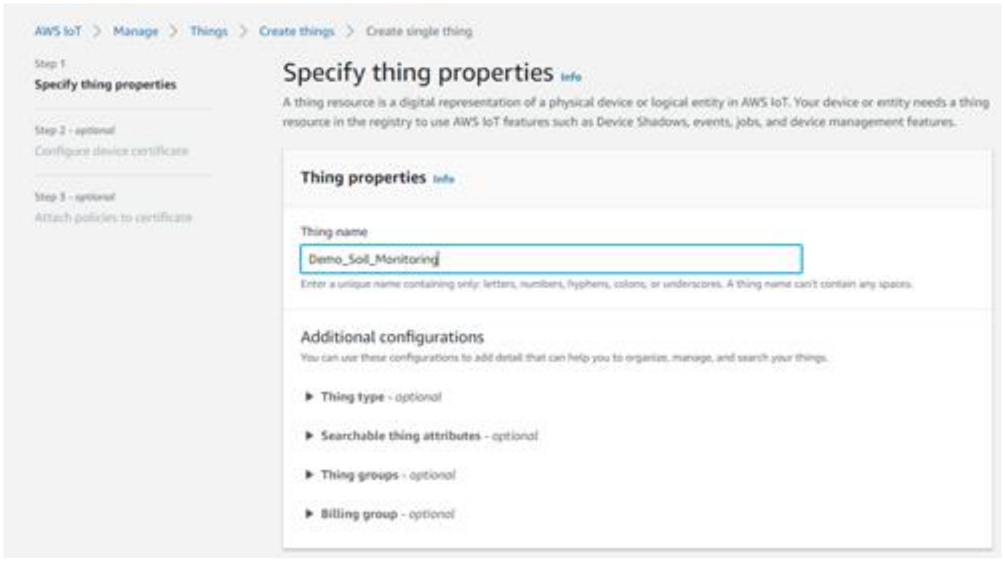
Table 4. Registering SEM IoT devices in the AWS IoT Core

Application	SAM	SWQM	SAQM
Connect IoT devices	Demo Soil Monitoring	Demo Water Quality Monitoring	Demo Air Quality Monitoring

Access policy	{
	  "Version": "2012-10-17",
	  "Statement": [
	    {
	      "Effect": "Allow",
	      "Action": "iot:*",
	      "Resource": "Each Applications ARN"
	    }
	  ]
	}		

Certificate Keys	Certificate.pem, PrivateKEY.pem, PublicKEY.pem, AmazonRootCA1.pem (Each application has own certificate keys)		

a. Register the connected IoT devices on the IoT Core. The name of the IoT devices is in the table 4 of “Connect IoT device” for this test.


AWS IoT > Manage > Things > Create things > Create single thing
Step 1
Specify thing properties
Step 2 - optional
Configure device certificate
Step 3 - optional
Attach policies to certificate
Info
Info

Specify thing properties

A thing resource is a digital representation of a physical device or logical entity in AWS IoT. Your device or entity needs a thing resource in the registry to use AWS IoT features such as Device Shadows, events, jobs, and device management features.

Thing properties

Thing name: Demo_Soil_Monitoring

Enter a unique name containing only: letters, numbers, hyphens, colons, or underscores. A thing name can't contain any spaces.

Additional configurations

You can use these configurations to add detail that can help you to organize, manage, and search your things.

▶ Thing type - optional

▶ Searchable thing attributes - optional

▶ Thing groups - optional

▶ Billing group - optional

Figure 4-1. Create a Soil_Monitoring device

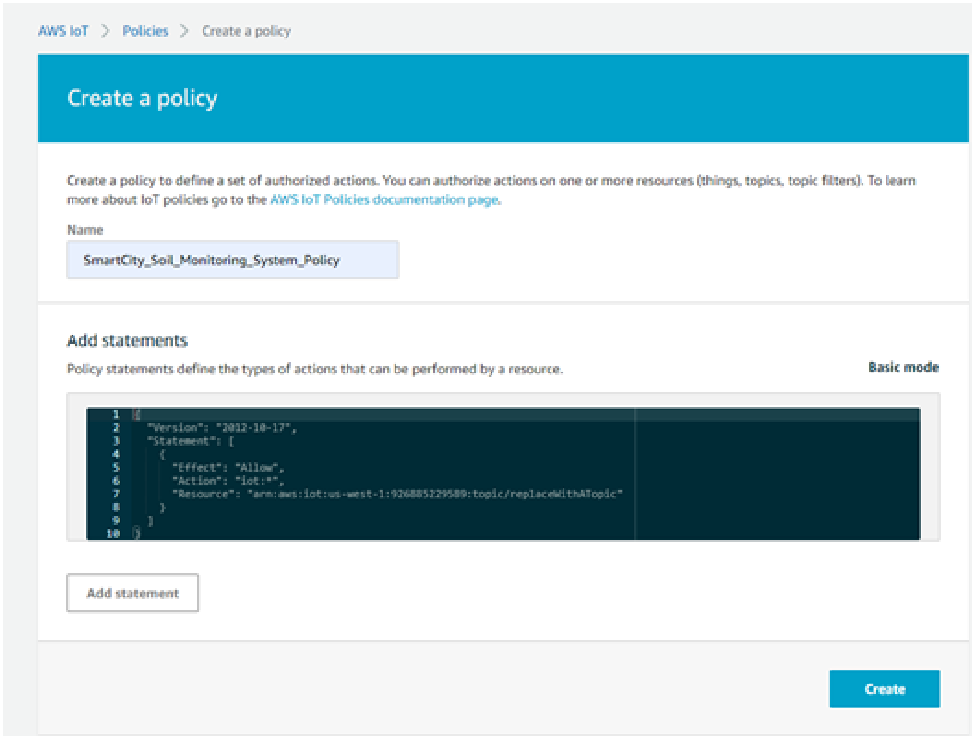
b. Creating the policy is for the set of the authorization for action. This policy is created by the JSON language. "Action": "iot:*" means allowing actions related to all the Internet of Things.

"Resource":arn:aws:iot:us-west-1:926885229589:topic/Demo_Soil_Monitoring"

"Resource":arn:aws:iot:us-west-1:926885229589:topic/Demo_Water_Quality_Monitoring"

"Resource":arn:aws:iot:us-west-1:926885229589:topic/Demo_Air_Quality_Monitoring"

Three above resources are a system rule that specifies resources and actions for a specific access function. There is a server or file that can be accessed through this, and the operation is not performed by performing "Allow" or "Deny" through the resource.



Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the AWS IoT Policies documentation page.

Name: SmartCity_Soil_Monitoring_System_Policy

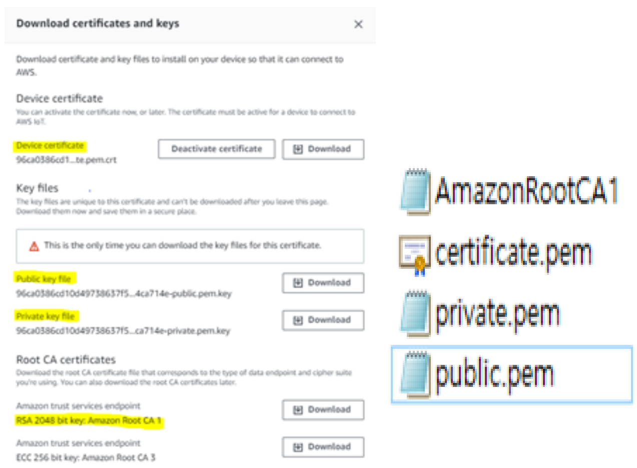
Policy statements define the types of actions that can be performed by a resource.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:*",
      "Resource": "arn:aws:iot:us-west-1:926885229589:topic/replaceWithATopic"
    }
  ]
}
```

Figure 4-2. Create a policy for IoT device creation

c. Connect the registered IoT devices and create the access policy.

d. Download Device certificate.pem, publickey.pem, privatekey.key, and AmazonRootCA1 (RSA 2048 bit key).



Download certificate and key files to install on your device so that it can connect to AWS.

Device certificate

You can activate the certificate now, or later. The certificate must be active for a device to connect to AWS IoT.

Device certificate
96ca0386cd1...te.pem.crt

Key files

The key files are unique to this certificate and can't be downloaded after you leave this page. Download them now and save them in a secure place.

This is the only time you can download the key files for this certificate.

Public key file
96ca0386cd10d49738637f5...4ca714e-public.pem.key

Private key file
96ca0386cd10d49738637f5...ca714e-private.pem.key

Root CA certificates

Download the root CA certificate file that corresponds to the type of data endpoint and cipher suite you're using. You can also download the root CA certificates later.

Amazon trust services endpoint
RSA 2048 bit key: Amazon Root CA 1

Amazon trust services endpoint
ECC 256 bit key: Amazon Root CA 3

Figure 4-3. Download credential keys for IoT device

2. Create device shadow and MQTT test

Table 5. Creating device shadow and MQTT test with topic

Connected IoT Device	Demo_Soil_Monitoring	Demo_Water_Quality_Monitoring	Demo_Air_Quality_Monitoring
Device Shadow - MQTT topic prefix	$aws/things/"Connected IoT Device"/shadow		
MQTT topic	$aws/things/"Connected IoT Device"/shadow/update, $aws/things/"Connected IoT Device"/shadow/get, $aws/things/"Connected IoT Device"/shadow/get/update		
JSON query for MQTT test	{ "state" : { "desired" : { "welcome" : null }, "reported" : {	{ "state" : { "desired" : { "welcome" : null }, "reported" : {	{ "state" : { "desired" : { "welcome" : null }, "reported" : {

	"welcome" : null, "SoilMoisture" : "60", "temperature": “39C”, "AirHumidity": "30%", "Device_id": "SAM_01" } } }	"welcome" : null, "pHLevel" : 9, "temperature": "25C", "MountedLevel": "80%", "Turbidity": "30 NTU", "Device_id": "SWQM_1" } } }	"welcome" : null, "AQI" : 155, "Temp": "25C", "PM2.5": 5.0, "PM10": 54, "device_id": "SAQM_1" } } }
QoS Level	Level 0 or 1		

a. Create device shadow

The AWS IoT Core allows tracking and communicating with all devices in the applications through the device shadow feature, even if IoT devices don't have an Internet connection. (AWS IoT core n.d) The device shadow allows connected devices to sync their state with the AWS. The device shadow can also get, update, or delete the state of device information by using MQTT topics.

b. MQTT topic

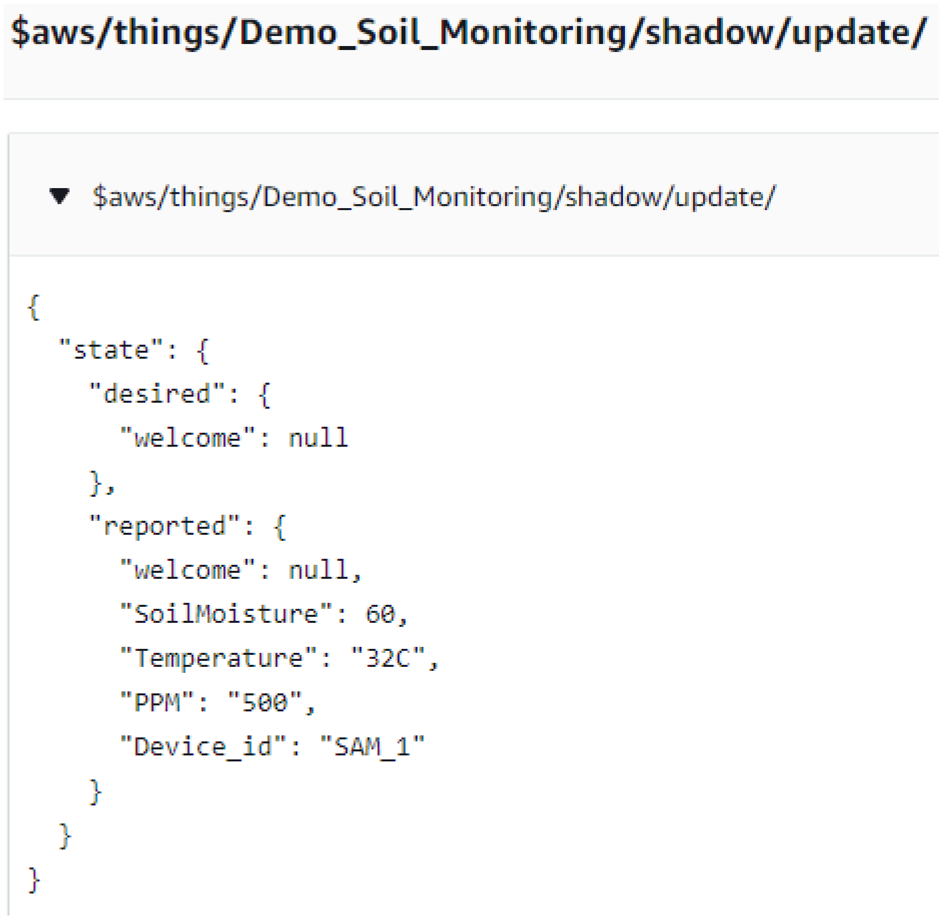
The device shadow service uses these reserved MQTT topics to get, update, and delete a device's state information in the device shadow. MQTT topics for this device shadow allow applications to publish and subscribe to MQTT messages that interact with the IoT device’s device shadow.

c. MQTT test

The MQTT test is to monitor MQTT messages delivered to the AWS account by the client. Connected IoT devices to the AWS IoT Core post MQTT messages identified by item to deliver the status of the environmental parameters to the MQTT message. The AWS IoT Core also publishes MQTT messages to notify devices and applications of changes and events.

The JSON queries for MQTT test of each application are created as a demo version analysis. Especially, the environment parameters are chosen based on research. The information and source about environmental parameters in Table 3.

The MQTT test subscribed and published a total of 3 topics: /update, /get, get/accepted in the EM application. Levels of QoS were able to select 0 and 1. QoS level 0 is chosen because it will deliver the message at most once. MQTT payload display is auto-format JSON payloads because it improves readability. For example, $aws/things/Demo_Soil_Monitoring/shadow/update. It publishes a request state document to this topic to update the device's shadow


$aws/things/Demo_Soil_Monitoring/shadow/update/

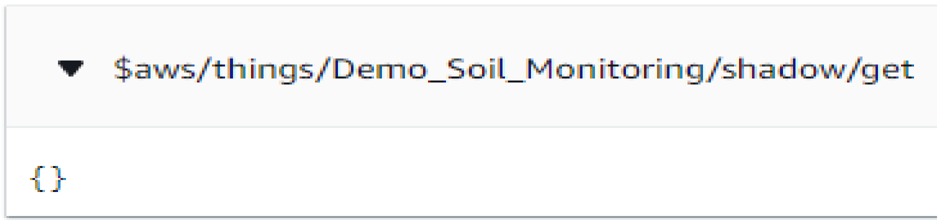
▼ $aws/things/Demo_Soil_Monitoring/shadow/update/

```
{
  "state": {
    "desired": {
      "welcome": null
    },
    "reported": {
      "welcome": null,
      "SoilMoisture": 60,
      "Temperature": "32C",
      "PPM": "500",
      "Device_id": "SAM_1"
    }
  }
}
```

Figure 5-1. Update state document of Soil_Monitoring topic

$aws/things/Demo_Soil_Monitoring/shadow/get

It publishes an empty message to this topic to get the device's shadow.

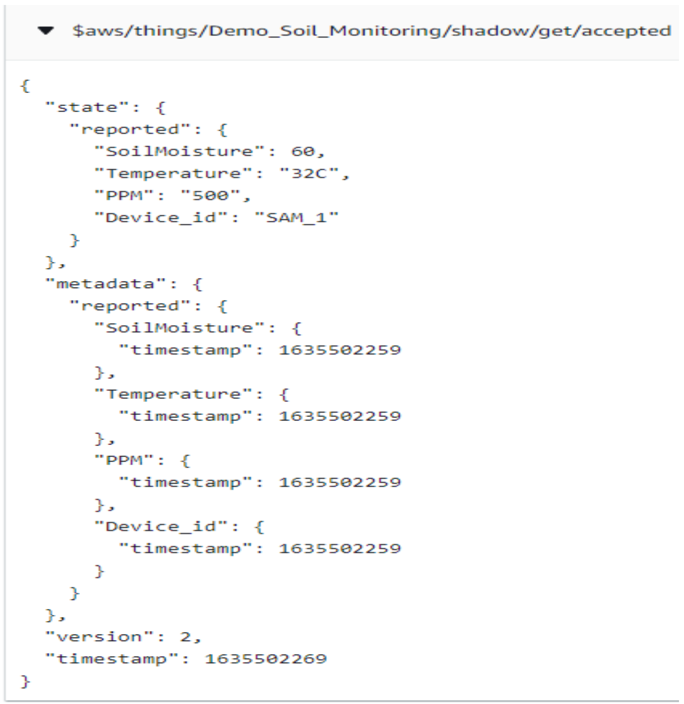

▼ $aws/things/Demo_Soil_Monitoring/shadow/get

```
{}
```

Figure 5-2. Empty message to Soil_Monitoring topic

$aws/things/Demo_Soil_Monitoring/shadow/get/accepted

AWS IoT Core publishes a response shadow document to this topic when returning the device's shadow.



$aws/things/Demo_Soil_Monitoring/shadow/get/accepted

