Microsoft

DevOps for IoT

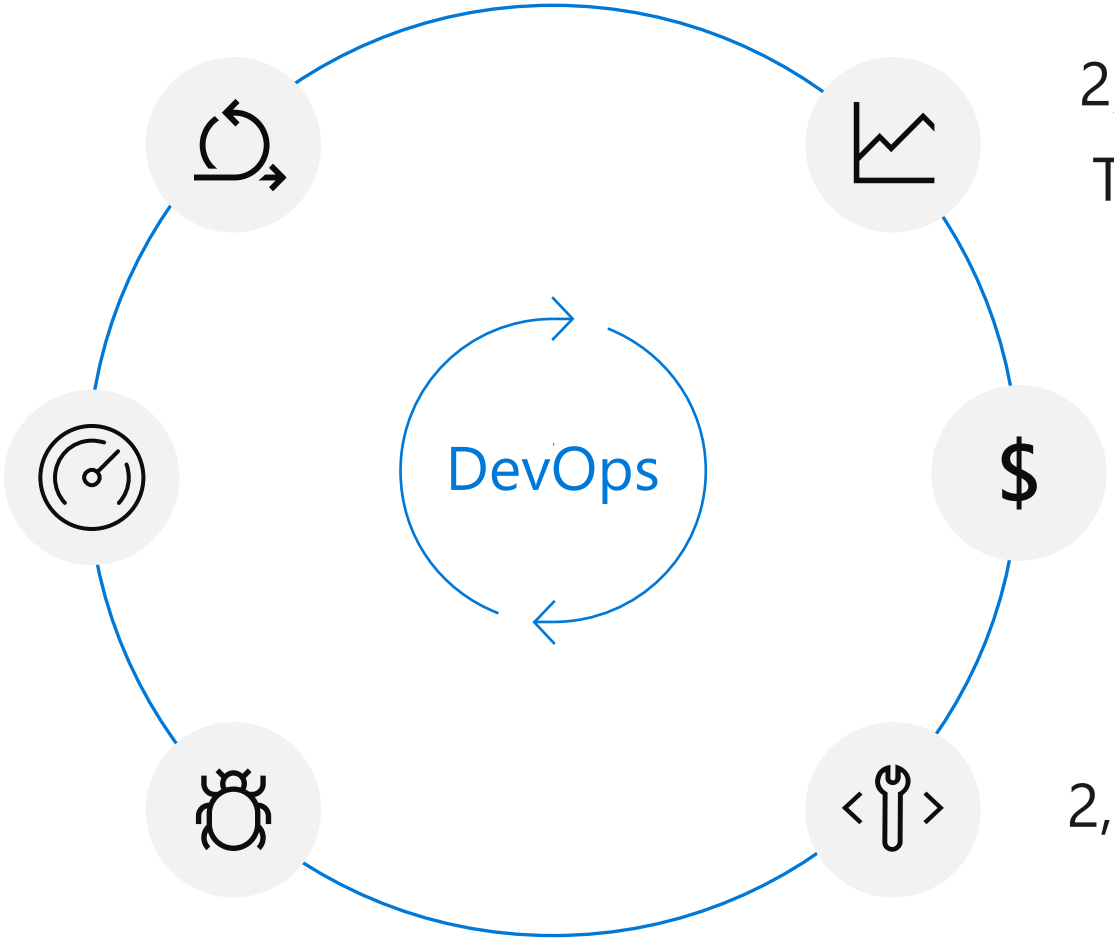
Alex Yochev
@alxayo

High Performance DevOps Companies Achieve...

46x Deployment Frequency

Faster Time to Market

7x Lower Change Failure Rate



2,555x Faster Lead Time For Changes

Increased Revenue

2,604x Faster Mean Time to Recover

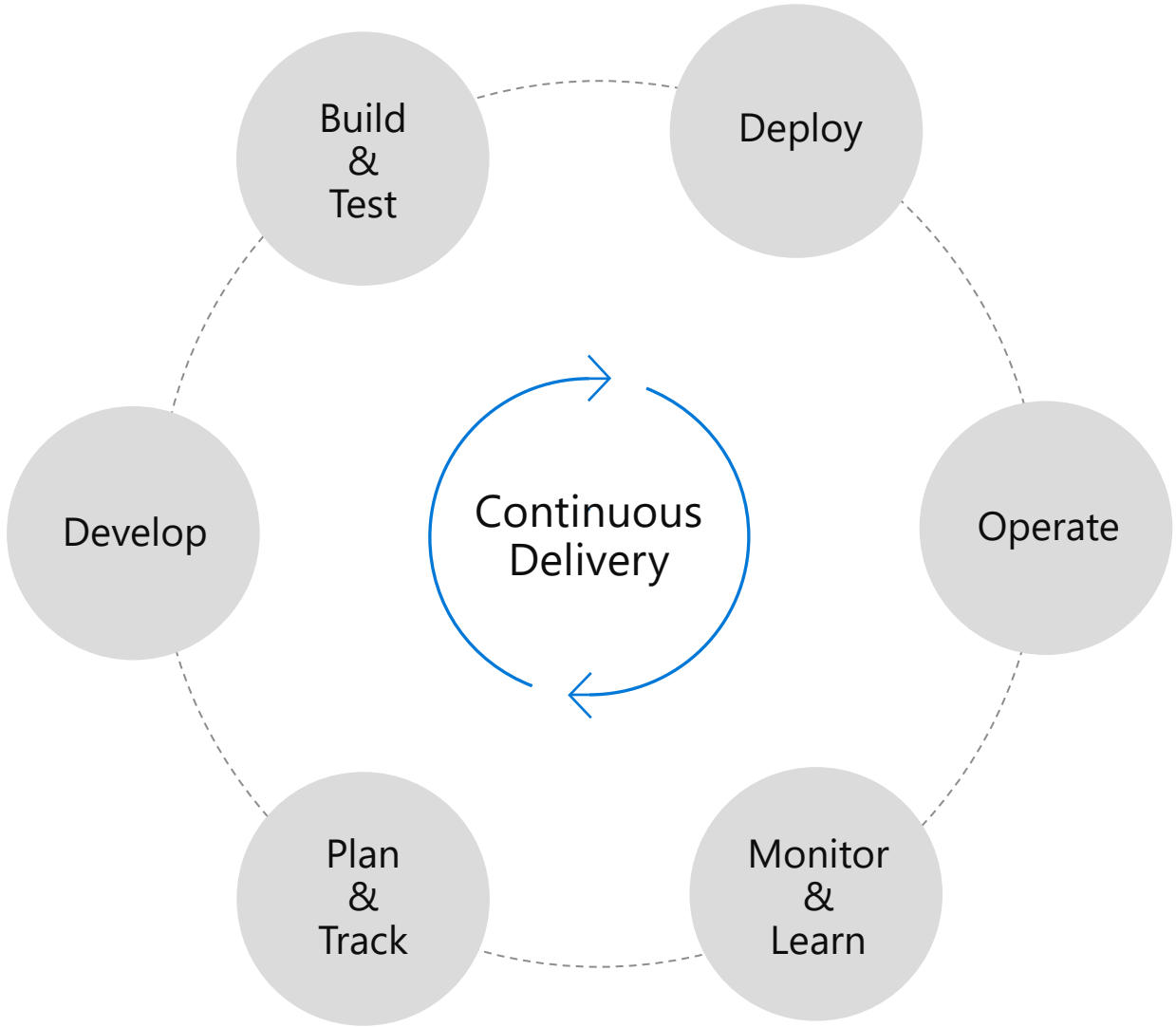
Microsoft

Source: 2018 Accelerate: State of DevOps: Strategies for a New Economy." N. Forsgren, J. Humble, G. Kim. DevOps Research and Assessment (DORA)

What is DevOps?

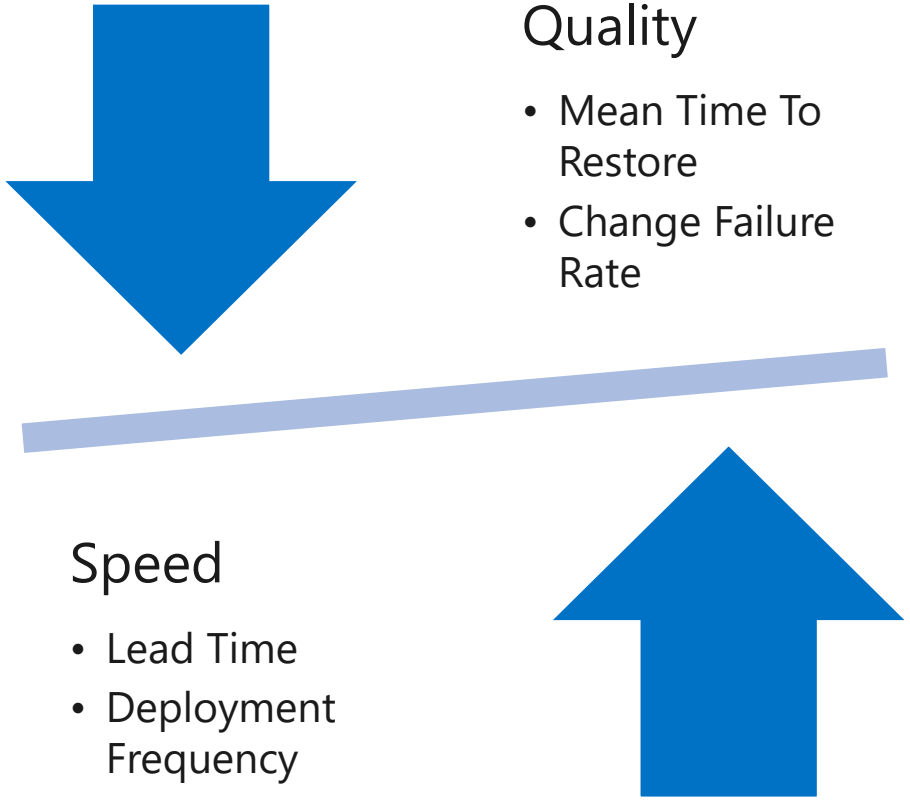
People. Process. Products.

