The background features a complex digital environment with glowing blue lines, data points, and various text elements like 'CONNECT', 'DIGITAL', 'BUSINESS', 'CONNECTION', 'SOCIAL', 'MARKET', 'INTERNET', 'TECHNOLOGY', 'IDEA', and 'DATA'. The overall aesthetic is high-tech and futuristic.

Aplicación de tecnologías y arquitecturas serverless para el desarrollo de soluciones IoT

Objetivos

- Investigar los modelos *Cloud*.
- Investigar las posibles arquitecturas *IoT*.
- Investigar las arquitecturas *serverless* para soluciones *IoT*.
- Implementar un caso de uso real con *serverless IoT*.
- Validar la solución.

Definición

“La computación en la nube es un modelo para permitir el acceso de red ubicuo, conveniente y bajo demanda a un grupo compartido de recursos de computación configurables (por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios) que pueden aprovisionarse y liberarse rápidamente con mínimo esfuerzo de gestión o interacción por parte del proveedor de servicios. Este modelo de nube promueve la disponibilidad y se compone de cinco características esenciales, tres modelos de servicio y cuatro modelos de implementación”

Características

- Auto servicio bajo demanda
- Amplio acceso a la red
- Agrupación de recursos
- Rápida elasticidad
- Servicio medido

Modelos de servicio

IaaS	CaaS	PaaS	FaaS	
Functions	Functions	Functions	Functions	Customer managed
Application	Application	Application	Application	Customer managed Unit of Scale
Runtime	Runtime	Runtime	Runtime	Abstracted By Vendor
Containers (optional)	Containers	Containers	Containers	
Operating System	Operating System	Operating System	Operating System	
Virtualization	Virtualization	Virtualization	Virtualization	
Hardware	Hardware	Hardware	Hardware	