```
{
  "state": {
    "reported": {
      "SoilMoisture": 60,
      "Temperature": "32C",
      "PPM": "500",
      "Device_id": "SAM_1"
    }
  },
  "metadata": {
    "reported": {
      "SoilMoisture": {
        "timestamp": 1635502259
      },
      "Temperature": {
        "timestamp": 1635502259
      },
      "PPM": {
        "timestamp": 1635502259
      },
      "Device_id": {
        "timestamp": 1635502259
      }
    }
  },
  "version": 2,
  "timestamp": 1635502269
}
```

Figure 5-3. Response shadow of Soil_Monitoring

3. Create Rule and Action

a. Create Rule and query statement

Table 6. Creating rule and action for email notification

Rule	SAM rule	SWQM rule	SAQM rule
Rule query by SQL	SELECT*FROM '$aws/things/Demo_Soil_Monitoring/shadow/update/accepted' WHERE state.reported.SoilMoisture > 50	SELECT*FROM '$aws/things/Demo_Water_Quality_Monitoring/shadow/update/accepted' WHERE state.reported.pHLevel > 8.6	SELECT*FROM '$aws/things/Demo_Air_Quality_Monitoring/shadow/update/accepted' WHERE state.reported.AQI > 150
Format of Notification	Raw data by email		
Endpoint (email)	003870099@coyote.csusb.edu		

Create a rule to evaluate messages sent by IoT devices and specify what to do when a message is received. Names of rule are in Table 6. Rules were created to notify data changes and critical environmental parameters.

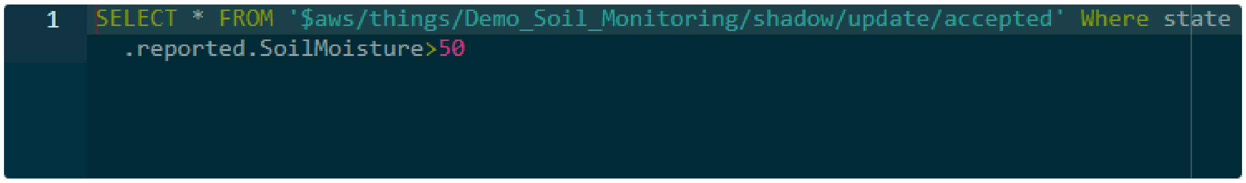
Rule query statement indicates the source of the messages that will be processed with the created rule. Rule query statement uses the Structured Query Language (SQL) version 2016-03-23. The rule of SQL statement is SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>.

SELECT*FROM

'$aws/things/Demo_Soil_Monitoring/shadow/update/accepted' WHERE

state.reported.moisture > 5 is a good example of the SQL rule in

Demo_Soil_Monitoring.



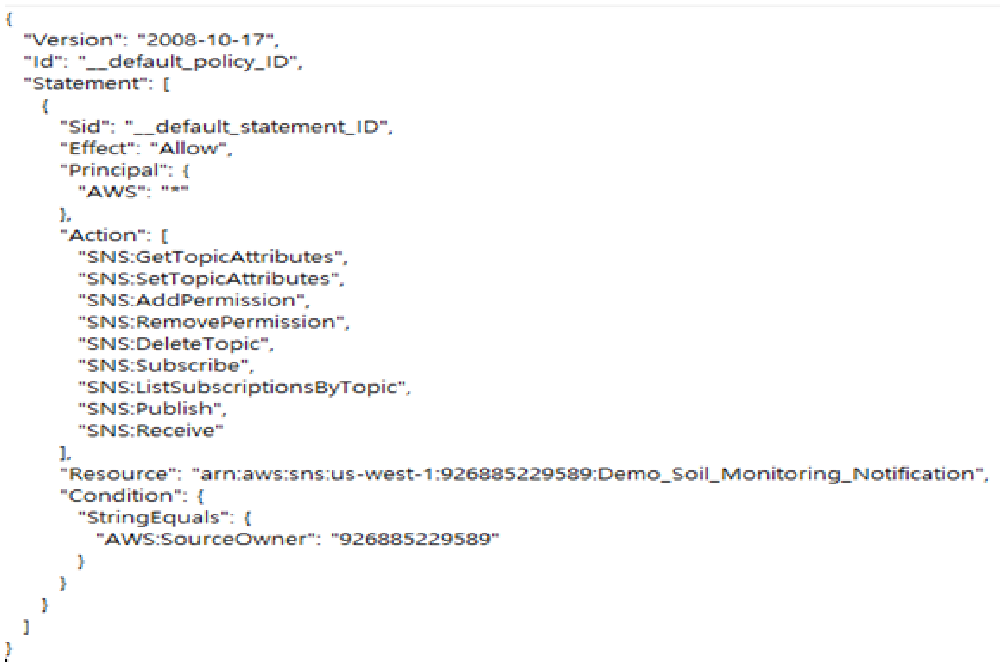
```
SELECT * FROM '$aws/things/Demo_Soil_Monitoring/shadow/update/accepted' Where state
  .reported.SoilMoisture>50
```

Figure 6-1. Rule query statement of Soil_Monitoring

b. Add action for notification by using the AWS Simple Notification Service (SNS)

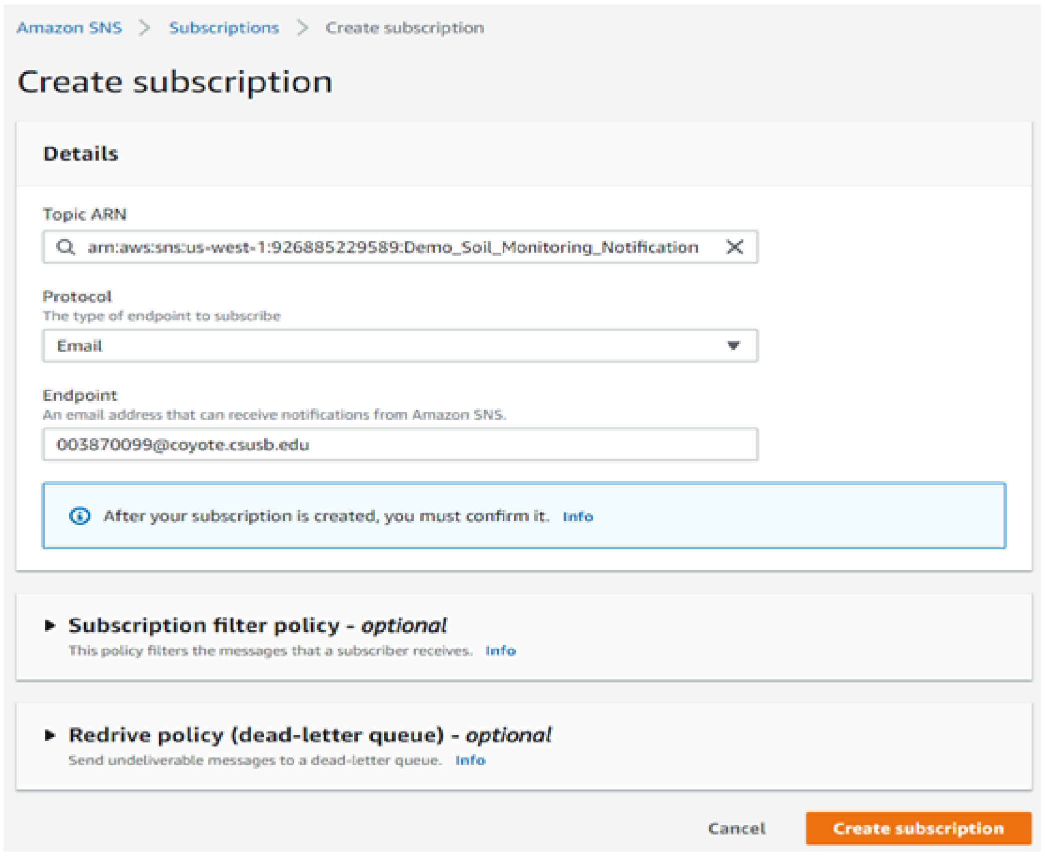
The SQL rule above allows the selection actions that can be performed when matching the inbound message. The SNS action is a functional addition that allows messages to arrive, such as storing in the AWS S3 database, calling cloud functions in the AWS CloudWatch, or push notification in the AWS SNS. The statement of SQL can be found in Table 6.

The SEM systems have enabled receiving notification of data changes via email through the AWS SNS. The format of an e-mail notification is raw because raw data is the first raw measurement data obtained by the sensor. Figure 8-2 is an example of an access policy for roles to grant AWS IoT. The access policy defines who can access the MQTT topic. By default, only the topic owner can publish or subscribe to the topic.


```
{
  "Version": "2008-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "__default_statement_ID",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "SNS:GetTopicAttributes",
        "SNS:SetTopicAttributes",
        "SNS:AddPermission",
        "SNS:RemovePermission",
        "SNS:DeleteTopic",
        "SNS:Subscribe",
        "SNS:ListSubscriptionsByTopic",
        "SNS:Publish",
        "SNS:Receive"
      ],
      "Resource": "arn:aws:sns:us-west-1:926885229589:Demo_Soil_Monitoring_Notification",
      "Condition": {
        "StringEquals": {
          "AWS:SourceOwner": "926885229589"
        }
      }
    }
  ]
}
```
Figure 6-2. Role to grant AWS IoT access policy

c. Configure AWS Simple Notification Service (SNS)

The AWS SNS can find the SEM applications notification topic. Subscription settings are required to receive a notification. The protocols for receiving notification vary widely from Amazon Kinesis, SQS, Lambda, Email, Email-JSON, HTTP, HTTPS, Application endpoint, and SMS. Demo_Soil_Monitoring test has been tested to receive notification through email as an example.


Amazon SNS > Subscriptions > Create subscription

Create subscription

Details

Topic ARN

arn:aws:sns:us-west-1:926885229589:Demo_Soil_Monitoring_Notification

Protocol

The type of endpoint to subscribe

Email

Endpoint

An email address that can receive notifications from Amazon SNS.

003870099@coyote.csusb.edu

After your subscription is created, you must confirm it. Info

▸ Subscription filter policy - *optional*

This policy filters the messages that a subscriber receives. Info

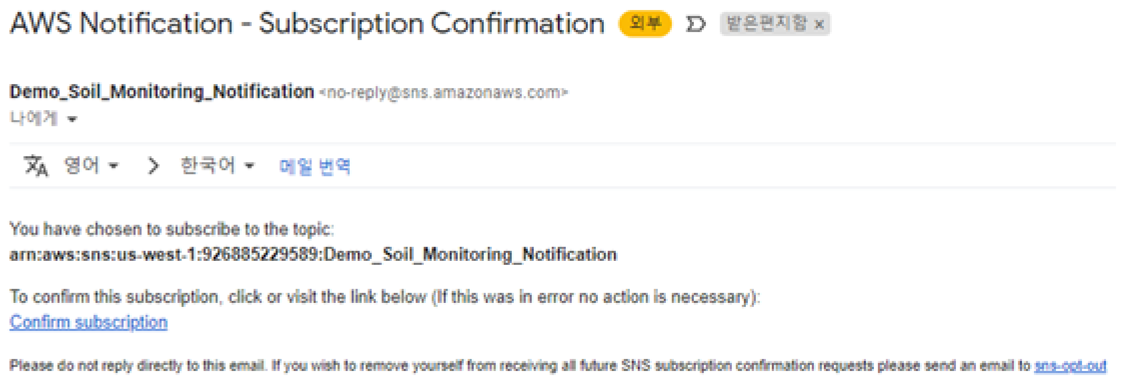
▸ Redrive policy (dead-letter queue) - *optional*

Send undeliverable messages to a dead-letter queue. Info

Figure 6-3. Create subscription of SNS through email

d. Confirm subscription of AWS notification

When Completing the Create subscription, a confirmation email will be received to the endpoint email address Figure 8-3 is describing creating subscriptions. After receiving the confirmation email in Figure 8-4, click the 'Confirm subscription' link to complete the AWS SNS connection setting up through the rule and action settings of the AWS IoT Device.



AWS Notification - Subscription Confirmation 외부 받은편지함 ×

Demo_Soil_Monitoring_Notification <no-reply@sns.amazonaws.com>
나에게

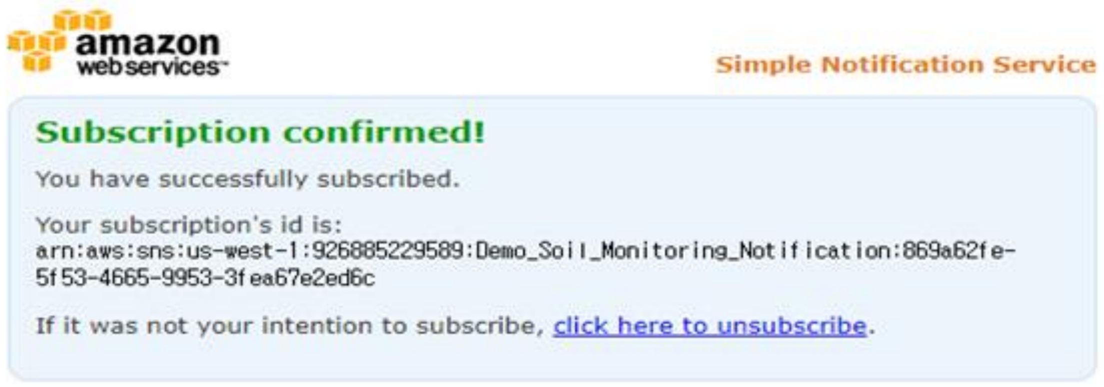
영어 > 한국어 메일 번역

You have chosen to subscribe to the topic:
arn:aws:sns:us-west-1:926885229589:Demo_Soil_Monitoring_Notification

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):
Confirm subscription

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to sns-opt-out

Figure 6-4. Confirm notification email



Simple Notification Service

Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:
arn:aws:sns:us-west-1:926885229589:Demo_Soil_Monitoring_Notification:869a62fe-5f53-4665-9953-3fea67e2ed6c

If it was not your intention to subscribe, click here to unsubscribe.

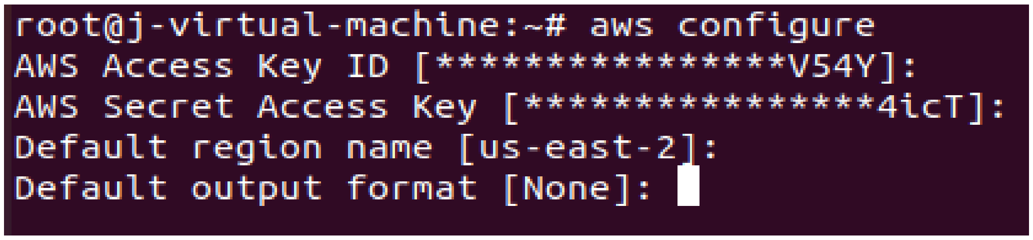
Figure 6-5. Notification of subscription message

4. Sending device data to AWS IoT Core

State document data of each SEM application obtained through the MQTT test. The MQTT test is operated on the Ubuntu Linux 20.04 LTS operating system through the AWS Cloud Service, EC2 virtual machine. In the Virtual Machine, the MQTT test is conducted through AWS SDK Python. Below is the order of the MQTT test through the virtual machine

a. AWS configure

Configure AWS server in the virtual machine with AWS access key ID, AWS secret Access key, Default region name, and Default output format. AWS Access Key ID and AWS Secret Access Key are created by AWS Identity and Access Management (IAM).

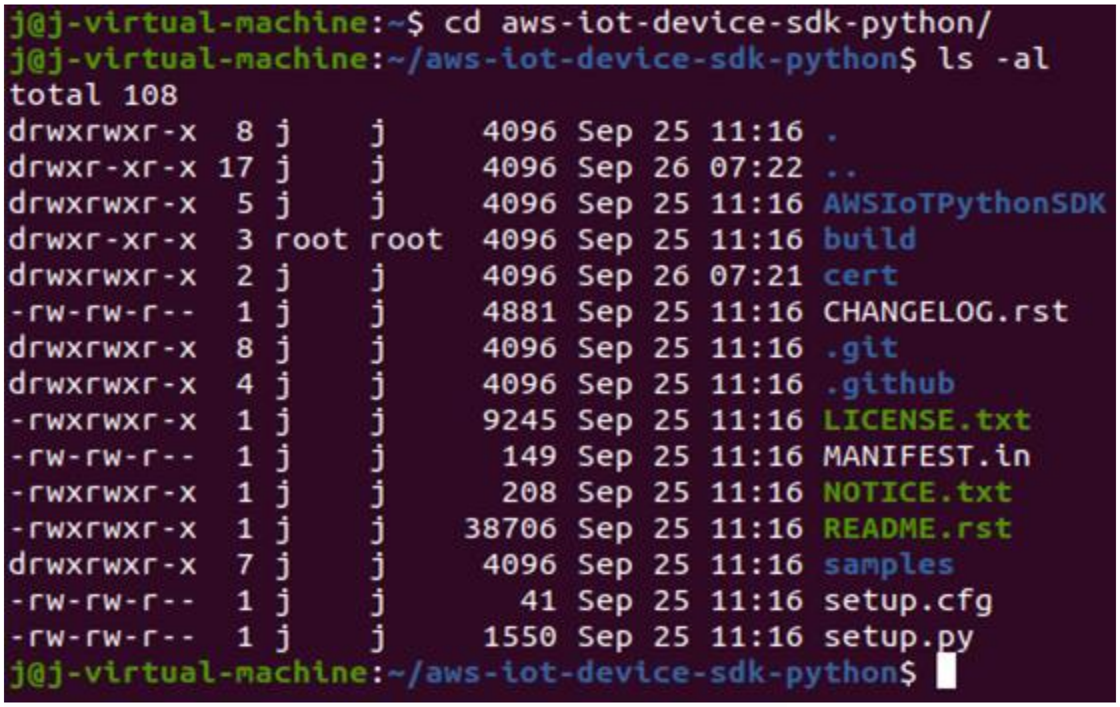


```
root@j-virtual-machine:~# aws configure
AWS Access Key ID [****************V54Y]:
AWS Secret Access Key [****************4icT]:
Default region name [us-east-2]:
Default output format [None]:
```

Figure 7-1. Configure AWS in virtual machine

b. Download AWSIoTSDK Python3 in the EC2 virtual machine

AWS configuration and AWS IoT SDK Python can connect the AWS IoT Core and SEM applications IoT devices.

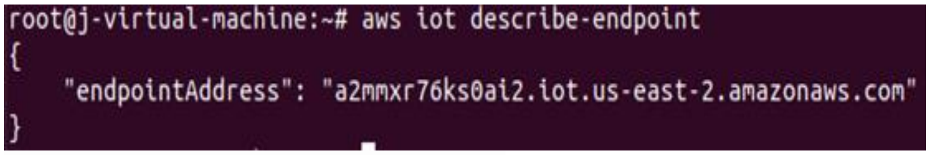


```
j@j-virtual-machine:~$ cd aws-iot-device-sdk-python/
j@j-virtual-machine:~/aws-iot-device-sdk-python$ ls -al
total 108
drwxrwxr-x  8 j    j     4096 Sep 25 11:16 .
drwxr-xr-x 17 j    j     4096 Sep 26 07:22 ..
drwxrwxr-x  5 j    j     4096 Sep 25 11:16 AWSIoTPythonSDK
drwxr-xr-x  3 root root  4096 Sep 25 11:16 build
drwxrwxr-x  2 j    j     4096 Sep 26 07:21 cert
-rw-rw-r--  1 j    j     4881 Sep 25 11:16 CHANGELOG.rst
drwxrwxr-x  8 j    j     4096 Sep 25 11:16 .git
drwxrwxr-x  4 j    j     4096 Sep 25 11:16 .github
-rwxrwxr-x  1 j    j     9245 Sep 25 11:16 LICENSE.txt
-rw-rw-r--  1 j    j      149 Sep 25 11:16 MANIFEST.in
-rwxrwxr-x  1 j    j      208 Sep 25 11:16 NOTICE.txt
-rwxrwxr-x  1 j    j    38706 Sep 25 11:16 README.rst
drwxrwxr-x  7 j    j     4096 Sep 25 11:16 samples
-rw-rw-r--  1 j    j       41 Sep 25 11:16 setup.cfg
-rw-rw-r--  1 j    j     1550 Sep 25 11:16 setup.py
j@j-virtual-machine:~/aws-iot-device-sdk-python$ 
```

Figure 7-2. AWS IoT SDK python in the virtual machine

c. Endpoint and keys for the certificate of the IoT device.

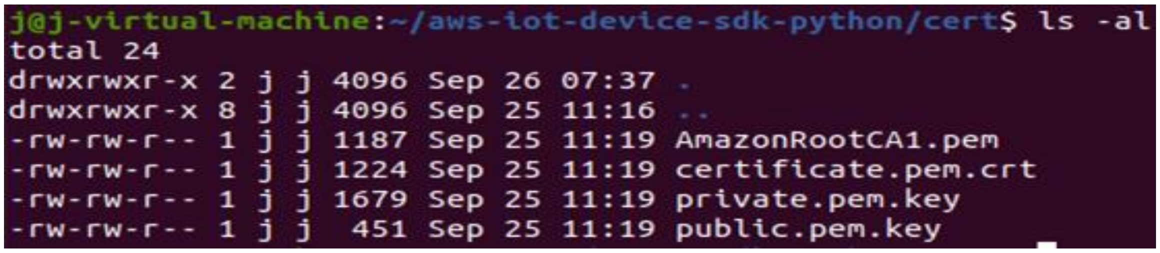
State document python file needs endpoint to connect AWS IoT MQTT to shadow Client before the MQTT test through state document python file proceeds.



