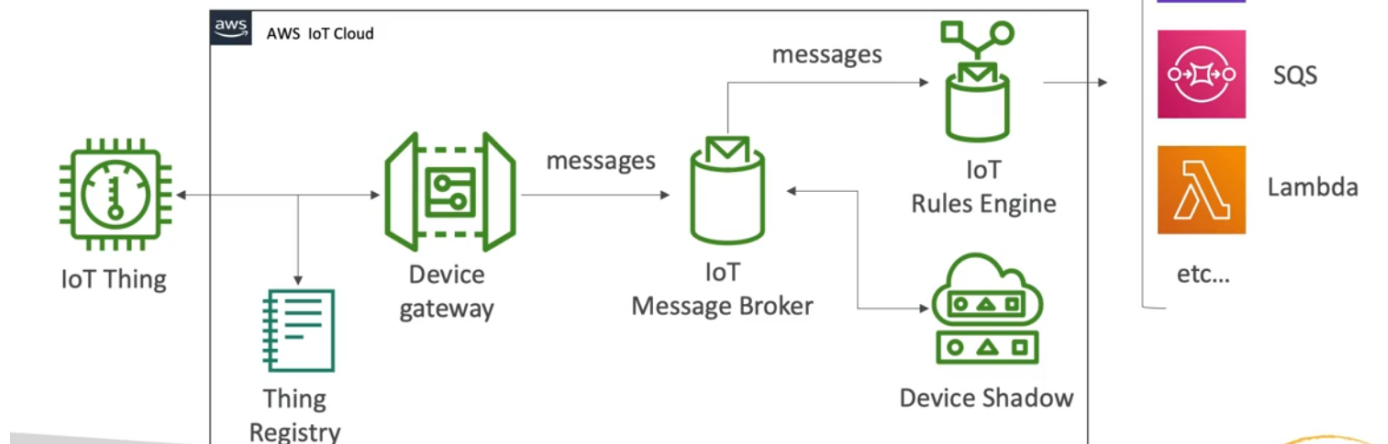


# IOT things

## IoT Overview

- We deploy IoT devices ("Things")
- We configure them and retrieve data from them



It acts as a bridge, enabling devices to connect to the cloud-based IoT platform or applications. Here are some key aspects of an IoT device gateway

\*\*\*IoT Device gateway \*\*\*

**Device Communication:** The IoT device gateway serves as a communication interface for IoT devices. It supports various communication protocols, such as MQTT (Message Queuing Telemetry Transport), HTTP, CoAP (Constrained Application Protocol), or custom protocols, depending on the requirements of the IoT system.

**Data Aggregation:** The gateway collects data from multiple IoT devices, aggregates it, and prepares it for transmission to the cloud. It can buffer and batch data to optimize network utilization and reduce latency.

**Protocol Translation:** IoT devices often use different communication protocols, which may not be directly compatible with the cloud platform or applications. The gateway performs protocol translation, converting data from device-specific protocols to standardized formats that the cloud can understand.

**Security and Authentication:** IoT device gateways play a crucial role in enforcing security measures. They provide secure communication channels, implement authentication and authorization mechanisms, and ensure data encryption to protect sensitive information.

**Device Management:** The gateway facilitates device management tasks such as device registration, provisioning, firmware updates, and configuration management. It can handle device authentication

and authorization for secure access to cloud services.

**Edge Computing:** Some IoT gateways have edge computing capabilities, allowing them to perform data processing, analytics, or filtering at the edge of the network. This reduces the amount of data sent to the cloud and enables real-time or low-latency processing.

**Offline Operation:** IoT device gateways often have the ability to operate offline or in intermittent connectivity scenarios. They can store data locally and transmit it to the cloud when the network connection is available.

