AWS re:Invent

WIN316

Building AWS IoT applications using .NET

Sundararajan Narasiman
Partner Solutions Architect
Amazon Web Services

Matt Luttrell
Cloud Application Architect
Amazon Web Services

AWS re:Invent

aws

Agenda

- Why AWS IoT Core and .NET
- AWS IoT Core protocols and integration
- Options to publish or subscribe with AWS IoT Core
- Demos

Why AWS IoT Core and .NET

- AWS IoT Core is a service for building connected applications

- AWS IoT Core supports device SDKs in programming languages such as Java, C, Python, and JavaScript

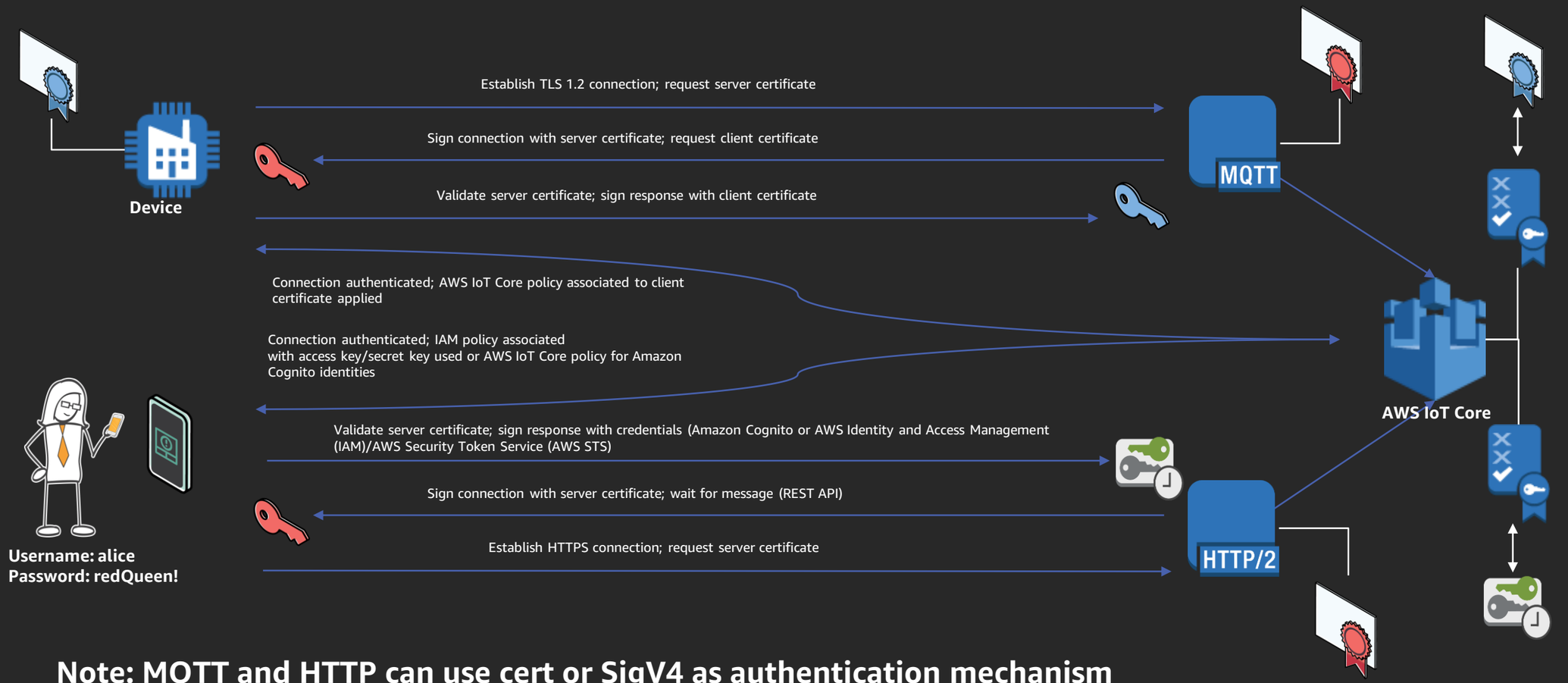
Why AWS IoT Core and .NET, continued

- Large IoT edge gateways are found in use cases like smart city command and control centers and building management systems
- Microsoft .NET occupies a significant footprint in the technology landscape of large enterprises
- No out-of-the-box support for AWS IoT Device SDK for .NET Framework or .NET Core

AWS IoT Core protocols and integration

	Protocol	Authentication	Port
1	MQTT	X.509 certificate	8883, 443
2	HTTP	X.509 certificate	8443
3	HTTP	SigV4	443
3	MQTT over WebSockets	SigV4	443

AWS IoT Core device authentication and authorization



Note: MQTT and HTTP can use cert or SigV4 as authentication mechanism

Demo

Links for source code

Links for source code

- https://github.com/aws-samples/aws-iot-core-http-sigv4-dotnet-app
- https://github.com/aws-samples/aws-iot-dotnet-publisher-http
- https://github.com/aws-samples/iot-dotnet-publisher-consumer
- https://github.com/aws-samples/aws-iot-core-dotnet-app-mqtt-over-websockets-sigv4

Thank you!

AWS re:Invent

aws

Please complete the session survey in the mobile app.

How to make .NET code handshake with AWS IoT Core using X509 Certificate

- Identify protocol for connection – HTTPS / MQTTS
- Identify Port for connection
- Convert Device Certificate, Root Certificate and Private Key to Windows format
- Implement Authentication using X509 Certificates
- Initiate MQTT Connection
- Define various MQTT events – Connect, Subscribe, Publish, Error and Disconnect
- Implement the publish / subscribe mechanism

How to make .NET code handshake with AWS IoT Core using AWS SigV4 authentication

- Identify protocol for connection – HTTPS / MQTTS
- Identify Port for Connection
- Implement AWS SigV4 authentication using AccessKey and SecretKey
- Initiate MQTT Connection
- Define various MQTT events – Connect, Subscribe, Publish, Error and Disconnect
- Implement the publish / subscribe mechanism