Core Tenets of IoT

April 2016

Notices

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

Contents

Abstract

This paper outlines core tenets that should be considered when developing a strategy for the Internet of Things (IoT). The paper helps customers understand the benefits of Amazon Web Services (AWS) and how the AWS cloud platform can be the critical component supporting the core tenets of an IoT solution. The paper also provides an overview of AWS services that should be part of an overall IoT strategy. This paper is intended for decision makers who are learning about Internet of Things platforms.

Overview

One of the value propositions of an Internet of Things (IoT) strategy is the ability to provide insight into context that was previously invisible to the business. But before a business can develop a strategy for IoT, it needs a platform that meets the foundational principles of an IoT solution.

AWS believes in some basic freedoms that are driving organizational and economic benefits of the cloud into businesses. These freedoms are why more than a million customers already use the AWS platform to support virtually any cloud workload. These freedoms are also why the AWS platform is proving itself as the primary catalyst to any Internet of Things strategy across commercial, consumer, and industrial solutions.

AWS customers working across such a spectrum of solutions have identified core tenets vital to the success of any IoT platform. These core tenets are agility, scale, cost, and security; which have been shown as essential to the long-term success of any IoT strategy.

This whitepaper defines these tenets as:

- Agility – The freedom to quickly analyze, execute, and build business and technical initiatives in an unfettered fashion
- Scale – Seamlessly expand infrastructure regionally or globally to meet operational demands
- Cost – Understand and control the costs of operating an IoT platform
- Security – Secure communication from device through cloud while maintaining compliance and iterating rapidly

By using the AWS platform, companies are able to build agile solutions that can scale to meet exponential device growth, with an ability to manage cost, while building on top of some of the most secure computing infrastructure in the world. A company that selects a platform that has these freedoms and promotes these core tenets will improve organizational focus on the differentiators of its business and the strategic value of implementing solutions within the Internet of Things.

Core Tenets of IoT

Agility

A leading benefit companies seek when creating an IoT solution is the ability to efficiently quantify opportunities. These opportunities are derived from reliable sensor data, remote diagnostics, and remote command and control between users and devices. Companies that can effectively collect these metrics open the door to explore different business hypotheses based on their IoT data. For example, manufacturers can build predictive analytics solutions to measure, test, and tune the ideal maintenance cycle for their products over time. The IoT lifecycle is comprised of multiple stages that are required to procure, manufacture, onboard, test, deploy, and manage large fleets of physical devices. When developing physical devices, the waterfall-like process introduces challenges and friction that can slow down business agility. This friction coupled with the up-front hardware costs of developing and deploying physical assets at scale often result in the requirement to keep devices in the field for long periods of time to achieve the necessary return on investment (ROI).

With the ever-growing challenges and opportunities that face companies today, a company's IT division is a competitive differentiator that supports business performance, product development, and operations. In order for a company's IoT strategy to be a competitive advantage, the IT organization relies on having a broad set of tools that promote interoperability throughout the IoT solution and among a heterogeneous mix of devices. Companies that can achieve a successful balance between the waterfall processes of hardware releases and the agile methodologies of software development, can continuously optimize the value that's derived from their IoT strategy.

Scalability and Global Footprint

Along with an exponential growth of connected devices, each *thing* in the Internet of Things communicates packets of data that require reliable

connectivity and durable storage. Prior to cloud platforms, IT departments would procure additional hardware and maintain underutilized, overprovisioned capacity in order to handle the increasing growth of data emitted by devices, also known as telemetry. With IoT, an organization is challenged with managing, monitoring, and securing the immense number of network connections from these dispersed, connected devices.

In addition to scaling and growing a solution in one regional location, IoT solutions require the ability to scale globally and across different physical locations. IoT solutions should be deployed in multiple physical locations to meet the business objectives of a global enterprise solution such as data compliance, data sovereignty, and lower communication latency for better responsiveness from devices in the field.

Cost

Often the greatest value of an IoT solution is in the telemetric and contextual data that is generated and sent from devices. Building on-premise infrastructure requires upfront capital purchase of hardware; it can be a large, fixed expense that does not directly correlate to the value of the telemetry that a device will produce sometime in the future. To balance the need to receive telemetry today with an uncertain value derived from telemetric data in the future, an IoT strategy should leverage an elastic and scalable cloud platform. With the AWS platform, a company pays only for the services it consumes without requiring a long-term contract. By leveraging a flexible, consumption based pricing model, the cost of an IoT solution and the related infrastructure can be directly accessed alongside the business value delivered by ingesting, processing, storing, and analyzing the telemetry received by that same IoT solution.