> DevOps is the union of **people**, **process**, and **products** to enable continuous delivery of value to your end users.



Microsoft

Balancing Quality and Speed



Quality

- Mean Time To Restore
- Change Failure Rate

Speed

- Lead Time
- Deployment Frequency

LEAD TIME

The time it takes to design and validate a product or feature, and the time to deliver the feature to customers.

DEPLOYMENT FREQUENCY

The frequency with which software is deployed to the end users (i.e. in production or in an app store)

MEAN TIME TO RESTORE (MTTR)

How quickly is the service restored in the event of (an inevitable) failure

CHANGE FAILURE RATE

The percentage of changes to production (code and infrastructure) that fail

DevOps for IoT and IIoT

"Using a well-designed automated **DevOps toolchain, you can streamline security testing, performance testing and other testing processes that are much more difficult to do manually**," said David Linthicum, chief cloud strategy officer at **Deloitte** Consulting.

What's more, **deployment tools can deal with the complexity of pushing code and data out to thousands of devices**, using automation to administer updates and checking they have the correct device configuration, Linthicum said.

The upshot is that DevOps improves stability and security while reducing deployment and operating costs.

Assessing the Advantages of DevOps For IoT Performance Testing (2021, iotworldtoday.com):
https://www.iotworldtoday.com/2021/07/06/assessing-the-advantages-of-devops-for-iot-performance-testing

DevOps capabilities and practices

Continuous planning

- Objectives & Key Results (OKR)
- Lean product discovery
- Lean product definition
- Release planning
- Sprint planning
- Agile requirements
- Security requirements
- Architecture design
- Capacity planning
- UX architecture design
- Threat modeling
- Prioritization & estimation
- Demos & Retrospectives

Continuous integration

- Behavior-driven development (BDD)
- Test-driven development (TDD)
- Microservices & container development
- Mono-repo & Multi-repo
- Unit testing & code coverage
- Version control
- Git pull request
- Trunk-based policies
- Security static code scan
- CredScan
- Open Source software (OSS) component compliance
- Build parallel & serial pipeline

Continuous delivery

- Release pipeline
- Secure infrastructure deployment
- IaaS deployment
- PaaS deployment
- SaaS deployment
- Shared services
- Infrastructure as code (IaC)
- Change management
- Configuration management
- Release management
- Blue-green deployments
- Canary deployments
- Feature flags
- Trunk production ready

Continuous operations

- Site reliability engineering (SRE)
- Telemetry & monitoring
- Application performance monitoring
- Auto Failover, scaling, & DR
- Modern service management
- Secure access & application data
- High availability, security, cost & performance advisory
- Secure DevOps ChatOps
- Shift-right testing
- Secrets management
- Governance & GDPR support
- Automation & AIOps
- Continuity & resilience

Continuous quality ☆☆☆	Quality requirements	Shift-left testing	Governance & standards	Test automation	Compliance & audits	Shift right testing
Continuous security	Security architecture	Access & identity management	Application & data security	Security infrastructure	Secure operations	Governance, risk, & compliance
Continuous collaboration	Collaborative culture	Alignment & autonomy	Kanban collaboration	Wiki & Teams collaboration	ChatOps collaboration	Feature Team & SRE
Continuous improvement	Lead time & cycle time	Deployment frequency	Mean time to restore (MTTR)	Change fail percentage	Continuous feedback	Value stream mapping

Connection Between IoT and DevOps - CI

The **continuous integration** makes sure that the Internet of things (IoT) applications that belong to DevOps are considered as a **holistic system** or application. All these constituent parts depend upon one another, even **components that aren't under direct control, like sensors** that are embedded in machines.

This is slightly different from the traditional methods to DevOps, where **all constituents are under direct control**.

What is the Connection Between IoT and DevOps?:
https://www.kovair.com/blog/connection-between-iot-and-devops/

Microsoft

Connection Between IoT and DevOps - CD

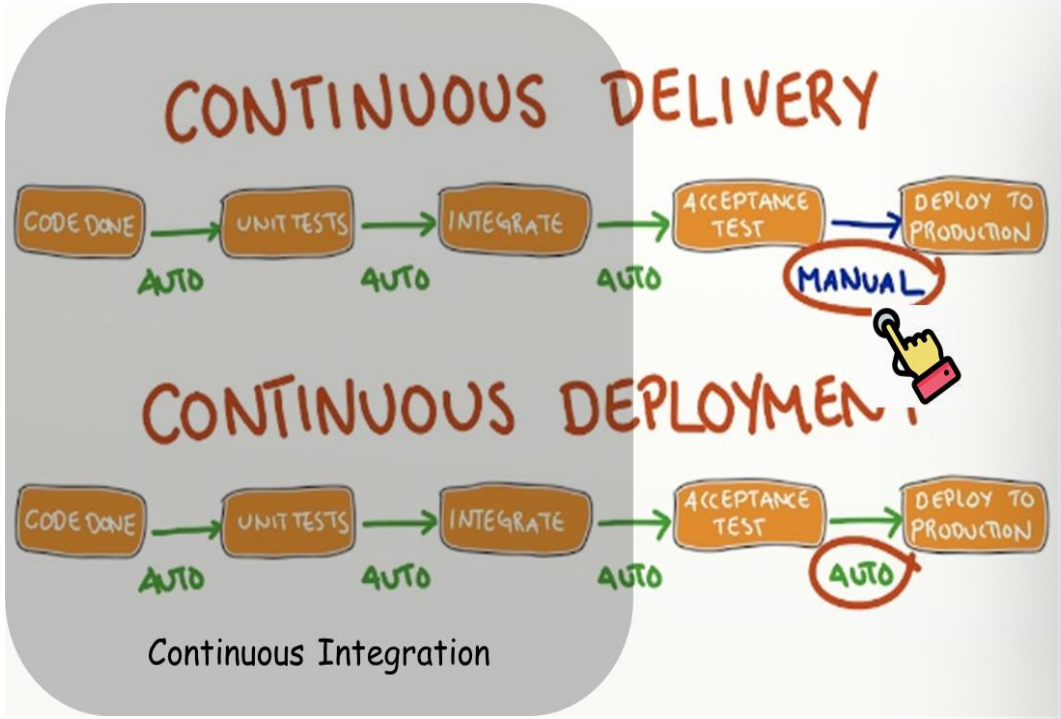
Deployment is considered to be the **most significant component** when it comes to **DevOps IoT**.

The application requires **living on the platform as well as the functioning and playing using remote devices**.

The **deployment processes get complicated** due to this. There can be some updates to the **application that exists on a central cloud** platform, and maybe even **firmware updates that exist on a device or remote sensor**

What is the Connection Between IoT and DevOps?:
https://www.kovair.com/blog/connection-between-iot-and-devops/

What is Continuous Integration



Continuous Integration is a Mindset and a Team Strategy

"… a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible. Many teams find that this approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly."

Source: https://martinfowler.com/articles/continuousIntegration.html

What are the Goals of Continuous Integration

The main purpose of Continuous Integration is to prevent developers stepping over each other code and eliminate integration issues.

CI Goals:

#1: Harness collaboration

#2: Enable parallel development

#3: Minimize integration debt

#4: Act as a quality gate

#5: Automate Everything!

Elements of Continuous Integration

Version Control System

- ❑ Application / Test Code
- ❑ Configuration
- ❑ Database Definition
- ❑ Seed / Test Data
- ❑ Build / Deployment Scripts

Branching Strategy

- ❑ Have a Branching strategy
- ❑ Keep changes small
- ❑ Commit frequently
- ❑ At least once per day for each developer

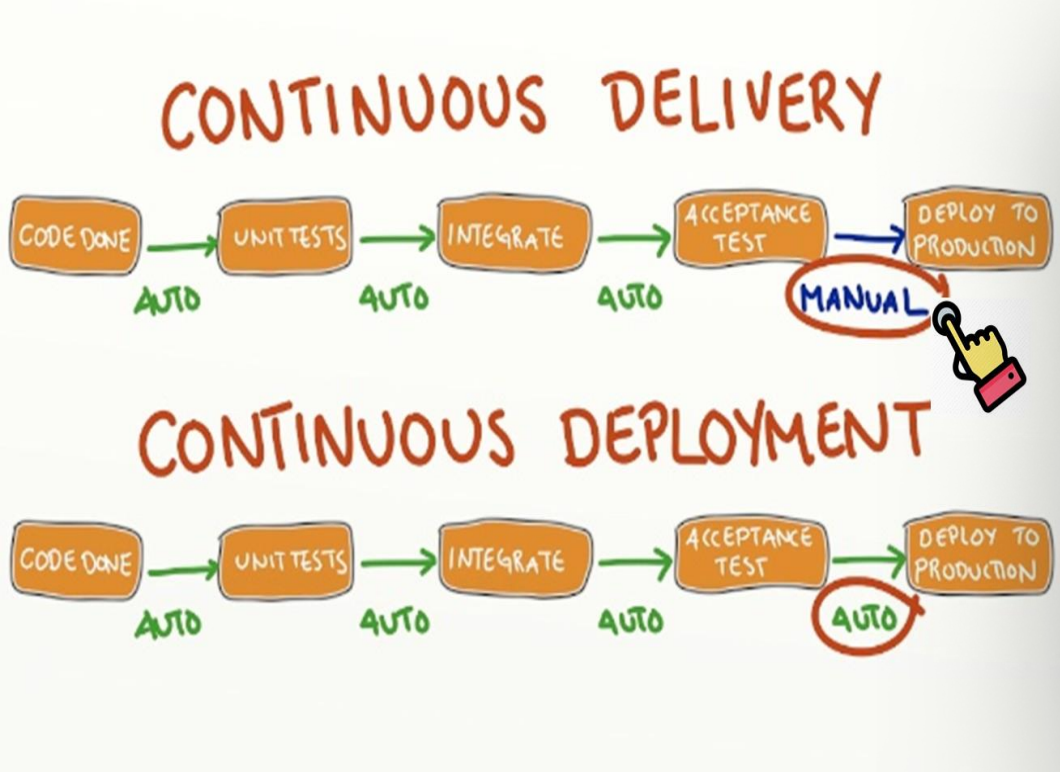
Automated Build

- ❑ Runs after every commit/check-in
- ❑ Checks quality of the commit (Shift Left Testing)
- ❑ Separate Build machine (Build Server)
- ❑ Must run fast

Agreement of the Team

Continuous integration is a practice, not a tool. It requires a degree of commitment and discipline from your development team.