**Scalability and Deployment Flexibility:** Gateways are designed to handle large-scale deployments with a high number of connected devices. They can be deployed in various configurations, such as standalone gateways, gateway clusters, or as part of edge computing infrastructure

IOT Think register

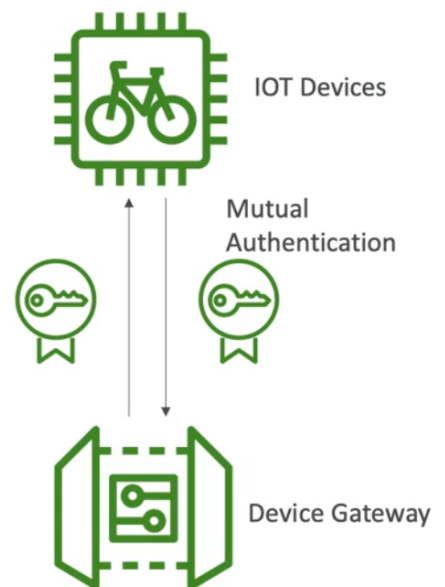
# IoT Thing Registry = IAM of IoT



- All connected IoT devices are represented in the AWS IoT registry
- Organizes the resources associated with each device in the AWS Cloud
- Each device gets a unique ID
- Supports metadata for each device (ex: Celsius vs Fahrenheit, etc...)
- Can create X.509 certificate to help IoT devices connect to AWS
- IoT Groups: group devices together and apply permissions to the group

# Authentication

- 3 possible authentication methods for Things:
  - Create X.509 certificates and load them securely onto the Things
  - AWS SigV4
  - Custom tokens with Custom authorizers
- For mobile apps:
  - Cognito identities (extension to Google, Facebook login, etc...)
- Web / Desktop / CLI:
  - IAM
  - Federated Identities

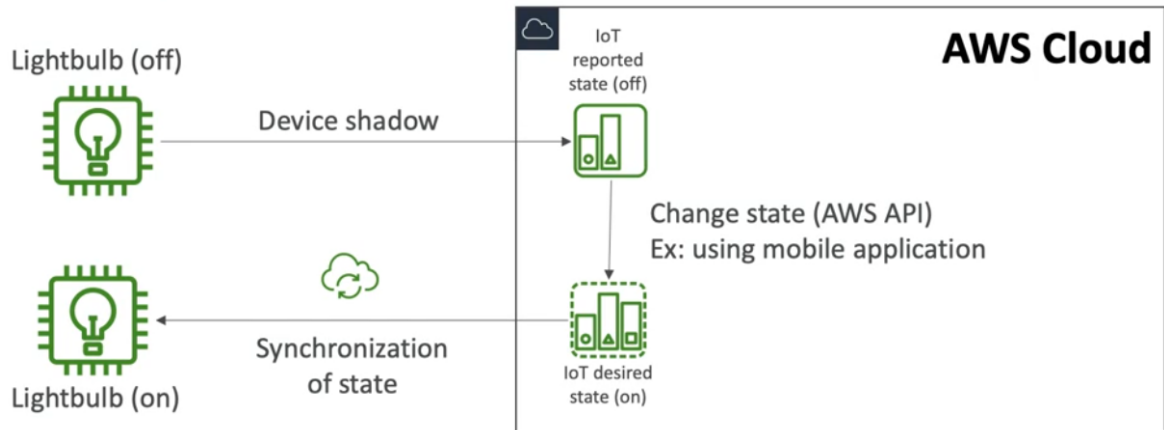


# Authorization

- AWS IoT policies:
  - Attached to X.509 certificates or Cognito Identities
  - Able to revoke any device at any time
  - IoT Policies are JSON documents
  - Can be attached to groups instead of individual Things.
- IAM Policies:
  - Attached to users, group or roles
  - Used for controlling IoT AWS APIs

