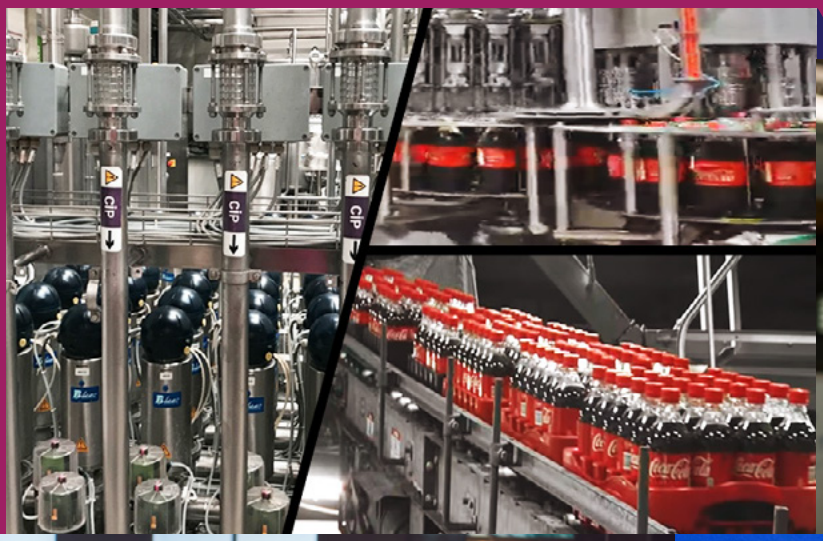
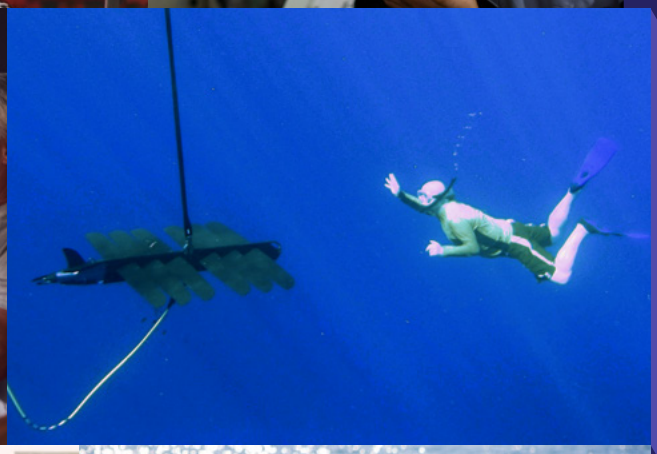
Architecture Monthly

November 2021

IoT for the Edge





EDITOR'S NOTE

Internet of Things (IoT) for the edge encompasses so many devices and industries, that we couldn't pick just one photo for our cover. We include IoT use cases from manufacturing, fitness, ocean research, and agriculture. And these represent only a fraction of what is possible. By moving certain workloads to the edge, your devices communicate with local compute resources and can respond more quickly to changes. AWS edge services deliver data processing, analysis, and storage close to your endpoints, allowing you to deploy APIs and tools to locations outside AWS data centers. You can harness the data generated by your IoT edge devices and enable them to act intelligently with AWS IoT services.

We'd like to thank our experts, Olawale Oladehin, Head of Worldwide Solutions Architect - IoT, Maggie Tallman, Worldwide Go-To-Market Manager - IoT & Robotics, and Richard Elberger, IoT Principal Technologist, AWS. We are also pleased to have a contribution by one of our Customers, Jaime González, Chief Technology Officer, Pentasoft. Special thanks go to Ryan Burke, Sr. Application Architect, and Channa Samynathan, Specialist Solutions Architect – IoT, for their invaluable help shepherding this issue.

Please give us your feedback! Include your comments on the Amazon Kindle page. You can view past issues and reach out to aws-architecture-monthly@amazon.com anytime with your questions and comments.

Jane Scolieri, Managing Editor

TABLE OF CONTENTS

NOTICES

ASK AN EXPERT

Maggie Tallman, Worldwide Go-To-Market Manager - IoT & Robotics
Olawale Oladehin, Head of Worldwide Solutions Architect – IoT

Do you see different trends in IoT in cloud versus at the edge?

We see more of a continuum building between cloud and edge. Customers build incredible applications that use Amazon Web Services (AWS) for data processing, analytics, storage, Internet of Things (IoT), and machine learning (ML). Increasingly, edge computing use cases (like ML inference used in Industrial IoT and manufacturing) are requiring processes be closer to end users and IoT devices.

Edge services are infrastructure and software that deliver data processing, analysis, and storage as close to the endpoint as necessary. This includes deploying services, APIs, and tools to locations outside AWS data centers. Services can also be deployed onto customer-owned infrastructure and IoT devices. As such, edge and IoT are likely to increasingly interoperate. An architectural pattern becoming more common is an edge computing environment placed near sensors that generate data. An example is achieving faster incident detection by forward deploying optimized ML models to an edge gateway with AWS IoT Greengrass.

To realize the benefits of an edge-to-cloud continuum, customers want similar consistency, security, and reliability that they expect from their global cloud infrastructure. You only need to build an application once. You will then have the flexibility to deploy it on the cloud, in your data centers, and at the edge. In any location or platform, you will get consistent performance with centralized management and governance. This significantly shortens the development lifecycle, reduces development costs, increases agility, and improves scalability.

One emerging trend where the lines between IoT and edge are starting to blur is in the convergence of information technology (IT) and operational technology (OT). IoT-enabled devices and connected equipment drive the adoption of edge solutions where infrastructure and applications are being placed within operations facilities. Increasingly, use cases like predictive maintenance and factory floor automation require real-time inference. Often these environments are disconnected from, or only partially connected to the cloud.

Architects have more choice than ever in how they build out capabilities along the edge-to-cloud continuum. Services such as AWS Outposts and AWS Wavelength bring managed cloud services and infrastructure closer to physical points of operation. AWS Panorama is an ML appliance that organizations use to bring computer vision (CV) to on-premises cameras to make predictions locally with high accuracy and low latency. AWS Snowcone is ruggedized,

secure, and purpose built for use outside of a traditional data center. With AWS IoT SiteWise Edge software, you can locally collect, organize, process, and monitor equipment data. Architects have increasing flexibility when selecting where to run workloads.

What are the general architecture pattern trends for IoT at the edge?

Customers are connecting their physical assets to the digital world to inform data-driven decisions. Edge solutions encompass a wide range of industries, each with a unique set of requirements and nuances. At AWS, we group these solutions into six patterns: Hybrid Edge, Industrial Edge, Connected Devices, Edge Networking, Rugged Edge, and 5G. Each use case provides customers the ability to create real-time, low-latency systems.

Hybrid solutions bring together cloud services and infrastructure nearer to devices. This enables highly responsive applications like connected vehicles and smart factories. These solutions consist of services such as AWS Outposts and Amazon EKS Anywhere. Edge Networking moves traffic to the AWS edge network, where customers can use services like Amazon CloudFront to benefit from perimeter protection and edge compute. For Industrial and Rugged Edge, customers are running applications in ruggedized environments that constrict connectivity or power. Customers can collect and process data locally with compute applications using AWS Snow Family. Customers use services like AWS IoT SiteWise to ingest and visualize data from industrial equipment. For each of these patterns, customers are beginning with the purpose-built solution that solves their use case at the edge. They augment those primary services with services like AWS Lambda and Amazon Managed Grafana to deliver consistent computing paradigms and visualization for their workloads.

When putting together an AWS architecture to solve business problems specifically for IoT customers, what are some of the considerations?

Edge applications rely on the cloud for storage, but must also do some processing close to the point of operation. This enables outcomes with optimized latency and cost. For example, industrial machines operating on assembly lines must detect and adjust to variations in the dimensions and quality of materials in real time. Self-driving cars must be able to ingest and process information offline. The edge is moving the point of processing to where you need it. Concepts of cloud computing like security, agility, and consistency must be considered when building edge applications.

Customers can translate best practices learned from the cloud to their edge workloads. Use the same security principles for encryption in transit and encryption at rest that you'd typically use in the cloud. Mirror best practices in managing PKI on premises and in the cloud, such as maintaining current cipher suites and limiting access to encryption keys. Determine how to best manage your physical hardware and what levels of redundancy you need for your workload. Think about the cloud as a set of technology services from which patterns emerge to guide designing for an edge environment.

What's your outlook for IoT, and what role will cloud play in future development efforts?

We see the IoT business as an on-ramp for big data processing, app modernization, and AI/

ML. The proliferation of devices means that our customers increasingly need solutions to connect them and manage the data they generate. There are two key trends that we think are important drivers of the future of IoT and the role of the cloud:

1. **Merging between AI and IoT (or AIOT).** This is occurring within virtually every industry. This merging will test how much data devices can process, and the boundaries of that processing. With the cloud, the smart products of today will evolve to the connected robots and vehicles of tomorrow.

   Andy Jassy, Amazon President and CEO, made a strong case for the evolution of IoT, edge, and cloud:

   > "When we think about 10 years from now and when we think about hybrid, we don't think the on-premises part is going to be in data centers. We think the on-premises part will be billions of these devices that sit at the edge—in our houses, in our offices, in factories and oil fields and agricultural fields and planes and ships, and automobiles—everywhere. These devices have relatively little CPU and relatively little disc, and so the cloud becomes disproportionately important in implementing all of those devices."

   Technologies such as 5G, LoRaWAN, and digital twins will also contribute to the merging of AI and IoT.

2. **Merging of IoT and OT, or industrial IoT (IIoT).** IIoT brings machine data, computing power, and people together to improve the performance and productivity of industrial processes. With IIoT, industrial companies can digitize processes, transform business models, and improve performance and productivity. We are finding that IIoT workloads are typically new workloads for the cloud enabled by increasing maturity of the edge-to-cloud continuum.

   Early adopters are using AWS IoT services and technologies with a wide variety of services to deliver productivity gains in smart manufacturing. IIoT is driving Industry 4.0 and the factory of the future. This gives manufacturers the ability to automate and optimize their operating efficiency. For instance, robotics and automated machinery can work more efficiently and accurately, when reinforced by local intelligent workloads.

   We are very much at "Day 1" with IIoT. Demand for cross-site visibility and smarter insights from IoT data will drive integrating on-premises IIoT workloads with the cloud. In time, most digital data will come from IoT-connected sensors and devices.

ABOUT THE EXPERTS

Maggie Tallman is a Worldwide Go-To-Market Manager for AWS IoT & Robotics, responsible for leading teams who help AWS customers leverage our capabilities in Robotics, KVS, Snow Edge Compute, and IoT Public Sector. Prior to AWS, Maggie held executive roles across both global startups and Fortune 100 companies, spanning operations, business development, product management, and developer relations. Earlier in her career, she worked at HP as Division Manager for the Internet Services Group. In her spare time, Maggie enjoys hiking, biking, and yoga, and has been a Board Director at several national dance companies and animal shelters. She earned her MBA from Notre Dame and makes it back every so often for a football game.

Olawale "Wale" Oladehin is the Head of Worldwide Solutions Architect – IoT (Internet of Things) at AWS. He leads a team of world-class, customer facing Solutions Architects focusing on IoT, Robotics, and streaming video worldwide. Wale has 15 years of experience working across ecommerce, video, and embedded systems. He has a passion for helping customers innovate through technology to achieve their business outcomes. He holds a Bachelor's degree in Computer Science from Princeton University.

CUSTOMER CONVERSATIONS

Jaime González, Chief Technology Officer, Pentasoft

What are the main barriers for businesses adopting IoT?

The Internet of Things (IoT) has expanded. Now the term can be applied to many household and industrial devices that communicate with each other to streamline processes. The main challenges of the IoT industry currently relate to security, cost of implementation, connectivity, complexity, and regulatory standards.

Security. As more devices are connected, the risk of malware increases. Suppliers try to introduce new connected devices quickly with a predominance of functionality over security. The good news is that proven technologies like end-to-end encryption and token-based authentication, features that are well suited for IoT applications, help address the problem.

Cost. Sensors and actuators in IoT projects are not usually expensive, but monitoring complex environments can involve thousands of devices. Low Powered Wide Area Networks (LPWAN) can bring down the cost of devices and connectivity. However, there are scenarios that require higher bandwidth, such as cellular, Wi-Fi, or wired internet. This can impact the cost of a project.

Connectivity. The future of IoT must rely on decentralizing IoT networks by moving functionality to the edge, such as using fog computing models. The variety of platforms make it difficult to find a foundational layer of connectivity. However, some protocols (MQTT, CoAP, XMPP, and OPC-UA, among others) are a good sign of standards on the rise in this industry.

Complexity. Understanding the benefits of concepts like predictive maintenance is straightforward, but engineering a way to accomplish that objective is not. That's because an IoT system often consists of a wide array of components ranging from security to analytics. It is difficult to establish clear workflows for product development because companies

don't have experience implementing IoT technologies.

Regulatory standards. The legal issues involved in IoT projects include information flows across borders, conflicts between surveillance devices and customer privacy, data retention policies, or security breaches. The development speed of the IoT technology frequently exceeds the regulatory environment, creating a perception of unfair or deceptive behaviors to consumers.

Which use cases are more approachable as businesses lean further into edge technologies?

Our experience with customers outlines some common use cases in industrial IoT projects, such as affordable near real-time device monitoring with proactive alerts and predictive maintenance. There is also a demand for a powerful central IoT repository for storing asset data, performing metrics calculations, and extracting insights from sensor data.

With these scenarios in mind, we developed Neuron, our SaaS solution implemented as a collection of serverless microservices. Neuron relies on AWS IoT Core and AWS IoT SiteWise. Neuron Edge, running on AWS IoT Greengrass, is the component performing intensive edge computing tasks. These tasks include data cleansing, data aggregation, delta upload to the AWS IoT SiteWise repository, alarms evaluation with AWS IoT Events, field protocol implementation, and many others. This component is the cornerstone of the low-cost pricing schema offered by Neuron. It reduces data trips to the cloud and implements advanced functionality at the edge, such as sensors and actuators management, or real-time alarms.

What are the general architecture pattern trends for IoT at the edge?

Edge computing is more decentralized and distributed than traditional cloud computing because of IoT's inherent mobility requirements. That is the source of a new set of reference architectures that take a layered approach to decentralize edge computing.

These architectures usually have three distinct layers: *device*, *edge*, and *cloud*. Let's focus on the edge layer, which is responsible for:

- Receiving, processing, and forwarding data from the device layer
- Providing local services such as edge security
- Edge data cleansing, preprocessing, and analysis
- IoT process optimization

The **edge** layer can in turn be divided into three sublayers, or levels of responsibility:

Low: Contains *edge controllers* that collect data from the device, perform preliminary data thresholding, and implement control flow down to the devices. It supports a wide array of communication protocols and interfaces. Upper layers receive operational instructions or data-driven decisions, which are then transmitted to the devices through this level.

Medium: Contains *edge gateways* and is responsible for exchanging data with the low and high levels. It has more storage and computing resources compared to the low level. Data and intelligence derived from the data processing can be cached locally to support future processing.

High: Contains powerful *edge servers* responsible for performing more complex and critical data processing. It makes decisions based on the data collected from the medium

level. It processes bulk data by using more complex machine learning algorithms and analyzes data from different equipment to achieve process optimization, usually with longer latency.

The real value of these architectural patterns lies in the swift development of actionable decisions. This can be achieved by uncovering intelligence and insights from data generated at the edge.

Do you see different trends in IoT at the edge versus IoT in the cloud?

As the volume of data to be processed explodes, alternatives to the traditional IoT model of sending all data to the cloud for processing are required. Edge computing tries to bring data processing as close to an IoT device as possible. That can mean improved latency, performance, cost, and security advantages for companies. Rather than sending data to be processed on cloud servers, the computation takes place on the device or in the local network itself. This reduces the impact on your time and resources.

One current trend is that edge processing power and data storage could all be combined to enable analytics and AI. These processes require very fast response times or involve the processing of large real-time datasets that are impractical to send to the cloud. Beyond performance and latency advantages, this can also be the most economical architectural choice. As much of this data may be ephemeral, a round trip to the cloud may not actually create any value.

A significant benefit of this new model is that it allows companies to have the best of both worlds. You can sense, capture, and analyze massive amounts of data at the point of origin. At the same time, you can obtain global visibility, management, and deeper analysis in the cloud. Organizations using a hybrid cloud strategy and edge computing in tandem will gain greater flexibility and consistency.

What's your outlook for IoT, and what role will the edge technologies play in future development efforts?

I am a strong advocate of IoT as the key factor of a new era of distributed intelligence with an increasing number of connected devices. Putting compute power closer to data sources at the edge is the most effective way to manage the volume of data now being generated. Edge computing will also play a pivotal role in enabling businesses to realize the advantages of emerging technology, like 5G and AI.

IoT will bring digital and physical technologies together. This has great potential to transform industrial processes due to the amounts of raw data generated by machines. Some examples are condition-based monitoring and predictive maintenance.

As the telco industry prepares for 5G, it is also important to consider the fundamental role that the edge will play in delivering those services. The high-bandwidth and low-latency capabilities of 5G make the shorter distance between the device and the edge even more efficient. This allows enterprises to capitalize on massive amounts of data. The volume and complexity of 5G services will surge at the edge. Pushing computing closer to data acquisition is the only way to achieve the ultra-low latency outcomes enabled by 5G.

References

[1] H. Washizaki, S. Ogata, A. Hazeyama, T. Okubo, E. B. Fernandez, and N. Yoshioka, "Landscape of Architecture and Design Patterns for IoT Systems," IEEE Internet of Things Journal, vol. 7, no. 10, pp. 10091–10101, Oct. 2020, doi: 10.1109/jiot.2020.3003528.

[2] F. Dahlqvist, M. Patel, A. Rajko, and J. Shulman, "Growing opportunities in the Internet of Things," McKinsey & Company, Jul. 22, 2019.