Security

The foundation of an IoT solution starts and ends with security. Since devices may send large amounts of sensitive data and end users of IoT applications may also have the ability to directly control a device, the security of things must be a pervasive design requirement. IoT solutions should not just be designed with security in mind, but with security controls permeating every layer of the solution. Security is not a static formula; IoT applications must be able to continuously model, monitor, and iterate on security best practices. In the Internet of Things, the attack surface is different than traditional web infrastructure. The pervasiveness of ubiquitous computing means that IoT

vulnerabilities could lead to exploits that result in the loss of life, for example from a compromised control system for gasoline pipelines or power grids.

A competing dynamic for IoT security is the lifecycle of a physical device and the constrained hardware for sensors, microcontrollers, actuators, and embedded libraries. These constrained factors may limit the security capabilities each device can perform. With these additional dynamics, IoT solutions must continuously adapt their architecture, firmware, and software to stay ahead of the changing security landscape. Although the constrained factors of devices can present increased risks, hurdles and potential tradeoffs between security and cost, building a secure IoT solution must be the primary objective for any organization.

AWS Services for IoT Solutions

The AWS platform provides a foundation for executing an agile, scalable, secure and cost-effective IoT strategy. In order to achieve the business value that IoT can bring to an organization, customers should evaluate the breadth and depth of AWS services that are commonly used in large-scale, distributed IoT deployments. AWS provides a range of services to accelerate time to market: from device SDKs for embedded software, to real-time data processing and event-driven compute services.

In these sections, we will cover the most common AWS services used in IoT applications, and how these services correspond to the core tenets of an IoT solution.

AWS IoT

The Internet of Things cannot exist without *things*. Every IoT solution must first establish connectivity in order to begin interacting with devices. AWS IoT is an AWS managed service that addresses the challenges of connecting, managing, and operating large fleets of devices for an application. The combination of scalability of connectivity and security mechanisms for data transmission within AWS IoT provides a foundation for IoT communication as part of an IoT solution. Once data has been sent to AWS IoT, a solution is able to leverage an ecosystem of AWS services spanning databases, mobile services, big data, analytics, machine learning and more.

Device Gateway

A device gateway is responsible for maintaining the sessions and subscriptions for all connected devices in an IoT solution. The AWS IoT Device Gateway enables secure, bi-directional communication between connected devices and the AWS platform over MQTT, WebSockets, and HTTP. Communication protocols such as MQTT and HTTP enable a company to utilize industry standard protocols instead of using a proprietary protocol that would limit future interoperability.

As a publish and subscribe protocol, MQTT inherently encourages scalable, fault-tolerant communication patterns and fosters a wide range of communication options among devices and the Device Gateway. These message patterns range from communication between two devices to broadcast patterns where one device can send a message to a large field of devices over a shared topic. In addition, the MQTT protocol exposes different levels of Quality of Service (QoS) to control the retransmission and delivery of messages as they are published to subscribers. The combination of publish and subscribe with QoS not only opens the possibilities for IoT solutions to control how devices interact in a solution, but also drive more predictability in how messages are delivered, acknowledged, and retried in the event of network or device failures.

Shadows, Device Registry, and Rules Engine

AWS IoT consists of additional features that are essential to building a robust IoT application. The AWS IoT service includes the Rules Engine, which is capable of filtering, transforming, and forwarding device messages as they are received by the Device Gateway. The Rules Engine utilizes a SQL-based syntax that selects data from message payloads and triggers actions based on the characteristics of the IoT data. AWS IoT also provides a Device Shadow that maintains a virtual representation of a device. The Device Shadow acts as a message channel to send commands reliably to a device, and store the last-known state of a device in the AWS platform.

For managing the lifecycle of a fleet of devices, AWS IoT has a Device Registry. The Device Registry is the central location for storing and querying a predefined set of attributes related to each thing. The Device Registry supports the creation of a holistic management view for an IoT solution to control the associations between things, shadows, permissions, and identities.