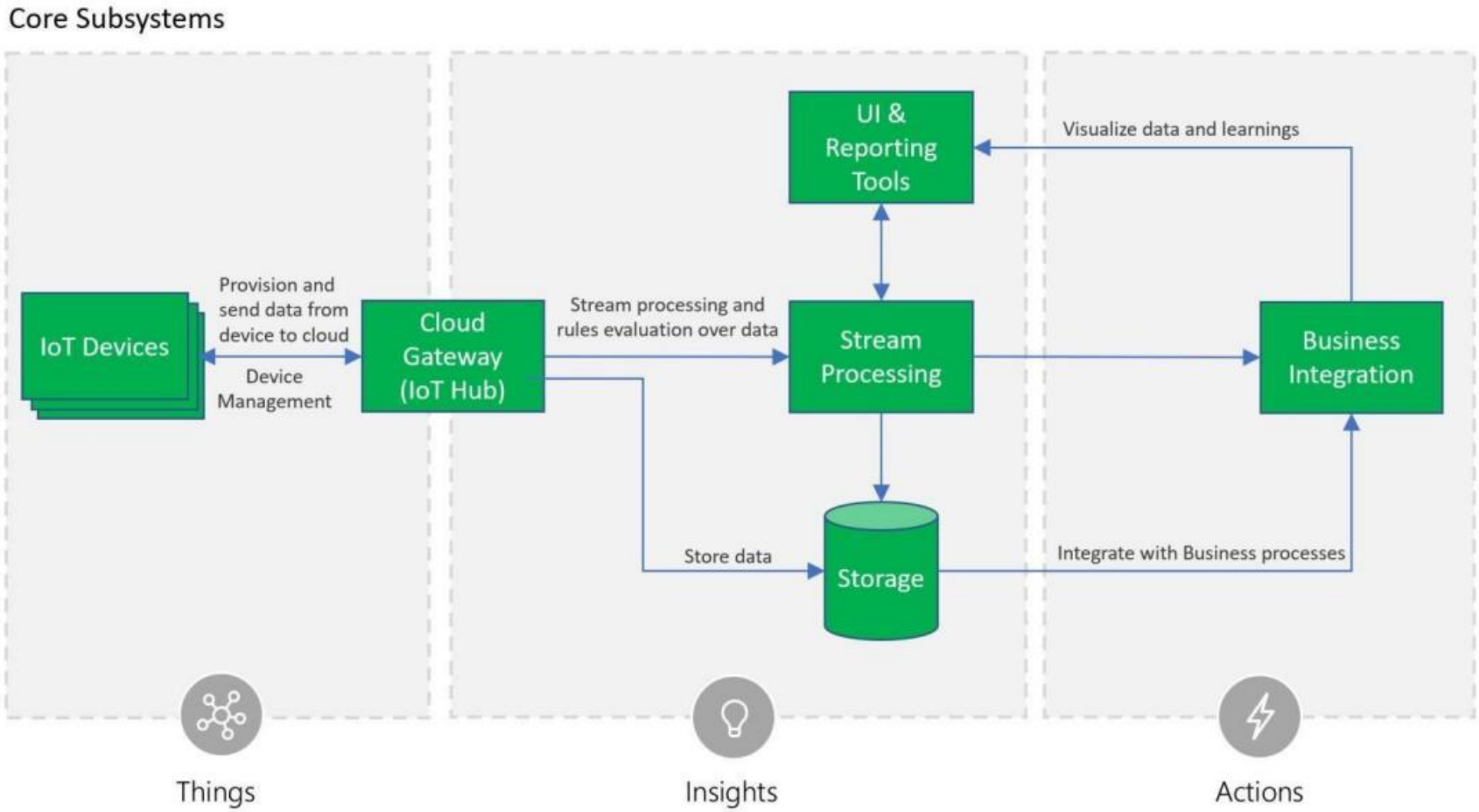
What is Continuous Delivery



Continuous delivery is a software engineering approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time and, when releasing the software, doing so manually. It aims at building, testing, and releasing software with greater speed and frequency. The approach helps reduce the cost, time, and risk of delivering changes by allowing for more incremental updates to applications in production. A straightforward and repeatable deployment process is important for continuous delivery.

Source: https://en.wikipedia.org/wiki/Continuous_delivery

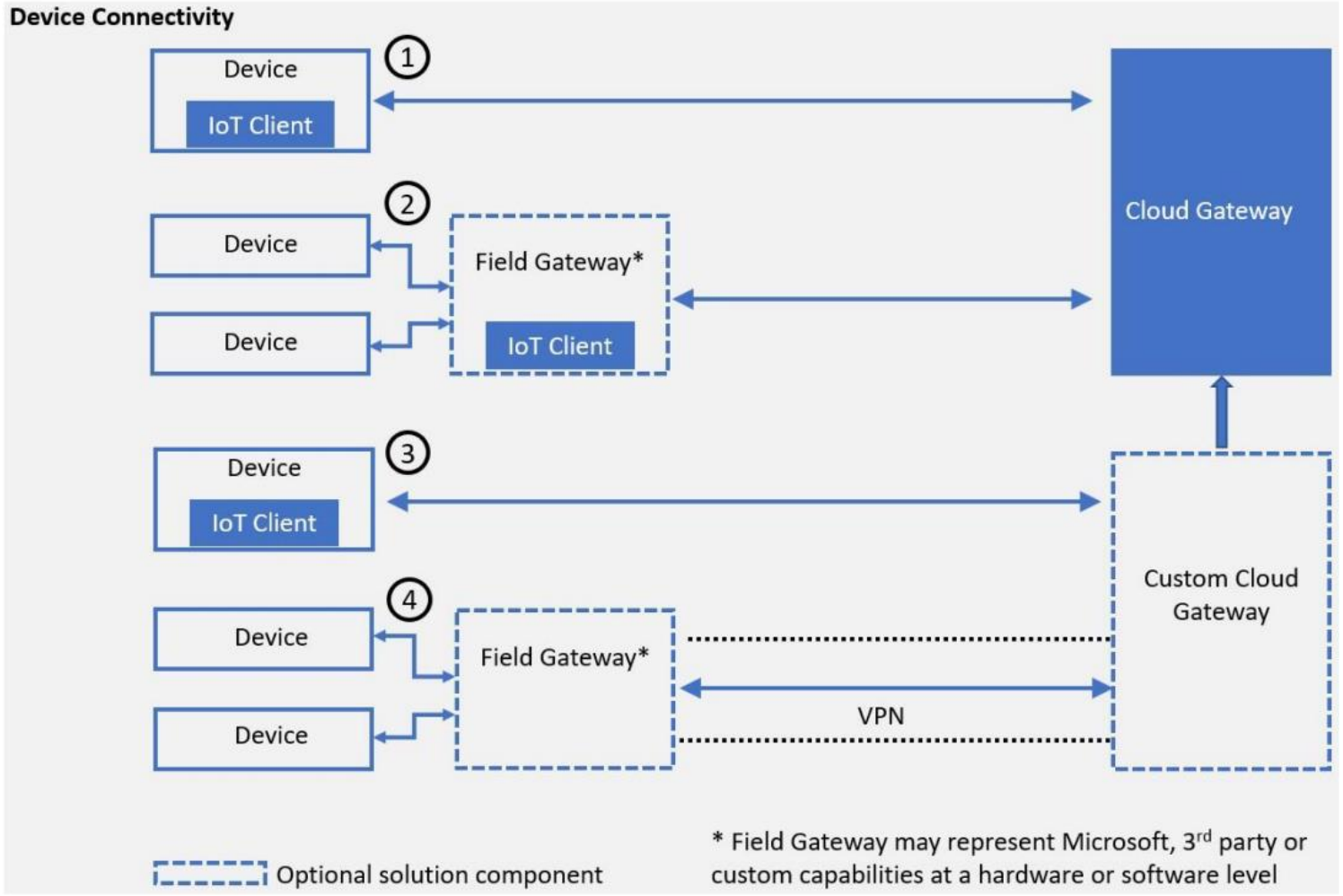
Microsoft Azure IoT Reference Architecture



Microsoft Azure IoT Reference Architecture V 2.1.1 (2021)
https://azure.microsoft.com/en-us/resources/microsoft-azure-iot-reference-architecture/