[3] "IoT Architecture: Topology and Edge Compute Considerations." https://www.digi.com/blog/post/iot-architecture-topology-and-edge-compute/ (accessed Sep. 09, 2021).

[4] F. Borelli, "Architectural Software Patterns for the Development of IoT Smart Applications," Universidade Federal do ABC, Mar. 2020.

[5] A. Calihman, "Architectures in the IoT Civilization," NetBurner, Jan. 30, 2019. https://www.netburner.com/learn/architectural-frameworks-in-the-iot-civilization/ (accessed Sep. 28, 2021).

[6] M. Kashaboina, "An intelligent edge: A game changer for IoT," TechTarget, Sep. 15, 2021.

ABOUT THE CUSTOMER

Jaime González is the Chief Technology Officer at Pentasoft. He began his career in IBERIA airlines (IAG Group) developing software modules for airport ground operations. In 2005, he co-founded Pentasoft, a software development company specializing in serverless SaaS solutions on AWS. He is passionate about serverless architectures and software design patterns based on microservices. Jaime has a Bachelor's Degree in Computer Science from the Universidad Politécnica of Madrid with further studies in networking and data science.

ASK AN EXPERT, HARDWARE SECURITY

Richard Elberger, IoT Principal Technologist, AWS

Why do our customers use hardware security for the IoT? What problems does it solve?

Over the last decade we've witnessed an increase in IoT security issues. These issues can be against a single device or an entire IoT system. A compromised IoT device is usually a result of an increased *attack surface* with any number of exposed *attack* vectors.

Network communications, required by IoT, increase the attack surface. Artifacts, such as a password, are used to authenticate and authorize communications across the network. If passwords are somehow accessed, bad actors can gain the needed information to leverage an attack vector, such as a server that the device communicates with. Perpetrators can lift information about the application from the device, such as the application code from flash memory. This can increase the attack surface, since they are then able to grasp the application scope.

Hardware security, either using discrete components, embedded enclaves, or Physical Unclonable Function (PUF), plays a vital role in protecting customers against both approaches. The *root of trust* (a source that can always be trusted within a cryptographic system) must be generated by the hardware security and never be read into main memory. It should have physical security so information cannot be directly read into other components such as main memory and application flash. And it should have a cryptography library that performs and accelerates encryption operations.

Hardware security protects the client private key used in the Transport Layer Security (TLS) for network authentication and authorization. Hardware security protects the private key used for flash disk encryption, secure boot, and verifies firmware over-the-air updates.

What are the major trends with IoT device hardware security today?

Although the Trusted Platform Module (TPM) has been a component of high-end assets and laptops for almost two decades, the adoption of IoT hardware security hasn't been nearly as pervasive. This can be because of cost, as hardware security also increases both hardware and software design complexity.

In such cases, we are seeing semiconductor companies include hardware security in microprocessors and microcontrollers. This makes their use much more common since increased security benefits everyone in the IoT value chain. Some examples include the Espressif ESP32-SE and the Xilinx Kria. Within the popular ESP32-SE module, Espressif has designed in the Microchip ECC608A secure element. Similarly, Xilinx has designed Zynq® UltraScale+™ MPSoC with an Infineon TPM 2.0.

With the hardware security built into the main compute module, IoT devices can now be designed with decreased hardware complexity.

We see this trend across classic TPMs, secure elements, secure enclaves, and other embedded security approaches.

Companies such as Espressif and Xilinx then provide integration guidance to software designers and engineers. They share implementation best practices that reduce software complexity. These designs are not common across all semiconductors. I would encourage customers to look for modules that offer reduced complexity, like the Espressif and Xilinx modules, as well as others like the Microchip SAMA527 Wireless System-on-Module (SOM). Similarly, Espressif and SiFive deliver secure enclave technologies and Arm TrustZone serves as the foundation for Arm's Platform Security Architecture (PSA).

What has AWS IoT been doing to help customers use hardware security effectively?

There are three things that AWS IoT has been doing to help customers use hardware security effectively.

1. **Working with partners.** AWS Partners provide a spectrum of hardware security solutions. The AWS Partner Network works with our partners to verify and demonstrate AWS IoT integrations. You can view our partner hardware security solutions on the AWS Partner Device Catalog.
2. **Ensuring that IoT provisioning works seamlessly.** We have the provisioning options our customers need to tackle a wide variety of use cases. Since 2016, AWS IoT provided Just-in-Time Registration (JITR) and Just-in-Time Provisioning (JITP). Other options include Fleet Provisioning and Bulk Registration. The option you choose really depends on the type of hardware security you have and the user experience you want for the IoT device.
3. **Ensuring the device software works with hardware security.** AWS provides the software our customers need in order to integrate AWS device software with hardware security to ensure end-to-end IoT security. This includes AWS IoT Device SDKs, FreeRTOS, and AWS IoT Greengrass.

What are some of the decisions I need to make when selecting hardware security?

You'll need to make decisions that involve security cryptographic strength, materials used in manufacturing, and consider options such as easy-to-use software development kits (SDKs).

The main thing you're protecting in hardware security is the **private key**. The private key represents the device identity. Your security hardware must permit secure connection software, like mbedTLS, to use that key without ever loading it into the device main memory. Virtually all modules, enclaves, and secure partitions do this differently. When the protection and cryptography is safer and faster, the module will typically be more expensive. The module you choose corresponds to the value of the asset and the overall IoT system that you're protecting. In other words, the choice will be different if a connected light bulb is installed in a home bathroom or a submarine.

Speaking of submarines, let's discuss **environmental conditions**. The packaging of the hardware security chip, and other materials, may need to withstand extreme environmental conditions such as moisture, pressure, and

temperature. These constraints will be stated in the hardware security datasheet.

Another consideration is the **form factor**. This is especially important when the hardware security is not included in the microprocessor or microcontroller. It is possible that the available package (the enclosure around the chip that goes on the board) might not fit in your industrial design. In this case, you may want to consider a microcontroller that has built-in hardware security.

Where can customers find more information about designing in hardware security if they're not already doing so?

There are at least three places where customers can seek help about using hardware security modules with AWS IoT.

A bit more technical understanding might be required to make the right hardware security or secure enclave decision for your use case. Here are a few blogs I would recommend to get started: an AWS blog about modules manufactured by Microchip, a blog about modules manufactured by Infineon, and a blog focusing on Arm TrustZone.

You will need to understand how modules usually make their way through the device manufacturing process. AWS recently published Device Manufacturing and Provisioning with X.509 Certificates in AWS IoT Core that leads customers through the typical process.

Once you're comfortable with the hardware security landscape, find out which hardware security modules work well with AWS IoT so you can begin prototyping. The AWS Partner Device Catalog lists IoT hardware that is qualified for AWS IoT Core. Navigate to the AWS Partner Device Catalog and search for "hardware security." Each module maker will have its own design-in requirements.

ABOUT THE EXPERT

Richard Elberger is an IoT Principal Technologist at Amazon Web Services. As a prolific speaker, periodic writer, and tireless embedded technology addict, he creates content and builds community for IoT and Cloud practitioners globally. Richard maintains and contributes to multiple IoT-related open source projects (FreeRTOS, meta-aws, and ThingPress) which help customers build and deliver amazing IoT solutions on AWS.

WHITEPAPER

Security at the Edge: Core Principles

September 24, 2021

Abstract

Today's business leaders know that it is critical to ensure the security of their environments, and the security present in traditional cloud networks is extended to workloads at the edge. This whitepaper provides security executives the foundations for implementing a defense in depth security strategy at the edge by addressing three areas of security at the edge:

- AWS services at AWS edge locations
- How those services and others can be used to implement the best practices outlined in the design principles of the AWS Well-Architected Framework Security Pillar
- The security aspects of additional AWS edge services, which customers can use to help secure their edge environments or expand operations into new, previously unsupported environments

Together, these elements offer core principles for designing a security strategy at the edge, and demonstrate how AWS services can provide a secure environment extending from the core cloud to the edge of the AWS network and out to customer edge devices and endpoints.

Introduction to edge computing

Security is the top priority at AWS. The high security bar set by AWS services covers customers as they expand their use of AWS services to bring workloads out to the edge to use its growing number of capabilities and applications.

Edge computing comprises elements of geography and networking, and brings computing closer to the user. Edge takes place at or near the physical location of either the user or the source of the data. By placing computing services close to these locations, the user benefits from faster, more reliable services.

This paper discusses AWS services that are available to provide a secure environment, from the core cloud to the edge of the AWS network, and out to customer edge devices and endpoints. Many of the AWS services that provide security capabilities to the edge reside at AWS edge locations, or as close to customers' edge devices and endpoints as necessary. AWS edge locations are a worldwide network of data centers that run with AWS at physical locations directly connected to the expanding AWS global infrastructure.

AWS edge services provide infrastructure and software that deliver data processing, analysis, and storage as close to the endpoint as necessary. This includes deploying AWS Managed Services, APIs, and tools to locations outside AWS data centers, and even onto customer-owned infrastructure and devices. AWS enables customers to build high-performance applications that rely on the cloud for data processing and storage, but also need to process or store some data close to where it is generated to deliver ultra-low latency, intelligent, real-time responsiveness, and reduce the amount of data transfer.

Every AWS customer is unique, and "edge" can mean something different to different customers. Edge use cases and technology can range from autonomous vehicles, medical devices, oil rig sensors, industrial robots,

nautical GPS, and meteorological devices. Mobile phones and robot vacuums are also examples of edge devices.

The objective of AWS edge services is to provide consistent capabilities and customer experience from the edge to the cloud. AWS uses the same programming model for the cloud, on-premises infrastructure, and local devices. This gives you the choice of centralized control or de-centralized control, with decentralized implementation. You have access to the same environment to develop, connect, deploy, manage, and secure with the same tools, regardless of where your workloads are located.

Security at the edge

AWS provides services and features you can use to help you create secure architectures, workloads, and services to elevate your security from edge to cloud. Security at AWS starts with core infrastructure, which is built for the cloud and designed to meet the most stringent security requirements in the world. For example, all data flowing across the AWS global network that interconnects data centers and Regions is automatically encrypted at the physical layer before it leaves AWS secured facilities.

At the edge, AWS offers services that address the different aspects of edge security, including preventive security mechanisms like encryption and access control, continuous monitoring mechanisms like configuration auditing, and physical security like tamper-evident enclosures. Customers that need to store and process data on premises, or in countries where there is no AWS Region, can do so securely with AWS edge services. This capability can help you comply with data handling or data residency requirements.

AWS Cloud security principles are fundamental and apply regardless of where an organization operates. These principles are discussed in detail in a later section of this whitepaper. AWS offerings combine a high security bar with agility to adapt rapidly as needed. AWS customers working at the edge have access to over 200 fully featured, integrated cloud and device services, many of which have specific edge capabilities.

AWS services with Points of Presence (PoP) at edge locations — globally scaled and connected through the AWS network backbone — provide a more secure, performant, and available experience. AWS also offers services that run on the edge, which enable you to deliver content. AWS

edge services, which provide infrastructure and software that deliver data processing, analysis, and storage at endpoints comprise a comprehensive set of cloud services that support the secure deployment and management of edge devices.

Security at the edge has the same principles as cloud security. By extending cloud services to the edge, AWS gives you a way to operate safely, with strong security infrastructure and safeguards. AWS-owned infrastructure is monitored 24/7 to help safeguard the confidentiality, integrity, and availability of our customers' data. Moving cloud workloads to edge devices or endpoints provides you with more control and visibility, and mitigates risk.

Media and entertainment at the edge

The media and entertainment industry provides natural examples of customers who need to focus on securing their content delivery at the edge. For example, Amazon CloudFront provides streaming services with low latency, sustained high throughput, lower rebuffering rates, and integration with other AWS services, while also securely distributing content globally. For more information, see Amazon CloudFront for Media & Entertainment.

A defense in depth model (for example, using multiple independent layers of specialized security controls) provides layers of protection. In addition to the design principles of the AWS Well-Architected Framework's Security Pillar, this paper highlights three aspects of edge protection whose PoP is at AWS edge locations. The three highlighted edge protections that help secure the connection points between the origin infrastructure, edge services, and customer edge devices or applications are:

- Secure content delivery
- Network and application layer protection
- Distributed Denial of Service (DDoS) mitigation

The design principles also cover the security of edge devices and applications. A comprehensive defense in depth strategy should include services that account for the security of both AWS edge locations, and edge devices and applications.

Secure content delivery

Secure content delivery provides content, such as data, videos, applications, and APIs, quickly and securely to customers. These should be delivered over secure transport, using the recommended version of Transport Layer Security (TLS) to encrypt communications between endpoints. If necessary, there are a number of methods that you can use to help secure that same content through restricted access, including signed URLs, signed cookies, and token authentication.

Amazon CloudFront, a global content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to viewers with low latency and high transfer speeds, addresses these areas of security when it is deployed at AWS edge locations.

To create a more secure CDN, organizations can gain protection against L3/L4 DDoS attacks using AWS Shield. AWS also offers AWS Shield Advanced, which provides additional detection and mitigation against large and sophisticated DDoS attacks, nearreal-time visibility into attacks, and

integration with AWS WAF, a web application firewall service, to protect against application layer (L7) attacks. Together, these services create a flexible, layered security perimeter.

CloudFront offers security capabilities, including field-level encryption and HTTPS support, seamlessly running with AWS Shield Advanced, AWS WAF, and Amazon Route 53 to protect against multiple types of attacks, including network and application layer DDoS attacks. For more details about CloudFront and Route 53, see the Appendix.

Network and application layer protection

Edge networks are architected outside of the security perimeters of traditional cloud. Extending security to edge end devices requires network and application security and continuous monitoring, as well as encryption of data in transit and at rest.

Edge customers should define trust boundaries for networks and accounts, and verify secure system configurations and other policy-enforcement points, including web application firewalls (WAFs) and API gateways. This can be done by blocking well-known exploits, implementing protections specific to applications, responding to new threats, and performing ongoing monitoring.

There are two important aspects to network and application layer protection at the edge:

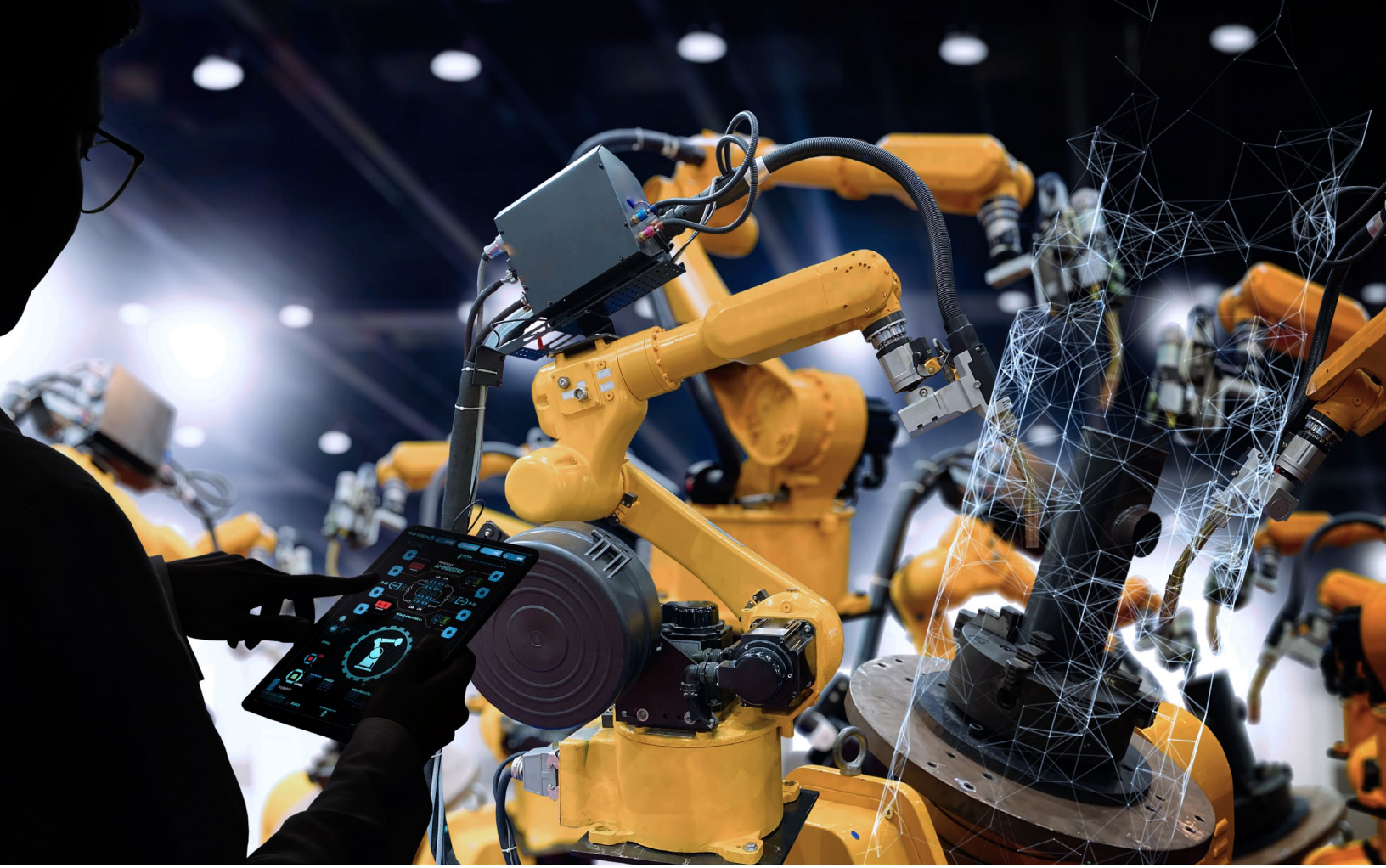
- Protections from well-known exploits and attacks that could affect an organization's applications
- Visibility and control of workloads

Manufacturing at the edge

Edge computing offers manufacturers opportunities to collect, process, and analyze data to enable predictive maintenance, improve quality control, and enhance worker safety with near-real-time alerts, industrial robot fleet management, and simulation. Although these edge applications can increase efficiency and keep costs down, they should be protected against security events. AWS WAF provides security rules to help protect these edge applications against common security attacks. AWS Shield Advanced helps protect against DDoS attacks.