```
root@j-virtual-machine:~# aws iot describe-endpoint
{
    "endpointAddress": "a2mmxr76ks0ai2.iot.us-east-2.amazonaws.com"
}
```

Figure 7-3. AWS IoT Core endpoint

Also, a state document python file needs to authorize shadow credentials by device certificate, private key, and RootCA1 RSA 2048 bit key.

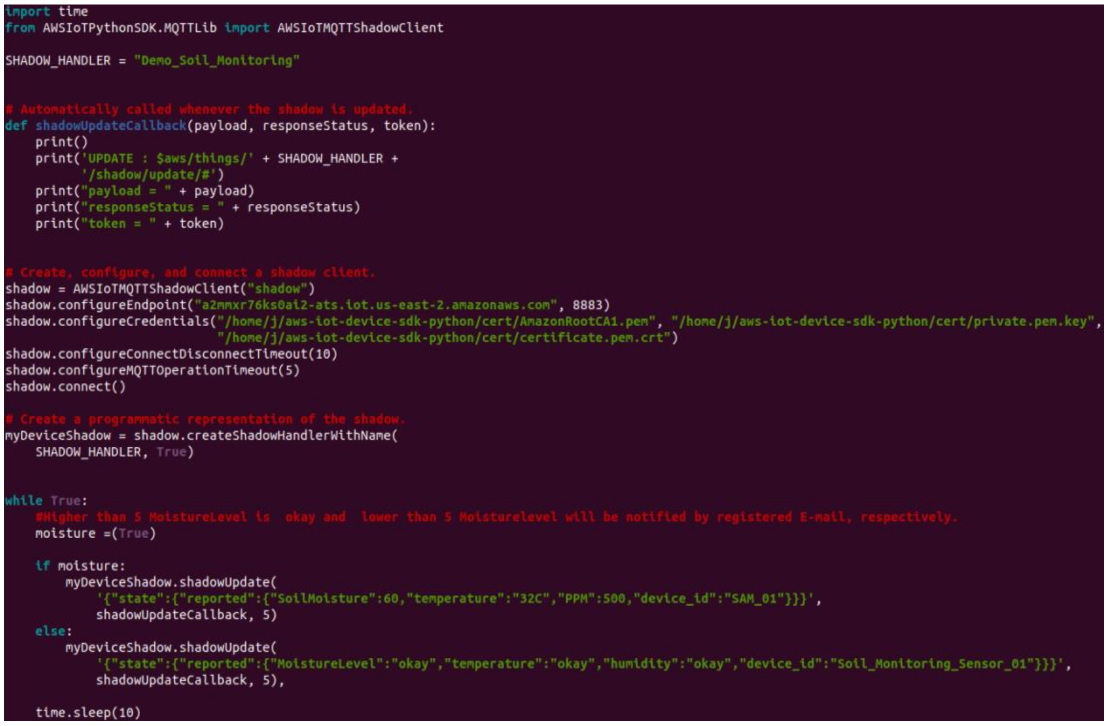


```
j@j-virtual-machine:~/aws-iot-device-sdk-python/cert$ ls -al
total 24
drwxrwxr-x 2 j j 4096 Sep 26 07:37 .
drwxrwxr-x 8 j j 4096 Sep 25 11:16 ..
-rw-rw-r-- 1 j j 1187 Sep 25 11:19 AmazonRootCA1.pem
-rw-rw-r-- 1 j j 1224 Sep 25 11:19 certificate.pem.crt
-rw-rw-r-- 1 j j 1679 Sep 25 11:19 private.pem.key
-rw-rw-r-- 1 j j  451 Sep 25 11:19 public.pem.key
```

Figure 7-4. Soil Monitoring's credential keys

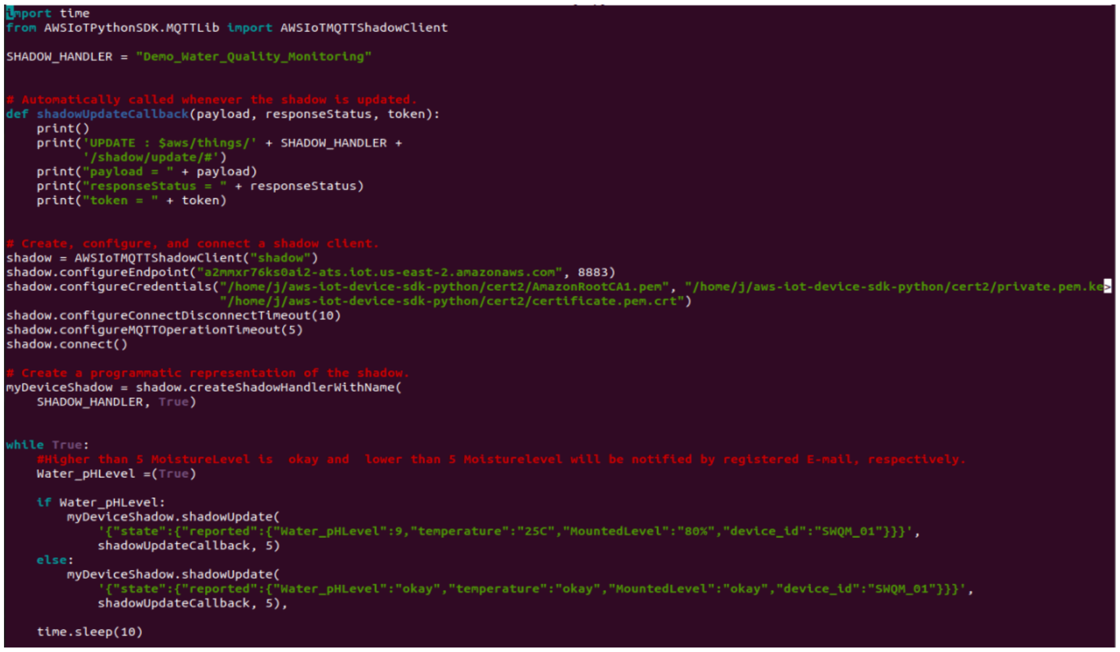
d. State document python file to operate SEM applications.

Creating a state document python file is for data transmission in the virtual machine for IoT devices. Shadow handlers must write the exact registered name on the AWS IoT Core. Write the exact endpoint address and port number 8883 for the MQTT test described in Figure 9-3. Device certificate, private key, and RootCA1 RSA 2048-bit key files that are described in Figure 9-4 need to be downloaded and recognize the file path route in the state document python file to authorize the connection between the virtual machine and AWS IoT Core.



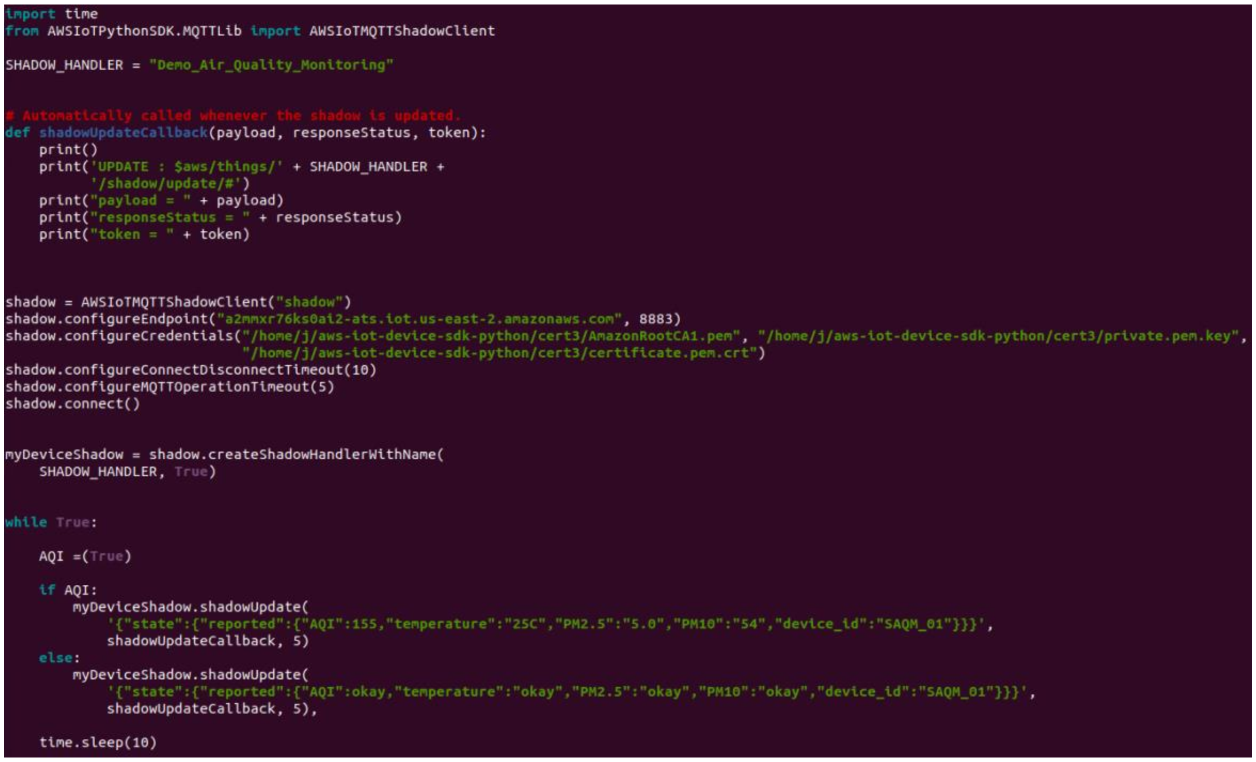
```python
import time
from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTShadowClient

SHADOW_HANDLER = "Demo_Soil_Monitoring"

# Automatically called whenever the shadow is updated.
def shadowUpdateCallback(payload, responseStatus, token):
    print()
    print('UPDATE : $aws/things/' + SHADOW_HANDLER +
          '/shadow/update/#')
    print("payload = " + payload)
    print("responseStatus = " + responseStatus)
    print("token = " + token)

# Create, configure, and connect a shadow client.
shadow = AWSIoTMQTTShadowClient("shadow")
shadow.configureEndpoint("a2mmxr76ks0ai2-ats.iot.us-east-2.amazonaws.com", 8883)
shadow.configureCredentials("/home/j/aws-iot-device-sdk-python/cert/AmazonRootCA1.pem", "/home/j/aws-iot-device-sdk-python/cert/private.pem.key",
                            "/home/j/aws-iot-device-sdk-python/cert/certificate.pem.crt")
shadow.configureConnectDisconnectTimeout(10)
shadow.configureMQTTOperationTimeout(5)
shadow.connect()

# Create a programmatic representation of the shadow.
myDeviceShadow = shadow.createShadowHandlerWithName(
    SHADOW_HANDLER, True)


while True:
    #Higher than 5 MoistureLevel is  okay and  lower than 5 Moisturelevel will be notified by registered E-mail, respectively.
    moisture =(True)

    if moisture:
        myDeviceShadow.shadowUpdate(
            '{"state":{"reported":{"SoilMoisture":60,"temperature":"32C","PPM":500,"device_id":"SAM_01"}}}',
            shadowUpdateCallback, 5)
    else:
        myDeviceShadow.shadowUpdate(
            '{"state":{"reported":{"MoistureLevel":"okay","temperature":"okay","humidity":"okay","device_id":"Soil_Monitoring_Sensor_01"}}}',
            shadowUpdateCallback, 5),

    time.sleep(10)
```

Figure 7-5. Python file for soil



```python
import time
from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTShadowClient

SHADOW_HANDLER = "Demo_Water_Quality_Monitoring"

# Automatically called whenever the shadow is updated.
def shadowUpdateCallback(payload, responseStatus, token):
    print()
    print('UPDATE : $aws/things/' + SHADOW_HANDLER +
          '/shadow/update/#')
    print("payload = " + payload)
    print("responseStatus = " + responseStatus)
    print("token = " + token)

# Create, configure, and connect a shadow client.
shadow = AWSIoTMQTTShadowClient("shadow")
shadow.configureEndpoint("a2mmxr76ks0ai2-ats.iot.us-east-2.amazonaws.com", 8883)
shadow.configureCredentials("/home/j/aws-iot-device-sdk-python/cert2/AmazonRootCA1.pem", "/home/j/aws-iot-device-sdk-python/cert2/private.pem.ke>
                            "/home/j/aws-iot-device-sdk-python/cert2/certificate.pem.crt")
shadow.configureConnectDisconnectTimeout(10)
shadow.configureMQTTOperationTimeout(5)
shadow.connect()

# Create a programmatic representation of the shadow.
myDeviceShadow = shadow.createShadowHandlerWithName(
    SHADOW_HANDLER, True)


while True:
    #Higher than 5 MoistureLevel is  okay and  lower than 5 Moisturelevel will be notified by registered E-mail, respectively.
    Water_pHLevel =(True)

    if Water_pHLevel:
        myDeviceShadow.shadowUpdate(
            '{"state":{"reported":{"Water_pHLevel":9,"temperature":"25C","MountedLevel":"80%","device_id":"SWQM_01"}}}',
            shadowUpdateCallback, 5)
    else:
        myDeviceShadow.shadowUpdate(
            '{"state":{"reported":{"Water_pHLevel":"okay","temperature":"okay","MountedLevel":"okay","device_id":"SWQM_01"}}}',
            shadowUpdateCallback, 5),

    time.sleep(10)
```

Figure 7-6. Python file for water quality



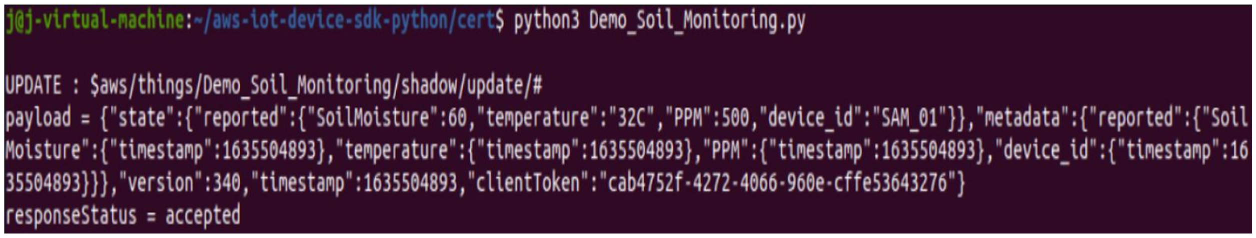
```python
import time
from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTShadowClient

SHADOW_HANDLER = "Demo_Air_Quality_Monitoring"


# Automatically called whenever the shadow is updated.
def shadowUpdateCallback(payload, responseStatus, token):
    print()
    print('UPDATE : $aws/things/' + SHADOW_HANDLER +
          '/shadow/update/#')
    print("payload = " + payload)
    print("responseStatus = " + responseStatus)
    print("token = " + token)



shadow = AWSIoTMQTTShadowClient("shadow")
shadow.configureEndpoint("a2mmxr76ks0ai2-ats.iot.us-east-2.amazonaws.com", 8883)
shadow.configureCredentials("/home/j/aws-iot-device-sdk-python/cert3/AmazonRootCA1.pem", "/home/j/aws-iot-device-sdk-python/cert3/private.pem.key",
                            "/home/j/aws-iot-device-sdk-python/cert3/certificate.pem.crt")
shadow.configureConnectDisconnectTimeout(10)
shadow.configureMQTTOperationTimeout(5)
shadow.connect()


myDeviceShadow = shadow.createShadowHandlerWithName(
    SHADOW_HANDLER, True)


while True:

    AQI =(True)

    if AQI:
        myDeviceShadow.shadowUpdate(
            '{"state":{"reported":{"AQI":155,"temperature":"25C","PM2.5":"5.0","PM10":"54","device_id":"SAQM_01"}}}',
            shadowUpdateCallback, 5)
    else:
        myDeviceShadow.shadowUpdate(
            '{"state":{"reported":{"AQI":okay,"temperature":"okay","PM2.5":"okay","PM10":"okay","device_id":"SAQM_01"}}}',
            shadowUpdateCallback, 5),

    time.sleep(10)
```

Figure 7-7. Python file for air quality

5. Operating python file and getting an email notification

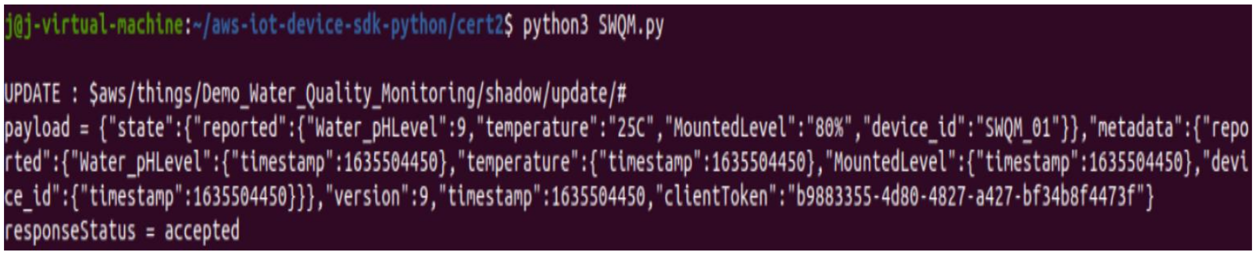
a. Operate state document python file in Linux Virtual Machine

The AWS IoT Core MQTT test, which was conducted by running a state document python file in a virtual machine, allows receiving notification of the changed data set through SNS when the “MoistureLevel” is less than 5. When the “MoistureLevel” is less than 5, the criteria for receiving notification emails are executed by the Rule query statement in Figure 8-1.



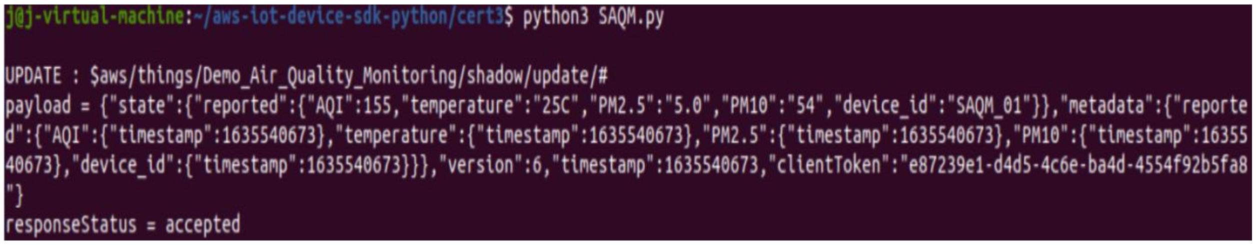
```
j@j-virtual-machine:~/aws-iot-device-sdk-python/cert$ python3 Demo_Soil_Monitoring.py

UPDATE : $aws/things/Demo_Soil_Monitoring/shadow/update/#
payload = {"state":{"reported":{"SoilMoisture":60,"temperature":"32C","PPM":500,"device_id":"SAM_01"}},"metadata":{"reported":{"Soil
Moisture":{"timestamp":1635504893},"temperature":{"timestamp":1635504893},"PPM":{"timestamp":1635504893},"device_id":{"timestamp":16
35504893}}},"version":340,"timestamp":1635504893,"clientToken":"cab4752f-4272-4066-960e-cffe53643276"}
responseStatus = accepted
```

Figure 7-8. Operating soil document



```
j@j-virtual-machine:~/aws-iot-device-sdk-python/cert2$ python3 SWQM.py

UPDATE : $aws/things/Demo_Water_Quality_Monitoring/shadow/update/#
payload = {"state":{"reported":{"Water_pHLevel":9,"temperature":"25C","MountedLevel":"80%","device_id":"SWQM_01"}},"metadata":{"repo
rted":{"Water_pHLevel":{"timestamp":1635504450},"temperature":{"timestamp":1635504450},"MountedLevel":{"timestamp":1635504450},"devi
ce_id":{"timestamp":1635504450}}},"version":9,"timestamp":1635504450,"clientToken":"b9883355-4d80-4827-a427-bf34b8f4473f"}
responseStatus = accepted
```

Figure 7-9. Operating water quality document

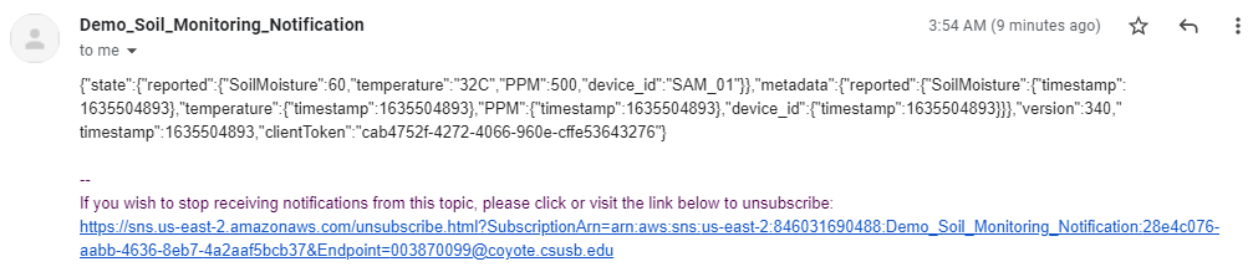


```
j@j-virtual-machine:~/aws-iot-device-sdk-python/cert3$ python3 SAQM.py

UPDATE : $aws/things/Demo_Air_Quality_Monitoring/shadow/update/#
payload = {"state":{"reported":{"AQI":155,"temperature":"25C","PM2.5":"5.0","PM10":"54","device_id":"SAQM_01"}},"metadata":{"reporte
d":{"AQI":{"timestamp":1635540673},"temperature":{"timestamp":1635540673},"PM2.5":{"timestamp":1635540673},"PM10":{"timestamp":16355
40673},"device_id":{"timestamp":1635540673}}},"version":6,"timestamp":1635540673,"clientToken":"e87239e1-d4d5-4c6e-ba4d-4554f92b5fa8
"}
responseStatus = accepted
```

Figure 7-10. Operating air quality document

b. The e-mail notification by operating state document python file

Because the “MoistureLevel” of the State document python file is 4, which is less than 5, an E-mail notification can be received to an email address registered with the SNS service of the AWS.

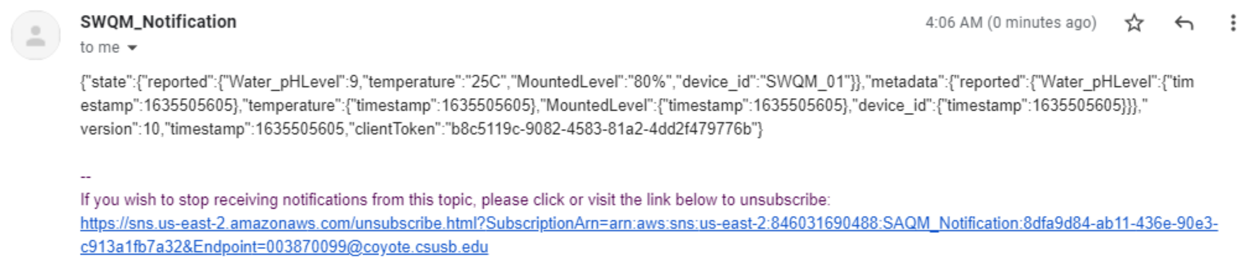


Demo_Soil_Monitoring_Notification 3:54 AM (9 minutes ago)
to me

{"state":{"reported":{"SoilMoisture":60,"temperature":"32C","PPM":500,"device_id":"SAM_01"}},"metadata":{"reported":{"SoilMoisture":{"timestamp":1635504893},"temperature":{"timestamp":1635504893},"PPM":{"timestamp":1635504893},"device_id":{"timestamp":1635504893}}},"version":340,"timestamp":1635504893,"clientToken":"cab4752f-4272-4066-960e-cffe53643276"}

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-east-2.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-2:846031690488:Demo_Soil_Monitoring_Notification:28e4c076-aabb-4636-8eb7-4a2aaf5bcb37&Endpoint=003870099@coyote.csusb.edu

Figure 7-11. Email notification for Soil_Monitoring

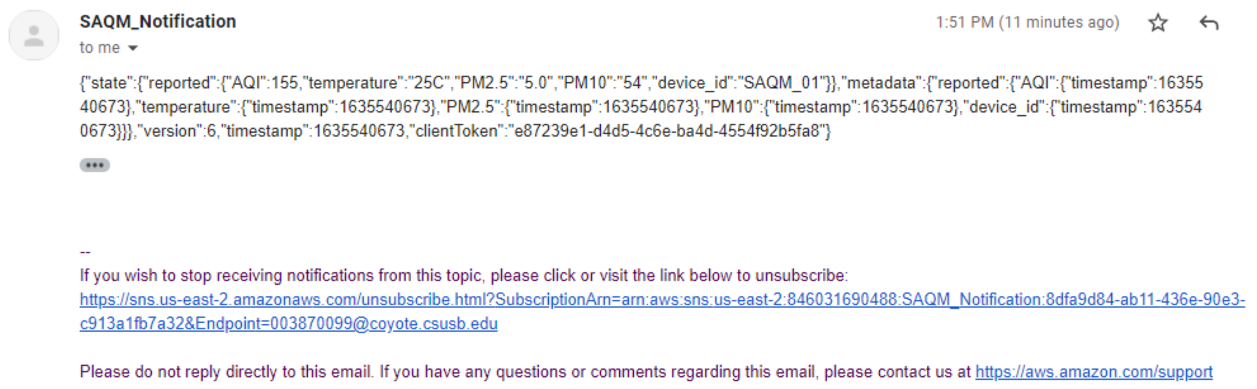


SWQM_Notification 4:06 AM (0 minutes ago)
to me

{"state":{"reported":{"Water_pHLevel":9,"temperature":"25C","MountedLevel":"80%","device_id":"SWQM_01"}},"metadata":{"reported":{"Water_pHLevel":{"timestamp":1635505605},"temperature":{"timestamp":1635505605},"MountedLevel":{"timestamp":1635505605},"device_id":{"timestamp":1635505605}}},"version":10,"timestamp":1635505605,"clientToken":"b8c5119c-9082-4583-81a2-4dd2f479776b"}

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-east-2.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-2:846031690488:SAQM_Notification:8dfa9d84-ab11-436e-90e3-c913a1fb7a32&Endpoint=003870099@coyote.csusb.edu

Figure 7-12. Email notification for Water_Quality_Monitoring



SAQM_Notification 1:51 PM (11 minutes ago)
to me

{"state":{"reported":{"AQI":155,"temperature":"25C","PM2.5":"5.0","PM10":"54","device_id":"SAQM_01"}},"metadata":{"reported":{"AQI":{"timestamp":1635540673},"temperature":{"timestamp":1635540673},"PM2.5":{"timestamp":1635540673},"PM10":{"timestamp":1635540673},"device_id":{"timestamp":1635540673}}},"version":6,"timestamp":1635540673,"clientToken":"e87239e1-d4d5-4c6e-ba4d-4554f92b5fa8"}

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-east-2.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-2:846031690488:SAQM_Notification:8dfa9d84-ab11-436e-90e3-c913a1fb7a32&Endpoint=003870099@coyote.csusb.edu

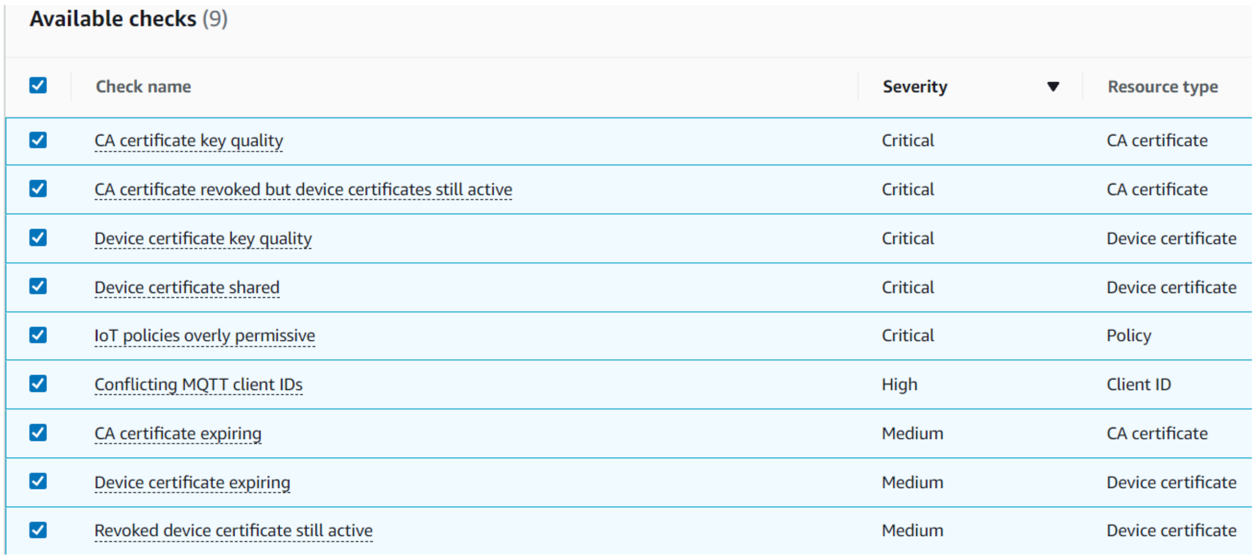
Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support

Figure 7-13. Email notification for Air_Quality_Monitoring

6. Defend IoT device in AWS IoT Core

All operations through AWS IoT Core's defender can receive email notification through SNS. The AWS IoT Core audit function allows checking the security and status of Certificate Authority (CA) certificates and device certificates for IoT access policy, and MQTT message conflicts. Figure 10-1 is a result of the audit checklist. Figure 10-2 is an SNS email notification through the result of the audit check.

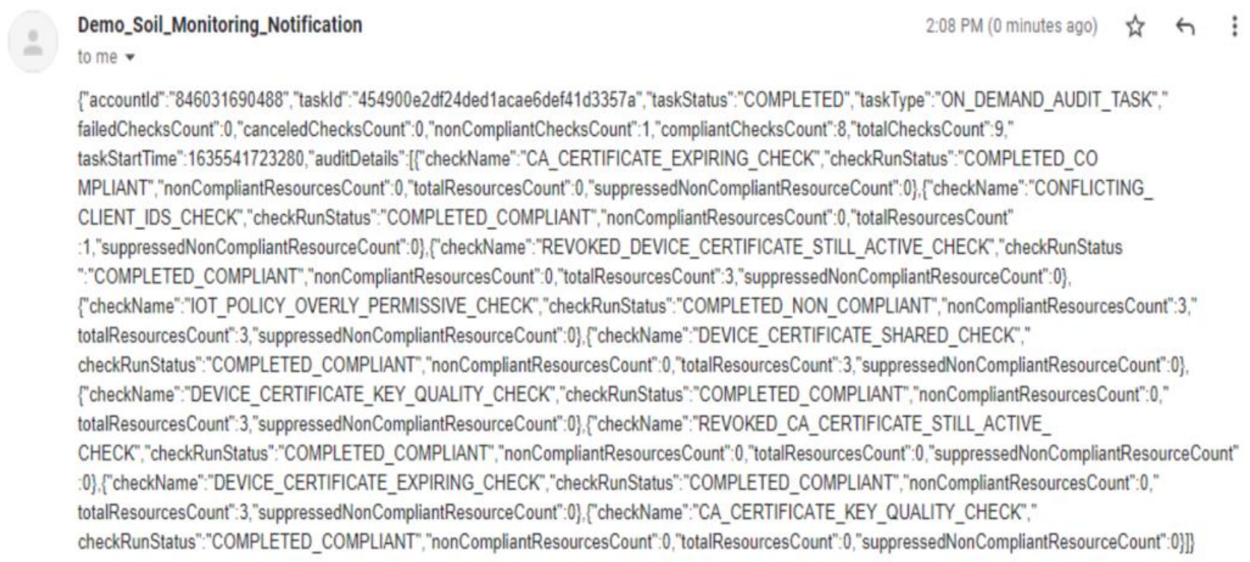
a. Audit the status device.



Available checks (9)

✓	Check name	Severity ▼	Resource type
✓	CA certificate key quality	Critical	CA certificate
✓	CA certificate revoked but device certificates still active	Critical	CA certificate
✓	Device certificate key quality	Critical	Device certificate
✓	Device certificate shared	Critical	Device certificate
✓	IoT policies overly permissive	Critical	Policy
✓	Conflicting MQTT client IDs	High	Client ID
✓	CA certificate expiring	Medium	CA certificate
✓	Device certificate expiring	Medium	Device certificate
✓	Revoked device certificate still active	Medium	Device certificate

Figure 8-1. Result of an audit checklist



Demo_Soil_Monitoring_Notification

to me

2:08 PM (0 minutes ago)