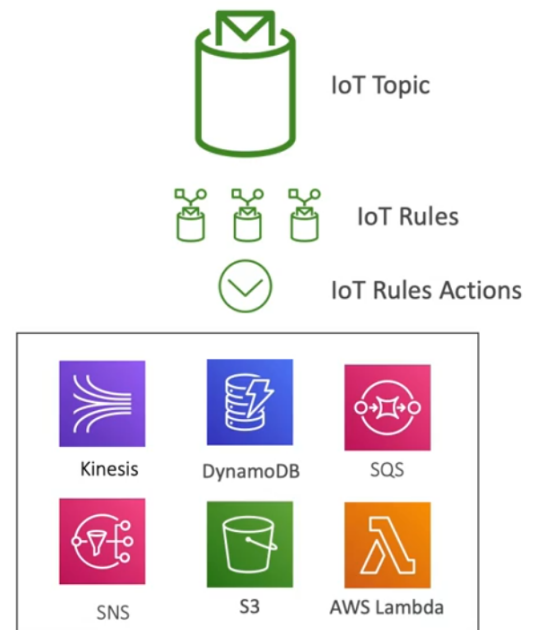
# Device Shadow

- JSON document representing the state of a connected Thing
- We can set the state to a different desired state (ex: light on)
- The IoT thing will retrieve the state when online and adapt



## Rules Engine

- Rules are defined on the MQTT topics
- Rules = when it's triggered | Action = what it does
- Rules use cases:
  - Augment or filter data received from a device
  - Write data received from a device to a **DynamoDB** database
  - Save a file to **S3**
  - Send a push notification to all users using **SNS**
  - Publish data to a **SQS** queue
  - Invoke a **Lambda** function to extract data
  - Process messages from a large number of devices using **Amazon Kinesis**
  - Send data to the Amazon **Elasticsearch Service**
  - Capture a **CloudWatch metric** and Change a **CloudWatch alarm**
  - Send the data from an MQTT message to **Amazon Machine Learning** to make predictions based on an Amazon ML model
  - & more
- Rules need IAM Roles to perform their actions



# IoT Greengrass

- IoT Greengrass brings the compute layer to the device directly
- You can execute AWS Lambda functions on the devices:
  - Pre-process the data
  - Execute predictions based on ML models
  - Keep device data in sync
  - Communicate between local devices
- Operate offline
- Deploy functions from the cloud directly to the devices



**AWS IoT Device Management:** AWS IoT Device Management simplifies the management of IoT devices at scale. It provides features like device onboarding, provisioning, over-the-air updates, and remote management of device configurations. It helps you keep track of device inventory, monitor device health, and perform diagnostics.

**AWS IoT Analytics:** AWS IoT Analytics allows you to process, analyze, and gain insights from IoT data. It provides tools for building IoT data pipelines, filtering and transforming data, and running analytics queries. IoT Analytics integrates with other AWS services like Amazon S3, Amazon QuickSight, and Amazon Athena for data storage, visualization, and advanced analytics.

**AWS IOT EVENT :** AWS IoT Events enables you to define event patterns based on the data received from IoT devices. You can define conditions and rules to detect specific events or anomalies in the data stream. For example, you can define rules to detect a sudden temperature rise, a drop in pressure, or a malfunctioning device

**AWS IoT Button:** The AWS IoT Button is a programmable button that connects to AWS services. It allows you to trigger actions, events, or workflows with a single press. The button can be customized to perform tasks like ordering products, controlling smart home devices, or sending notifications.

**Amazon FreeRTOS:** Amazon FreeRTOS is an open-source operating system for microcontrollers that simplifies IoT device development. It provides libraries, middleware, and connectivity modules for secure communication with AWS IoT Core. FreeRTOS helps in building low-power devices, edge devices, and IoT gateways.

**AWS IoT Device SDKs:** AWS provides SDKs (Software Development Kits) for various programming languages like Python, Java, JavaScript, C++, and more. These SDKs facilitate device integration, communication, and interaction with AWS IoT services.

**AWS IOT SITEWISE : AWS IoT SiteWise simplifies the process of collecting, organizing, and analyzing data from industrial equipment. It enables you to gain actionable insights, optimize operations, and drive efficiency in your industrial processes. By leveraging the capabilities of AWS IoT SiteWise, you can monitor and manage your industrial assets effectively and make data-driven decisions to improve performance and reliability.**