A WAF deployed at AWS edge locations can help to set fundamental protections, customize them to the applications, and help organizations quickly visualize actions so they can create a dynamic security posture. With AWS WAF, you can use the AWS preconfigured rules (Managed Rules), use Marketplace Rules, or create your own custom rules to protect against common attack vectors. AWS Managed Rules give you protection against common web application attacks. They are curated by multiple points of intelligence across multiple sources within AWS.

Marketplace Rules are written, updated, and managed by third-party security experts, and can be used on their own or in conjunction with AWS Managed Rules. AWS WAF, which integrates with AWS Shield Advanced at no extra cost, provides easy setup, low operation overhead, minimal latency impact, and customizable security. It also uses advanced automation to analyze web logs, identify malicious requests, and automatically update security rules.



In addition to preventing incidents, visibility into traffic coming into and out of a network is a second key aspect of network and application layer protection. There are multiple options available to get insights and metrics: CloudWatch metrics, sampled web requests, and logs.

With CloudWatch, you can monitor web requests and web access control lists (ACLs) and rules. CloudWatch collects and processes raw data from AWS WAF and Shield Advanced into readable, near-real-time metrics. AWS WAF supports full logging of all web requests inspected by the service, which can then be stored in the cloud for compliance and auditing purposes, and used for debugging and additional forensics. You can also integrate the logs with your security information and event management (SIEM) and log analysis tools.

For more details about AWS WAF, see the Appendix.

View full whitepaper online

CASE STUDY

Seafloor Systems Saves 4 Hours of Labor per Robot Build Using AWS IoT Greengrass

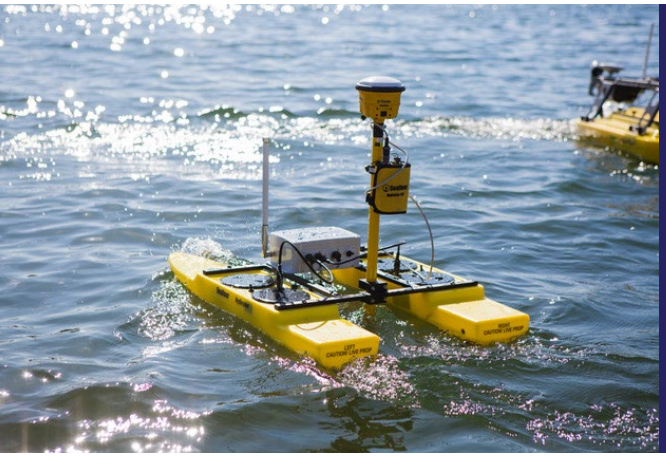
2021

Hydrographic robot developer Seafloor Systems needs to build and manage its fleet of uncrewed survey vessels (USVs) efficiently to effectively scale. The startup creates bespoke autonomous robotic boats for its global customer base. Remotely operated, these boats collect data to chart bodies of water. Seafloor Systems wanted a globally scalable set of tools and resources to streamline the development and production of its robots.

Seafloor Systems found those tools and resources on Amazon Web Services (AWS). Using AWS Internet of Things (IoT) solutions and AWS RoboMaker, which provides a fully managed cloud infrastructure for robotic developers to simulate, test, and securely deploy robotic applications at scale, the startup can build and update its robots more simply, quickly, and cost effectively to better meet customer needs.

Exploring AWS RoboMaker and AWS IoT Solutions to Enhance Application Delivery

Seafloor Systems serves hydrographers at research institutions and companies globally. In addition to its USVs, it offers instruments, software, and support for hydrographic projects of all sizes. The startup began using AWS in January 2020. "We're a small team, so we're always balancing development, deployment, and building up our R&D," explains Marcos Barrera, lead robotics and artificial intelligence research engineer at Seafloor Systems. "You can't build an infrastructure and have people in disparate locations work well together without a strong backbone and scalability. That's one reason I was drawn to AWS."



"As our technology stack has grown, we wouldn't be able to manufacture at the scale at which we sell without using AWS IoT Greengrass."

Marcos Barrera

Lead Robotics and Artificial Intelligence Research Engineer, Seafloor Systems

In 2020, Barrera began exploring AWS IoT Greengrass, an IoT open-source edge runtime and cloud service, to deploy updates to robots in the field. AWS IoT Greengrass would provide the agility Seafloor Systems needed to improve its applications quickly and securely—enabling it to better respond to customer needs and add more functionality without having to start from scratch. However, Barrera also needed a way to perform simulations from home during the COVID-19 pandemic. After searching for a solution, he began to explore application development on AWS RoboMaker.

By running simulations on AWS RoboMaker, Seafloor Systems could quickly test a variety of use cases on its robots without the complexity. This solution would eliminate the need to test software on physical robots in remote locations, enabling the company to reduce costs and increase the speed of its delivery. Because AWS RoboMaker and AWS IoT Greengrass operate on cloud infrastructure, Seafloor Systems could use the flexibility of the cloud to streamline and scale the development of its robot fleets.

AWS RoboMaker is also compatible with Robot Operating System, an open-source collection of frameworks for robot software development. This was an important consideration for Barrera; with this compatibility, Seafloor Systems could collaborate with a community of roboticists. Robot Operating System compatibility would also enable greater modularity, allowing the startup to implement and test different functionalities without replacing its existing work.

Deploying Robots Faster and More Simply on AWS

Barrera initially used the first version of AWS IoT Greengrass to manage software bundling. It wasn't until joining the private beta in November 2020 for AWS IoT Greengrass Version 2, which adds ease-of-use functionality in response to customer requests, that Barrera saw how the service could change how Seafloor Systems builds its robots on a large scale. AWS IoT Greengrass Version 2 builds on the ability of AWS IoT Greengrass to package a software stack by adding modularity, letting users set their stacks up as a network of dependent components—a capability that complemented the startup's bespoke designs. When building a new robot, it can package all system components and reuse as much as possible, increasing efficiency. When Seafloor Systems needs to exchange a thruster system or sensor package, for example, it can do so remotely instead of having to start from scratch. "It's less work and really surgical," says Barrera. "And no one on my team has to fully understand the nuances of all the interconnections. We can just plug in different services for the robot, and then all the necessary drivers, connections, and algorithms are there."

In December 2020, Seafloor Systems deployed a proof of concept for deploying and managing software at the edge in its robots, using AWS RoboMaker and AWS IoT Greengrass. A key benefit of the solution is its ease of use, especially because Seafloor Systems maintains unified product development across physical locations. The solution saves at least 4 hours of manual labor in building the software stack of each robot. Given Seafloor Systems' current fleet of 2,000 active HyDrones, its smallest and best-selling USV, that savings adds up quickly. "As our technology stack has grown, we wouldn't be able to manufacture at the scale at which we sell without using AWS IoT Greengrass," says Barrera. Using AWS IoT Greengrass also enables Seafloor Systems to deploy applications to robots in remote

locations in times ranging from minutes to 1 hour, compared to the days or weeks it would previously take. The company saves approximately $1,500 per robot deployment by using AWS.

Seafloor Systems plans to use AWS RoboMaker for much of its robotic development and simulation. The company is exploring simulation pipelines for maritime robotics, using AWS services to test field use cases without the complexity, cost, and difficulty of performing identical live tests on the open water—something Barrera sees becoming increasingly crucial as the field evolves. "Using AWS RoboMaker for simulation enables us to use the latest and greatest research as it becomes available," says Barrera. "That's going to be a differentiator for us." Using AWS RoboMaker could also pose new opportunities for Seafloor Systems' work with partner universities and institutions. Students and researchers who don't have access to USVs could instead learn by running simulations using the dimensions and software stacks of Seafloor Systems' USVs.

Taking Advantage of IoT Innovations

On AWS, Seafloor Systems is prepared for the future of IoT in maritime robotics. Barrera imagines an IoT network in which the boats collect data and share it to a federated dataset in the cloud that everyone can access to learn about a location. Then, a global machine learning model could perform analytics at the edge using AWS IoT Greengrass.

Using AWS services, Seafloor Systems can not only build and deploy its robots more quickly but also facilitate innovation so that its robots can work for more use cases. For example, a New Zealand research institution used a HyDrone and computer vision to identify invasive aquatic plants in freshwater areas that couldn't be seen with the naked eye. "Hydrographic scientists and businesses will always need our boats," says Barrera. "But there's this whole other world of pure robotics that benefits from being able to put sensors out in the world to collect data. There's going to be so much advancement in various fields just because we were able to give them robots to do these things."

Seafloor™

About Seafloor Systems

Founded in 1999 and based in California, Seafloor Systems offers hydrographic instruments like hydroacoustic sonar equipment, a growing fleet of uncrewed survey vessels, software, and support to hydrography projects of all sizes.

Benefits of AWS

- Saves at least 4 hours of manual labor per robot built
- Deploys applications to remote robots in minutes to 1 hour, versus days to weeks
- Streamlines production with modular software management
- Maintains unified product development across physical locations
- Saves money and time by relying on simulation, not live tests
- Enables partner institutions to do research with simulations
- Saves approximately $1,500 per robot deployment

Read case study online

IMPLEMENTATION GUIDE

Monitoring River Levels Using LoRaWAN

About this guide

Authorities around the world have the important responsibility of monitoring river and sea levels, so both public institutions and private citizens can be better informed of flood risks. This implementation guide demonstrates how AWS IoT Core for LoRaWAN can be used in conjunction with a qualified gateway device from AWS Advanced Technology Partner Laird Connectivity to install a private long range wide-area network (LoRaWAN) capable of collecting environmental monitoring data, such as river levels.

Overview

Both public and private sector organizations play a crucial role in managing the risk to life and property from flooding. To illustrate the size of the task faced by such authorities, Flooding in England: national assessment of flood risk, published by the Environment Agency, identified that one in six properties in England is at risk of flooding. Furthermore, it reported that rising sea levels and increasingly severe and frequent rainstorms caused by climate change mean that the risk of flooding will only increase.

As part of a comprehensive approach, authorities commonly undertake monitoring of river and sea levels at a finite number of fixed monitoring stations, providing both immediate and longer-term profiling of risk from rising water levels. To facilitate even greater geographical coverage, low-power wide-area networks (LPWAN) technologies such as LoRaWAN give organizations additional flexibility to deploy low-cost, low-power sensors without depending on existing power or telecoms infrastructure.

This implementation guide demonstrates how AWS IoT Core for LoRaWAN can be leveraged alongside the Laird Connectivity Sentrius RG1xx LoRaWAN Gateway to deploy a private LoRaWAN network capable of collecting environmental sensor readings from a fleet of geographically distributed microcontrollers.

Before you begin

- **Long range (LoRa)** is a wireless radio communication technology which operates in the license-free, sub-gigahertz radio frequency band. Due to the technology's focus on achieving longer range and lower power consumption compared to other wireless connectivity standards such as Bluetooth, Wi-Fi or mobile broadband, LoRa has found widespread use to meet a variety of Internet of Things (IoT) use cases where there is a compelling requirement to implement an LPWAN. In such deployments, the LPWAN is often used to facilitate communication between geographically distributed, low-cost, power-constrained devices such

as battery-operated sensor units that are positioned in remote locations with challenges in access.

- **Long range wide-area network (LoRaWAN)** provides the protocols for the upper layers of the LPWAN. It builds on the lower physical foundations provided by LoRa technology, including its hardware, to manage end-to-end communication between devices that participate in the overall network. Additionally, LoRaWAN allows data payloads to wirelessly flow between devices participating in the network and centralized gateways responsible for routing the traffic.
- **AWS IoT Core for LoRaWAN** is a fully managed feature that removes the undifferentiated heavy-lifting of instantiating and operating a private LoRaWAN network by enabling customers to build a fully serverless, scalable, and secure LoRaWAN-based application that tightly integrates with AWS services, including AWS IoT Core.

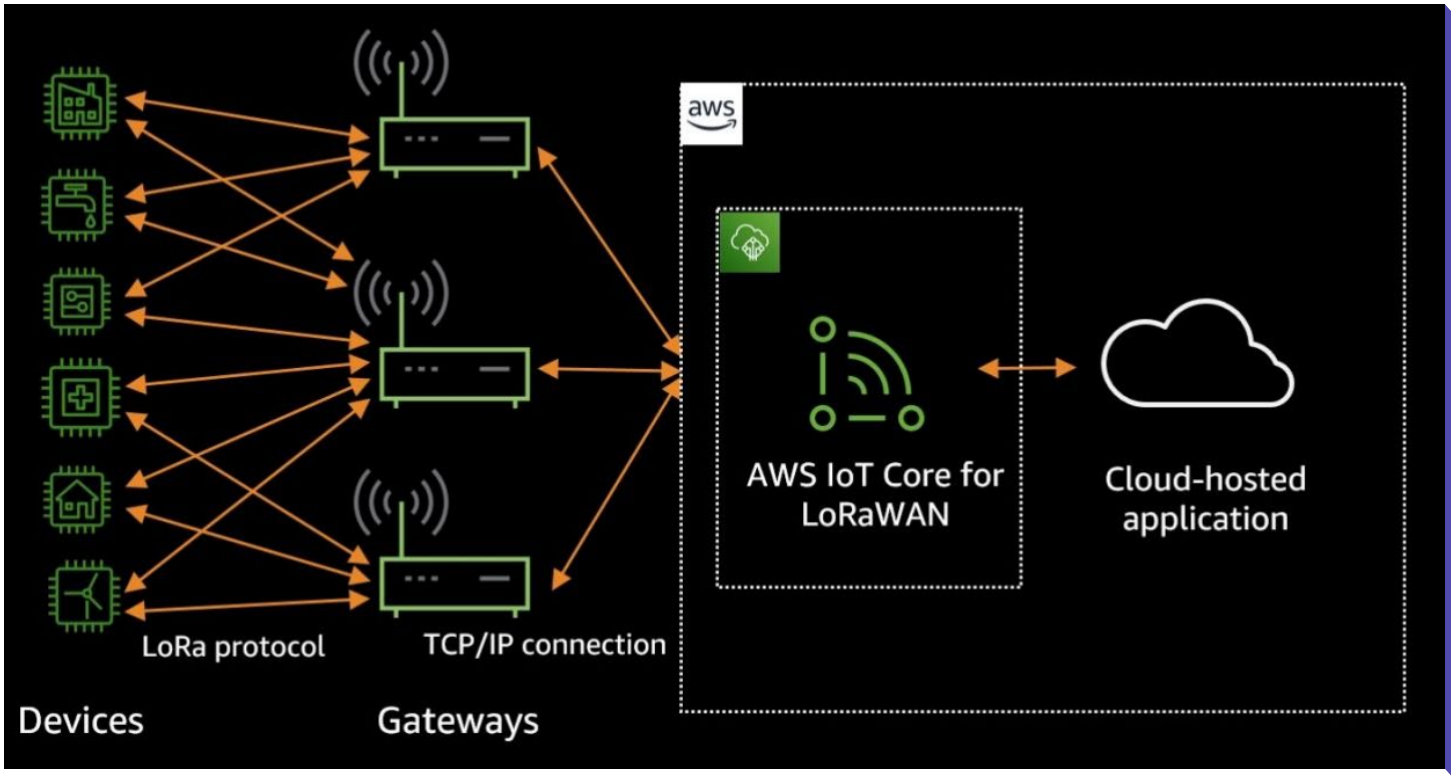


Figure 1 – AWS IoT Core for LoRaWAN overview

Cost

All AWS services included in this implementation guide have a pay-as-you-go pricing model, which scales relative to the demand placed on the application. There are no upfront or monthly commitments.

There are no additional charges for using AWS IoT Core for LoRaWAN, beyond AWS IoT Core charges incurred from messaging. However, if additional AWS IoT Core features are used in

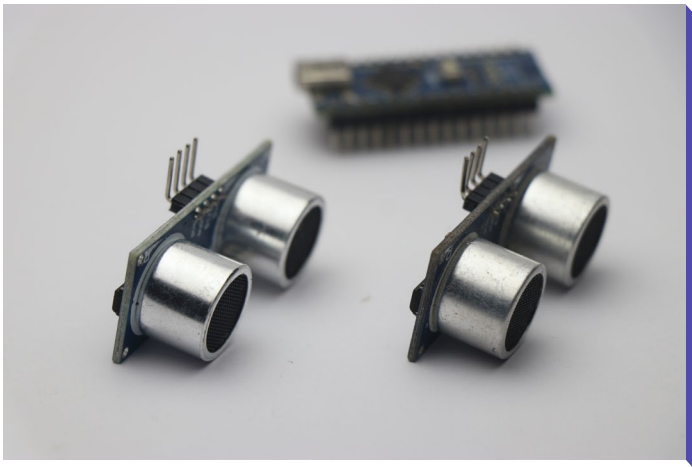
conjunction with the deployment, connectivity, device shadow, registry, and rules engine, charges may apply.

The cost of AWS Lambda, used in the solution to decode LoRaWAN payloads, is based on the number of times the function is run, and the duration it runs for.

Architecture overview

LoRaWAN gateway

To facilitate LoRaWAN connectivity, deploy a Laird Connectivity Sentrius RG1xx LoRaWAN gateway, which is a gateway device qualified for use with AWS IoT Core, and provides a range of up to ten miles for connecting devices. This gateway is registered in AWS IoT Core for LoRaWAN, and configured to receive data payloads from the remote IoT device wirelessly.



Once configured and registered, the gateway communicates with AWS IoT Core for LoRaWAN over a fixed internet connection using two distinct protocols: Configuration and Update Service (CUPS) and WebSocket Secure (WSS).