Security and Identity

For connected devices, an IoT platform should utilize concepts of identity, least privilege, encryption, and authorization throughout the hardware and software development lifecycle. AWS IoT encrypts traffic to and from the service over Transport Layer Security (TLS) with support for most major cipher suites. For identification, AWS IoT requires a connected device to authenticate using a X.509 certificate. Each certificate must be provisioned, activated, and then installed on a device before it can be used as a valid identity with AWS IoT. In order to support this separation of identity and access for devices, AWS IoT provides IoT Policies for device identities. AWS IoT also utilizes AWS Identity and Access Management (AWS IAM) policies for AWS users, groups, and roles. By using IoT Policies, an organization has control over allowing and denying communications on IoT topics for each specific device's identity. AWS IoT policies, certificates, and AWS IAM are designed for explicit, whitelist configuration of the communication channels of every device in a company's AWS IoT ecosystem.

Event Driven Services

In order to achieve the tenets of scalability and flexibility in an IoT solution, an organization should incorporate the techniques of an event-driven architecture. An event-driven architecture fosters scalable and decoupled communication through the creation, storage, consumption, and reaction to events of interest that occur in an IoT solution. Messages that are generated in an IoT solution should first be categorized and mapped to a series of events. An IoT solution should then associate these events with business logic that executes commands and possibly generates additional events in the IoT system. The AWS platform provides several application services for building a distributed, event-driven IoT architecture.

Foundationally event-driven architectures rely on the ability to durably store and transfer events through an ecosystem of interested subscribers. In order to support decoupled event orchestration, the AWS platform has several application services that are designed for reliable event storage and highly scalable event driven computation. An event-driven IoT solution should utilize Amazon Simple Queue Service (Amazon SQS), Amazon Simple Notification Service (Amazon SNS), and AWS Lambda as foundational application components for creating simple and complex event workflows. Amazon SQS is a fast, durable, scalable, and fully managed message queuing service. Amazon SNS is a web service that publishes messages from an application and immediately delivers them to

subscribers or other applications. AWS Lambda is designed to run code in response to events while the underlying computer resources are automatically managed. AWS Lambda can receive and respond to notifications directly from other AWS services. In an event-driven IoT architecture, AWS Lambda is where the business logic is executed to determine when events of interest have occurred in the context of an IoT ecosystem.

AWS services such as Amazon SQS, Amazon SNS, and AWS Lambda can separate the consuming of events from the processing and business logic applied to those events. This separation of responsibilities creates flexibility and agility in an end-to-end solution. This separation enables the rapid modification of event trigger logic or the logic used to aggregate contextual data between parts of a system. Finally, this separation allows changes to be introduced in an IoT solution without blocking the continuous stream of data being sent between end devices and the AWS platform.

Automation and DevOps

In IoT solutions, the initial release of an application is the beginning of a long-term approach to constantly refine the business advantages of an IoT strategy. After the first release of an application, a majority of time and effort will be spent adding new features to the current IoT solution. With the tenet of remaining agile throughout the solution lifecycle, customers should evaluate services that enable rapid development and deployment as business needs change. Unlike traditional web architectures where DevOps technologies only apply to the backend servers, an IoT application will also require the ability to incrementally roll-out changes to disparate, globally connected devices. With the AWS platform, a company can implement server-side and device-side DevOps practices to automate operations.

Applications deployed in the AWS cloud platform can take advantage of several DevOps technologies on AWS. For an overview of AWS DevOps, we recommend reviewing the document *Introduction to DevOps on AWS* [1]. Although most solutions will differ in deployment and operations requirements, IoT solutions can utilize AWS CloudFormation to define their server-side infrastructure as code. Infrastructure treated as code has the benefits of being reproducible, testable, and more easily deployable across other AWS regions. Enterprise organizations that utilize AWS CloudFormation in addition to other DevOps tools greatly increase their agility and pace of application changes.

In order to design an IoT solution that adheres to the tenets of security and agility, organizations must also update their connected devices after they have been deployed into the environment. Firmware updates provide a company a mechanism to add new features to a device and are a critical path for delivering security patches during the lifetime of a device. To implement firmware updates to connected devices, an IoT solution should first store the firmware in a globally accessible service such as Amazon Simple Storage Service (Amazon S3) for secure, durable, highly-scalable cloud storage. Then the IoT solution can implement Amazon CloudFront, a global content delivery network (CDN) service, to bring the the firmware stored in Amazon S3 to the lower latency points of presence for connected devices. Finally, a customer can leverage the AWS IoT Shadow to push a command to a device to request that it download the new version of firmware from a pre-signed Amazon CloudFront URL that restricts access to the firmware objects available through the CDN. Once the upgrade is complete the device should acknowledge success by sending a message back into the IoT solution. By orchestrating this small set of services for firmware updates customers control their Device DevOps approach and can scale it in a way that aligns with their overall IoT strategy.