```
{"accountId":"846031690488","taskId":"454900e2df24ded1acae6def41d3357a","taskStatus":"COMPLETED","taskType":"ON_DEMAND_AUDIT_TASK","
failedChecksCount":0,"canceledChecksCount":0,"nonCompliantChecksCount":1,"compliantChecksCount":8,"totalChecksCount":9,"
taskStartTime":1635541723280,"auditDetails":[{"checkName":"CA_CERTIFICATE_EXPIRING_CHECK","checkRunStatus":"COMPLETED_CO
MPLIANT","nonCompliantResourcesCount":0,"totalResourcesCount":0,"suppressedNonCompliantResourceCount":0},{"checkName":"CONFLICTING_
CLIENT_IDS_CHECK","checkRunStatus":"COMPLETED_COMPLIANT","nonCompliantResourcesCount":0,"totalResourcesCount"
:1,"suppressedNonCompliantResourceCount":0},{"checkName":"REVOKED_DEVICE_CERTIFICATE_STILL_ACTIVE_CHECK","checkRunStatus
":"COMPLETED_COMPLIANT","nonCompliantResourcesCount":0,"totalResourcesCount":3,"suppressedNonCompliantResourceCount":0},
{"checkName":"IOT_POLICY_OVERLY_PERMISSIVE_CHECK","checkRunStatus":"COMPLETED_NON_COMPLIANT","nonCompliantResourcesCount":3,"
totalResourcesCount":3,"suppressedNonCompliantResourceCount":0},{"checkName":"DEVICE_CERTIFICATE_SHARED_CHECK","
checkRunStatus":"COMPLETED_COMPLIANT","nonCompliantResourcesCount":0,"totalResourcesCount":3,"suppressedNonCompliantResourceCount":0},
{"checkName":"DEVICE_CERTIFICATE_KEY_QUALITY_CHECK","checkRunStatus":"COMPLETED_COMPLIANT","nonCompliantResourcesCount":0,"
totalResourcesCount":3,"suppressedNonCompliantResourceCount":0},{"checkName":"REVOKED_CA_CERTIFICATE_STILL_ACTIVE_
CHECK","checkRunStatus":"COMPLETED_COMPLIANT","nonCompliantResourcesCount":0,"totalResourcesCount":0,"suppressedNonCompliantResourceCount"
:0},{"checkName":"DEVICE_CERTIFICATE_EXPIRING_CHECK","checkRunStatus":"COMPLETED_COMPLIANT","nonCompliantResourcesCount":0,"
totalResourcesCount":3,"suppressedNonCompliantResourceCount":0},{"checkName":"CA_CERTIFICATE_KEY_QUALITY_CHECK","
checkRunStatus":"COMPLETED_COMPLIANT","nonCompliantResourcesCount":0,"totalResourcesCount":0,"suppressedNonCompliantResourceCount":0}]}
```

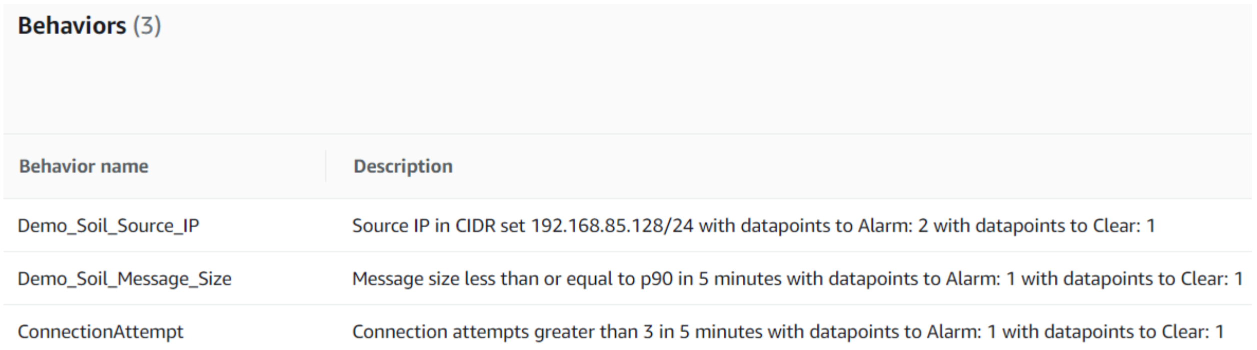
Figure 8-2. SNS email notification of audit checklist result

b. Detecting abnormal rules.

1) Source IP - IP address is outside the standard supply chain range. In other words, the device was stolen. It can lead to data leakage based on MQTT.

2) Message size - Message size is related to security through data transformation through MQTT-based data leakage. In other words, security problems in the AWS IoT cloud can occur.

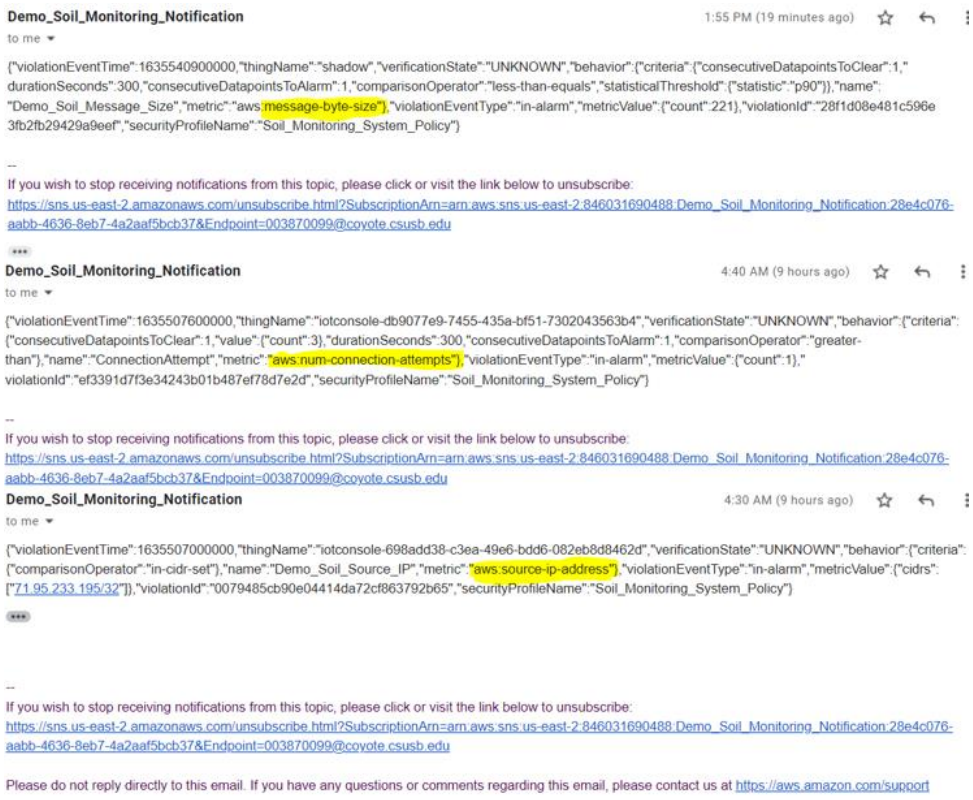
3) Connection attempt - Abnormal behavior in connection attempts points to the most likely attacker's stolen identity access scenario. That is, it is an impersonation attack. An impersonation attack is an attack that masquerades as a known or trusted entity in order to participate in the command and control of an IoT device. (*Security use cases - AWS IoT Core*, 2021)



Behaviors (3)

Behavior name	Description
Demo_Soil_Source_IP	Source IP in CIDR set 192.168.85.128/24 with datapoints to Alarm: 2 with datapoints to Clear: 1
Demo_Soil_Message_Size	Message size less than or equal to p90 in 5 minutes with datapoints to Alarm: 1 with datapoints to Clear: 1
ConnectionAttempt	Connection attempts greater than 3 in 5 minutes with datapoints to Alarm: 1 with datapoints to Clear: 1

Figure 8-3. IoT defender for detecting abnormal rules



Demo_Soil_Monitoring_Notification 1:55 PM (19 minutes ago)

to me

```
{"violationEventTime":1635540900000,"thingName":"shadow","verificationState":"UNKNOWN","behavior":{"criteria":{"consecutiveDatapointsToClear":1,"
durationSeconds":300,"consecutiveDatapointsToAlarm":1,"comparisonOperator":"less-than-equals","statisticalThreshold":{"statistic":"p90"}},"name":
"Demo_Soil_Message_Size","metric":"aws:message-byte-size"},"violationEventType":"in-alarm","metricValue":{"count":221},"violationId":"28f1d08e481c596e
3fb2fb29429a9eef","securityProfileName":"Soil_Monitoring_System_Policy"}
```

--

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-east-2.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-2:846031690488:Demo_Soil_Monitoring_Notification:28e4c076-aabb-4636-8eb7-4a2aaf5bcb37&Endpoint=003870099@coyote.csusb.edu

Demo_Soil_Monitoring_Notification 4:40 AM (9 hours ago)

to me

```
{"violationEventTime":1635507600000,"thingName":"iotconsole-db9077e9-7455-435a-bf51-7302043563b4","verificationState":"UNKNOWN","behavior":{"criteria":
{"consecutiveDatapointsToClear":1,"value":{"count":3},"durationSeconds":300,"consecutiveDatapointsToAlarm":1,"comparisonOperator":"greater-
than"},"name":"ConnectionAttempt","metric":"aws:num-connection-attempts"},"violationEventType":"in-alarm","metricValue":{"count":1},"
violationId":"ef3391d7f3e34243b01b487ef78d7e2d","securityProfileName":"Soil_Monitoring_System_Policy"}
```

--

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-east-2.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-2:846031690488:Demo_Soil_Monitoring_Notification:28e4c076-aabb-4636-8eb7-4a2aaf5bcb37&Endpoint=003870099@coyote.csusb.edu

Demo_Soil_Monitoring_Notification 4:30 AM (9 hours ago)

to me

```
{"violationEventTime":1635507000000,"thingName":"iotconsole-698add38-c3ea-49e6-bdd6-082eb8d8462d","verificationState":"UNKNOWN","behavior":{"criteria":
{"comparisonOperator":"in-cidr-set"},"name":"Demo_Soil_Source_IP","metric":"aws:source-ip-address"},"violationEventType":"in-alarm","metricValue":{"cidrs":
["71.95.233.195/32"]},"violationId":"0079485cb90e04414da72cf863792b65","securityProfileName":"Soil_Monitoring_System_Policy"}
```

--

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-east-2.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-2:846031690488:Demo_Soil_Monitoring_Notification:28e4c076-aabb-4636-8eb7-4a2aaf5bcb37&Endpoint=003870099@coyote.csusb.edu

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support

Figure 8-4. E-mail notification of three abnormal behaviors.

7. Security practices for the SEM IoT devices

a) Keep your device’s clock in sync

Securing the exact time on the device is related to the expiration date of the X.509 certificate. The watch of the device is used to validate the server certificate. Software on most systems includes Network Time Protocol (NTP) clients. Therefore, the device must wait for data synchronization, including the server's time, before connecting to the AWS IoT Core.

b) Validate Server

First, connect to allow Device to interact with AWS IoT Core. When the device is connected, AWS IoT Core sends a server certificate to the device. This Transport Layer Security (TLS) X.509 certificate has a fixed validity and provides a new certificate before it expires.

c) Single Identity per device.

Devices can generally apply for precise permissions through X.509 certificates. Since each device has a different nature, individual authentication can use subdivided access control, authorization, and policies. There is a need for a policy that allows users to narrow the scope of possible device-specific privileges to be identified and to subscribe and publish fixed topics.

d) Second AWS Region as backup

There are servers that are divided into so many regions of the AWS Cloud Platform. To eliminate security uncertainty from attackers, data can be protected by storing copies of data in a second AWS region as a backup. (Security best practices in AWS IoT Core - AWS IoT Core, n.d.)

Analysis by Using GCP IoT Core

The analysis by GCP IoT Core is based on Table 2 and 3, objective evaluation approaches in Chapter 3. In the GCP IoT Core analysis, GCP PowerShell is used for connections between IoT devices and GCP IoT Core through MQTT protocol communication.

Moreover, this analysis implements a management system that allows checking the specification and information of SEM IoT devices, which GCP IoT Core supports and manages connections. The analysis proceeds through Node.js of GCP SDK. The analysis of AWS IoT Core was conducted in Ubuntu 20.04 virtual machine through the AWS EC2's instance. However, GCP IoT Core's analysis allows convenient and efficient analysis because GCP IoT Core does not need to use virtual machines to connect the IoT device because the Secure Shell (SSH) is already connected to GCP PowerShell by port number 22 to alternate virtual machines. Also, GCP SDKs are already conveniently installed in GCP PowerShell.

Below are the steps which is the sequence of SEM application IoT device’s MQTT test configuration by using the GCP IoT Core:

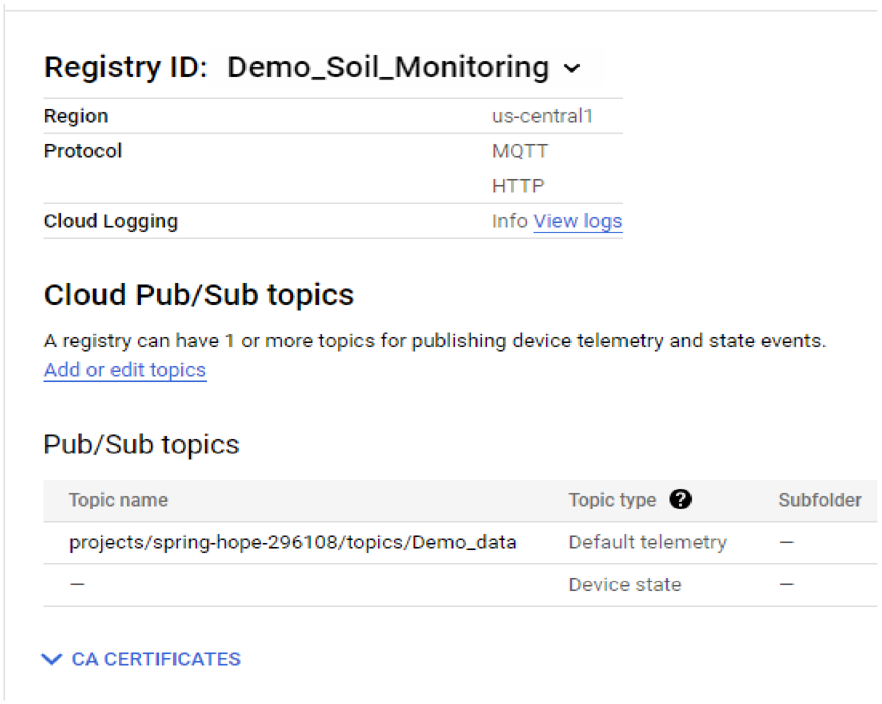
1. Create a device registry - To configure region, type of protocol, topic.

2. Create credentials - Private key and Server certificate key files.
3. Create and add the device to the registry
4. Run a Node.js to connect IoT device - subscribe, publish
5. Message for published telemetry topic, submitting for Project ID.

The step of the analysis

1. Create a device registry

The Creating a registry is a container of devices with shared properties. Each registry is created in a specific region and belongs to a project. The device registry configures protocols and Publish/Subscribe topic. Also, CA certificates can be added manually.



Registry ID: Demo_Soil_Monitoring

Region	us-central1
Protocol	MQTT HTTP
Cloud Logging	Info View logs

Cloud Pub/Sub topics

A registry can have 1 or more topics for publishing device telemetry and state events.
Add or edit topics

Pub/Sub topics

Topic name	Topic type	Subfolder
projects/spring-hope-296108/topics/Demo_data	Default telemetry	—
—	Device state	—

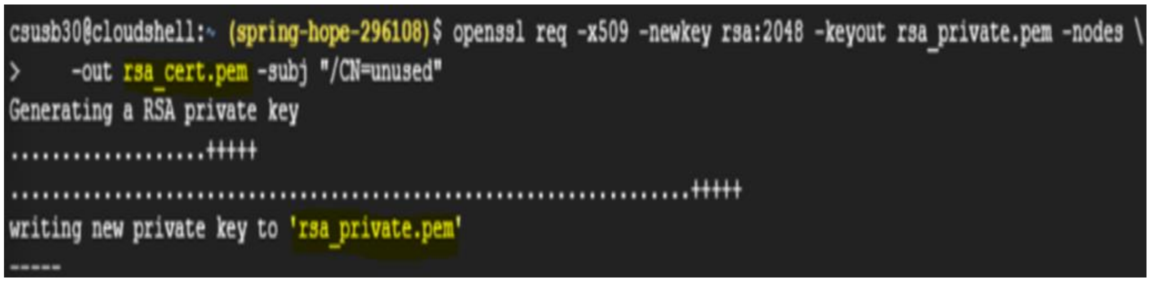
CA CERTIFICATES

Figure 9-1. Create a registry of Soil_Monitoring

2. Create credentials

a. Generating RSA keys.

RSA 256 keys can be generated by the command in GCP PowerShell. Or RSA 256 key can be downloaded from the GCP console. However, this way needs to make sure the file path when the Node.js command runs. Generating RSA 256 Keys is efficient to add the key manually. There are two keys that are ‘rsa_cert.pem’ and ‘rsa_private.pem’.

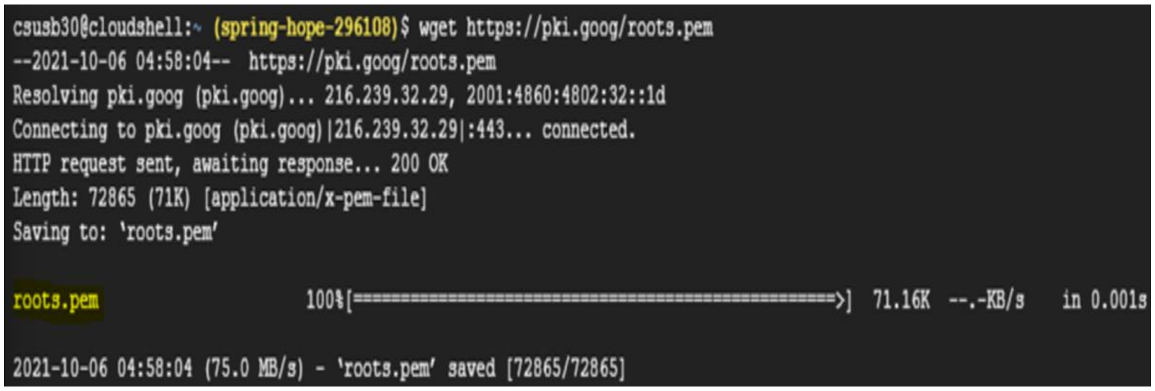