The CUPS protocol allows a supported LoRaWAN gateway to periodically retrieve configuration and software updates from a remote CUPS server. Although optional, its use is highly recommended, as it simplifies the management of LoRaWAN gateways. The LoRaWAN Basics Station software running on the gateway leverages CUPS to securely communicate with the managed CUPS server running in AWS over HTTPS, and retrieve endpoint information and certificates for the data plane.

Thereafter, actual data transfer is facilitated over the data plane using the LoRaWAN Network Server (LNS) protocol based on WebSocket Secure (WSS).

LoRaWAN device

To simulate a low-cost, low-power microcontroller monitoring a designated river level, a Pycom LoPy4 ESP32 development board equipped with a built-in Semtech SX1276 LoRa transceiver is used to upload data to the gateway. You will use an HC-SR04 ultrasonic distance sensor to approximate the distance to the water surface, and to send this measurement as a lean, two-byte payload. This paper provides a MicroPython application example which illustrates the distance capture and its subsequent transmission as a valid LoRaWAN payload.

AWS IoT Core for LoRaWAN

With a goal of building an end-to-end application, service and wireless device profiles are configured in AWS IoT Core for LoRaWAN, and the microcontroller registered as a wireless device.

To facilitate onward connectivity to additional AWS services, a destination accompanied by an AWS IoT rule is configured.

AWS Lambda decoder

Payloads received by LoRaWAN are base64 encoded. As such, you will use a decoder function deployed to AWS Lambda to decode the payload, construct a meaningful JSON payload, and republish this back to AWS IoT Core from where it can be forwarded to AWS services. This paper provides a Lambda decoder function example to demonstrate this conversion.

The Lambda decoder function allows other applications and devices to subscribe to messages arriving via LoRaWAN through the use of the MQTT protocol and a designated topic. Depending on the precise use case, the Lambda function could be modified to undertake alternative tasks, such as directly invoking an AWS SDK API call to forward data to other AWS services, or updating the device shadow.

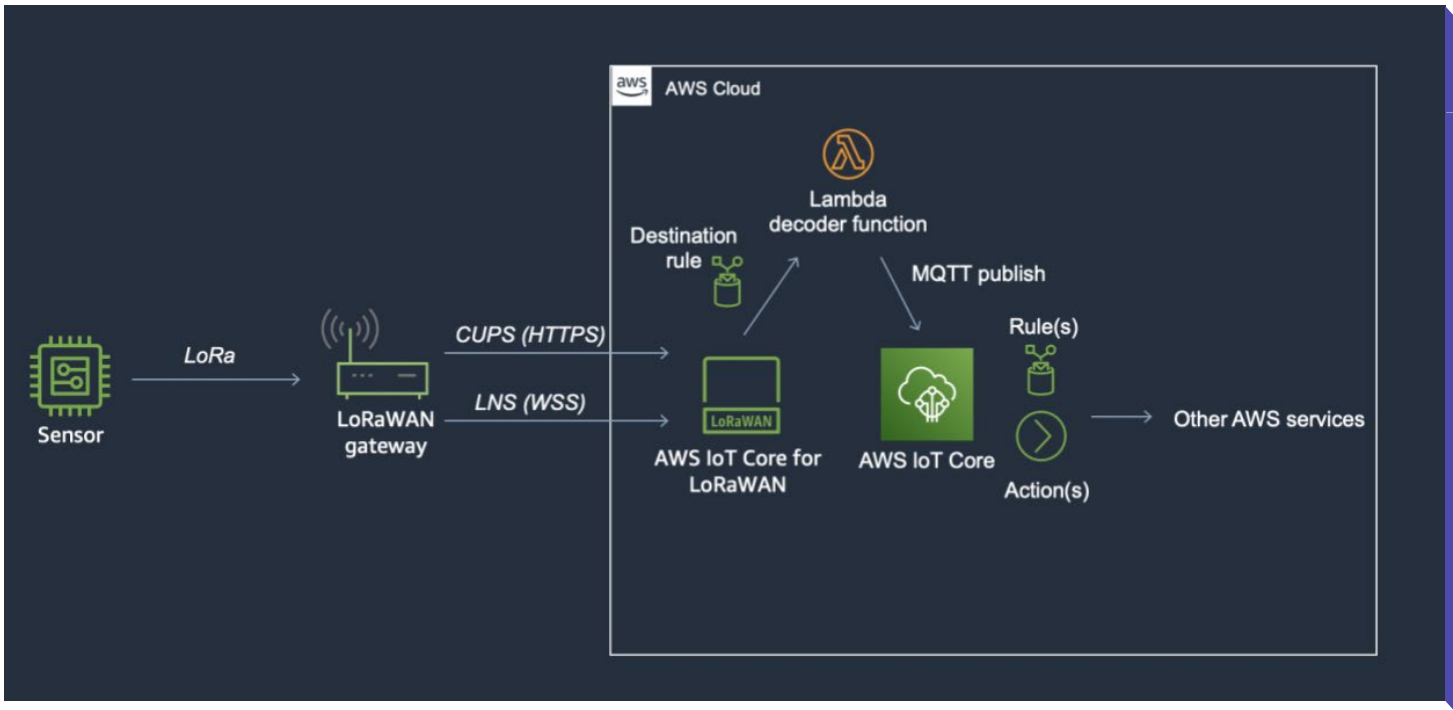


Figure 2 – Solution overview

[Read full implementation guide online]

Run ML inference on AWS Snowball Edge with Amazon SageMaker Edge Manager and AWS IoT Greengrass

by Raj Kadiyala and Nida Beig

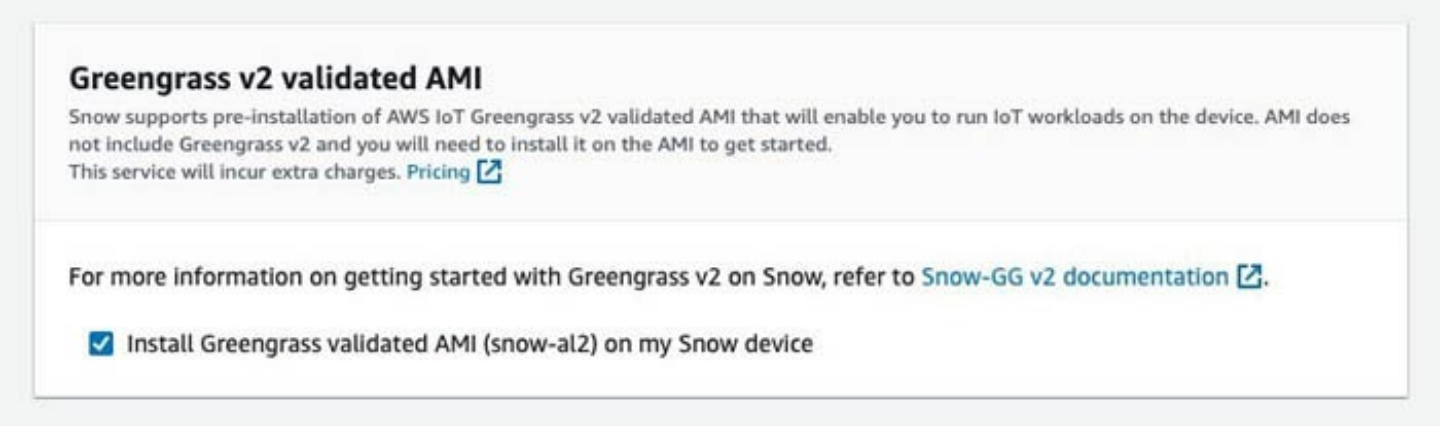
You can use AWS Snowball Edge devices in locations like cruise ships, oil rigs, and factory floors with limited to no network connectivity for a wide range of machine learning (ML) applications such as surveillance, facial recognition, and industrial inspection. However, given the remote and disconnected nature of these devices, deploying and managing ML models at the edge is often difficult. With AWS IoT Greengrass and Amazon SageMaker Edge Manager, you can perform ML inference on locally generated data on Snowball Edge devices using cloud-trained ML models. You not only benefit from the low latency and cost savings of running local inference, but also reduce the time and effort required to get ML models to production. You can do all this while continuously monitoring and improving model quality across your Snowball Edge device fleet.

In this post, we talk about how you can use AWS IoT Greengrass version 2.0 or higher and Edge Manager to optimize, secure, monitor, and maintain a simple TensorFlow classification model to classify shipping containers (connex) and people.

Getting started

To get started, order a Snowball Edge device (for more information, see Creating an AWS Snowball Edge Job). You can order a Snowball Edge device with an AWS IoT Greengrass validated AMI on it.

After you receive the device, you can use AWS OpsHub for Snow Family or the Snowball Edge client to unlock the device. You can start an Amazon Elastic Compute Cloud (Amazon EC2) instance with the latest AWS IoT Greengrass installed or use the commands on AWS OpsHub for Snow Family.



Launch and install an AMI with the following requirements, or provide an AMI reference on the Snowball console before ordering and it will be shipped with all libraries and data in the AMI:

- The ML framework of your choice, such as TensorFlow, PyTorch, or MXNet
- Docker (if you intend to use it)
- AWS IoT Greengrass
- Any other libraries you may need

Prepare the AMI at the time of ordering the Snowball Edge device on AWS Snow Family console. For instructions, see Using Amazon EC2 Compute Instances. You also have the option to update the AMI after Snowball is deployed to your edge location.

Install the latest AWS IoT Greengrass on Snowball Edge

To install AWS IoT Greengrass on your device, complete the following steps:

1. Install the latest AWS IoT Greengrass on your Snowball Edge device. Make sure dev_tools=True is set to have *ggv2 cli* See the following code:

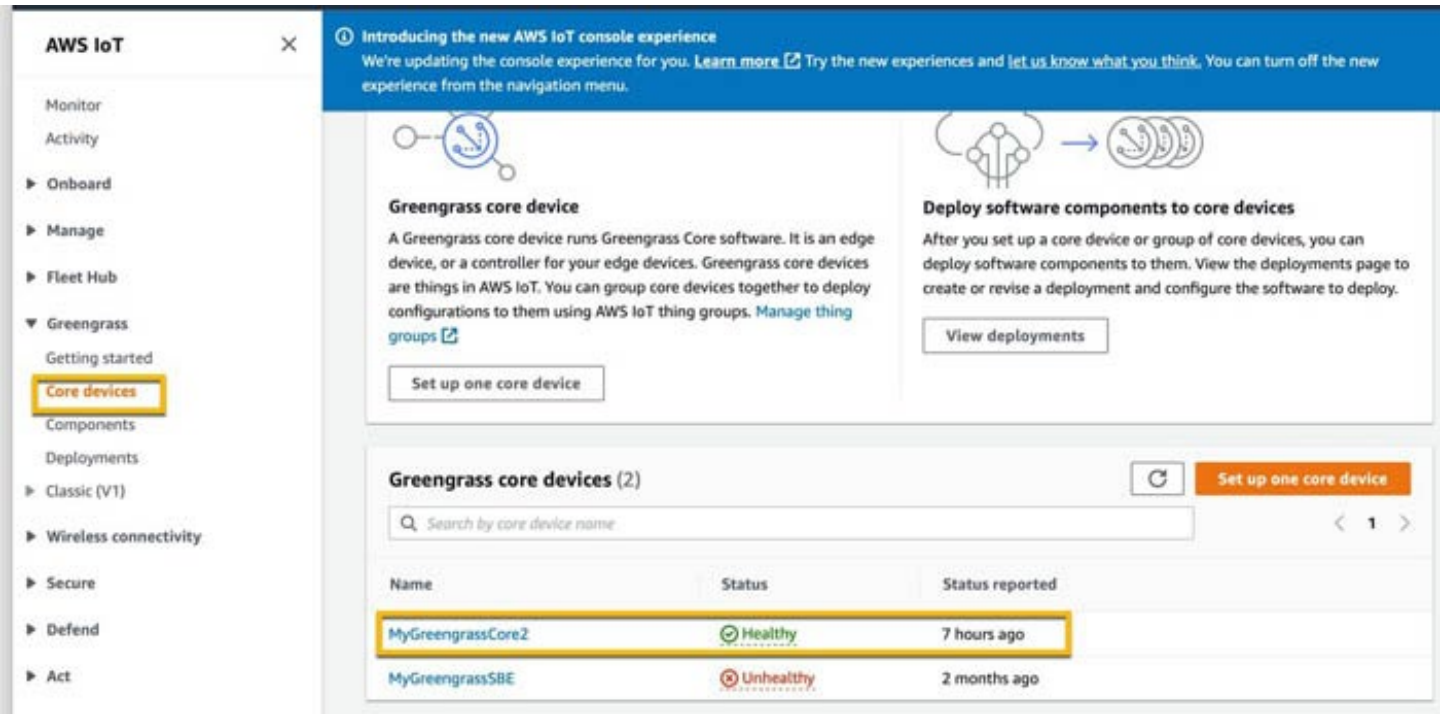
```
sudo -E java -Droot="/greengrass/v2" -Dlog.store=FILE
\ -jar ./MyGreengrassCore/lib/Greengrass.jar \ --aws-
region region \ --thing-name MyGreengrassCore \ --thing-
group-name MyGreengrassCoreGroup \ --tes-role-name
GreengrassV2TokenExchangeRole \ --tes-role-alias-name
GreengrassCoreTokenExchangeRoleAlias \ --component-default-user
ggc_user:ggc_group \ --provision true \ --setup-system-service
true \ --deploy-dev-tools true
```

We reference the --thing-name you chose here when we set up Edge Manager.

2. Run the following command to test your installation:

```
aws greengrassv2 help
```

3. On the AWS IoT console, validate the successfully registered Snowball Edge device with your AWS IoT Greengrass account.

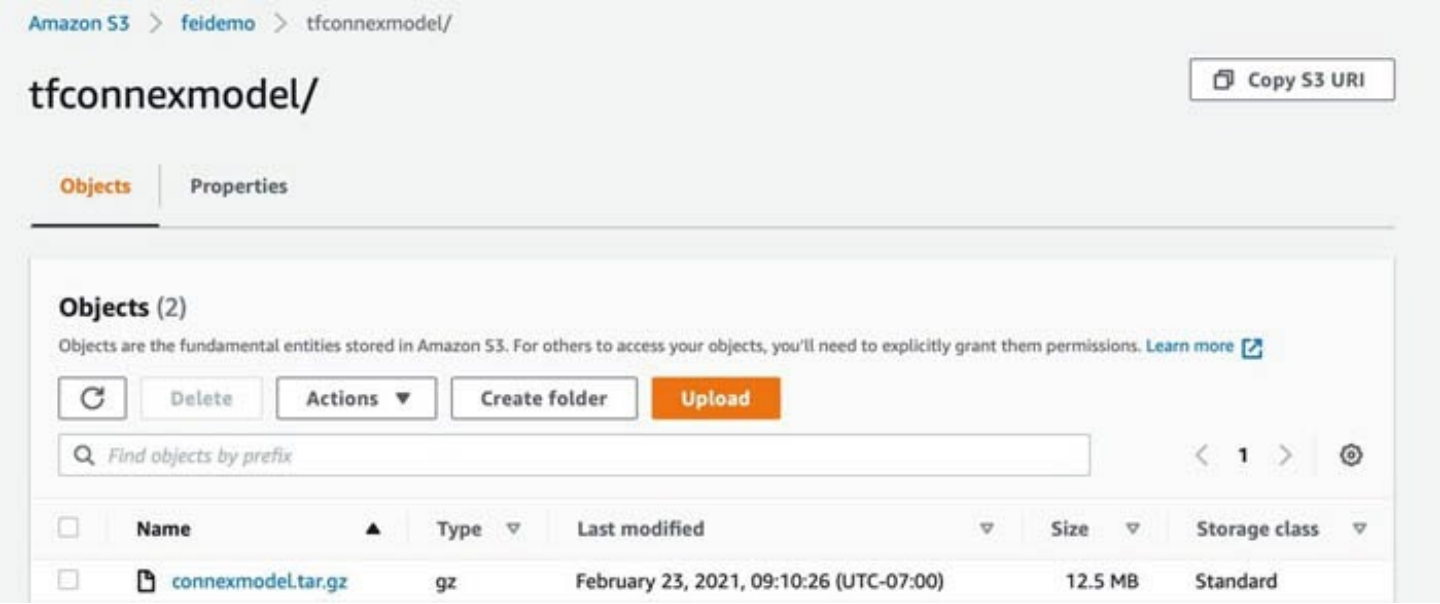


Optimize ML models with Edge Manager

We use Edge Manger to deploy and manage the model on Snowball Edge.

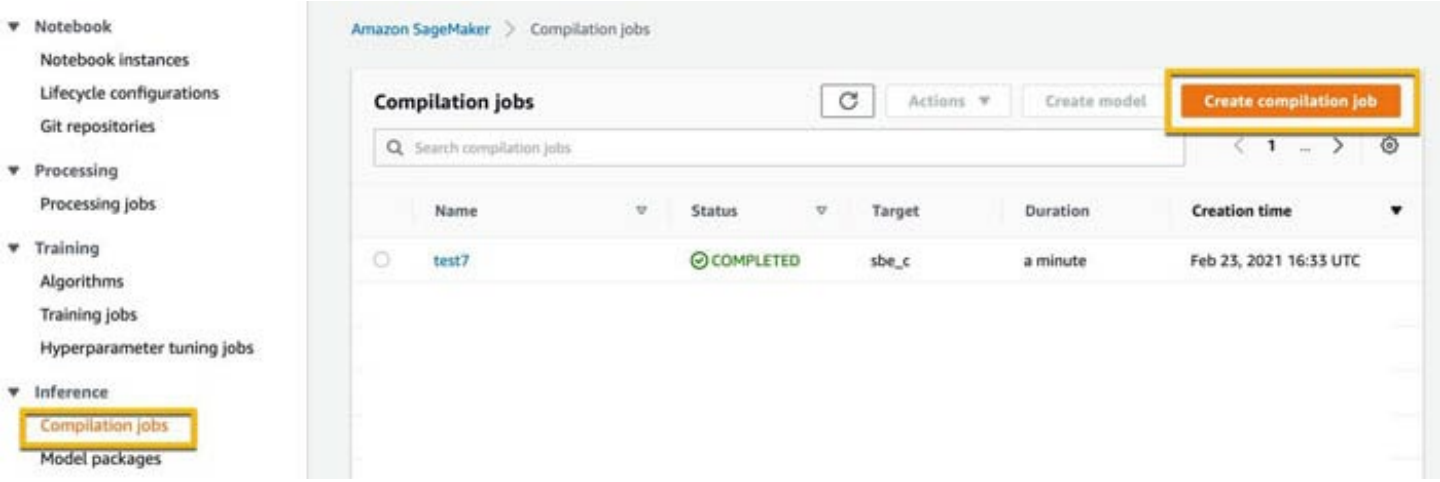
1. Install the Edge Manager agent on Snowball Edge using the latest AWS IoT Greengrass.
2. Train and store your ML model.