Cloud
Computing

Definición

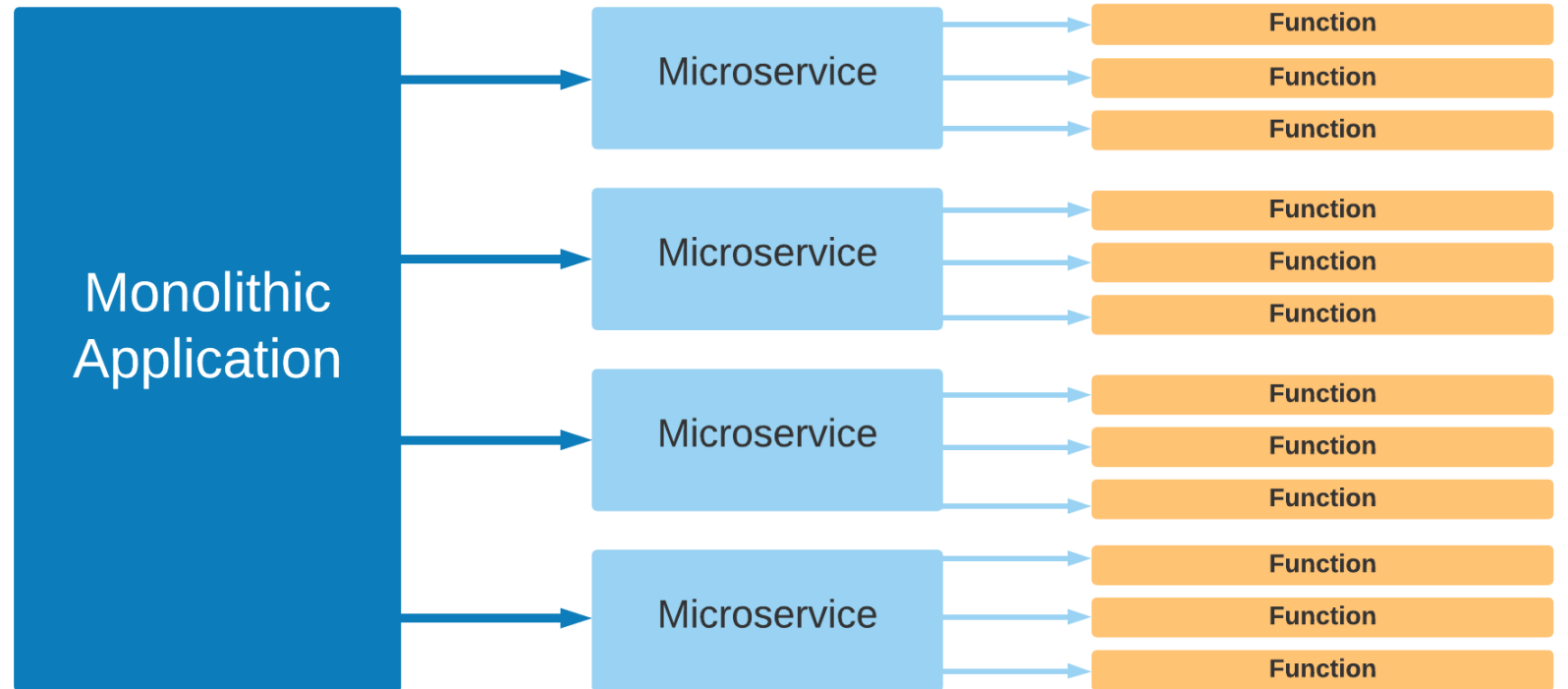


Características

- Auto servicio bajo demanda
- Amplio acceso a la red
- Agrupación de recursos
- Rápida elasticidad
- Servicio medido

Arquitectura
serverless

Tipos de arquitectura



Arquitectura
serverless

Principios



Arquitectura *serverless*

- Usar un servicio de computación para ejecutar código bajo demanda
- Desarrollar funciones sin estado de proposito único
- Diselo pasado en eventos *push*
- Crear *frontends* más robustos y potentes
- Adoptar servicios de terceros

Arquitectura *serverless*

Ventajas

- No más servidores
- Multitud de usos
- Bajo coste
- Menos Código
- Escalable y flexible

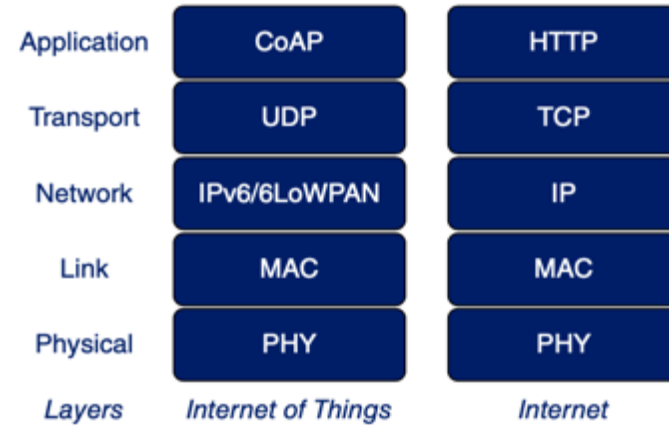
Desventajas

- No para todos
- Niveles de servicio
- Customización
- Dependencia del proveedor