```
csusb30@cloudshell:~ (spring-hope-296108)$ openssl req -x509 -newkey rsa:2048 -keyout rsa_private.pem -nodes \
>     -out rsa_cert.pem -subj "/CN=unused"
Generating a RSA private key
..................+++++
.................................................................+++++
writing new private key to 'rsa_private.pem'
-----
```

Figure 10-1. Generate RSA 256 private and certification keys

b. Server certification key.

Server certification files also can be generated by the command in the GCP PowerShell. Server certification file is to authenticate the security of the server when connecting the IoT device to the server of the GCP IoT Core.



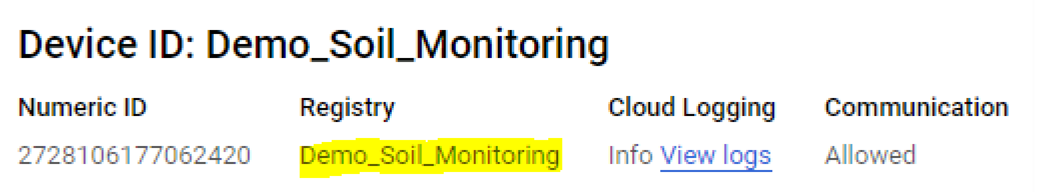
```
csusb30@cloudshell:~ (spring-hope-296108)$ wget https://pki.goog/roots.pem
--2021-10-06 04:58:04--  https://pki.goog/roots.pem
Resolving pki.goog (pki.goog)... 216.239.32.29, 2001:4860:4802:32::1d
Connecting to pki.goog (pki.goog)|216.239.32.29|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 72865 (71K) [application/x-pem-file]
Saving to: 'roots.pem'

roots.pem                100%[================================================>]  71.16K  --.-KB/s    in 0.001s

2021-10-06 04:58:04 (75.0 MB/s) - 'roots.pem' saved [72865/72865]
```

Figure 10-2. Generate server certification file

3. Create and add the device to the registry

a. Create Soil_Monitoring device

Create a virtual IoT device in the Demo_Soil_Monitoring registry that is created in Step1. As can be seen in Figure 11-1, the Demo_Soil_Monitoring registry enables telemetry with MQTT and HTTP through ‘projects/spring-hope-296108/topics/Demo_data’, a publish/subscribe topic.


Device ID: Demo_Soil_Monitoring

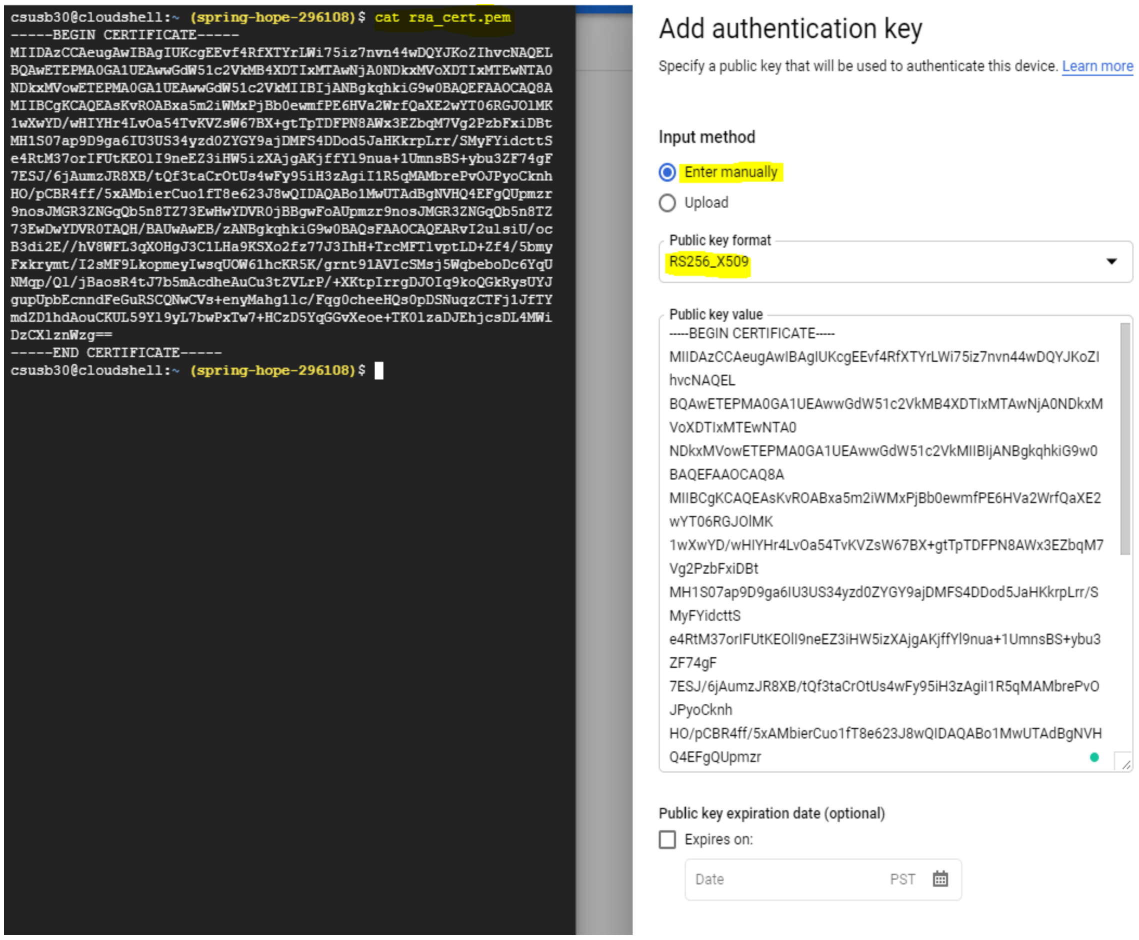
Numeric ID	Registry	Cloud Logging	Communication
2728106177062420	Demo_Soil_Monitoring	Info View logs	Allowed

Figure 11-1. Add an IoT device in the registry

b. Add a public key for IoT device certification

Add ‘rsa_cert.pem’ that is one of two RSA 256 keys in Figure 12-1. ‘rsa_cert.pem’ is the device certification file as the public key of the Demo_Soil_Monitoring device.

Open the ‘rsa_cert.pem’ file through the command in the GCP PowerShell. Copy the key value and move it manually to the GCP IoT Core console. The key format was RSA256_X509.



```
csusb30@cloudshell:~ (spring-hope-296108)$ cat rsa_cert.pem
-----BEGIN CERTIFICATE-----
MIIDAzCCAeugAwIBAgIUKcgEEvf4RfXTYrLWi75iz7nvn44wDQYJKoZIhvcNAQEL
BQAwETEPMA0GA1UEAwwGdW51c2VkMB4XDTIxMTAwNjA0NDkxMVoXDTIxMTEwNTA0
NDkxMVowETEPMA0GA1UEAwwGdW51c2VkMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A
MIIBCgKCAQEAsKvROABxa5m2iWMxPjBb0ewmfPE6HVa2WrfQaXE2wYT06RGJOlMK
1wXwYD/wHIYHr4LvOa54TvKVZsW67BX+gtTpTDFPN8AWx3EZbqM7Vg2PzbFxiDBt
MH1S07ap9D9ga6IU3US34yzd0ZYGY9ajDMFS4DDod5JaHKkrpLrr/SMyFYidcttS
e4RtM37orIFUtKEOlI9neEZ3iHW5izXAjgAKjffYl9nua+1UmnsBS+ybu3ZF74gF
7ESJ/6jAumzJR8XB/tQf3taCrOtUs4wFy95iH3zAgiI1R5qMAMbrePvOJPyoCknh
HO/pCBR4ff/5xAMbierCuo1fT8e623J8wQIDAQABo1MwUTAdBgNVHQ4EFgQUpmzr
9nosJMGR3ZNGqQb5n8TZ73EwHwYDVR0jBBgwFoAUpmzr9nosJMGR3ZNGqQb5n8TZ
73EwDwYDVR0TAQH/BAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAQEARvI2ulsiU/oc
B3di2E//hV8WFL3qXOHgJ3C1LHa9KSXo2fz77J3IhH+TrcMFTlvptLD+Zf4/5bmy
Fxkrymt/I2sMF9LkopmeyIwsqUOW61hcKR5K/grnt91AVIcSMsj5WqbeboDc6YqU
NMqp/Ql/jBaosR4tJ7b5mAcdheAuCu3tZVLrP/+XKtpIrrgDJOIq9koQGkRysUYJ
gupUpbEcnndFeGuRSCQNwCVs+enyMahg1lc/Fqg0cheeHQs0pDSNuqzCTFj1JfTY
mdZD1hdAouCKUL59Yl9yL7bwPxTw7+HCzD5YqGGvXeoe+TK0lzaDJEhjcsDL4MWi
DzCXlznWzg==
-----END CERTIFICATE-----
csusb30@cloudshell:~ (spring-hope-296108)$
```

Add authentication key

Specify a public key that will be used to authenticate this device. Learn more

Input method

[✓ Enter manually] Upload

Public key format: RS256_X509

Public key value:

```
-----BEGIN CERTIFICATE-----
MIIDAzCCAeugAwIBAgIUKcgEEvf4RfXTYrLWi75iz7nvn44wDQYJKoZI
hvcNAQEL
BQAwETEPMA0GA1UEAwwGdW51c2VkMB4XDTIxMTAwNjA0NDkxM
VoXDTIxMTEwNTA0
NDkxMVowETEPMA0GA1UEAwwGdW51c2VkMIIBIjANBgkqhkiG9w0
BAQEFAAOCAQ8A
MIIBCgKCAQEAsKvROABxa5m2iWMxPjBb0ewmfPE6HVa2WrfQaXE2
wYT06RGJOlMK
1wXwYD/wHIYHr4LvOa54TvKVZsW67BX+gtTpTDFPN8AWx3EZbqM7
Vg2PzbFxiDBt
MH1S07ap9D9ga6IU3US34yzd0ZYGY9ajDMFS4DDod5JaHKkrpLrr/S
MyFYidcttS
e4RtM37orIFUtKEOlI9neEZ3iHW5izXAjgAKjffYl9nua+1UmnsBS+ybu3
ZF74gF
7ESJ/6jAumzJR8XB/tQf3taCrOtUs4wFy95iH3zAgiI1R5qMAMbrePvO
JPyoCknh
HO/pCBR4ff/5xAMbierCuo1fT8e623J8wQIDAQABo1MwUTAdBgNVH
Q4EFgQUpmzr
```

Public key expiration date (optional)

☐ Expires on:

Date PST

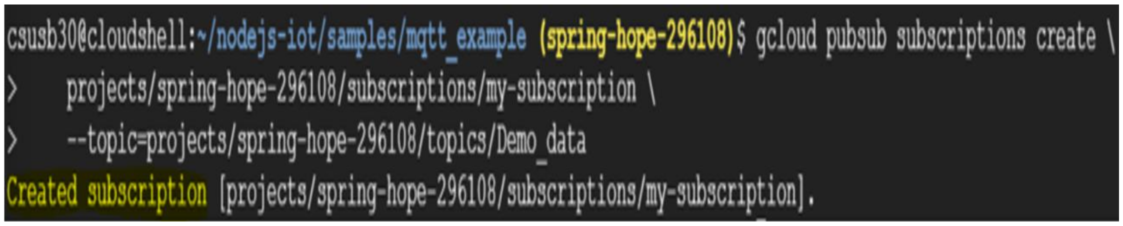
Figure 11-2. Device certification key to the console

4. Run a Node.js to connect IoT device

The SDK in the GCP PowerShell was run through Node.js to run this analysis. MQTT protocol communication is executed through project ID, region, and device topic.

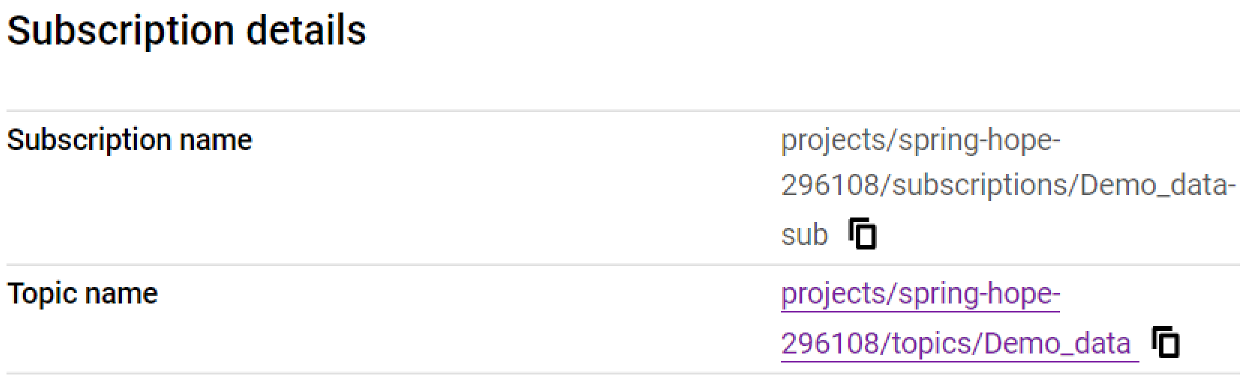
a. Create a subscription to the registry's pub/sub topic with project ID.

Figure 14-1 shows the process of creating publish/subscribe based on MQTT topic through gcloud. Next, Figure 14-2 shows the process of checking whether the created publish/subscribe is normally created in the GCP console. Figure 14-3 is a test whether MQTT communication protocol works normally based on the generated publish/subscribe. The MQTT test that works normally helps to extract the specification and information of IoT device in the later process.



```
csusb30@cloudshell:~/nodejs-iot/samples/mqtt_example (spring-hope-296108)$ gcloud pubsub subscriptions create \
>     projects/spring-hope-296108/subscriptions/my-subscription \
>     --topic=projects/spring-hope-296108/topics/Demo_data
Created subscription [projects/spring-hope-296108/subscriptions/my-subscription].
```

Figure 12-1. Creating a subscription

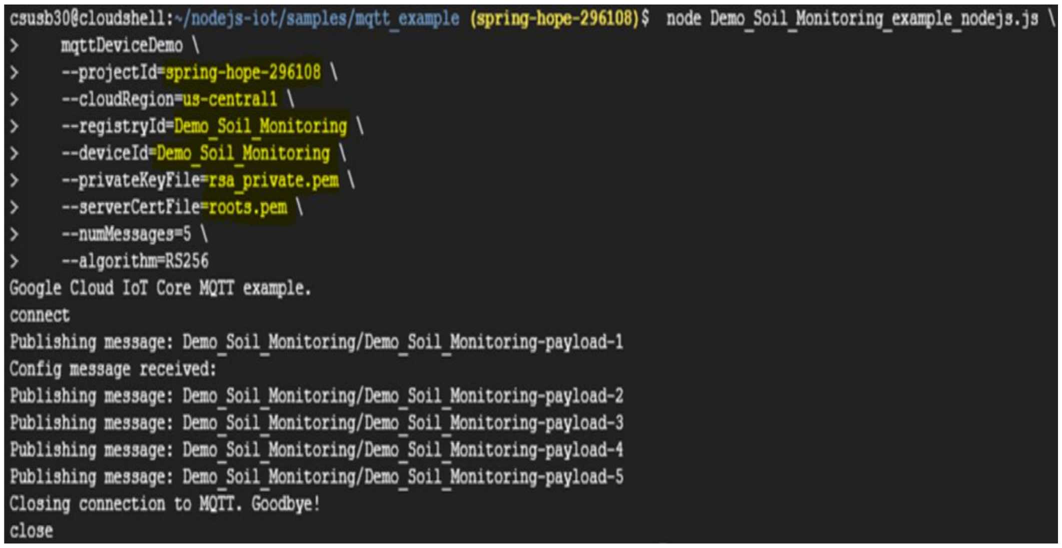


Subscription details

Subscription name	projects/spring-hope-296108/subscriptions/Demo_data-sub
Topic name	projects/spring-hope-296108/topics/Demo_data

Figure 12-2. Subscription details.

b. Run the MQTT test with a project ID, region, server certification file, and private key file.



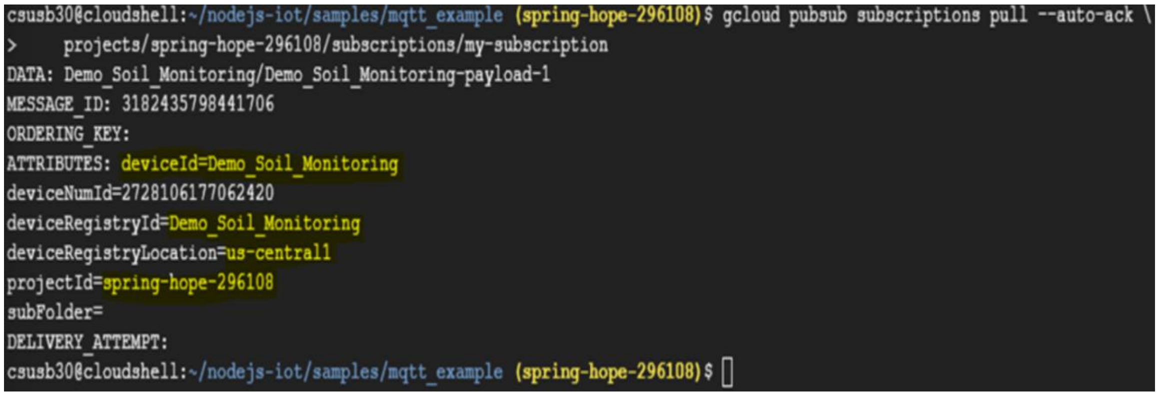
```
csusb30@cloudshell:~/nodejs-iot/samples/mqtt_example (spring-hope-296108)$  node Demo_Soil_Monitoring_example_nodejs.js \
>     mqttDeviceDemo \
>     --projectId=spring-hope-296108 \
>     --cloudRegion=us-central1 \
>     --registryId=Demo_Soil_Monitoring \
>     --deviceId=Demo_Soil_Monitoring \
>     --privateKeyFile=rsa_private.pem \
>     --serverCertFile=roots.pem \
>     --numMessages=5 \
>     --algorithm=RS256
Google Cloud IoT Core MQTT example.
connect
Publishing message: Demo_Soil_Monitoring/Demo_Soil_Monitoring-payload-1
Config message received:
Publishing message: Demo_Soil_Monitoring/Demo_Soil_Monitoring-payload-2
Publishing message: Demo_Soil_Monitoring/Demo_Soil_Monitoring-payload-3
Publishing message: Demo_Soil_Monitoring/Demo_Soil_Monitoring-payload-4
Publishing message: Demo_Soil_Monitoring/Demo_Soil_Monitoring-payload-5
Closing connection to MQTT. Goodbye!
close
```

Figure 12-3. MQTT test by node.js

5. Message for published telemetry topic, submitting for Project ID.

Through the MQTT test above, the Demo_Soil_Monitoring device can be checked the specification and information. Also, the connection between the Demo_Soil_Monitoring device and the GCP IoT Core is verified by the analysis.

Run the Pub/Sub subscription pull command to read the message that was published to the telemetry topic.