You can train your ML model using any framework of your choice and save it to an Amazon Simple Storage Service (Amazon S3) bucket. In the following screenshot, we use TensorFlow to train a multi-label model to classify connex and people in an image. The model used here is saved to an S3 bucket by first creating a .tar file.



After the model is saved (TensorFlow Lite in this case), you can start an [Amazon SageMaker Neo] compilation job of the model and optimize the ML model for Snowball Edge Compute (SBE_C).

3. On the SageMaker console, under **Inference** in the navigation pane, choose **Compilation jobs**.
4. Choose **Create compilation job**.



5. Give your job a name and create or use an existing role.

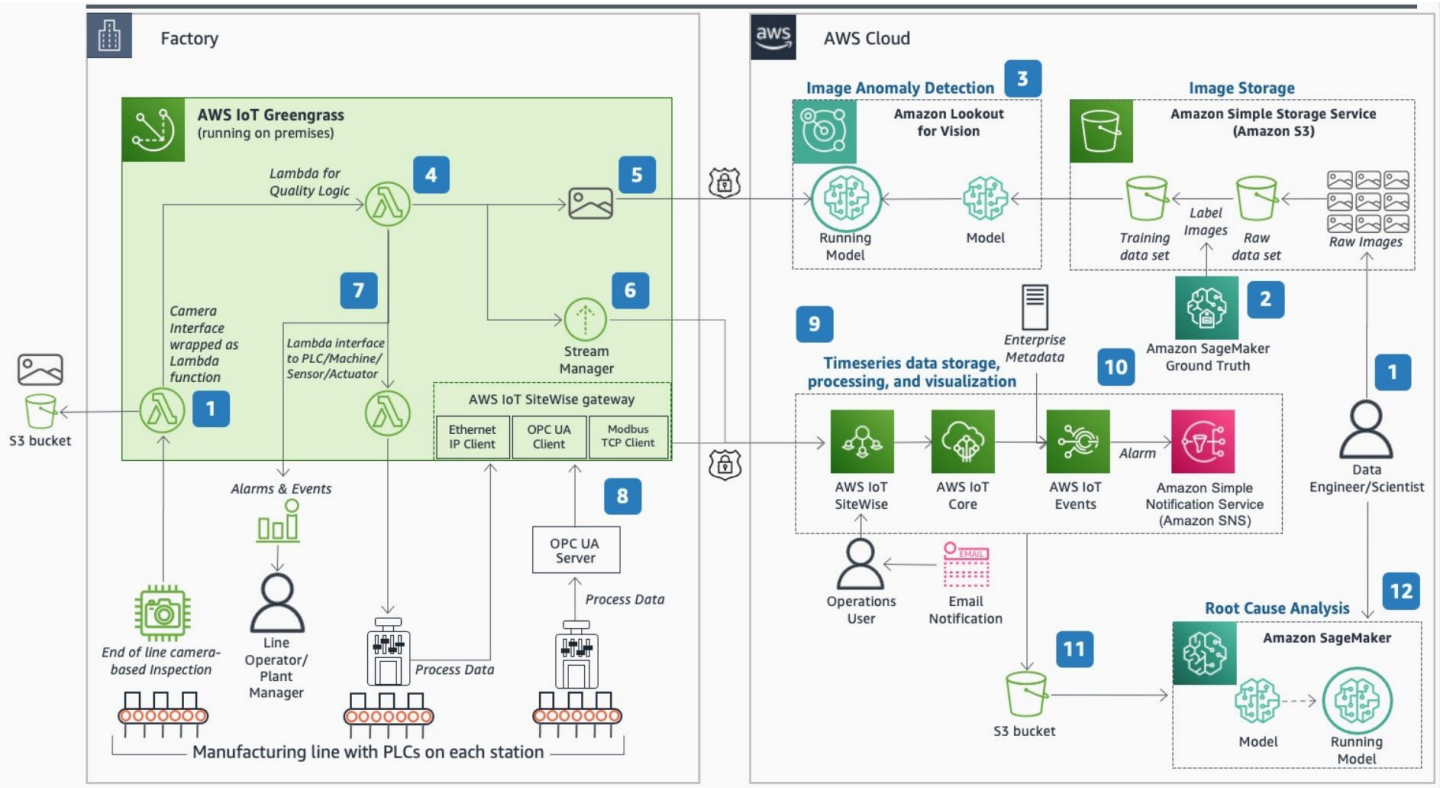
If you're creating a new [AWS Identity and Access Management (IAM)] role, ensure that SageMaker has access to the bucket in which the model is saved.

[Read full blog post online]

Using Computer Vision for Product Quality Analysis in Plants

Detect and act on product-defect-classification using AWS IoT and AI/ML services.

Use this architecture for camera-based, end-of-line quality inspection; defect-detection using image classification in the cloud; alert notifications; real-time actuation; and root cause analysis using process data and inferred vision results.



1. Auto-upload training images from the manufacturing line camera to **Amazon Simple Storage Service (Amazon S3)** using a **AWS Lambda** function running on **AWS IoT Greengrass**, or manually batch-upload training images to **Amazon S3**.
2. Use **Amazon SageMaker Ground Truth** to label training images.
3. Train a model using **Amazon Lookout For Vision (Lookout for Vision)** and deploy the trained model for running production inferences.
4. Feed live production images to **AWS Lambda** function on **AWS IoT Greengrass** to perform anomaly detection
5. Present camera image to **Lookout For Vision** for anomaly detection using inference API.
6. Feed inference metadata to **AWS IoT SiteWise** in the cloud for further processing via **AWS IoT Greengrass** stream manager.

7. Perform automated action on machine of concern *and/or* notify plant personnel from **AWS Lambda** function (using **AWS IoT Greengrass** connector for the **Amazon Simple Notification Service (Amazon SNS)**).
8. Ingest process data into **AWS IoT SiteWise** gateway running on **AWS IoT Greengrass** from machine/equipment using Open Platform Communications Unified Architecture (OPC UA) as the standard protocol. Modbus Transmission Control Protocol (TCP) and Ethernet IP are also natively supported, with the **AWS IoT SiteWise** gateway sending PLC data to cloud.
9. Compute Key Performance Indicator (KPI) metrics from process data in **AWS IoT SiteWise**. Create monitoring and KPI dashboards in **SiteWise Monitor** for operations user.
10. Create events from plant data and enterprise metadata by routing data to **AWS IoT Events** via **AWS IoT Core**, and send out email or text notifications to operations user using **Amazon SNS**.
11. Send process and vision inference data streams to **Amazon S3** for training root cause analysis models.
12. Train and run model inference to pinpoint root cause using **Amazon SageMaker**.

View reference architecture online

CASE STUDY

Coca-Cola İçecek Improves Operational Performance Using AWS IoT SiteWise

2021

Coca-Cola İçecek (CCI), one of the key bottlers in the Coca-Cola system, produces, distributes, and sells sparkling and still beverages of The Coca-Cola Company to 10 countries across Turkey, Pakistan, central Asia, and the Middle East, serving more than 400 million consumers. As part of CCI's digital strategy and vision, the company used Amazon Web Services (AWS) to transform its 26 bottling plants by building a digital twin.

CCI began this project in pursuit of its wider digital strategy. CCI's vision of investing in data and technology is to lead change and create value in the digital age through four key pillars: customer experience, asset optimization, people's experience, and innovation for growth. Asset optimization through digital technologies aims to create a continuous improvement engine to enable sustainability, efficiency, and quality for CCI's assets and operating model.

CCI wanted to use its digital twin to develop solutions that would improve asset optimization, enable sustainability, and avoid downtime, as well as bring more intelligence to its manufacturing processes. By using AWS services to improve communication between CCI operators and Internet of Things (IoT) devices, a network of physical objects embedded with software, sensors, and other technologies to exchange such IoT data, CCI created value for its business, community, and environment.

Unlocking Value from a Digital Twin in the Cloud

The purpose of the digital twin was to automate the shop floor and provide CCI with a holistic view of its manufacturing processes. With this information, CCI could improve line and asset utilization operations, and its operators could increase measurement accuracy and deploy preventative maintenance when necessary. "If we can locate failures and other maintenance issues before they happen, we can keep the plant up and running at all times and improve our utilization," says Suheyla Er Aksoy, asset optimization digital technology leader at CCI. A complete digital plant replica in the cloud would also help

"Working efficiently as a lean, agile team alongside AWS Professional Services helped accelerate our time to market."

Elif Ege

Digital Twin Product Manager

the company gain value through advanced analytics, artificial intelligence, and near-real-time asset monitoring.

To power the digital twin, CCI needed to connect data points from its assets and industrial IoT devices to the applications it needed to develop. Searching for a solution, the company turned to AWS. "We knew this was going to be a comprehensive solution that involved multiple aspects, such as IoT data modeling, processing, and performance," says Elif Ege, digital twin product manager at CCI. "AWS offered a flexible suite of services and provided the level of support we needed to build a scalable and configurable solution."

CCI chose AWS Professional Services—a global team of experts that helps businesses realize their desired outcomes on AWS—to build a scalable analytics solution using AWS IoT SiteWise, a managed service that makes it simple to collect, store, organize, and consume data from industrial equipment at scale to augment decision-making with data. As part of CCI's commitment to diversity, a diverse team spearheaded the project. "Diversity is critical to our sustainability, and inclusion is at the epicenter of our culture," says Aksoy. "We were able to form a highly competent and diverse leadership team to drive this product."

Building a Virtual Replica Using AWS IoT SiteWise

During the project's first phase, CCI used AWS Professional Services to build a solution for its clean-in-place (CIP) process, a critical sanitation process in the food and beverage industry that cleans interior surfaces of production lines and equipment without disassembly. "Our CIP is an everyday production process and crucial for industry hygiene and quality requirements," says Burcu Hacioglu, product owner for the digital twin product team at CCI. "We wanted to improve process efficiency by avoiding errors in time measurement, which could lead to CCI plants overusing energy and water."

CCI needed a way to collect and process enormous amounts of industrial data as well as build digital models of CIP assets and processes. To ingest equipment data for processing, CCI used AWS IoT SiteWise, which runs on AWS IoT Greengrass, an IoT open-source edge runtime and cloud service that helps developers build, deploy, and manage device software. AWS IoT SiteWise ingests a large amount of industrial data from CCI plants and enables operators to monitor processes at the edge using Grafana dashboards, an open-source analytics and interactive visualization web application. Amazon DynamoDB, a fast and flexible NoSQL database service for any scale, is used for state machine processing and calculates and compares operational data.

Using this solution, CCI operators receive access to digital representations of company assets and gain visibility into the CIP process in near real time. The IoT data is further enriched in the cloud using business rules and process-related information, which is powered by AWS services such as AWS IoT Analytics, a fully managed service that operationalizes analyses and scales automatically to support up to petabytes of IoT data. This enriched data is then stored in an industrial data lake and served to CCI operators and company stakeholders through Amazon Athena, an interactive query service that makes it easy to analyze such data on AWS using standard SQL.

CCI built the CIP digital solution in 2 months. "Working efficiently as a lean, agile team alongside AWS Professional Services helped

accelerate our time to market," Ege says. Within 4 months of deployment, CCI identified over 30 improvement opportunities that resulted in annual savings of 20 percent on electricity and 9 percent on water. Additionally, CCI estimates it will save 34 days of processing time per year using the digital twin solution, which will reduce overhead costs and further increase efficiency. "Between 2021 and 2023, we expect to regain 200 production days by implementing the digital twin of the CIP process," says Aksoy. "We also expect to optimize our filler-mixer machines, which have a high impact on our line utilization."

Scaling the Solution across the Company

By the end of February 2021, CCI deployed the CIP solution to four systems and expanded the digital twin to include filler-mixer machines in three of its plants. Using these solutions will help identify machine failures and improve production-line performance. To achieve its goals for asset optimization, human-machine collaboration, improved line utilization, and sustainability, CCI is working toward automating manufacturing processes in all 26 plants.

In the future, CCI will continue to transform its bottling operations at scale through near-real-time monitoring, process analytics, and artificial intelligence implementation. As CCI extends its digital twin across plants and lines, it will continue to develop machine learning models on AWS to drive intelligence on the shop floor. "With the digital twin, we engineered a data- and technology-based asset optimization approach," says Aksoy. "This accelerates our efforts toward autonomous smart manufacturing augmenting our teams with IoT data and AI."

About Coca-Cola İçecek

Coca-Cola İçecek is one of the key bottlers in the Coca-Cola system. Based in Istanbul, Turkey, the company produces and distributes beverages across Turkey, Pakistan, central Asia, and the Middle East, serving more than 400 million consumers.

Benefits of AWS

- Improved environmental sustainability of CIP process
- Saved 20% on energy annually
- Saved 9% on water annually
- Optimized CIP process time and cost performance
- Saves an estimated 34 days of process time annually

Read case study online

QUICK START

Industrial Machine Connectivity on AWS

Generate business value from your IIoT architecture

The Industrial Machine Connectivity (IMC) Quick Start helps you bring data from your Industrial Internet of Things (IIoT) assets to the Amazon Web Services (AWS) Cloud in a structured way. It's for developers and AWS Partners, regional and global systems integrators, independent software vendors, and original equipment manufacturers who want to generate immediate business value from an IIoT architecture.

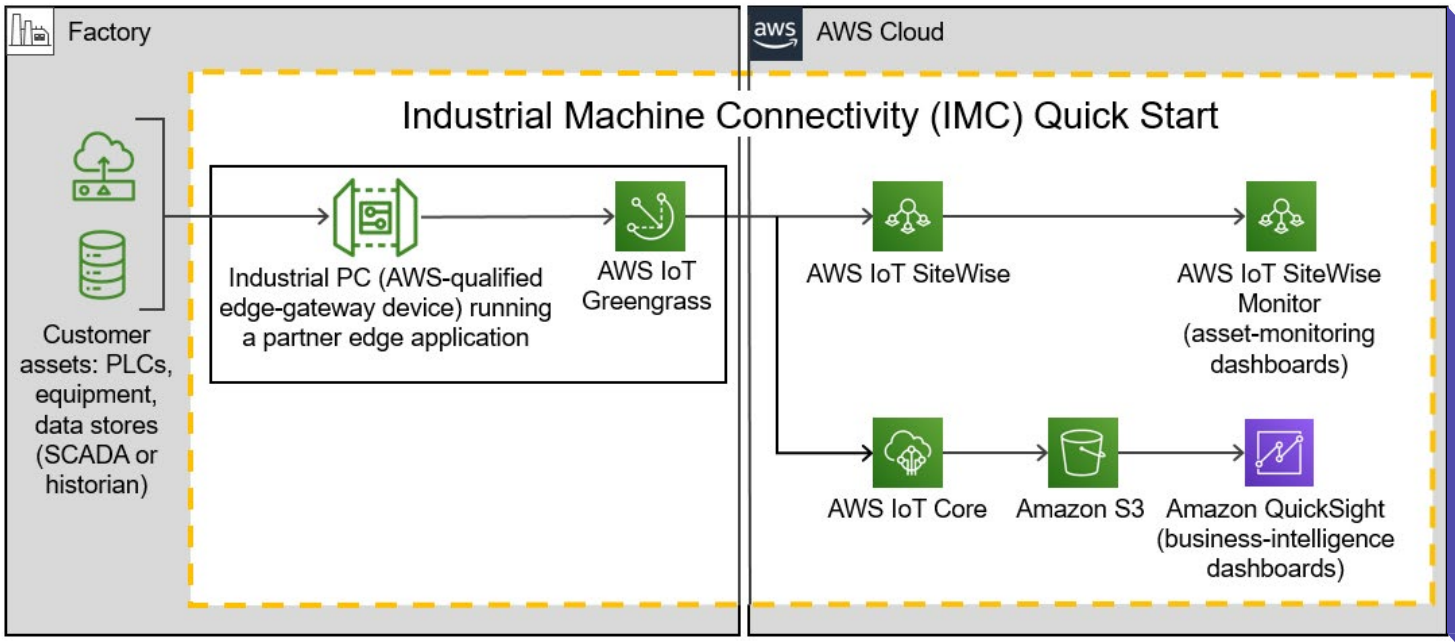
The primary objective of the IMC Quick Start is to help AWS Partners deliver a proof of concept that addresses a use case of high value to the customer. For example, the customer might want to start by visualizing near-real-time operational metrics and analyzing root causes when a line goes down. After a successful proof of concept, the partner and customer may work together to build out the production architecture to address other critical use cases.

The IMC architecture includes AWS managed IoT edge services and AWS-qualified edge hardware. You can use a range of programmable logic controllers (PLCs). And you can publish data over various protocols: HTTPS, MQTT (Message Queuing Telemetry Transport), and OPC Unified Architecture (UA). With this Quick Start, you can automate production rollout of a connected-factory architecture across multiple sites. You can organize, store, and manage your IIoT data in various ways:

- Create or transfer virtual assets.
- Create or transfer asset hierarchies.
- Create a time-series hot-data store.
- Transfer data from a historian or a SCADA (supervisory control and data acquisition) system.
- Archive cold data in Amazon Simple Storage Service (Amazon S3).