- Descentralización

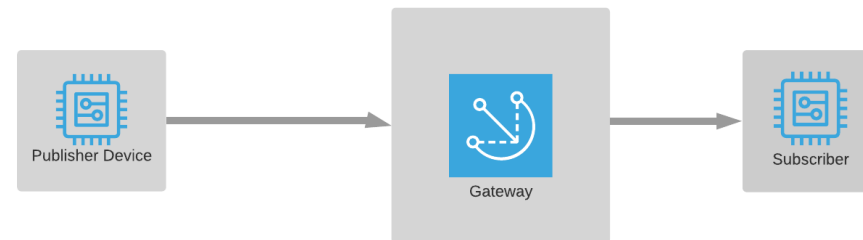


Capa de aplicación

- TCP/IP



- Publicación/Subscripción



Cloud Computing IoT

Ventajas

- Escalabilidad
- Movilidad de datos
- Time to market
- Seguridad
- Efectividad de costs

Desventajas

- Escalabilidad excesiva
- Disponibilidad
- Interoperabilidad
- Vendor-lock-in



IoT

Edge Computing y Fog Computing

- Amplia distribución geográfica, en contraste con la visión centralizada de la *Cloud*
- Modelo de suscripción usado por los miembros del *fog*
- Soporte y movilidad

Diferencias

Edge computing se ejecuta directamente en los dispositivos en los que el sensor está conectado o actúa como *gateway*

Fog computing mueve las actividades de computación *edge* a procesadores que están conectados a la *LAN*

¿Por qué?