Microsoft

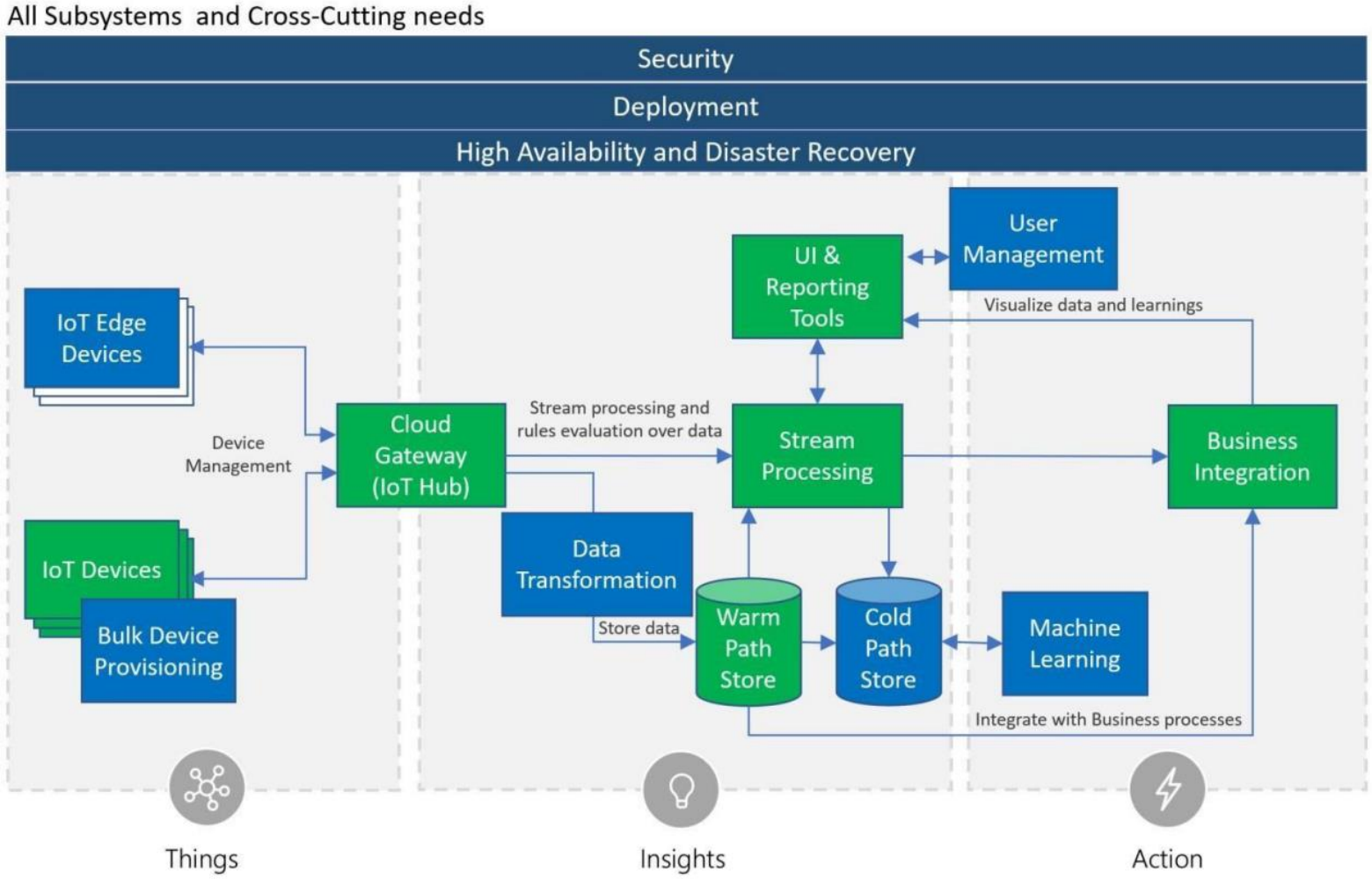
Device Connectivity



Microsoft Azure IoT Reference Architecture V 2.1.1 (2021)
https://azure.microsoft.com/en-us/resources/microsoft-azure-iot-reference-architecture/

Microsoft

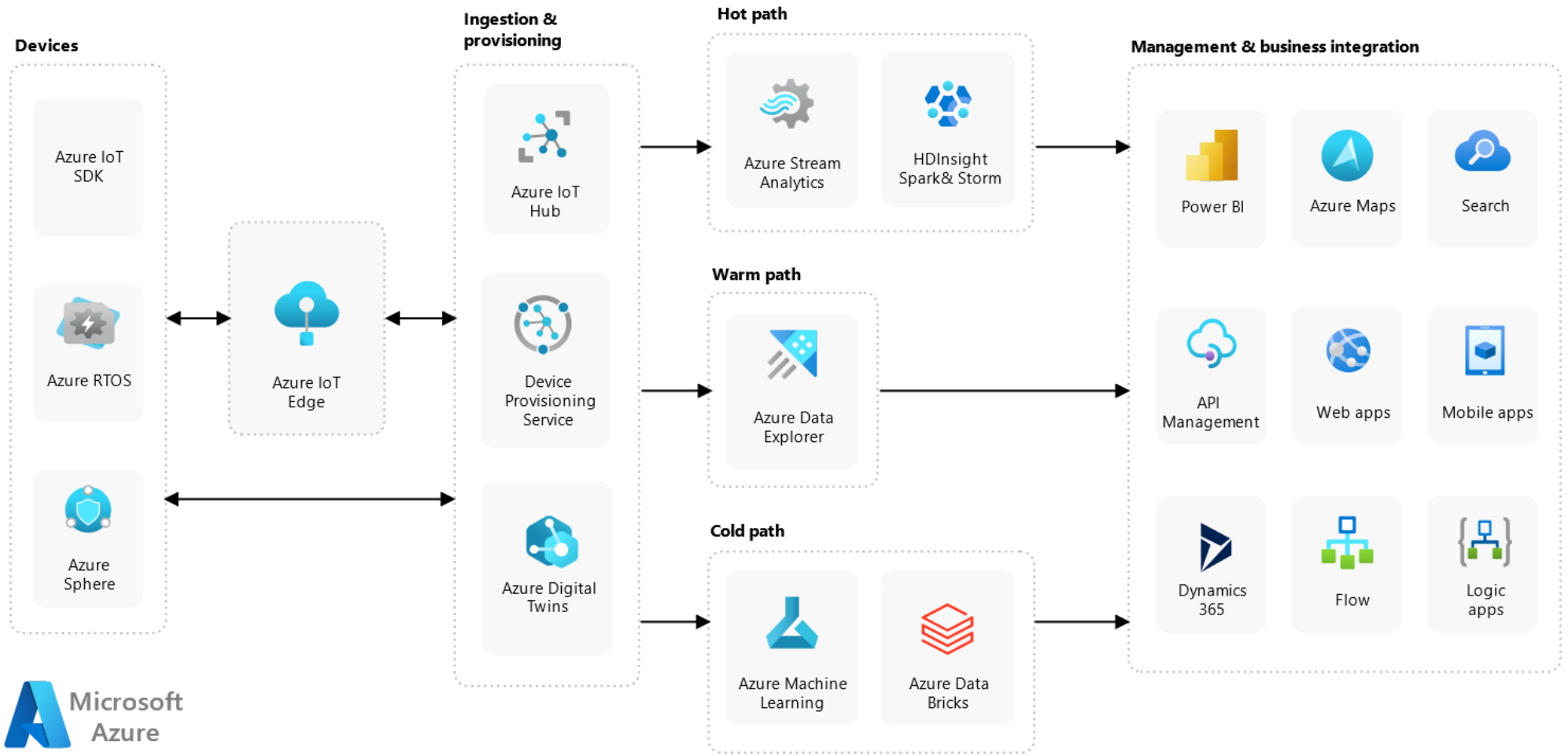
Microsoft Azure IoT Reference Architecture



Microsoft Azure IoT Reference Architecture V 2.1.1 (2021)
https://azure.microsoft.com/en-us/resources/microsoft-azure-iot-reference-architecture/