When you launch the IMC Quick Start, an AWS CloudFormation template automates the deployment of resources into your AWS account. You deploy this Quick Start in either virtual mode (for evaluation and training) or physical mode (for customer deployments). The mode you choose depends on whether your edge hardware is virtual or physical. The architecture for virtual edge hardware includes an Amazon Elastic Compute Cloud (Amazon EC2) instance. The architecture for physical edge hardware includes an industrial PC on the customer's premises. The mode determines the way you configure connectivity and security. All other cloud-based resources are largely the same for virtual and physical deployments.

This diagram shows a high-level view of a physical deployment. The dotted orange box outlines the IMC Quick Start's main components.



- In the factory:
  - AWS IoT Greengrass runs on an industrial PC (an AWS-qualified edge-gateway device). AWS IoT Greengrass ingests data from a partner edge application, such as Inductive Automation's Ignition or PTC's KEPServerEX.
  - The partner edge application translates the data from the customer assets—including PLCs, equipment, and data stores (SCADA or historian)—into industrial protocols.
- In the AWS Cloud:
  - AWS IoT SiteWise stores the metadata for the asset-model hierarchy of the industrial assets on the factory floor. It also contains a managed database for the time-series data generated by these assets.
    - After the hierarchy is defined in AWS IoT SiteWise, the partner edge application continuously ingests the asset data and transmits it to the AWS Cloud through a SiteWise connector within AWS IoT Greengrass.
    - AWS IoT SiteWise serves as the hot-storage tier for both time-series data and metadata. All this data, including the metadata, is accessible to applications that can generate business value from it.
    - The AWS IoT SiteWise Monitor feature enables you to build dashboards to visualize near-real-time time-series data stored in AWS IoT SiteWise's time-series database.
  - AWS IoT Core receives and routes MQTT messages either directly from the partner edge application or from the AWS IoT Greengrass core.
  - Amazon S3 can serve as a cold-storage tier for data.
  - Amazon QuickSight lets you build custom business-intelligence dashboards and visualizations for data stored in the S3 bucket.

View full quick start online

View deployment guide

BLOG

Automated Device Provisioning to AWS IoT Core Using 1NCE Global SIM

by Gaurav Gupta and Jan Sulaiman

Internet of Things (IoT) requires reliable and secure connectivity from wireless networks to share data. Connecting an IoT device to the cloud with long-range wireless technology that can operate reliably for up to 10 years on a single battery charge is becoming popular.

This is especially true in various high-scaling use cases like smart cities, smart meters, asset tracking, smart agriculture, smart building, and various other industrial applications.

AWS IoT Core offers a managed cloud services to ingest trillions of messages from billions of devices, and easily and securely interact with other AWS cloud services and other devices.

In high-scaling use cases, high-throughput standards like 5G-NR (>1Gbps) or LTE-Advanced (300Mbps) are an overkill. Instead, these remote and sometimes mobile applications need low-power, narrowband standards such as LTE Cat 1, LTE-M, and NB-IoT.

Such cellular Low Power Wide Area Network (LPWAN) technology supports data transfer in small, infrequent data packets ranging in size from 10-1000 bytes at speeds of < 375Kbps.

In this post, we'll describe how you can use the 1NCE IoT Connectivity Suite to take the complexity out of IoT projects and overcome the challenges of cellular IoT adoption.

1NCE is an AWS Advanced Technology Partner that offers managed connectivity services for low bandwidth IoT applications.

Current Challenges with Scaling Cellular IoT

Based upon a customer analysis conducted in March 2020 with more than 2,500 customers of 1NCE, it was identified that businesses often face three significant challenges while ramping up IoT solutions:

1. Device security and authentication, and how to manage secure device authentication across cloud and platform environments to allow automation.
2. Data protocol support for low-data IoT protocols like UDP or CoAP isn't readily available.
3. Device integration and management, especially the first setup and configuration of devices, requires additional effort to handle. Customers also lack the required staff with technical knowledge.

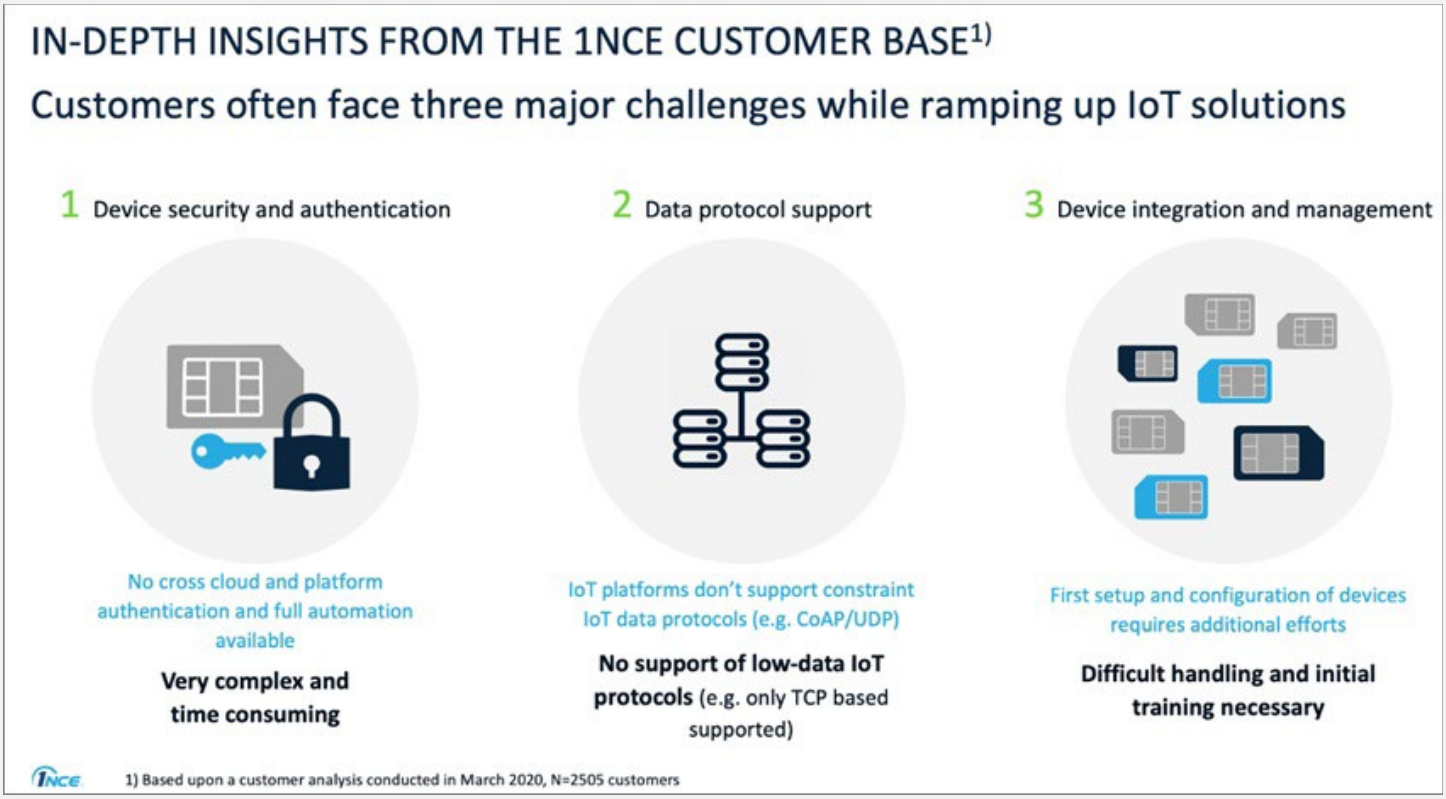


Figure 1 – 1NCE customer insights.

1NCE IoT Connectivity Suite

In August 2020, 1NCE launched a comprehensive set of services to connect cellular IoT devices to Amazon Web Services (AWS). 1NCE IoT Connectivity Suite is built using AWS services and is fully integrated with AWS IoT. The solution helps customers who need to deploy massive and scalable IoT solutions efficiently and fast.

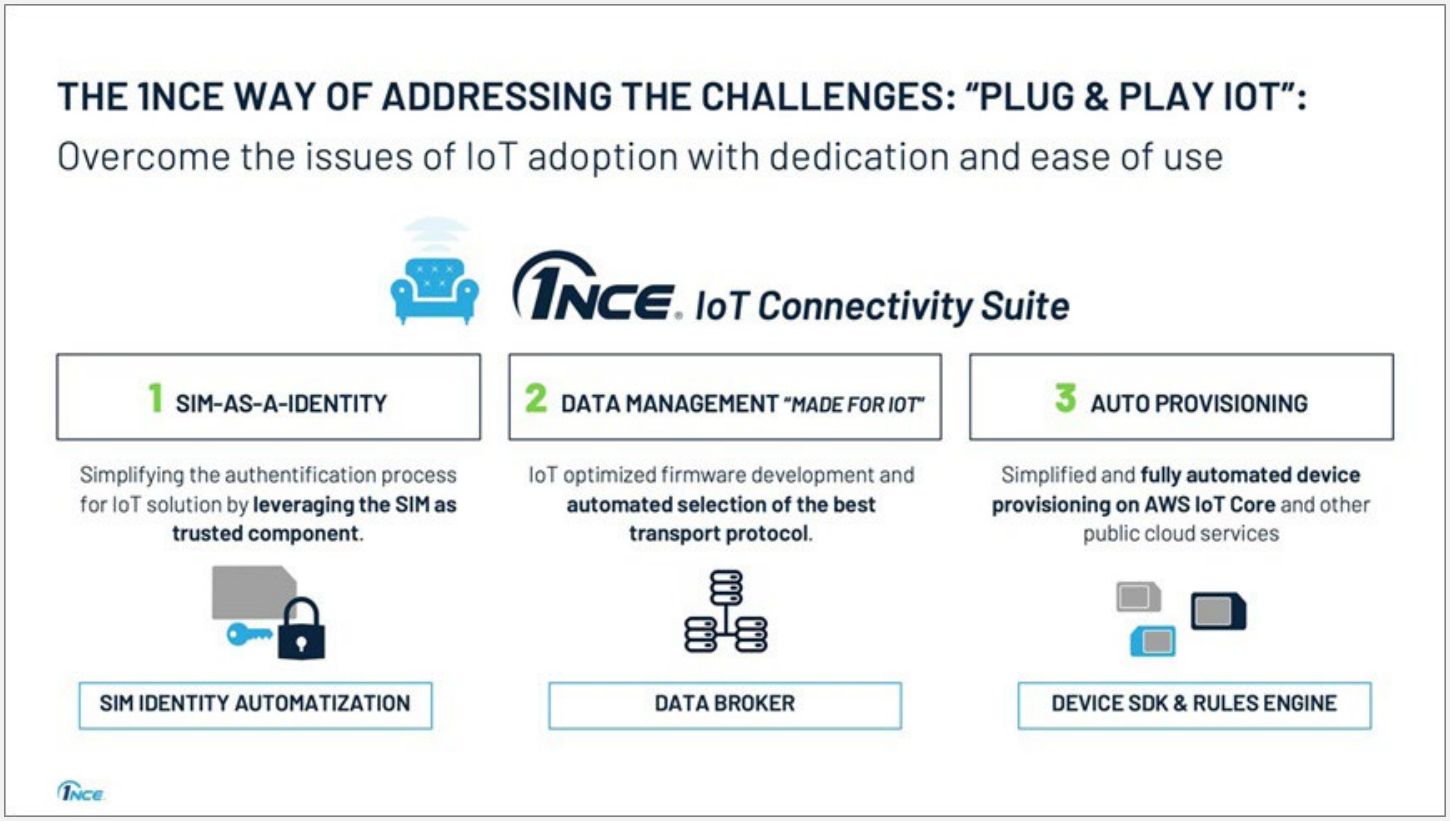


Figure 2 – 1NCE IoT Connectivity Suite.

The 1NCE solutions offers three main elements, starting with SIM-as-an-Identity to help you map the identities of individual IoT devices using a 1NCE SIM card. This allows you to automate the device onboarding to AWS IoT Core fully.

Second, real "made for IoT" data management enables you to use IoT-optimized transport protocols while still using all of AWS IoT Core's advanced capabilities.

Finally, a simplified and fully automated device setup and integration using prebuilt Blueprints and a simple-to-operate Data Broker connects IoT data to the right applications.

Customer Journey

You can order the 1NCE SIM with the 1NCE IoT Flat Rate from AWS Marketplace, or through the 1NCE Shop.

After receiving the SIM Cards and logging into the 1NCE Customer Portal, navigating to **Connectivity Suite > Account Connection > Via AWS Console** (shown in *Figure 3*) will provide you an AWS CloudFormation template to execute in your desired AWS region.

From there, **Connectivity Suite > Account Connection > Via CLI** provides you with the AWS Command Line Interface (CLI) commands for advanced users.

As soon as the CloudFormation template is successfully deployed, the SIM-as-an-Identity service will automatically start creating AWS IoT Things for all of your SIM Cards.

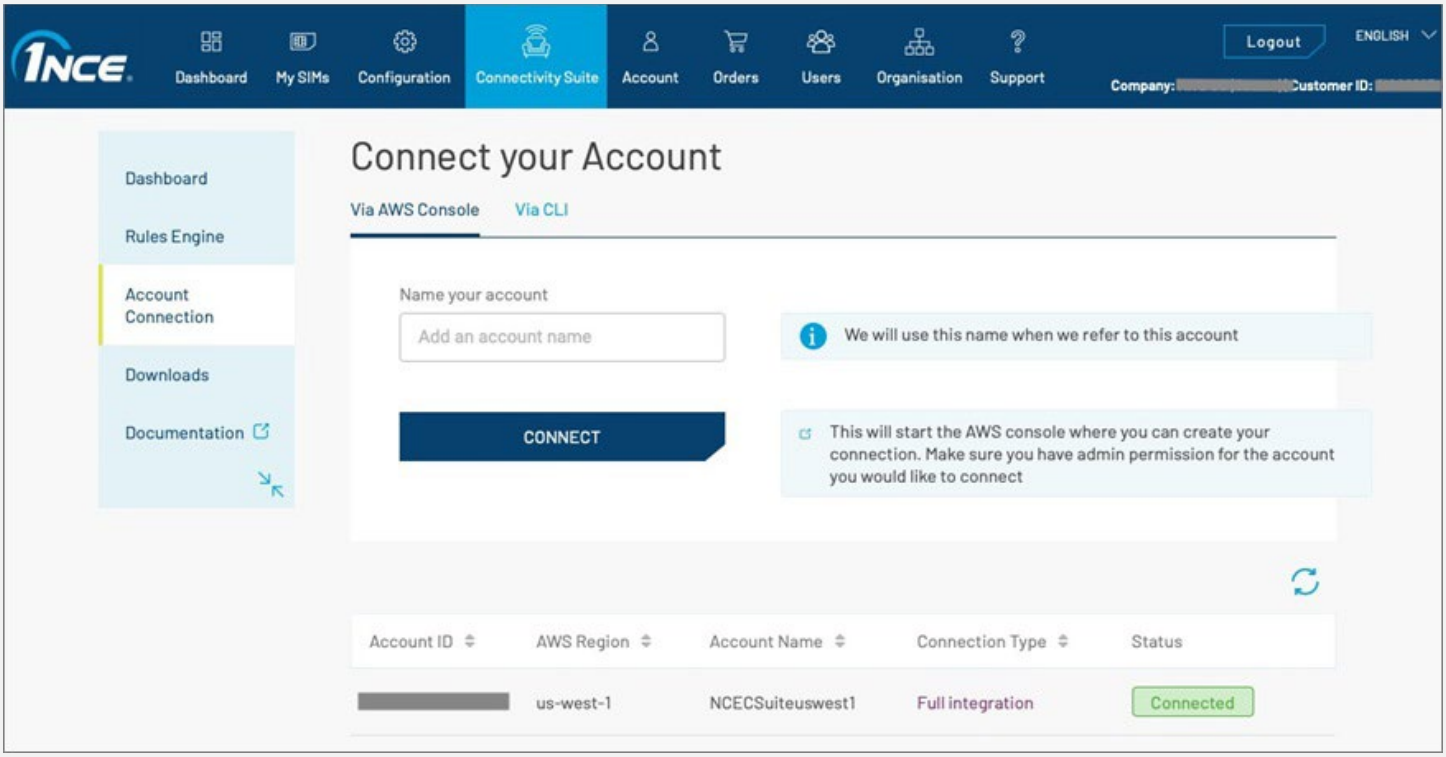


Figure 3 – AWS IoT integration with AWS CloudFormation.

You are now ready to insert the 1NCE SIM into your devices. The fastest interaction with all of the provided services is using one of the 1NCE Blueprints SDK provided for different devices; for example, the Blueprint for PyCOM GPy.

The Blueprints enable bootstrapping the devices when attached to the network, and automatically deliver an AWS IoT Core X.509 certificate, key and AWS IoT Core endpoints.

Certificate files are stored in the flash memory of the device and are reloaded during each device's power cycle. This process ensures a potential threat cannot hijack the certificate and key. After the bootstrap, devices can start sending data directly to AWS IoT Core.

For cases when you are not relying on MQTT, but instead use UDP or CoAP, the devices' bootstrapping and downloading of an X.509 certificate are unnecessary. Here the SIM Cards, as part of the secured 1NCE Mobile Core Network, act as a trusted element and serve as the authenticator.

Read full blog online

SOLUTION

Machine to Cloud Connectivity Framework

What does this AWS Solutions Implementation do?