- Las redes IoT suelen estar compuestas de millones de nodos, a través de distintas zonas geográficas, utilizando una arquitectura serverless permite eliminar toda la gestión sobre los proveedores Cloud.
- Las redes IoT suelen tener variaciones de carga donde a ciertas horas del día o debido a determinados eventos se puede desencadenar que se active un gran número de dispositivos.
- La adopción de metodologías serverless permite a las organizaciones centrarse en su valor comercial principal, agilidad de cambios y el time to market en vez de en la orquestación y el mantenimiento.

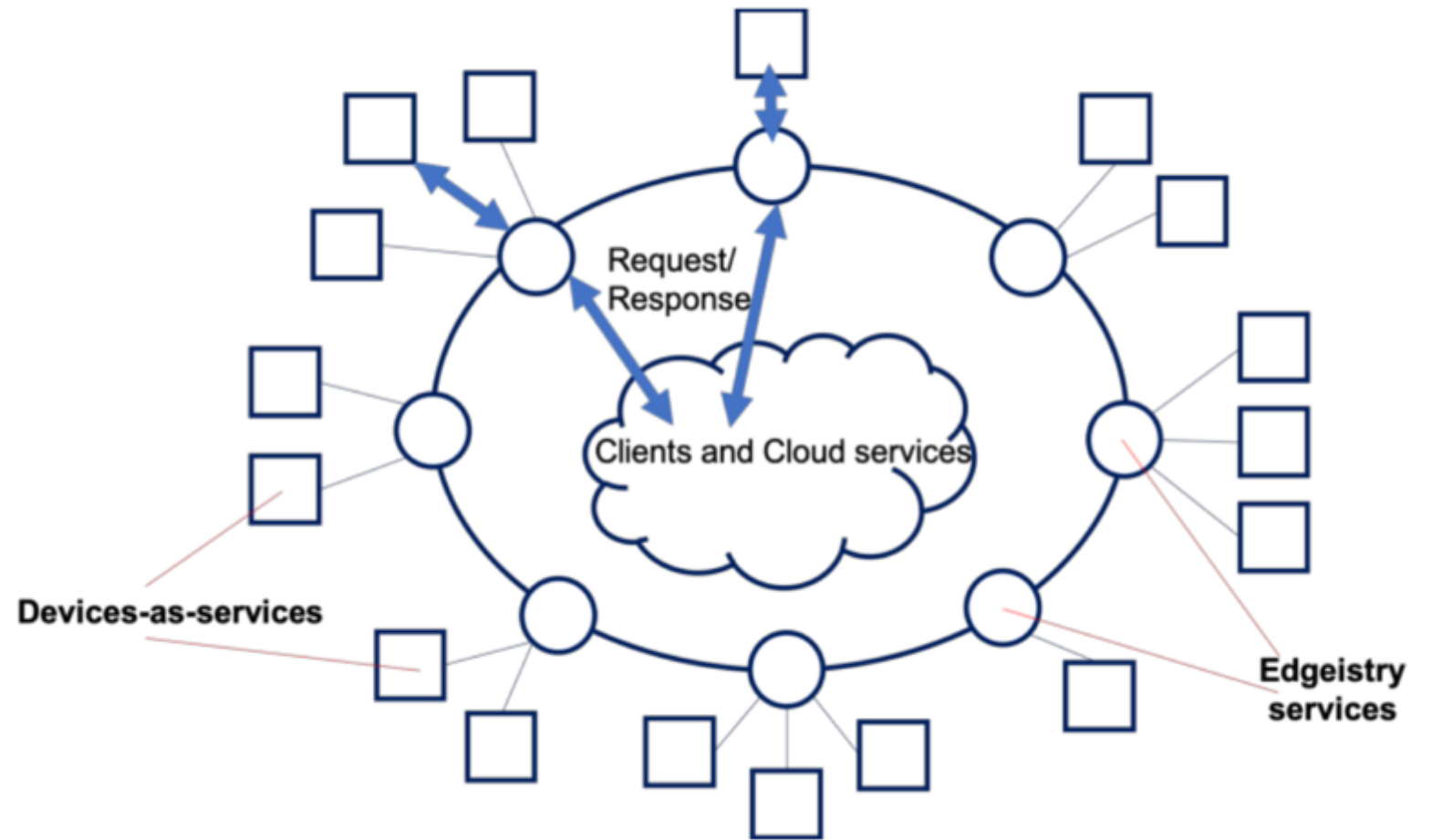
Desafíos

- Agrupación de recursos y elasticidad rápida
- Seguridad
- Provisionado automático y gestión a escala
- Programación de recursos Edge escasamente desacoplados y escasos
- Desarrollo de aplicación deviceless
- Gobierno centrado en Edge,