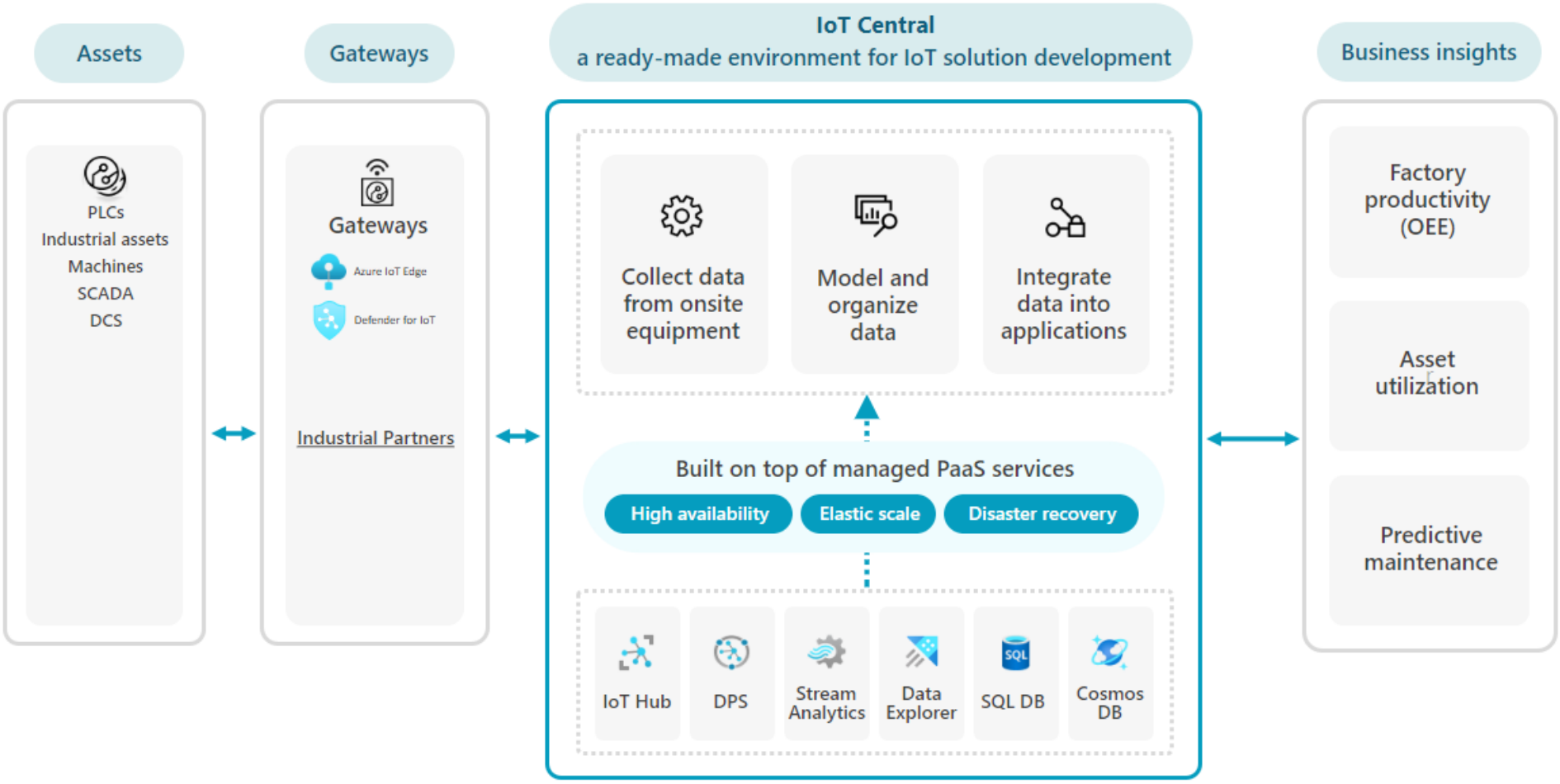
Microsoft

Azure IoT Reference Architecture



Microsoft

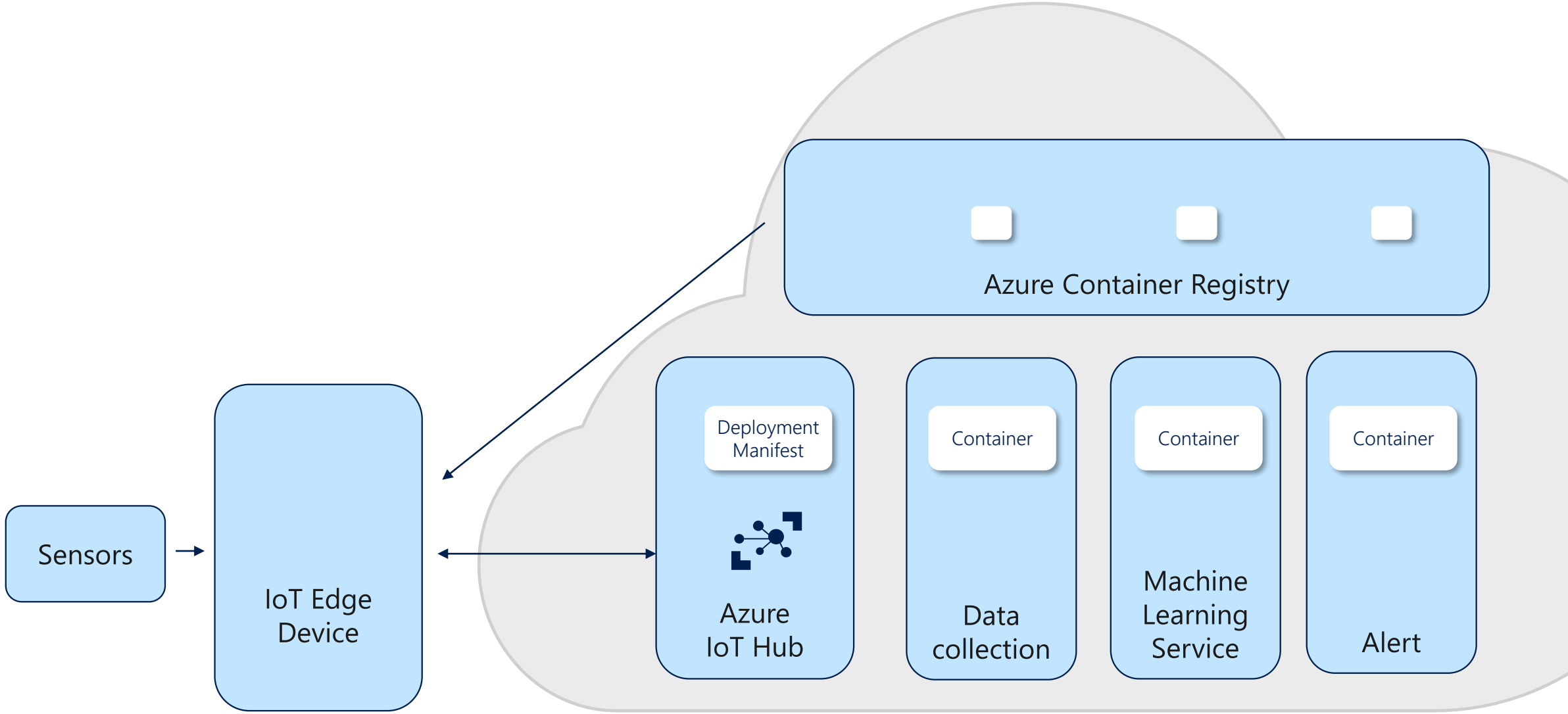
IoT (IIoT) architecture with Azure IoT Central



Industrial IoT (IIoT) architecture patterns with Azure IoT Central 04/29/2022
https://docs.microsoft.com/en-us/azure/iot-central/core/concepts-iiot-architecture

Microsoft

Bringing Compute Where The Data As

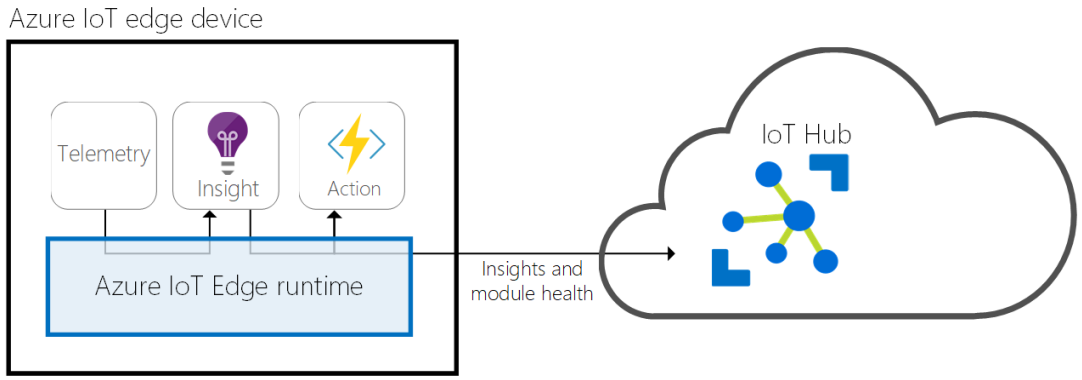


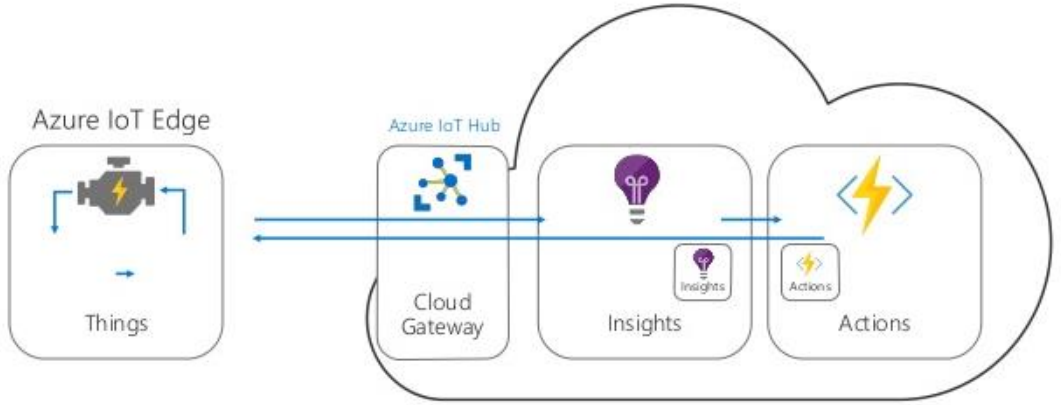
Microsoft

Azure IoT Edge

- Built on container technology as ‘modules’
- Modules support Python, NodeJS, .Net Core, Java, & C
- Low-latency AMQP / MQTT data transport
- Operate in offline / intermittent network conditions
- Supports Linux X64 | ARM32/64 , Windows X64

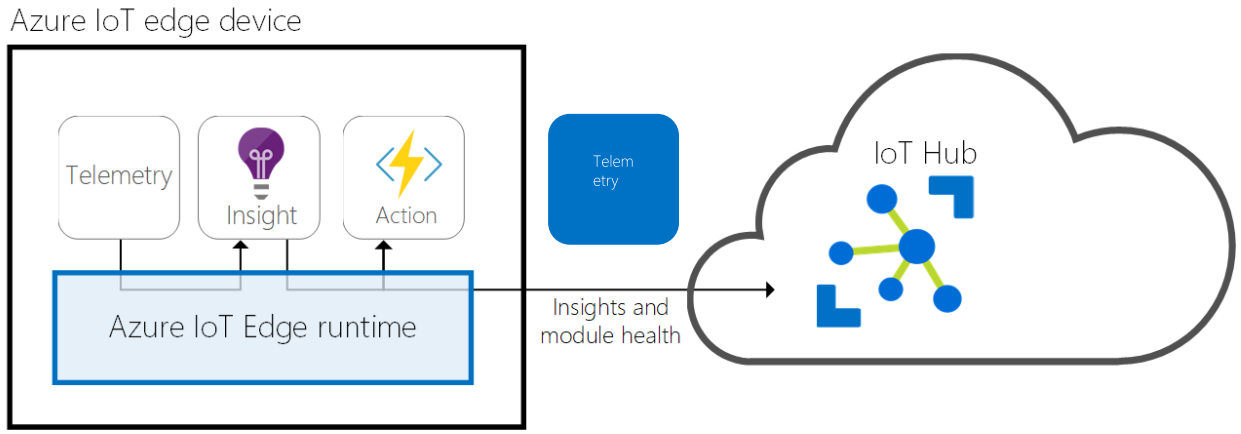
OSS and available @ https://github.com/Azure/iotedge