This solution implementation provides operational technology (OT) managers with secure machine and industrial equipment connectivity to the AWS Cloud. This solution automatically deploys and configures AWS IoT Greengrass and provides integration with AWS IoT SiteWise. Machine and industrial asset telemetry data can then be published to AWS IoT SiteWise and Amazon Simple Storage Service (Amazon S3), populating an industrial data lake with machine telemetry to support insights through visualization, analytics, and machine learning. This solution provides support for OPC Data Access (OPC-DA) and OPC Unified Architecture (OPC-UA) protocols.

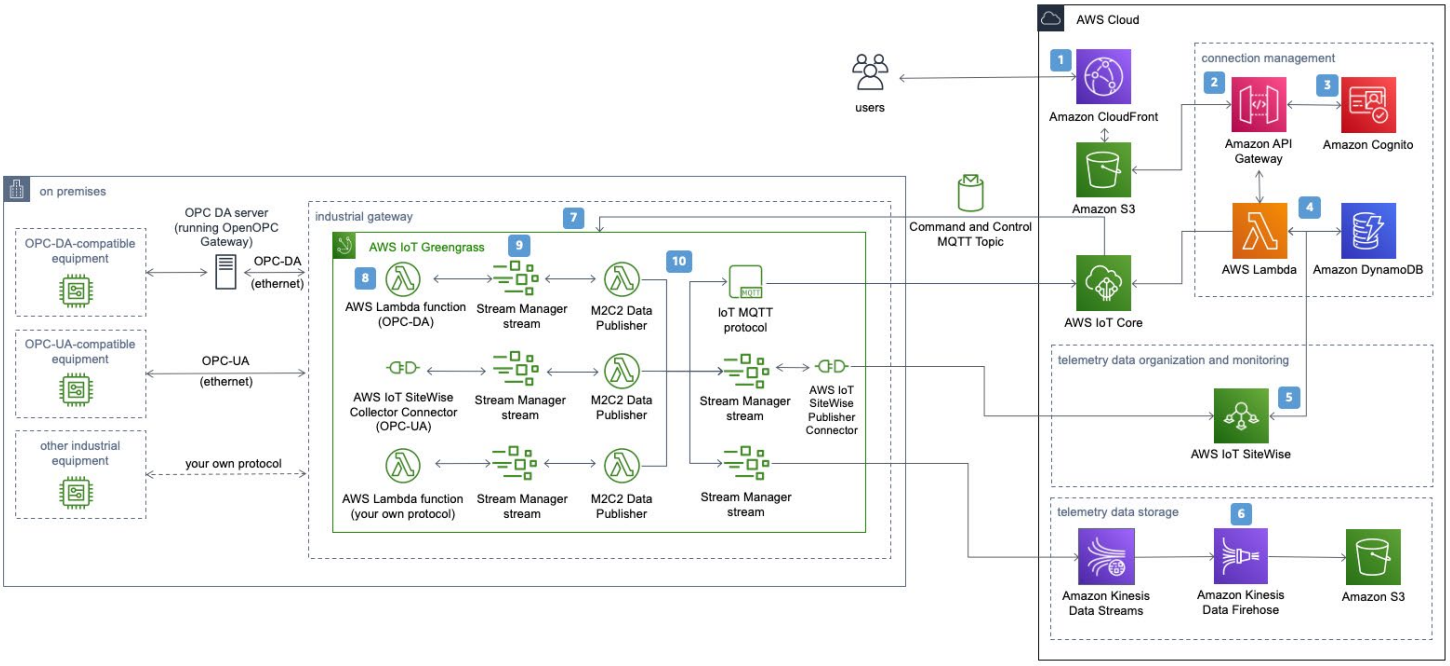
View related content

Benefits

- **Automate deployment of AWS IoT Greengrass**
  Automate the creation and configuration of AWS IoT Greengrass resources on edge devices to simplify implementation, helping them get set up and running quickly.
- **Customize with different secure protocols**
  Robust data ingestion from sources using the AWS IoT SiteWise Connector with either the OPC DA or OPC UA protocol. Use this solution as a reference to build secure cloud connectivity for additional industrial protocols based on the requirements of your factory equipment.
- **Publish data to multiple destinations in AWS Cloud**
  Send data to an AWS IoT Core topic or AWS IoT SiteWise for analytics and monitoring; or store your data in an industrial data lake using Amazon S3 to leverage additional analytics and machine learning services. The creation and management of connections between the industrial data sources and the AWS services is provided by a web user interface included with the solution.

AWS Solutions Implementation overview

The diagram below presents the architecture you can automatically deploy using this solution's implementation guide and accompanying AWS CloudFormation template.



Machine to Cloud Connectivity Framework architecture

The AWS CloudFormation template deploys the following infrastructure:

1. An Amazon CloudFront user interface that deploys into an Amazon Simple Storage Service (Amazon S3) bucket configured for web hosting.
2. An Amazon API Gateway API provides the user interface for client requests.
3. An Amazon Cognito user pool authenticates the API requests.
4. AWS Lambda functions power the user interface, as well as the configuration and deployment mechanism for AWS IoT Greengrass and AWS IoT SiteWise gateway resources. These Lambda functions send messages to the *m2c2/job/<connectionName>* AWS IoT topic (AWS IoT Core), then AWS IoT Greengrass subscribes to the messages. Amazon DynamoDB tables store the connection metadata.
5. An AWS IoT SiteWise gateway configuration for any OPC UA data sources.
6. An Amazon Kinesis Data Streams data stream, Amazon Kinesis Data Firehose, and an Amazon S3 bucket to store telemetry data.
7. AWS IoT Greengrass is installed and used on an on-premises industrial gateway to run protocol connector Lambda functions to connect and read telemetry data from your OPC UA and OPC DA servers.
8. Lambda functions are deployed onto AWS IoT Greengrass Core software on the industrial gateway to connect to the servers and to send the data to the configured destination(s).
9. Lambda functions that collect the telemetry data write to AWS IoT Greengrass stream manager streams. The publisher Lambda functions read from the streams.
10. Publisher Lambda functions forward the data to the appropriate endpoint.

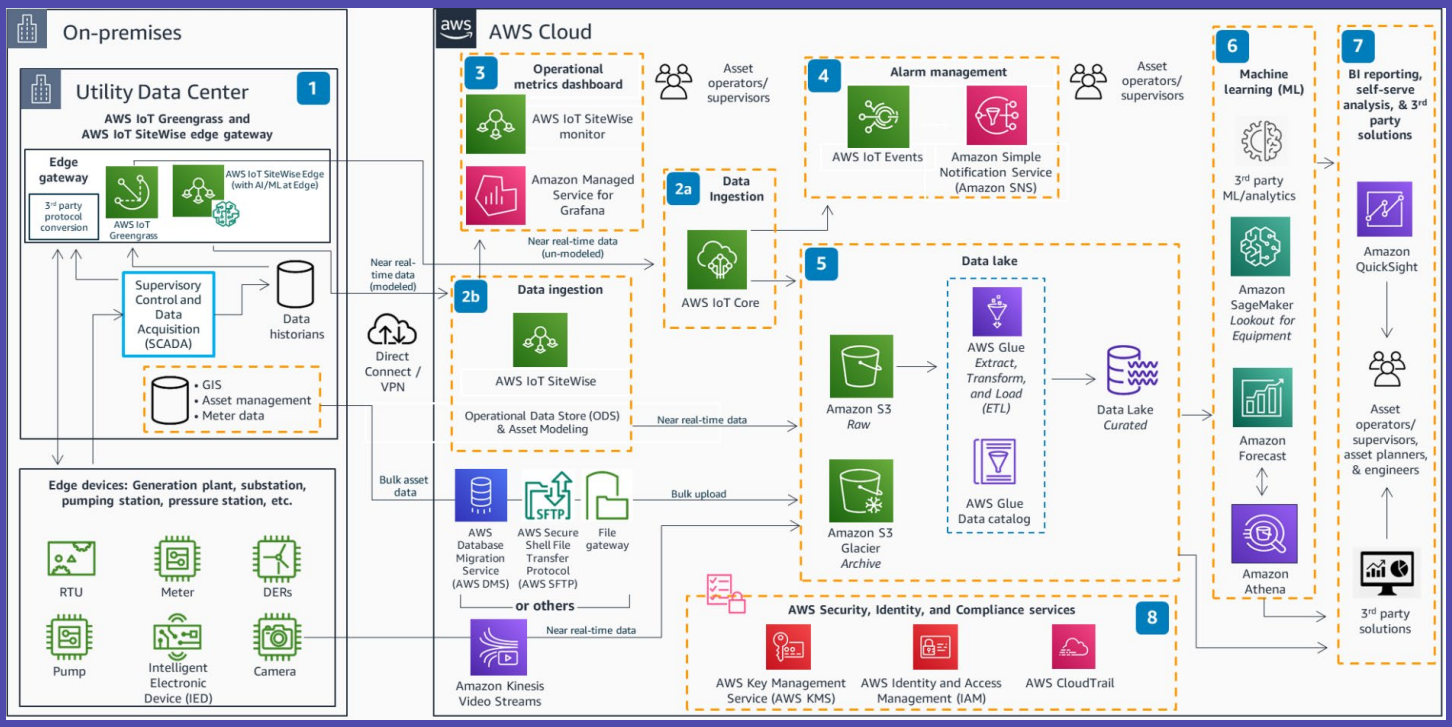
View implementation guide online

View solution online

REFERENCE ARCHITECTURE

Predictive Equipment Health for Utilities

Build a modern, end-to-end, field-to-cloud solution for ingestion of near-real-data from utility assets, devices, and a multitude of common utility systems using AWS services. Use analytics and machine learning services, operating on a data lake, to derive insights and predict the health and longevity of your assets.



1. Data sources on the edge utilize **AWS IoT Greengrass** and **AWS IoT SiteWise** edge for seamless connectivity and data preparation from Supervisory Control and Data Acquisition (SCADA), 3rd party protocol converters, Geographical Information System (GIS), data historians, and edge devices. **AWS IoT SiteWise** runs compiled machine learning (ML) models for local inference and actioning.

2a. Data is ingested directly from asset to **AWS IoT Core**, for non-asset modelled data.

2b. Data is ingested at scale with asset modelling in **AWS IoT SiteWise**

3. Real-time operational dashboard of *data* (critical asset performance metrics) via **AWS IoT SiteWise** monitor or **Amazon Managed Service for Grafana (AMG)**.
4. Build detector models in **AWS IoT Events** to continuously monitor the state of assets and issue immediate email and SMS alerts to operational staff via **Amazon Simple Notification Service (Amazon SNS)**.

5. **Amazon Simple Storage Service (Amazon S3)** serves as the data lake and *single version of truth* for all consumers. **AWS Glue** performs ETL functions and builds the data catalog. Infrequently-accessed data is moved to **Amazon S3 Glacier** for cost-effective archival.
6. Curated data from the data lake is utilized by Amazon AI/ML services (e.g. **Amazon SageMaker** and **Amazon Forecast**) or 3rd party ML services for predictive health analysis and assessment. The results can be readily consumed by asset owners and/or 3rd party asset applications.
7. Detailed Business Intelligence (BI) reporting occurs via **Amazon QuickSight** and 3rd party solutions (GIS, Asset Management, and Tableau).
8. All communication is fully secured, traceable, authenticated, and encrypted by AWS Security, Identity, and Compliance services.

View reference architecture online

BLOG

Autonomous vehicle data collection with AWS Snowcone and AWS IoT Greengrass

by Thomas Kriechbaumer

Self-driving and self-flying vehicles — autonomous cars, airplanes, and drones — require vast amounts of data to fulfill their promise of a safe mode of transportation for goods and people. Connected vehicles and the Internet of Things (IoT) have a strong influence on the way we collect and process low-bandwidth telemetry data, in addition to high-bandwidth sensor data for autonomy. Telemetry data must be ingested, analyzed, and acted upon in near-real time within seconds of the event happening. In contrast, cameras, lidars, and radars together can produce tens and hundreds of terabytes of data per hour, typically used for offline processing and machine learning.

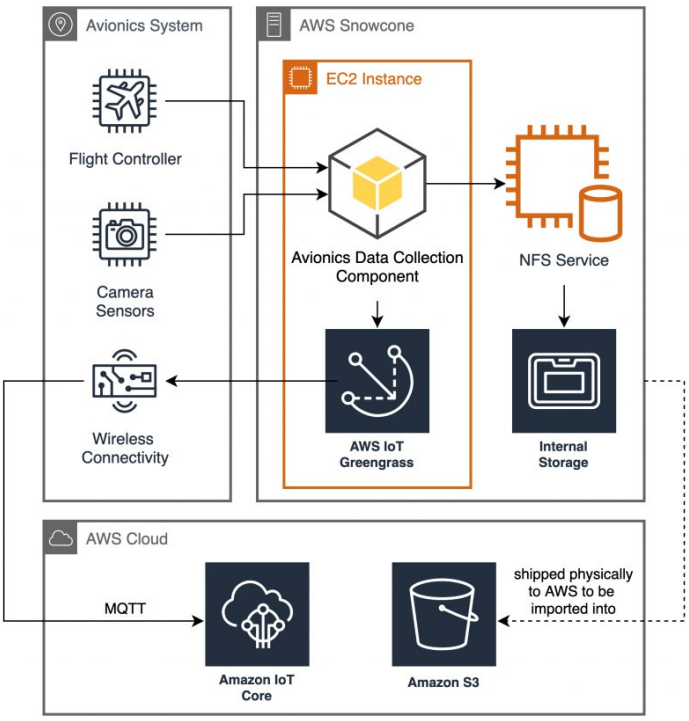
Common system architecture patterns for vehicles, airborne and ground-based alike, differentiate between safety-critical and non-safety-critical workloads. Autopilot, engine control, and fuel management are common safety-critical tasks, which have hard real-time requirements. Collecting and storing high-bandwidth sensor data can be classified as non-critical when other systems do not depend on the availability of these data streams during operation.

In this blog post, I describe an autonomous vehicle, such as a quadcopter or helicopter, with a split-responsibility system. The first, an avionics system, is in charge of safety-critical tasks that validates the correct operation of the vehicle within its design envelope. The second system (outside of the critical path) sends telemetry for real-time tracking and bulk data collection for research and development purposes. I demonstrate using AWS Snowcone in place of this second system to tap into a vehicle's sensor stream, transform real-time information into telemetry events, and persist all data for later ingestion into Amazon S3. Together with AWS IoT and AWS IoT Greengrass, we provide an edge computing environment with over-the-air update capabilities. Offloading these non-safety-critical tasks to a dedicated AWS Snowcone device allows the vehicle to operate securely and safely, while also providing advanced data collection and processing functionality for real-time fleet management and offline data analysis on vast amounts of high-bandwidth sensor data.

Walkthrough

First, I show how to configure a Snowcone to start and bootstrap a local Amazon EC2 instance, and how to provision this instance to run AWS IoT Greengrass 2.0 for edge computing tasks. Together with AWS IoT Core and a custom AWS IoT Greengrass component, I demonstrate how to send real-time vehicle location and health tracking telemetry events to the cloud, and how to ingest high-bandwidth sensor data into a Snowcone for offline transfer to an S3 bucket.

The following architecture diagram depicts the individual parts of the solution:



This blog post includes a demo application packaged as an AWS IoT Greengrass component: *AvionicsDataCollection* is a Python-based application that demonstrates how to read avionics state information and generates a telemetry event with live location data and engine metrics. These events are periodically sent to AWS IoT Core. Camera data, or any other high-bandwidth sensor data streams, are queried periodically and persisted to the internal Snowcone storage attached via NFS. This data will later be ingested into Amazon S3 when the device is shipped back to AWS. Your vehicle, sensors, and avionics system will have dedicated interfaces to extract information in a non-safety-critical environment.

I walk through the following main steps of this solution:

1. Configure IAM, security, and permissions-related configuration for AWS IoT resources in your AWS account.
2. Configuring the avionics system before first flight.
3. Automate provisioning of a local EC2 instance on a Snowcone device.
4. Install and provision AWS IoT Greengrass inside a local EC2 instance.
5. Deploy a new AWS IoT Greengrass component over-the-air with a demo application.

Once these steps are done, you will have a Snowcone device with AWS IoT Greengrass running inside an EC2 instance. The AWS IoT Greengrass core device automatically deployed the *AvionicsDataCollection* application, which sends periodic telemetry events to AWS IoT Core. It also writes high-bandwidth data into the Snowcone for later ingestion into Amazon S3. All steps are fully automated and require no human interaction after the initial configuration.

Read full blog online

SOLUTION

AWS Connected Vehicle Solution

What does this AWS Solutions Implementation do?

Amazon Web Services (AWS) enables automotive manufacturers and suppliers to build serverless IoT applications that gather, process, analyze, and act on connected vehicle data, without having to manage any infrastructure. With AWS IoT, customers can connect vehicles and devices to the AWS Cloud securely, with low latency and with low overhead.