```
csusb30@cloudshell:~/nodejs-iot/samples/mqtt_example (spring-hope-296108)$ gcloud pubsub subscriptions pull --auto-ack \
>     projects/spring-hope-296108/subscriptions/my-subscription
DATA: Demo_Soil_Monitoring/Demo_Soil_Monitoring-payload-1
MESSAGE_ID: 3182435798441706
ORDERING_KEY:
ATTRIBUTES: deviceId=Demo_Soil_Monitoring
deviceNumId=2728106177062420
deviceRegistryId=Demo_Soil_Monitoring
deviceRegistryLocation=us-central1
projectId=spring-hope-296108
subFolder=
DELIVERY_ATTEMPT:
csusb30@cloudshell:~/nodejs-iot/samples/mqtt_example (spring-hope-296108)$ 
```

Figure 13-1. Pull the subscription

According to the ordering key in Figure 15-1, the SEM IoT device connected to the GCP IoT Core checks the device name, Identification, registry key, location, and project ID through MQTT communication protocol connection to obtain credibility and reliability of the connection between the two technologies. can. In other words, the analysis through GCP IoT Core shows that there is no problem with the connection between the SEM IoT device and the GCP IoT Core, and there is no doubt that it works normally.

Comparison Based on The Analysis

According to Gartner's Magic Quadrant for Cloud Infrastructure and Platform Services, AWS IoT Core, Microsoft Azure IoT Hub, and GCP IoT Core, the three highest-ranked cloud platforms. The analysis is different between the services and functions of the AWS and GCP IoT Core. As a result of fundamental analysis, AWS and GCP IoT Core had their strengths and weaknesses.

Table 7. Comparison of AWS IoT Core and GCP IoT Core

	AWS IoT Core	GCP IoT Core
Authentication and Authorization	Strength: The AWS IoT Core authentication procedures build security through stronger authentication procedures due to complex custom authorization.	Strength: The GCP IoT Core supports authentication through public/private key pairs and JSON web tokens. This method confirms that the device can work with the platform.
	Weakness: AWS provides a more complex custom authorization scheme to authenticate devices and tasks using selected strategies.	Weakness: The authentication process can be directly confirmed by the user, but it is not familiar and easy to use.
Communication Protocols	Both AWS and Google Cloud support the use of HTTP and MQTT. However, MQTT protocol communication is used in both platform’s analysis.	
Data Analytics	Python, SQL, JSON	Node.js
Device Management	Both AWS and Google Cloud IoT Core are analyzed as one-to-one connections.	
SDKs	AWS provides a larger number of fully developed SDKs in different languages.	Google provides only one end-to-end sample along with numerous other samples for individual tasks. Also, not all samples are available in all languages.

CHAPTER SIX:

CONCLUSION AND FUTURE WORK

Conclusion

The establishment of environmental domains in smart cities conducted practical use technologies and case studies on how IoT devices are applied to cloud computing. Smart environment monitoring services were reviewed through cloud platform IoT service domain and other service domains, which are the main technologies.

The first research question is “What technologies are needed for smart environmental services?” Smart environment service is one of the services provided for the construction of a smart city. The most feasible and convenient technology to build a smart environment service to citizens is environmental monitoring that can be provided using sensors and wireless Internet based IoT devices, telemetry communication protocol for transmitting data, and cloud computing service applications.

The second research question is "How is the smart environmental monitoring data collected, analyzed and monitored?" WSN-based IoT devices transmit, store, analyze and monitor data to a cloud platform through MQTT protocol, a telemetry communication protocol. Various cloud platforms are provided with service models such as SaaS, PaaS, and IaaS, and construction models such as private, public, and hybrid clouds. The cloud service is designed so that users can choose various options to receive the services they need.

The third research question is " What applications of Smart Environment are there and how were they used?" Smart Agricultural Monitoring (SAM), Smart Water Quality Monitoring (SWQM), and Smart Air Quality Monitoring (SAQM) are being used through WSN-based IoT devices and cloud service platforms in smart city environments. In addition, by reviewing the actual use cases of the above three monitoring systems, the project could know the purpose and reason of the monitoring system.

The fourth research question is " Which components enable the connection of IoT devices with cloud service platforms?" This project studied communication essential elements such as QoS, API, and SDK based on MQTT. These four factors are what enable each cloud service platform to connect with IoT devices.

This study studies the cloud platform used to connect IoT devices to the cloud through MQTT, QoS, API, and SDK elements. The cloud platforms studied are AWS IoT Core, and GCP IoT Core. Cloud platforms provide services through integrated technologies and unique technologies of each platform. AWS and GCP cloud platforms mentioned above verified the connection between IoT devices and cloud platforms through the MQTT protocol. As a result, the role of IoT services in cloud platforms is the MQTT brokers in which IoT devices communicate and become connection media.

The fifth research question is "Why and how are AWS and GCP used in the environmental monitoring system?" Tables 1 and 2 are objective evaluation approaches for AWS and GCP cloud platforms. Through this evaluation analysis, AWS and GCP can be a method for analysis by investigating. Furthermore, both cloud platforms can provide appropriate services through connection to IoT devices. In addition, the cloud platform's IoT services for environmental monitoring systems could obtain the functions and services of AWS IoT Core and GCP IoT Core through objective evaluation approaches.

The AWS IoT Core is the most widely used cloud platform and has various services that support IoT devices, so the AWS IoT Core is utilized to analyze the environmental monitoring system. Not only the AWS IoT Core but also other AWS services provide services suitable for the analysis of smart environmental monitoring systems.

In this analysis, MQTT tests confirmed how WSN-based IoT devices of environmental monitoring systems connect to AWS IoT Core to detect environmental conditions and status. The analysis is a system that sets standards for measuring the critical environmental parameters such as moisture level, water pH level, and Air Quality Index (AQI) and allows sending notifications such as e-mail or SMS when the collected data changes. The analysis made a list of authentication expiration, quality, and activity through the AWS IoT core defender's audit function. The audit function checks the security status daily or weekly. The IoT device's prevention system has been proposed to build security by transmitting notifications such as email or SMS. Furthermore, an IoT device’s abnormality detection can be detected by a prevention system. Security analysis also suggested the most basic and essential security method to protect against other threats.

The analysis of the GCP IoT Core is about communication through publish/subscribe topic by using MQTT protocol communication. The primary purpose is to identify the specifications and information of IoT devices currently connected to the GCP IoT Core through MQTT protocol communication to verify the reliability and credibility of the environmental monitoring devices.

In addition, the devices’ reliability and credibility were more emphasized because of the process of generating RSA 256 keys and transferring the key value to the device to authenticate directly. After operating the MQTT test, the specification and information of the environmental monitoring IoT devices were obtained as result values through the 'pub/sub subscriptions pull' command in the GCP PowerShell.

The analysis was not only possible to check the connection between the environmental monitoring IoT device and GCP IoT Core but also possible to check specifications and information such as device name, Identification, registry key, location, and project ID of the connected IoT device.

Future Work and Recommendations for the Project

The project topic is " Cloud and IoT for the Smart Environment of Smart City.” In this project, the services provided by the cloud platform are connected to IoT devices in a smart environment to study whether data can be stored, managed, analyzed, and protected. As mentioned in Chapter 3, the best cloud platform cannot be determined. However, users can make the best choice if they make a choice that suits their purpose of use. In addition, AWS and GCP cloud platforms were selected to analyze and monitor SEM applications through objective evaluation approaches.

According to the analysis results in Chapter 5, the connection between WSN-based IoT devices and AWS cloud platforms through the MQTT communication protocol has successfully led to SEM applications' collected data transmission, analysis, and monitoring. However, the connection and data analysis and management of IoT devices and AWS cloud platforms can be provided with more services. The connection between IoT device and AWS cloud platform is not only possible through MQTT over WSS and HTTP protocol, but each protocol can provide different unique services. AWS IoT Core message brokers can also provide a large number of public/subscription messages with multiple IoT devices. In other words, the connection between the two technologies can be one to many and many to many connections’ mechanisms. With many public/subscribe message brokers, many IoT devices can manage a large number of messages. Therefore, the connections of two technologies are suitable to build a real smart city’s smart environment.

The second analysis was performed by GCP IoT Core and used the MQTT communication protocol. IoT devices in a smart environment could check specifications and information of IoT devices through MQTT connection tests. Being able to extract accurate specification and information of IoT devices from connections through the MQTT communication protocol means that the exact IoT device is connected to the cloud platform. In other words, the reliability and credibility of IoT devices have been secured.

In the above two analyses, connecting an IoT device to a virtual machine on a cloud platform or a cloud-based command line (CLI) to transmit, store and process data is an essential process. The AWS IoT Core's virtual machine connection went through a complicated process such as "AWS-configure" to set up servers and regions in virtual machines through AWS EC2's instance service. On the other hand, the GCP IoT Core was able to operate through its own GCP PowerShell, which has already been authorized for IoT devices, servers, and regions. GCP also has a virtual machine service called Google compute engine, but if the user doesn't need additional application installations for the usage and want fast work processing, cost savings, easy usage, and time savings through no installation. GCP PowerShell will be the most suitable service.

AWS IoT Core's virtual machine connection went through a complicated process such as "AWS-configure" to set up servers and regions in virtual machines through AWS EC2's instance service. On the other hand, the GCP IoT Core was able to operate through its own GCP PowerShell, which has already been certified for devices, servers, and regions. Through this, GCP provides a simple and convenient working environment and cost savings for users. The analyzed project was a one-on-one mechanism MQTT test that connects IoT devices in one smart environment to AWS and GCP IoT Core. However, many IoT devices connection management uses one to many and many to many connection mechanisms and can efficiently manage IoT devices by provisioning many IoT devices on the cloud platform at once.

Observation

This is what I observed while connecting, managing, security, and analyzing a smart city's environmental plant management in the AWS and GCP IoT Core. There are three types of offers for the AWS free tier service. Free trial, 12 months free, and always free. All the various services I used to do the MQTT test in the analysis were conducted through the free tier service. However, before starting this analysis, I was not very familiar with cloud computing, services, and platforms, so I had to devote myself to a long period of time. AWS IoT Core was used the most. AWS IoT Core was available free of charge because 250,000 messages were free tier services. In addition, the AWS SNS for receiving email notifications was always a free service, so it was easy to use. However, Amazon EC2's virtualization machine usage was free of charge for 750 hours. My EC2's free service timed out, so I had to charge the service.

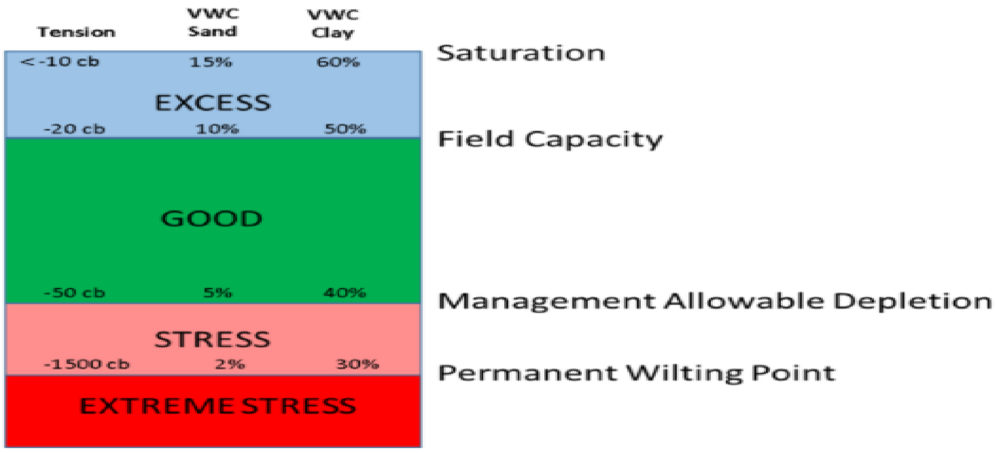
The GCP free trial service was not charged until the free usage limit. There was no expiration date for the free usage limit, but there may be changes. The Pub/Sub service for the MQTT test and IoT Core, which I mainly used, could use the free trial because the amount of free service data was very generous.

Analysis through AWS IoT Core was challenging to use because it had to use and understand other AWS services. On the other hand, analysis through GCP IoT Core was relatively simple and easy to understand

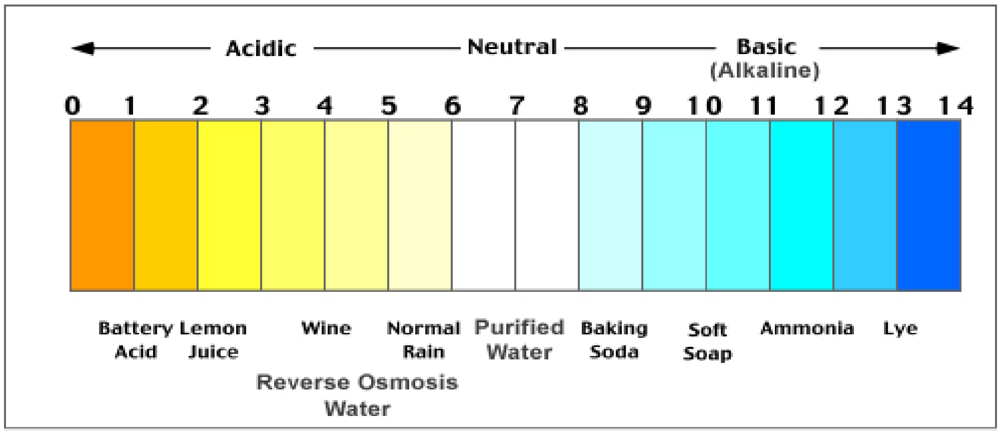
APPENDIX A:

ENVIRONMENTAL PARAMETER STANDARDS

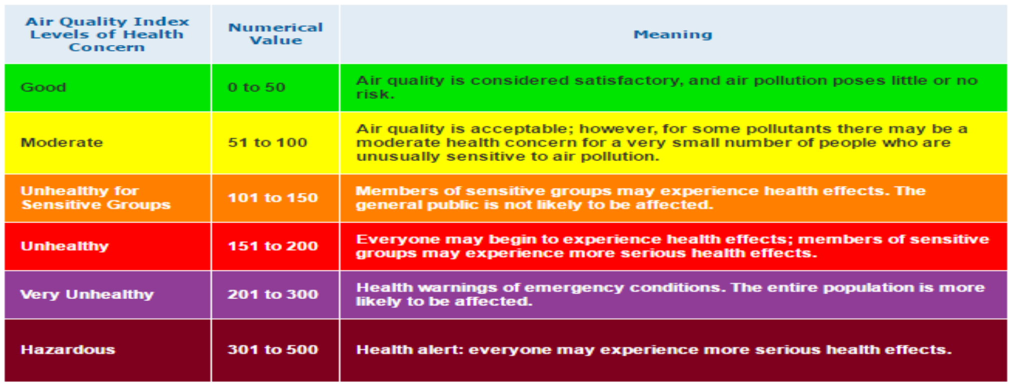
1. Soil Moisture Standards (Pitts, 2016).



2. Water pH Level Standards (FilterWater, 2017).



3. Air Quality Index Standards (AQI) (Hylton, 2016).



Air Quality Index Levels of Health Concern	Numerical Value	Meaning
Good	0 to 50	Air quality is considered satisfactory, and air pollution poses little or no risk.
Moderate	51 to 100	Air quality is acceptable; however, for some pollutants there may be a moderate health concern for a very small number of people who are unusually sensitive to air pollution.
Unhealthy for Sensitive Groups	101 to 150	Members of sensitive groups may experience health effects. The general public is not likely to be affected.
Unhealthy	151 to 200	Everyone may begin to experience health effects; members of sensitive groups may experience more serious health effects.
Very Unhealthy	201 to 300	Health warnings of emergency conditions. The entire population is more likely to be affected.
Hazardous	301 to 500	Health alert: everyone may experience more serious health effects.

REFERENCES

Abomhara, M., & Koien, G. M. (2014*).* Security and privacy in the internet of things: Current status and open issues. *2014 International Conference on Privacy and Security in Mobile Systems (PRISMS).* https://doi.org/10.1109/prisms.2014.6970594

Abu-Lughod, J. L., & Hay, R. (2007). *Third world urbanization. Routledge.*

Ahlgren, B., Hidell, M., & Ngai, E. C.-H. (2016). *Internet of things for smart cities: Interoperability and open data. IEEE Internet Computing, 20(6),* 52–56. https://doi.org/10.1109/mic.2016.124

Alhakbani, N., Hassan, M. M., Hossain, M. A., & Alnuem, M. (2014). A Framework of Adaptive Interaction Support in Cloud-Based Internet of Things (IoT) Environment. *Internet and Distributed Computing Systems,* 136–146. https://doi.org/10.1007/978-3-319-11692-1_12

Arroyo, P., Herrero, J., Suárez, J., & Lozano, J. (2019). Wireless Sensor Network Combined with Cloud Computing for Air Quality Monitoring. *Sensors, 19(3),* 691. https://doi.org/10.3390/s19030691

Athani, S., Tejeshwar, C. H., Patil, M. M., Patil, P., & Kulkarni, R. (2017, February 1). *Soil moisture monitoring using IoT enabled arduino sensors with neural* *networks for improving soil management for farmers and predict seasonal rainfall for planning future harvest in North Karnataka — India.* IEEE Xplore. https://doi.org/10.1109/I-SMAC.2017.8058385