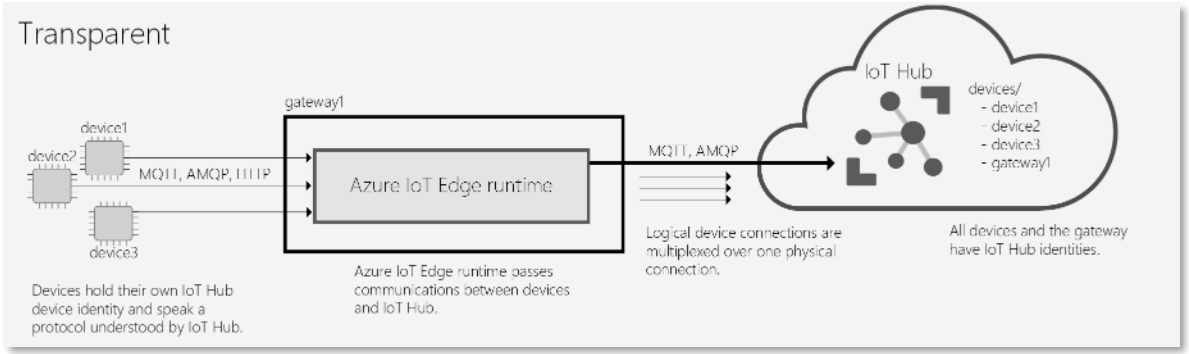
Microsoft

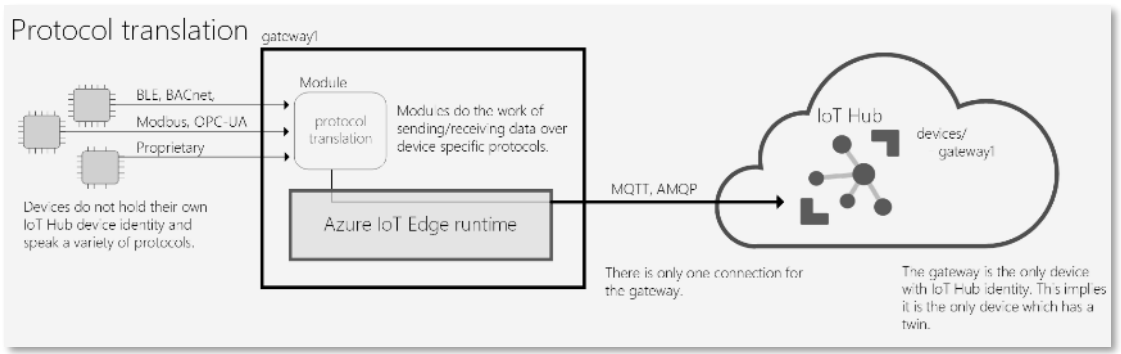
IoT Edge Runtime

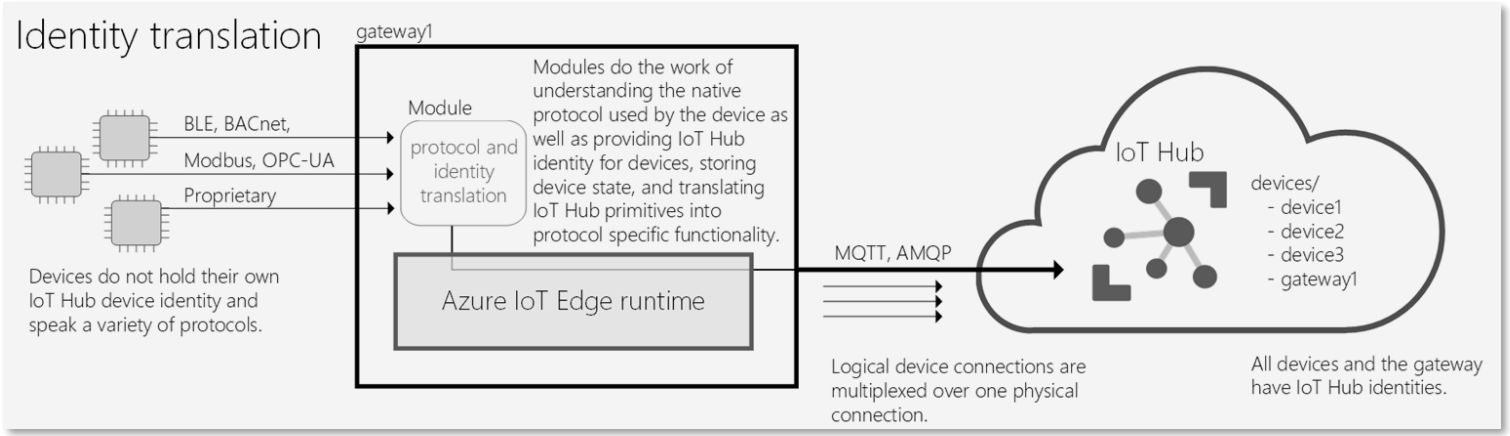


- Installs and updates workloads on the device.
- Maintains Azure IoT Edge security standards on the device.
- Ensures that IoT Edge modules are always running.
- Reports module health to the cloud for remote monitoring.
- Facilitates communication between downstream leaf devices and the IoT Edge device.
- Facilitates communication between modules on the IoT Edge device.
- Facilitates communication between the IoT Edge device and the cloud

Azure IoT Edge as a Gateway (Patterns)

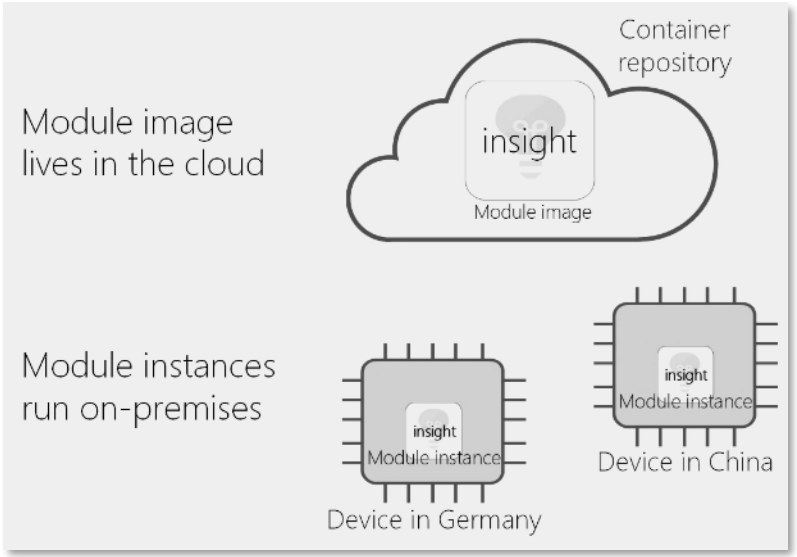






Microsoft

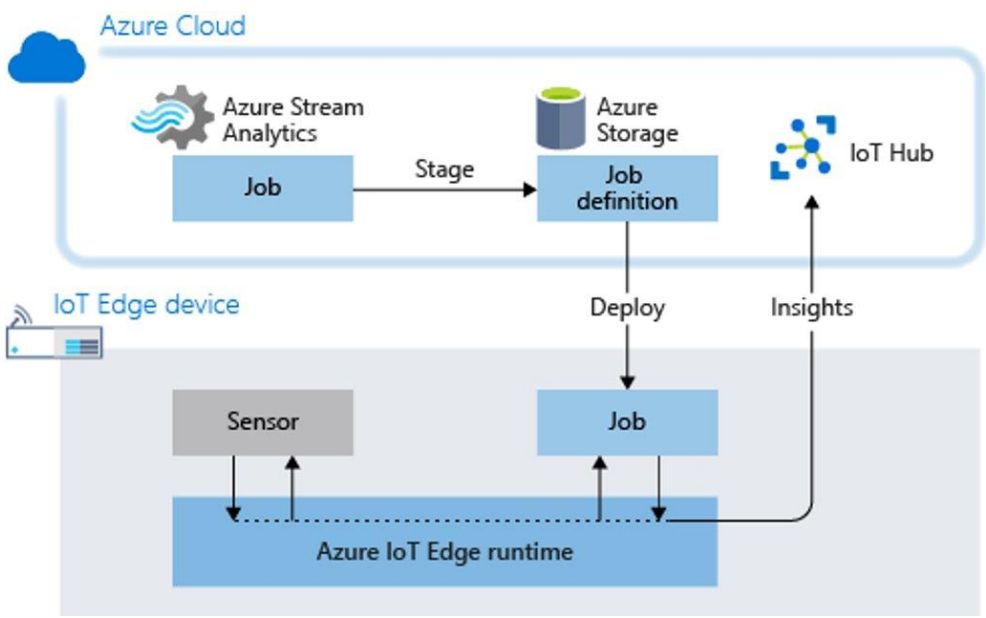
IoT Edge Module



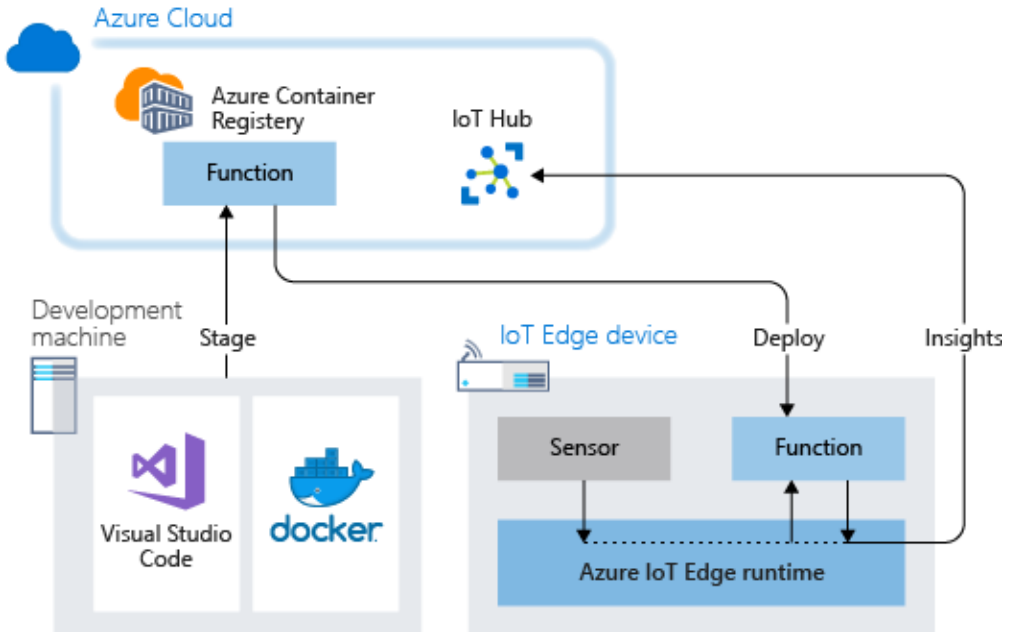
- A **module image** is a package containing the software that defines a module.
- A **module instance** is the specific unit of computation running the module image on an IoT Edge device. The module instance is started by the IoT Edge runtime.
- A **module identity** is a piece of information (including security credentials) stored in IoT Hub, that is associated to each module instance.
- A **module twin** is a JSON document stored in IoT Hub, that contains state information for a module instance, including metadata, configurations, and conditions.
- SDKs to develop custom modules in multiple languages (C#, C, Python, Java, Node.JS)