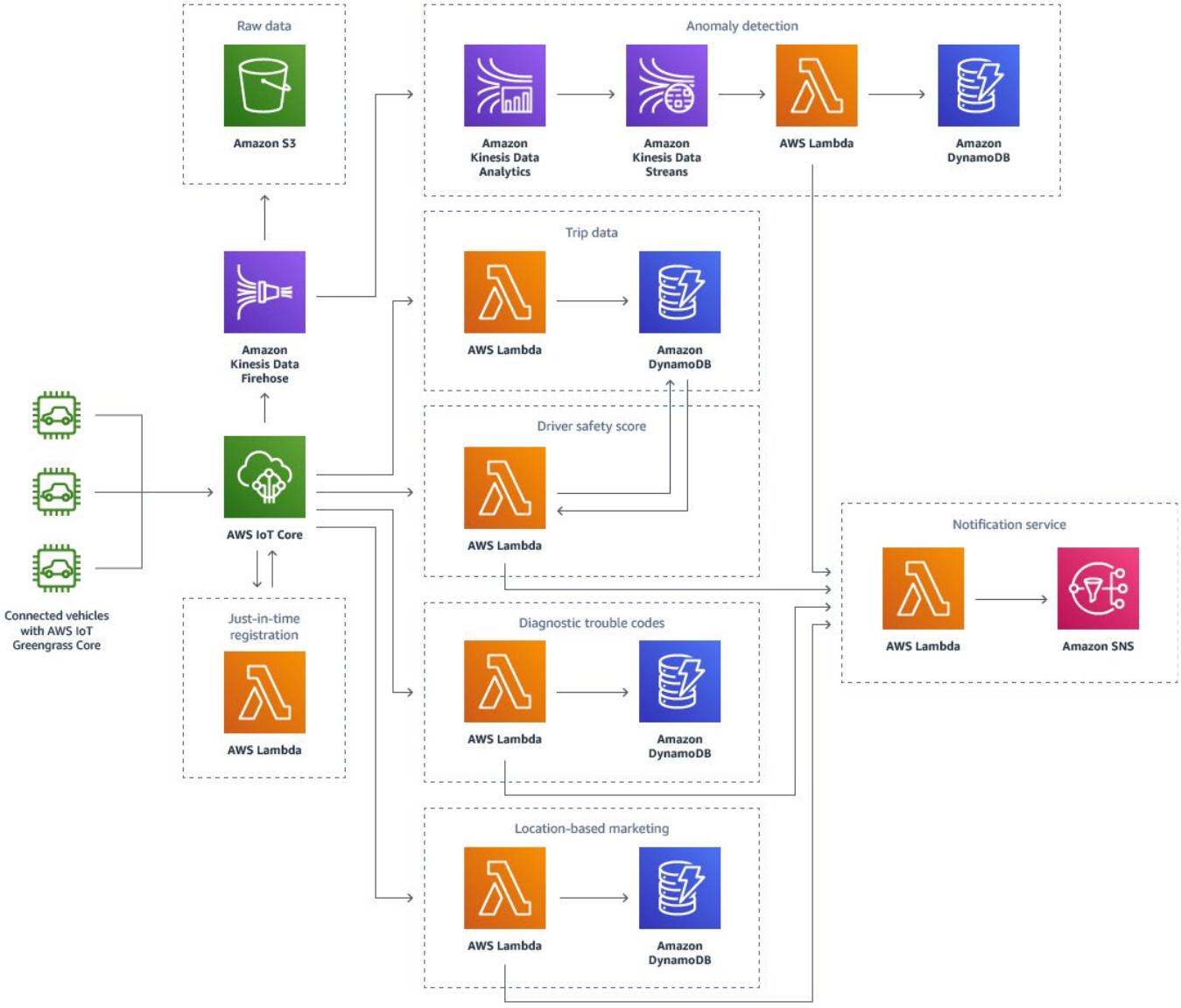
To help customers more easily develop and deploy a wide range of innovative connected vehicle services, AWS offers a connected vehicle solution that provides secure vehicle connectivity to the AWS Cloud, and a framework that helps customers integrate AWS IoT and AWS Greengrass into the Automotive Grade Linux (AGL) software stack.

Version 2.1.1 of the solution uses the most up-to-date Node.js runtime. Version 2.0 uses the Node.js 8.10 runtime, which reaches end-of-life on December 31, 2019. To upgrade to version 2.1.1, you can update the stack. For more information, see the deployment guide.

AWS Solutions Implementation overview

The connected vehicle solution includes capabilities for local computing within vehicles, sophisticated event rules, and data processing and storage. The solution is designed to provide a framework for connected vehicle services, allowing you to focus on extending the solution's functionality rather than managing the underlying infrastructure operations. You can build upon this framework to address a variety of use cases such as voice interaction, navigation and other location-based services, remote vehicle diagnostics and health monitoring, predictive analytics and required maintenance alerts, media streaming services, vehicle safety and security services, head unit applications, and mobile applications.

The diagram below presents the components and functionality you can build using the solution implementation guide and accompanying AWS CloudFormation template.



AWS Connected Vehicle Solution architecture

When AWS IoT receives a message, it authenticates and authorizes the message and the Rules Engine executes the appropriate rule on the message, which routes the message to the appropriate backend application.

An AWS IoT rule sends telematics data to an Amazon Kinesis Data Firehose delivery stream, which encrypts and streams raw vehicle telematics data to an Amazon S3 bucket. If an Amazon Kinesis Data Analytics application detects an anomaly, the record is sent to Amazon Kinesis Data Streams, which invokes an AWS Lambda function that parses the record, stores it in an Amazon DynamoDB table, and triggers an Amazon Simple Notification Service (Amazon SNS) notification to users.

The trip data AWS IoT rule invokes an AWS Lambda function that processes vehicle telematics data during a trip and stores it in a DynamoDB table.

The driver safety score AWS IoT rule detects the end of a trip and invokes an AWS Lambda function that processes aggregate trip data to generate a driver's safety score, trigger an Amazon SNS notification to the driver, and add the score to the trip data table.

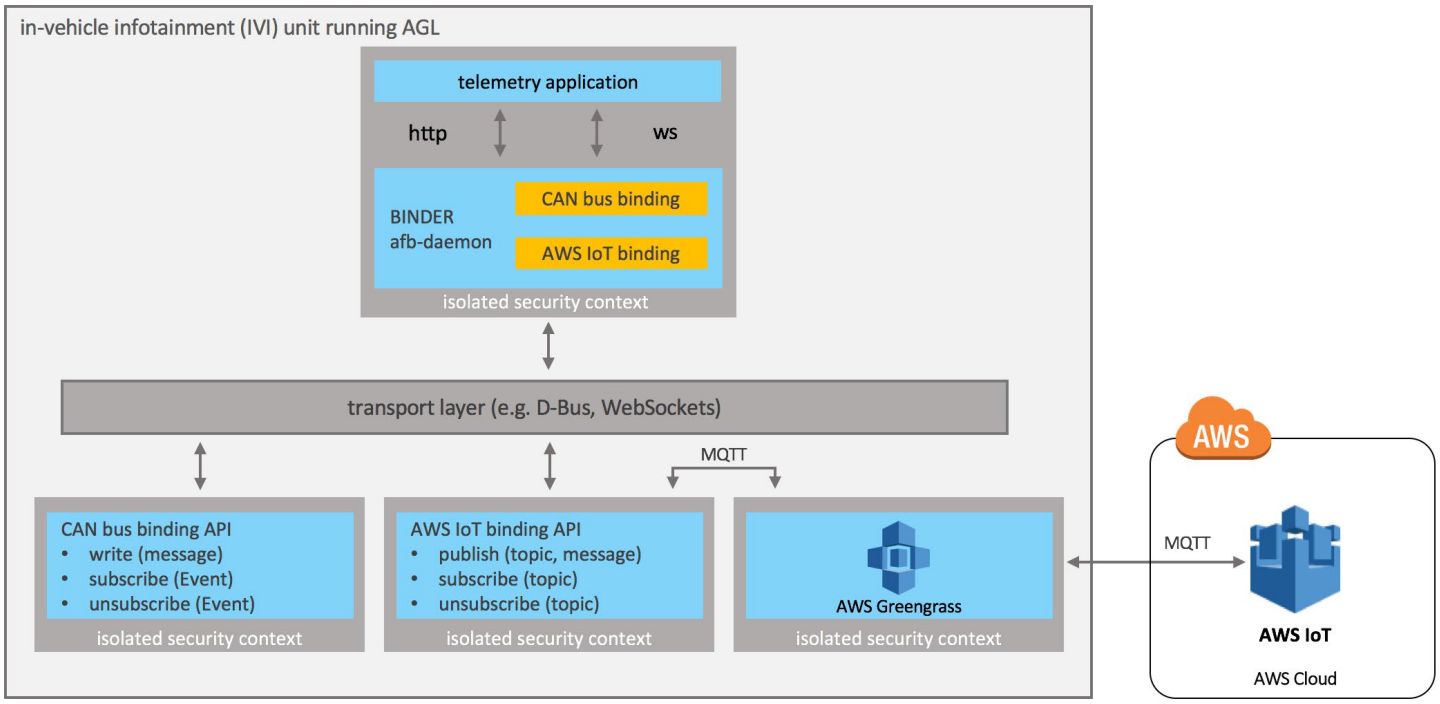
The diagnostic trouble code AWS IoT rule detects diagnostic trouble codes in the IoT topic and invokes Lambda functions that store the trouble code in a DynamoDB table, translate the trouble code into layman's terms, and trigger an Amazon SNS notification to the user.

The location-based marketing AWS IoT rule detects the location of the vehicle and invokes a Lambda function that determines whether the vehicle is near a point of interest. When the vehicle is near a point of interest, the function logs the location in a DynamoDB table and triggers an Amazon SNS notification to the user that contains an advertisement.

AWS IoT Framework for AGL

AWS IoT Framework for Automotive Grade Linux (AGL) helps you integrate AWS IoT and AWS Greengrass into the AGL software stack. The framework consists of AWS Greengrass Core and an AWS IoT binding service built using the AGL Application Framework and the AWS IoT Device SDK.

The diagram below shows how an application running on AGL can send telemetry data to AWS IoT using this framework.

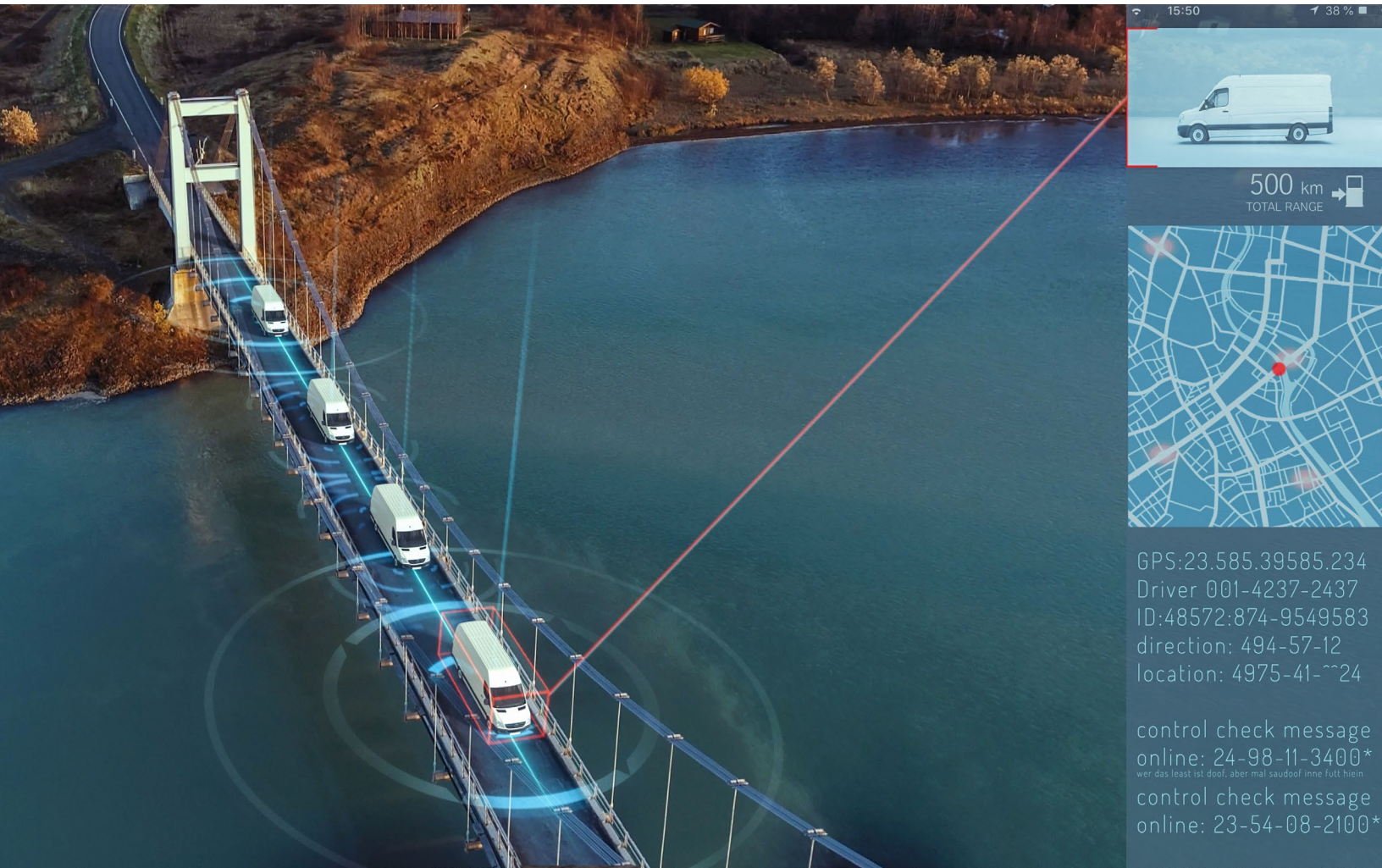


Features

- **Build an AGL image for the AWS IoT framework**
  With the steps outlined in the GitHub repository, you can build an AGL image for the AWS IoT framework. The framework includes all the components necessary to integrate AWS IoT and Greengrass into the AGL software stack.
- **Securely publish and receive messages**
  You can securely publish and receive messages such as vehicle telemetry between your applications running AGL and the AWS Cloud through AWS Greengrass Core using APIs that conform to the AGL security framework.

View solution online

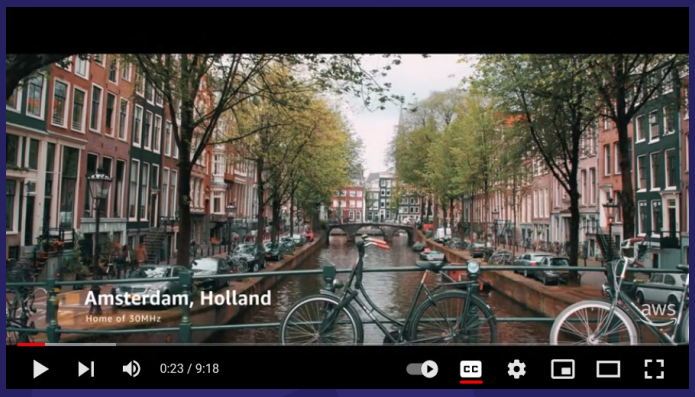
View implementation guide



VIDEOS

30MHz: Building A Smart Agriculture Solution For Indoor Farms And Greenhouses On AWS

30MHz has built a smart agriculture solution which provides growers with real-time remote crop monitoring solutions to help them optimize irrigation and ventilation, prevent disease or sunscald, improve pest management or predict shelf life.



Evolving at the Edge with the AWS Snow Family

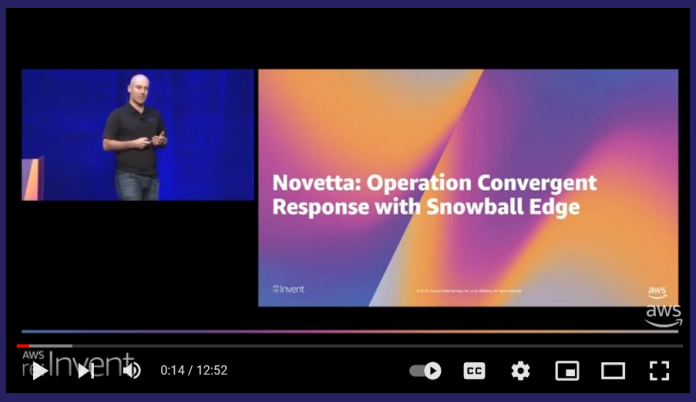
Many organizations have a whole new set of applications that are driving requirements for increased capabilities and performance at the edge of the cloud, or even beyond the edge of the network.

IoT All the Things | S3 E5 | Data Residency at the Edge: AWS Outposts Inside Out

AWS IoT experts meet AWS Outposts experts in the season finale of IoT All the Things. Find out how Outposts supports on-premises applications with low latency and local data processing requirements. And in a special segment, learn how you can run AWS Outposts with services like AWS IoT Greengrass.

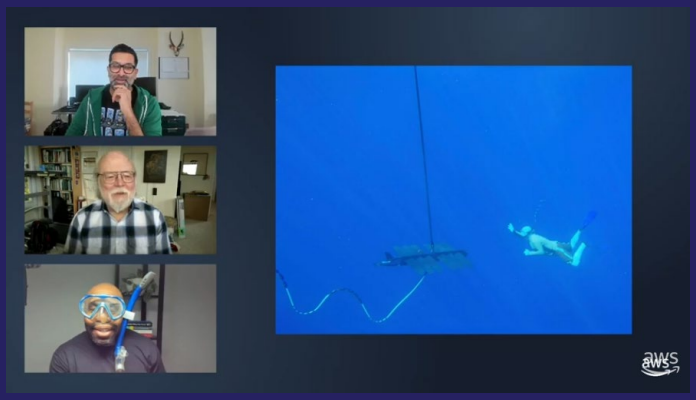
Orangetheory Fitness: Taking a Data-Driven Approach to Improving Health and Wellness (Special)

In this special edition of This is my Architecture, join builder Liz Dennett as she explores how Orangetheory has been able to scale it's studio based workouts and signature IoT hardware to create customized metrics to inspire and empower members.



Data Migration and Edge Computing with the AWS Snow Family

Learn how Novetta uses edge computing on AWS Snowball Edge to collect, analyze, and visualize data from IoT sensors and cameras in disaster response efforts. The Novetta solution keeps track of first responders and critical equipment in disaster zones. This talk describes Novetta's work with the AWS Disaster Response team and Verizon for Operation Convergent Response.



IoT All the Things | S3 E1 | All in with James Gosling: Behind the Scenes with AWS IoT Greengrass V2

This season premiere showcases a chat with James Gosling, Distinguished Engineer, taking you through the latest release of AWS IoT Greengrass. AWS IoT Greengrass V2 is an open source Internet of Things (IoT) edge runtime and cloud service that helps you build, deploy and manage IoT applications on your devices.