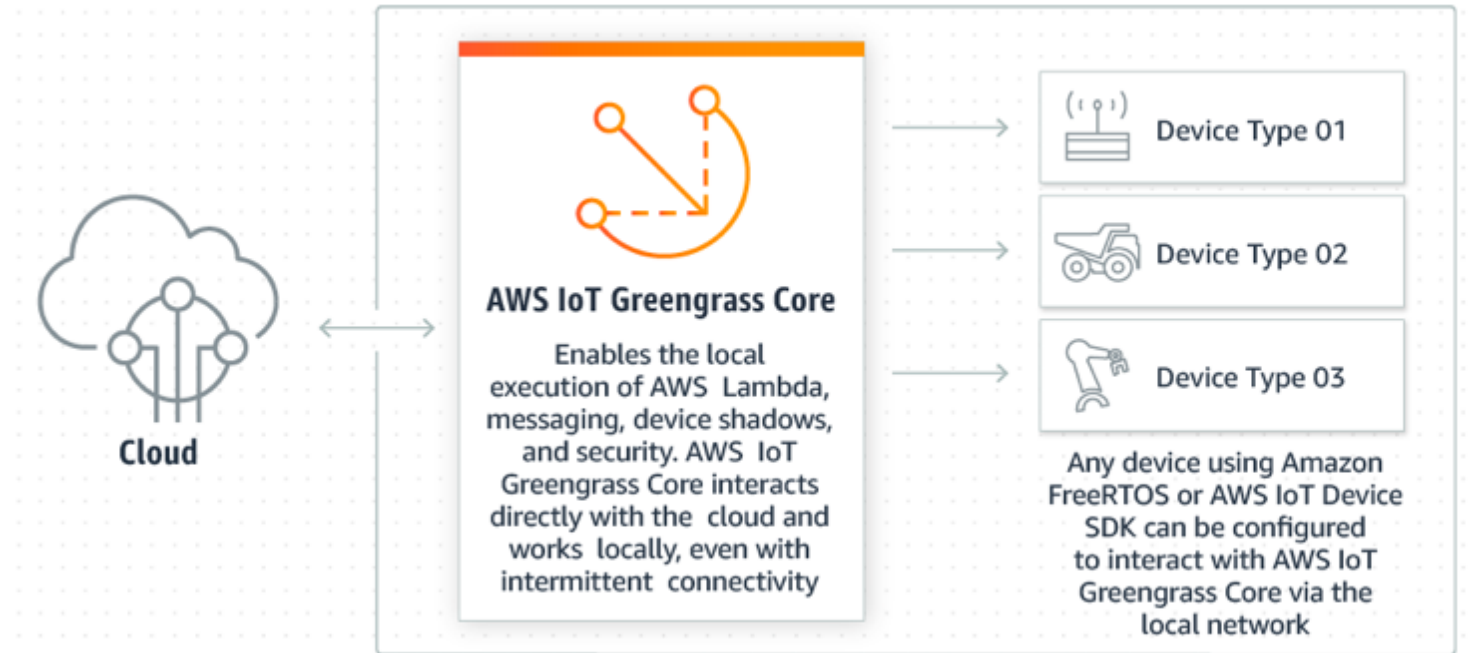
IoT y FaaS

Dispositivos como servicio



IoT y FaaS

AWS IoT Greengrass



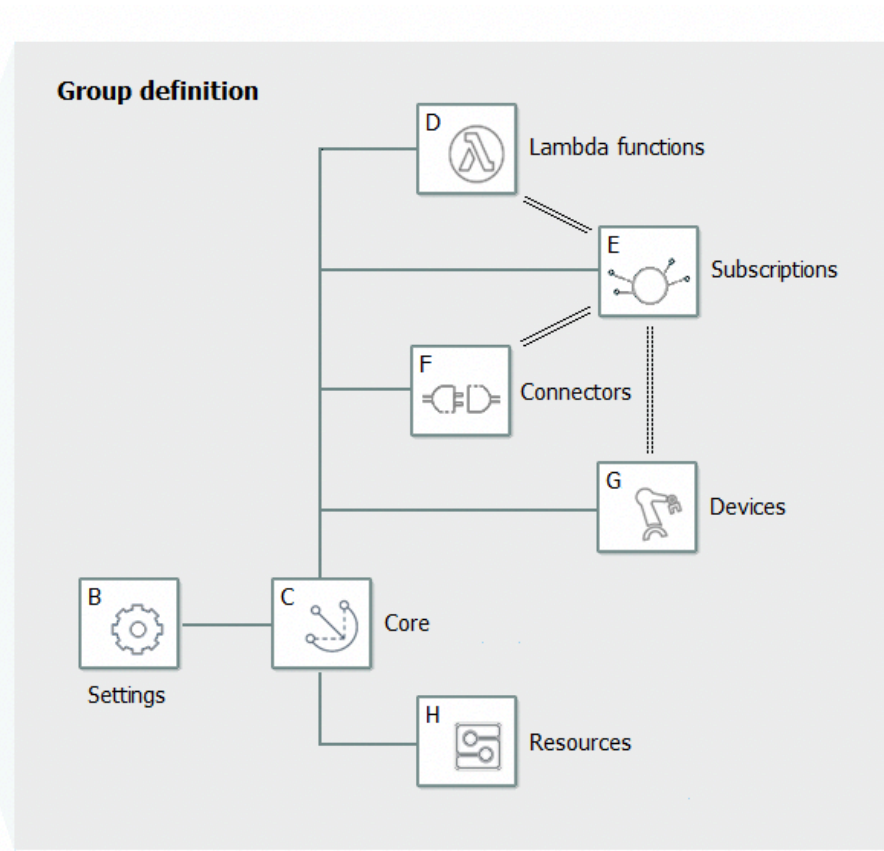
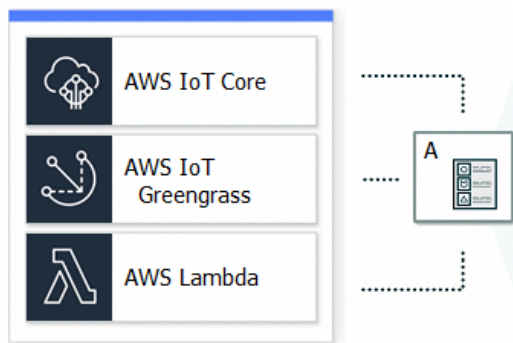
AWS IoT Greengrass