Azure Services as Edge Modules

Azure Stream Analytics



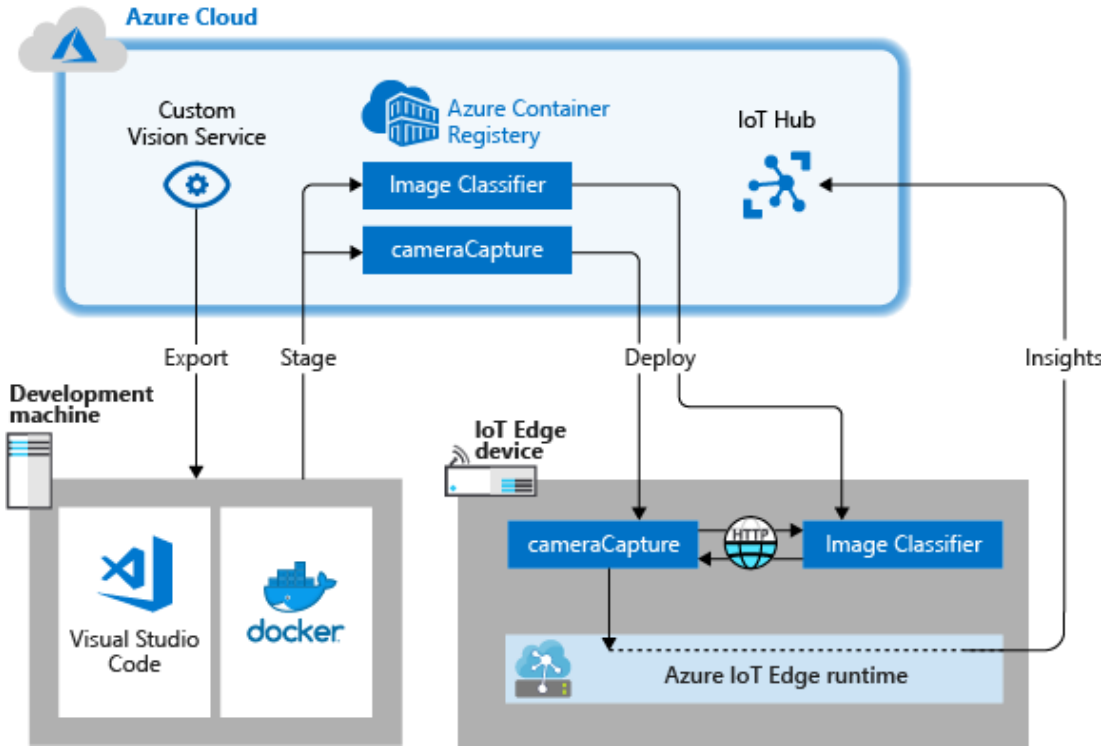
Azure Functions



Microsoft

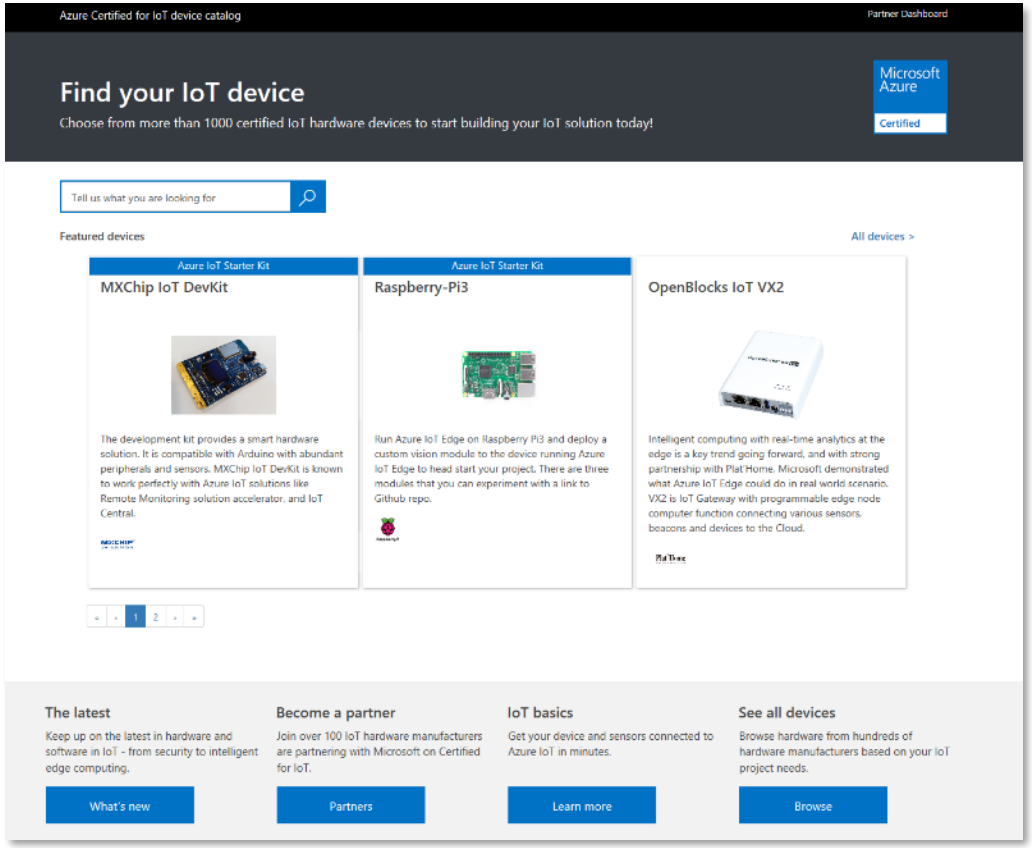
AI/ML Modules

- Cognitive Services including:
  - Text Analytics - Key Phrase Extraction, Language Detection, Sentiment Analysis
  - Face – Detection, Verification, and Emotion
  - Computer Vision – Text Recognition, custom vision module allows to finds whatever you train it to find
- Azure Machine Learning at the Edge
  - Export AML models to run a Edge modules



Microsoft

Certified Hardware for Azure IoT Edge



Azure certified for IoT device catalog

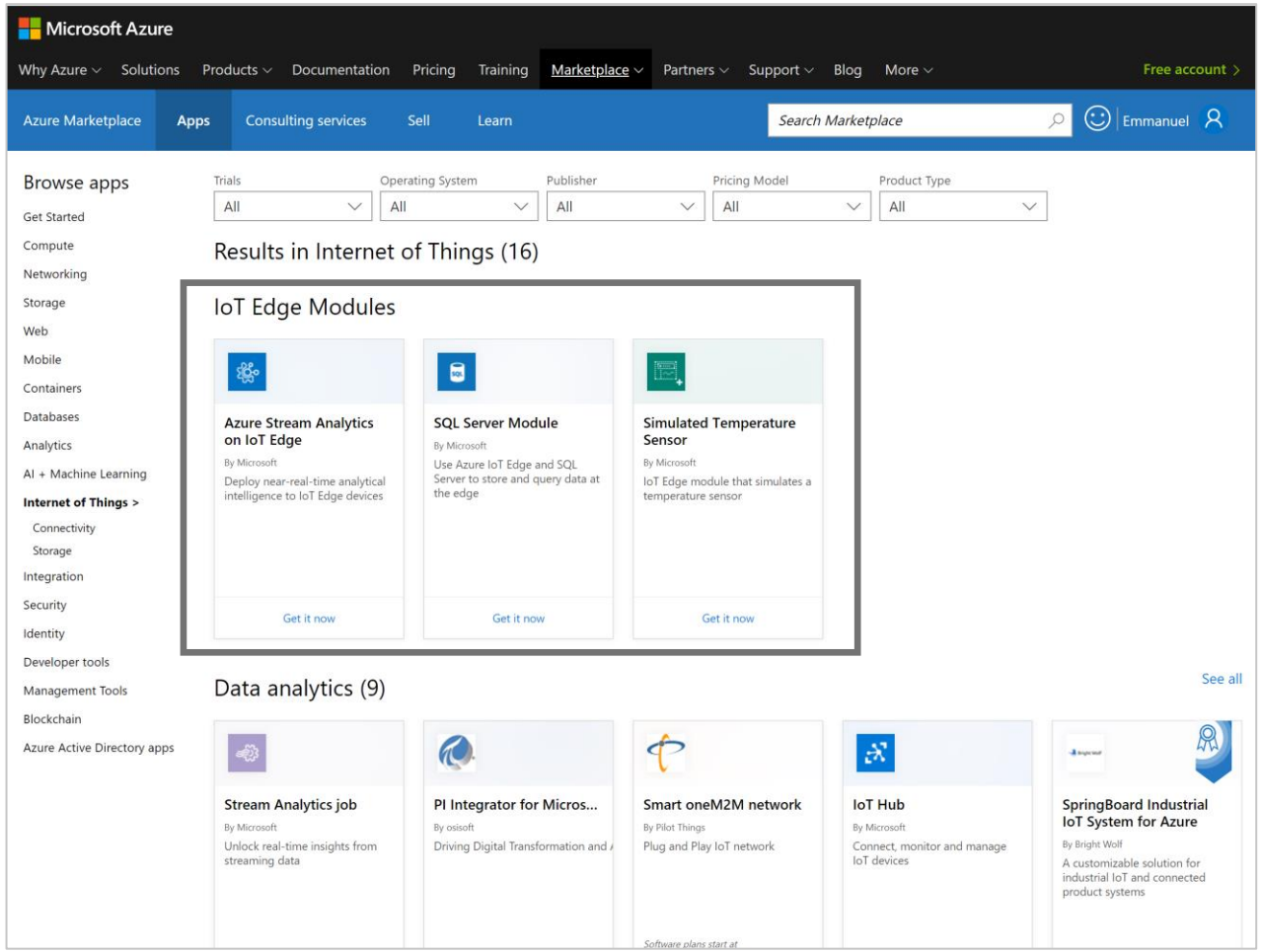
- Provides an easy and intuitive way to discover the right IoT device for intended use case
- More than 1000 devices already listed in the device catalog
- Start at https://catalog.azureiotsolutions.com/