In IoT, automation and DevOps procedures expand beyond the application services that are deployed in the AWS platform and include the connected devices that have been deployed as part of the overall IoT architecture. By designing a system that can easily perform regular and global updates for new software changes and firmware changes, organizations can iterate on ways to increase value from their IoT solution and to continuously innovate as new market opportunities arise.

Administration and Security

Security in IoT is more than data anonymization; it is the ability to have insight, auditability, and control throughout a system. IoT security includes the capability to monitor events throughout the solution, and react to those events to achieve the desired compliance and governance. Security at AWS is our number one priority. Through the AWS Shared Responsibility Model, an organization has the flexibility, agility, and control to implement their security requirements. [2] AWS manages the security **of** the cloud, while customers are responsible for security **in** the cloud. Customers maintain control over what security mechanisms they implement to protect their data, applications, devices, systems and networks. In addition, companies can leverage the broad set of security and administrative

tools that AWS and AWS partners provide to create a strong, logically isolated, and secure IoT solution for a fleet of devices.

The first service that should be enabled for monitoring and visibility is AWS CloudTrail. AWS CloudTrail is a web service that records AWS API calls for an account and delivers log files to Amazon S3. After enabling AWS CloudTrail, a solution should build security and governance processes that are based on the real-time input from API calls made across an AWS account. AWS CloudTrail provides an additional level of visibility and flexibility in creating and iterating on operational openness in a system.

In addition to logging API calls, customers should enable Amazon CloudWatch for all AWS services used in the system. Amazon CloudWatch allows applications to monitor AWS metrics and create custom metrics generated by an application. These metrics can then trigger alerts based off of those events. Along with Amazon CloudWatch metrics, there are Amazon CloudWatch Logs, which store additional logs from AWS services or customer applications, and can then trigger events based off of those additional metrics. AWS services, such as AWS IoT, directly integrate with Amazon CloudWatch Logs; these logs can be dynamically read as a stream of data and processed using the business logic and context of the system for real-time anomaly detection or security threats.

By pairing services like Amazon CloudWatch and Amazon CloudTrail with the capabilities of AWS IoT identities and policies, a company can immediately collect valuable data around security practices at the start of the IoT strategy and meet the needs for a proactive implementation of security within their IoT solution.

Bringing Services and Solutions Together

To better understand customer usage, predict future trends, or run an IoT fleet more efficiently, an organization needs to collect and process the potentially vast amount of data gathered from connected devices in addition to connecting with and managing large fleets of *things*.

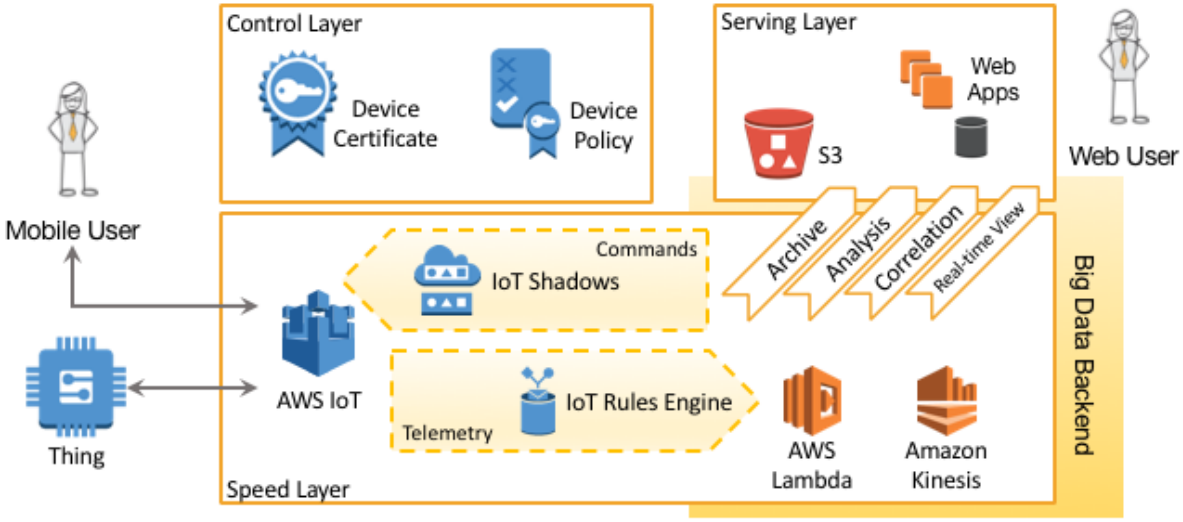
AWS provides a breadth of services for collecting and analyzing large scale datasets often called big data. These services may be integrated tightly within an IoT solution to support collecting, processing, and analyzing the solution's data, as well as proving or disproving hypotheses based upon IoT data. The ability to