Autor: T M Vinod Kumar. (2020). *Smart environment for smart cities.* Springer.

AWS IoT Core Features - Amazon Web Services. (n.d.). Amazon Web Services, Inc. Retrieved October 27, 2021, from https://aws.amazon.com/iot-core/features/?nc1=h_ls

AWS IoT Core Overview - Amazon Web Services. (n.d.). Amazon Web Services, Inc. https://aws.amazon.com/iot-core/?nc1=h_ls

AWS IoT Core Pricing - Amazon Web Services. (n.d.). Amazon Web Services, Inc. https://aws.amazon.com/iot-core/pricing/

AWS IoT Device SDKs, Mobile SDKs, and AWS IoT Device Client - AWS IoT Core. (n.d.). Docs.aws.amazon.com. https://docs.aws.amazon.com/iot/latest/developerguide/iot-sdks.html

AWS IoT Device Shadow Service - IoT Lens. (n.d.). Docs.aws.amazon.com. Retrieved September 28, 2021, from https://docs.aws.amazon.com/wellarchitected/latest/iot-lens/aws-iot-device-shadow-service.html

Bala, R., Gill, B., Smith, D., Ji, K., & Wright, D. (2021, July 27). *Magic Quadrant for Cloud Infrastructure and Platform Services.* Gartner.com. https://www.gartner.com/doc/reprints?id=1-271OE4VR&ct=210802&st=sb

Berkel, A. R., Singh, P. M., & van Sinderen, M. J. (2018). An information security architecture for smart cities. *Lecture Notes in Business Information Processing,* 167–184. https://doi.org/10.1007/978-3-319-94214-8_11

Bhabad, M., & Bagade, S. (2015). Internet of Things: Architecture, Security Issues and Countermeasures. *International Journal of Computer Applications*, *125*(14), 1–4. https://doi.org/10.5120/ijca2015906251

Bhoomika, K. N., Deepa, C., Rashmi, R. K., & Srinivasa, R. (2016). Internet of things for environmental monitoring. *Int. J. Adv. Netw. Appl,* 497-501.

Bigelow, S. (2019). What is Google Cloud Platform (GCP)? - Definition from WhatIs.com. *SearchCloudComputing.* https://searchcloudcomputing.techtarget.com/definition/Google-Cloud-Platform

Boretti, A., & Rosa, L. (2019). Reassessing the projections of the World Water Development Report. *Npj Clean Water, 2(1).* https://doi.org/10.1038/s41545-019-0039-9

Botta, A., de Donato, W., Persico, V., & Pescapé, A. (2016). Integration of Cloud computing and Internet of Things: *A survey. Future Generation Computer Systems, 56,* 684–700. https://doi.org/10.1016/j.future.2015.09.021

Bravo, J. (2014). *Department of Economic and Social Affairs.* https://esa.un.org/unpd/wup/Publications/Files/WUP2014-Highlights.pdf

Cloud Client Libraries. (2021). Google Cloud. https://cloud.google.com/apis/docs/cloud-client-libraries

Cloud IoT Core. (n.d.). Google Cloud. https://cloud.google.com/iot-core

Corbellini, S., Di Francia, E., Grassini, S., Iannucci, L., Lombardo, L., & Parvis, M. (2018). Cloud based sensor network for environmental monitoring. *Measurement, 118,* 354–361. https://doi.org/10.1016/j.measurement.2017.09.049

Corcoran, P. (2016). The internet of things: *Why now, and what's next? IEEE Consumer Electronics Magazine, 5(1),* 63–68. https://doi.org/10.1109/mce.2015.2484659

Definition of IoT Platforms - Gartner Information Technology Glossary. (n.d.). Gartner. https://www.gartner.com/en/information-technology/glossary/iot-platforms.

Deshmukh, A. D., & Shinde, U. B. (2016, August 1). *A low cost environment monitoring system using raspberry Pi and arduino with Zigbee.* IEEE Xplore. https://doi.org/10.1109/INVENTIVE.2016.7830096

Elmustafa, S. A. A., & Mujtaba, E. Y. (2019). Internet of things in smart environment: Concept, applications, challenges, and future directions. *World Scientific News, 134(1),* 1-51.

FilterWater. (2017, March 31). *pH Level of Water.* FilterWater.com. https://www.filterwater.com/t-ph-level-of-water.aspx

Folea, S. C., & Mois, G. (2015). A Low-Power Wireless Sensor for Online Ambient Monitoring. *IEEE Sensors Journal, 15(2),* 742–749. https://doi.org/10.1109/jsen.2014.2351420

Food and Agriculture Organization, UN. (2018). *Background* | *Global Symposium on Soil Pollution* | *Food and Agriculture Organization of the United Nations.* Www.fao.org. https://www.fao.org/about/meetings/global-symposium-on-soil-pollution/background/en/

Gayatri, M. K., Jayasakthi, J., & Anandha Mala, G. S. (2015, July 1). *Providing Smart Agricultural solutions to farmers for better yielding using IoT.* IEEE Xplore. https://doi.org/10.1109/TIAR.2015.7358528

Getting started | *Cloud IoT Core Documentation.* (n.d.). *Google Cloud. Retrieved September 28, 2021*, from https://cloud.google.com/iot/docs/how-tos/getting-started

Ghazal, T. M., Hasan, M. K., Alshurideh, M. T., Alzoubi, H. M., Ahmad, M., Akbar, S. S., Al Kurdi, B., & Akour, I. A. (2021). IoT for Smart Cities: Machine Learning

Approaches in Smart Healthcare—A Review. *Future Internet, 13(8),* 218. https://doi.org/10.3390/fi13080218

Gomes, M. M., Righi, R. da R., & da Costa, C. A. (2014). Future directions for providing better IoT infrastructure. *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication.* https://doi.org/10.1145/2638728.2638752

Gondchawar, N., & Kawitkar, R. (2016). IJARCCE IoT based Smart Agriculture. *International Journal of Advanced Research in Computer and Communication Engineering, 5*. https://doi.org/10.17148/IJARCCE.2016.56188

Google Cloud overview | *Overview.* (n.d.). Google Cloud. https://cloud.google.com/docs/overview

Google Kubernetes Engine (GKE). (n.d.). Google Cloud. https://cloud.google.com/kubernetes-engine

Google. (n.d.). *Compare AWS and Azure services to Google Cloud. Google Cloud.* https://cloud.google.com/free/docs/aws-azure-gcp-service-comparison

Hunkeler, U., Truong, H. L., & Stanford-Clark, A. (2008, January 1). MQTT-S — *A publish/subscribe protocol for Wireless Sensor Networks*. IEEE Xplore. https://doi.org/10.1109/COMSWA.2008.4554519

Hylton, M. (2016, July 5). AQI Calculations Overview- Ozone, PM2.5 and PM10. *AirNow Discussion Forum.* https://forum.airnowtech.org/t/aqi-calculations-overview-ozone-pm2-5-and-pm10/168

IBM Cloud Education. (2020, August 19). *What is an Application Programming Interface (API)?* Www.ibm.com. https://www.ibm.com/cloud/learn/api

IQ Air. (2020). *South Korea Air Quality Index (AQI) and Air Pollution information* | *AirVisual*. *Www.iqair.com.* https://www.iqair.com/south-korea

Jo, J. H., Sharma, P. K., Sicato, J. C. S., & Park, J. H. (2019). Emerging Technologies for Sustainable Smart City Network Security: Issues, Challenges, and Countermeasures*. Journal of Information Processing Systems, 15(4),* 765–784. http://jips-k.org/q.jips?cp=pp&pn=686

Johnson, K. (2018, July 19). *Environmental benefits of smart city solutions.* Foresight. https://www.climateforesight.eu/cities-coasts/environmental-benefits-of-smart-city-solutions/

Jorge.Bravo. (2014). *Department of Economic and Social Affairs.* https://esa.un.org/unpd/wup/Publications/Files/WUP2014-Highlights.pdf

Lueth, K. L. (2019, October 21). *Why it is called Internet of Things: Definition, history, disambiguation.* Iot-Analytics.com*.* https://iot-analytics.com/internet-of-things-definition/

Mahmoud, R., Yousuf, T., Aloul, F., & Zualkernan, I. (2015). Internet of things (IOT) security: Current status, challenges and prospective measures. *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST).* https://doi.org/10.1109/icitst.2015.7412116

Mamun, K. A., Islam, F. R., Haque, R., Khan, M. G. M., Prasad, A. N., Haqva, H., Mudliar, R. R., & Mani, F. S. (2019). Smart Water Quality Monitoring System Design and KPIs Analysis: *Case Sites of Fiji Surface Water. Sustainability, 11(24),* 7110. https://doi.org/10.3390/su11247110

Mandula, L. (2018, June 4). *Difference between cloud computing and internet of things.* Compare the Difference Between Similar Terms. Retrieved September 28, 2021, from https://www.differencebetween.com/difference-between-cloud-computing-and-internet-of-things/.

Manyika, J., Chui, M., Bisson, P., Woetzel, J., Dobbs, R., Bughin, J., & Aharon, D. (2020, February 13). *Unlocking the potential of the internet of things. McKinsey & Company.* Retrieved September 28, 2021, from https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/the-internet-of-things-the-value-of-digitizing-the-physical-world.

Markets & Markets. (2020, April). *IoT Cloud Platform Market Size, Share and Global Market Forecast to 2025* | *MarketsandMarkets*. Www.marketsandmarkets.com.

https://www.marketsandmarkets.com/Market-Reports/iot-cloud-platform-market-195182.html

Mazhelis, O., & Tyrvainen, P. (2014). A framework for evaluating Internet-of-Things platforms: Application provider viewpoint. *2014 IEEE World Forum on Internet of Things (WF-IoT).* https://doi.org/10.1109/wf-iot.2014.6803137

McClelland, C. (2016). The Industrial Internet of Things–*What's the Difference Between IoT and IIoT?. Retrieved January, 23, 2017.*

Mell, P., & Grance, T. (2011). *The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology Special Publication 800-145.* http://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf

Mohammed Sadeeq, M., Abdulkareem, N. M., Zeebaree, S. R., Mikaeel Ahmed, D., Saifullah Sami, A., & Zebari, R. R. (2021). IOT and cloud computing issues, challenges and opportunities: *A Review. Qubahan Academic Journal, 1(2),* 1–7. https://doi.org/10.48161/qaj.v1n2a36

MQTT - AWS IoT. (n.d.). Docs.aws.amazon.com. https://docs.aws.amazon.com/iot/latest/developerguide/mqtt.html

Mukesh, R., Sahay, K., Sukumaran, S., Amarnath, T., Nambi, D., Palani, Sahay, Sukumaran, M., Nambi, T., Palani, D., & Amarnath, S. (2019). *EasyChair Preprint Environmental Monitoring System Using IoT and Cloud Service at*

Real-Time Environmental Monitoring System Using IoT and Cloud Service at Real-Time. https://easychair.org/publications/preprint_open/5Lg1

Murad, Md. W., & Pereira, J. J. (2011, January 1). Malaysia: *Environmental Health Issues* (J. O. Nriagu, Ed.). ScienceDirect; Elsevier. https://www.sciencedirect.com/science/article/pii/B9780444522726005390

Nations, U. (2021). *International Day of Clean Air for blue skies*. United Nations. https://www.un.org/en/observances/clean-air-day

Nguyen, C. (2018, March 20). *A role for NIST in Smart City Technologies.* NIST. *Retrieved September 28, 2021,* from https://www.nist.gov/el/cyber-physical-systems/role-nist-smart-city-technologies.

Oracle. (2020). *What is the Internet of Things (IoT)?* Oracle.com. https://www.oracle.com/internet-of-things/what-is-iot

Palmer, S. (2020, September 25). *10 Best IoT Cloud Platforms I DevTeam.Space.* DevTeam.Space. https://www.devteam.space/blog/10-best-internet-of-things-iot-cloud-platforms/

Pasika, S., & Gandla, S. T. (2020). Smart water quality monitoring system with cost-effective using IoT. *Heliyon, 6(7), e04096.* https://doi.org/10.1016/j.heliyon.2020.e04096

Pflanzner, T., & Kertesz, A. (2016). A survey of IoT cloud providers. *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO).* https://doi.org/10.1109/mipro.2016.7522237

Pinto, A. R., Montez, C., Araújo, G., Vasques, F., & Portugal, P. (2014). An approach to implement data fusion techniques in wireless sensor networks using genetic machine learning algorithms. *Information Fusion, 15,* 90–101*.* https://doi.org/10.1016/j.inffus.2013.05.003

Pitts, L. (2016, March 13). *Monitoring Soil Moisture for Optimal Crop Growth.* Help Desk. https://observant.zendesk.com/hc/en-us/articles/208067926-Monitoring-Soil-Moisture-for-Optimal-Crop-Growth

Prasad, A. N., Mamun, K. A., Islam, F. R., & Haqva, H. (2015, December 1). *Smart water quality monitoring system. IEEE Xplore.* https://doi.org/10.1109/APWCCSE.2015.7476234

Pricing | *Cloud IoT Core.* (n.d.). Google Cloud. Retrieved October 27, 2021, from https://cloud.google.com/iot/pricing

Publishing over the MQTT bridge | *Cloud IoT Core Documentation.* (n.d.). Google Cloud. https://cloud.google.com/iot/docs/how-tos/mqtt-bridge

Rahmati, A., Shepard, C., Tossell, C., Zhong, L., & Kortum, P. (2015). Practical Context Awareness: *Measuring and Utilizing the Context Dependency of*

Mobile Usage. IEEE Transactions on Mobile Computing, 14(9), 1932–1946. https://doi.org/10.1109/tmc.2014.2365199

SDKs and Programming Toolkits for AWS. (n.d.). Amazon Web Services, Inc. https://aws.amazon.com/tools/

Security best practices in AWS IoT Core - AWS IoT Core. (n.d.). Docs.aws.amazon.com. Retrieved October 1, 2021, from https://docs.aws.amazon.com/iot/latest/developerguide/security-best-practices.htm

Security use cases - AWS IoT Core. (2021). Amazon.com. https://docs.aws.amazon.com/iot/latest/developerguide/dd-detect-security-use-cases.html

Shah, J., & Mishra, B. (2016, January 1). *IoT enabled environmental monitoring system for smart cities.* IEEE Xplore. https://doi.org/10.1109/IOTA.2016.7562757

Sinha, S. (2021, September 22). State of IoT 2021: *Number of connected IoT devices growing 9% to 12.3 billion globally, cellular IoT now surpassing 2 billion.* IoT Analytics. https://iot-analytics.com/number-connected-iot-devices/

Suma, N., Samson, S. R., Saranya, S., Shanmugapriya, G., & Subhashri, R. (2017). IOT based smart agriculture monitoring system*. International Journal on*

Recent and Innovation Trends in computing and communication, 5(2), 177-181.

Swinhoe, D. (2021, March 9). *City of Pittsburgh announces plans to migrate to Google Cloud.* Www.datacenterdynamics.com. https://www.datacenterdynamics.com/en/news/city-pittsburgh-announces-plans-migrate-google-cloud/

Telenor Group. (2017, April 20). *DTAC debuts the first IoT based agricultural solution.* Telenor Group. https://www.telenor.com/dtac-debuts-the-first-iot-based-agricultural-solution/

The graphical user interface of Porting Assistant for .NET is now open source. (n.d.). Amazon Web Services, Inc. Retrieved October 27, 2021, from https://aws.amazon.com/about-aws/whats-new/2021/01/graphical-user-interface-of-porting-assistant-for-dotnet-now-open-source/

The World Bank. (2017). *Climate-Smart Agriculture.* World Bank. https://www.worldbank.org/en/topic/climate-smart-agriculture

Tomás, J. P. (2017, September 22). *Air quality monitoring coming to Korea with $9 million investment.* Enterprise IoT Insights*.* https://enterpriseiotinsights.com/20170922/internet-of-things/air-quality-monitoring-korea-investment-tag23

Types of cloud computing. (2019). Amazon Web Services, Inc. https://aws.amazon.com/types-of-cloud-computing/

United Nations Economic Commission for Europe. (n.d.). Environmental Monitoring | UNECE. Unece.org. https://unece.org/environmental-monitoring

Using gateways | *Cloud IoT Core Documentation.* (n.d.). Google Cloud. https://cloud.google.com/iot/docs/how-tos/gateways

Verma, S. (2021, January 4). *Integrating IoT technology for effective environmental monitoring* - IoT Agenda. Internetofthingsagenda.techtarget.com. https://internetofthingsagenda.techtarget.com/blog/IoT-Agenda/Integrating-IoT-technology-for-effective-environmental-monitoring

What is MQTT (MQ Telemetry Transport)? - Definition from WhatIs.com. (2019). *What is MQTT (MQ Telemetry Transport)? - Definition from WhatIs.com.* IoT Agenda. https://internetofthingsagenda.techtarget.com/definition/MQTT-MQ-Telemetry-Transport

Yesner, R. (2020, January). IDC MarketScape: Worldwide IoT Applications Platforms for Smart Cities 2019–2020 Vendor Assessment. IDC: *The Premier Global Market Intelligence Company.* https://www.idc.com/getdoc.jsp?containerId=US43580918

Yi, W., Lo, K., Mak, T., Leung, K., Leung, Y., & Meng, M. (2015). A Survey of Wireless Sensor Network Based Air Pollution Monitoring Systems. *Sensors, 15(12),* 31392–31427. https://doi.org/10.3390/s151229859

Yin, C., Xiong, Z., Chen, H., Wang, J., Cooper, D., & David, B. (2015). A literature survey on smart cities. *Science China Information Sciences, 58(10),* 1–18. https://doi.org/10.1007/s11432-015-5397-4

Zehner, K., Griggs, N., Hiser, R., & Traub, N. (2016). " THE CLOUD ". *A Practical Framework for Understanding CloudComputing*. https://www.academia.edu/30948967/_THE_CLOUD_A_Practical_Framework_for_Understanding_Cloud_Computing.

Zhang, J., Rajendran, S., Sun, Z., Woods, R., & Hanzo, L. (2019). Physical Layer Security for the Internet of Things: Authentication and Key Generation. *IEEE Wireless Communications*, *26*(5), 92–98. https://doi.org/10.1109/mwc.2019.1800455