Características

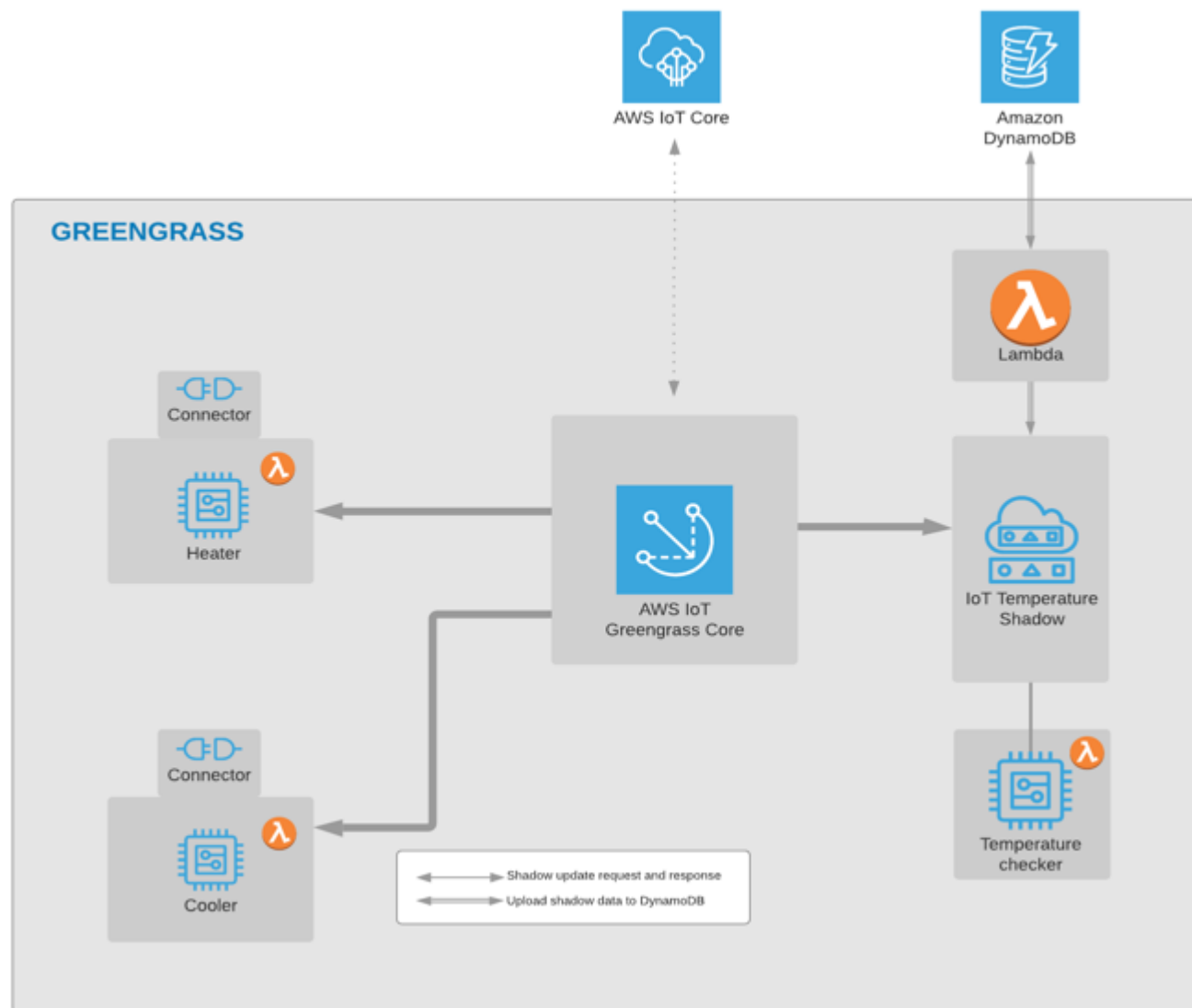
- Entorno de sobre la marcha ejecución Lambda
- Implementación de shadows
- Gestor de mensajes
- Gestor de grupos
- Agente de actualización
- Gestor de stream de datos
- Acceso a recursos locales
- Interfaz de Machine Learning
- Gestor de secretos locales
- Conectores con integraciones por defecto

Greengrass groups



AWS IoT
Greengrass

¿Qué se quiere resolver?



**AWS IoT
Greengrass**

Validación y Conclusión

¿Objetivos cumplidos?

- Investigar los modelos *Cloud*. ✓
- Investigar las posibles arquitecturas *IoT*. ✓
- Investigar las arquitecturas *serverless* para soluciones *IoT*. ✓
- Implementar un caso de uso real con *serverless IoT*. ✓
- Validar la solución. ✓

¿Desafíos cumplidos?

- Agrupación de recursos y elasticidad rápida ✓
- Seguridad ✓
- Provisionado automático y gestión a escala ✓
- Programación de recursos Edge escasamente desacoplados y escasos ✓
- Desarrollo de aplicación *deviceless* ✓
- Gobierno centrado en Edge ✓

GRACIAS