formulate and answer questions with the same platform one is using to manage fleets of *things* ultimately empowers an organization to avoid undifferentiated work and to unlock business innovations in an agile fashion.

The high-level, cohesive architectural perspective of an IoT solution that brings IoT, big data and other services together is called the Pragma Architecture. The Pragma Architecture is comprised of layers of solutions:

- Things - The device and fleet of devices
- Control Layer - The control point for access to the Speed Layer and the nexus for fleet management
- Speed Layer - The inbound, high-bandwidth device telemetry data bus and the outbound device command bus
- Serving Layer - The access point for systems and humans to interact with the devices in a fleet, to perform analysis, archive, and correlate data, and to use real-time views of the fleet.

Pragma Architecture



The Pragma Architecture is a single cohesive perspective of how the core tenets of IoT manifest as an IoT solution when using AWS services.

One scenario of a Pragma Architecture based IoT Solution is around processing of data emitted by devices; data also known as telemetry. In the diagram above, after a device authenticates using a device certificate obtained from the AWS IoT

service in the control layer, the device regularly sends telemetry data to the AWS IoT Device Gateway in the Speed Layer. That telemetry data is then processed by the IoT Rules Engine as an event to be output by Amazon Kinesis or AWS Lambda for use by web users interacting with the serving layer.

Another scenario of a Pragma Architecture based IoT Solution is to send a command to a device. In the diagram above, the user's application would write the desired command value to the target device's IoT Shadow. Then the AWS IoT Shadow and the Device Gateway work together to overcome an intermittent network to convey the command to the specific device.

These are just two device-focused scenarios from a broad tapestry of solutions that fit the Pragma Architecture. Neither of these scenarios address the need to process the potentially vast amount of data gathered from connected devices, this is where having an integrated Big Data Backend starts to become important. The Big Data Backend in this diagram is congruent with the entire ecosystem of real-time and batch-mode big data solutions that customers already leverage the AWS platform to create. Simply put, from the big data perspective IoT telemetry equals "ingested data" in big data solutions. If you'd like to learn more about big data solutions on AWS, please check below for a link to further reading.

There is a colorful and broad tapestry of big data solutions that companies have already created using the AWS platform. The Pragma Architecture shows that by building an IoT solution on that same platform, the entire ecosystem of big data solutions is available.

Summary

Defining your Internet of Things strategy can be a truly transformational endeavor that opens the door for unique business innovations. As organizations start striving for their own IoT innovations, it is critical to select a platform that promotes the core tenets: business and technical agility, scalability, cost, and security. The AWS platform over-delivers on the core tenets of an IoT solution by not just providing IoT services, but offering those services alongside a broad, deep, and highly regarded set of platform services across a global footprint. This over-delivery also brings freedoms that increase your business' control over its own destiny and enables your business' IoT solutions to more rapidly iterate toward the outcomes sought in your IoT strategy.

As next steps in evaluating IoT platforms, we recommend the *further reading* section below to learn more about AWS IoT, big data solutions on AWS, and customer case studies on AWS.

Contributors

The following individuals authored this document:

- Olawale Oladehin, Solutions Architect, Amazon Web Services
- Brett Francis, Principal Solutions Architect, Amazon Web Services

Further Reading

For additional reading, please consult the following sources:

- AWS IoT Service
- Getting Started with AWS IoT
- AWS Case Studies
- Big Data Analytics Options on AWS

Notes

[1] https://do.awsstatic.com/whitepapers/AWS_DevOps.pdf

[2] https://aws.amazon.com/compliance/shared-responsibility-model/