Expanded device catalog with IoT Edge certified hardware

- Capability based certification for extensibility and long-term sustainability
- Each capability has N number of levels. Level 1 being lowest
- Select the best device most suitable for your IoT application

Microsoft

Azure IoT Edge Module on Azure Marketplace



Customers:

Save development effort

Discover and integrate certified pre-built modules

Partners:

Showcase with wide reach

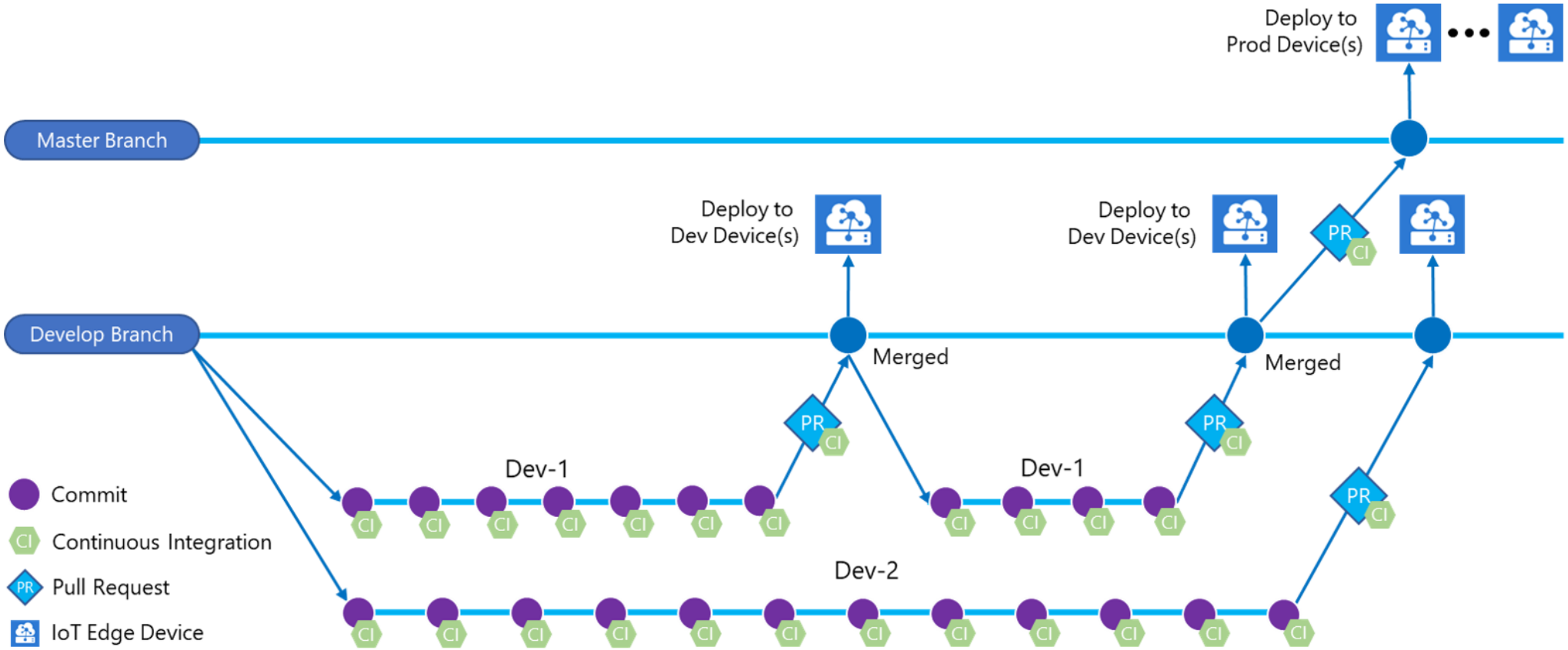
Certified modules gives IoT customers peace of mind for their project

Work with a leader in IoT

Market with Microsoft, and collaborate with other Microsoft IoT partners

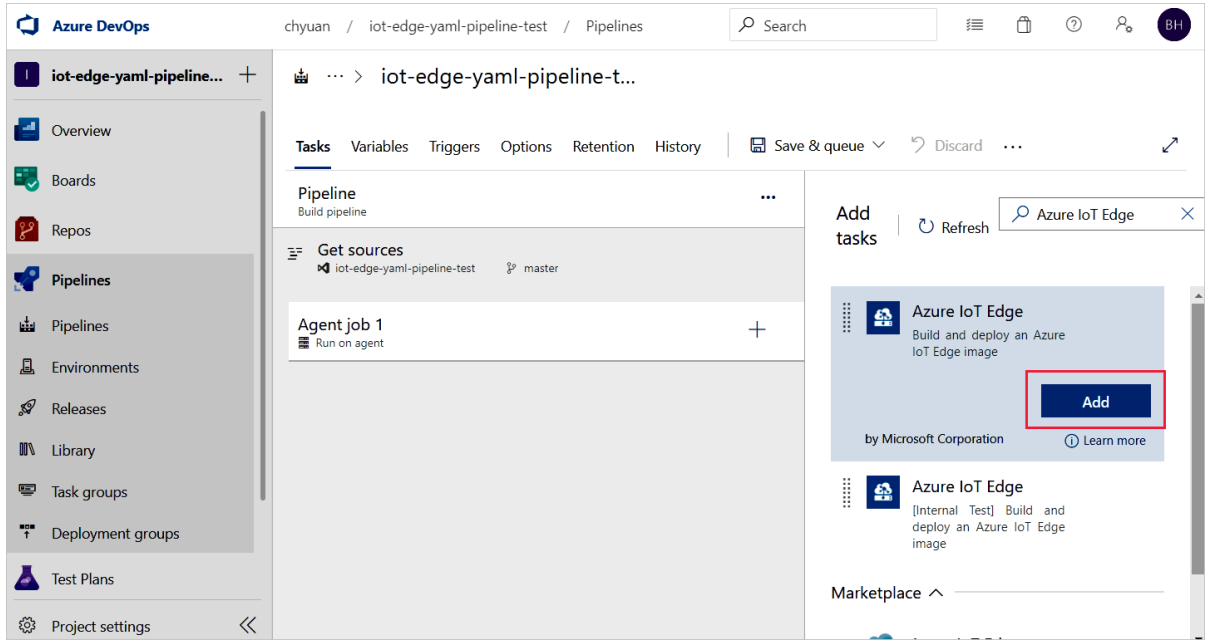
Microsoft

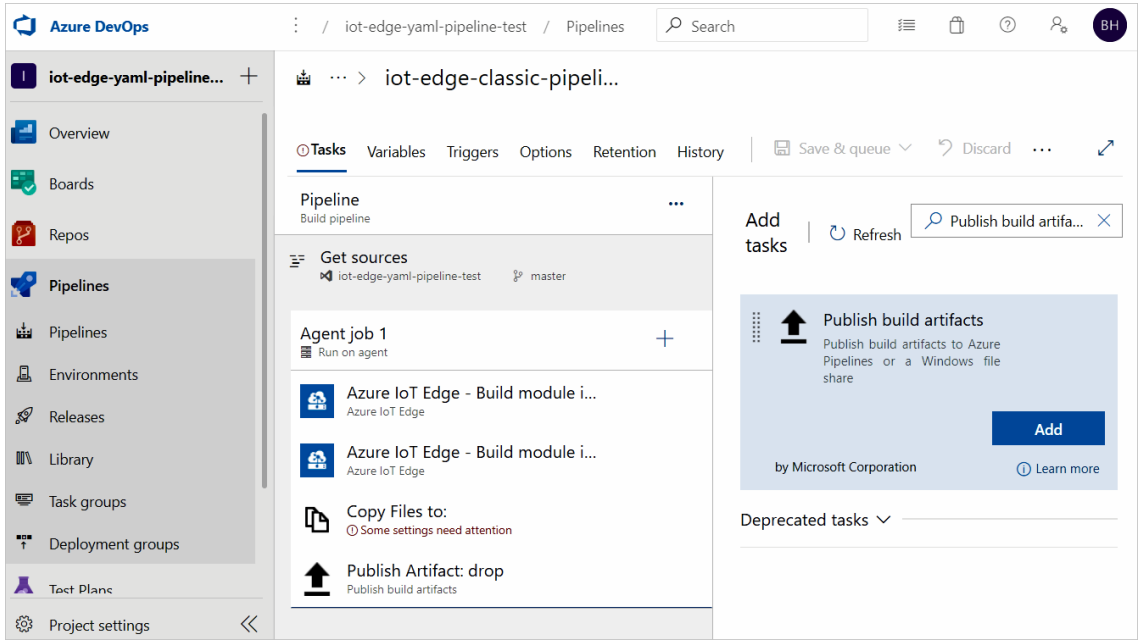
DevOps Process with IoT Edge



Microsoft

Implement CI/CD for IoT Edge

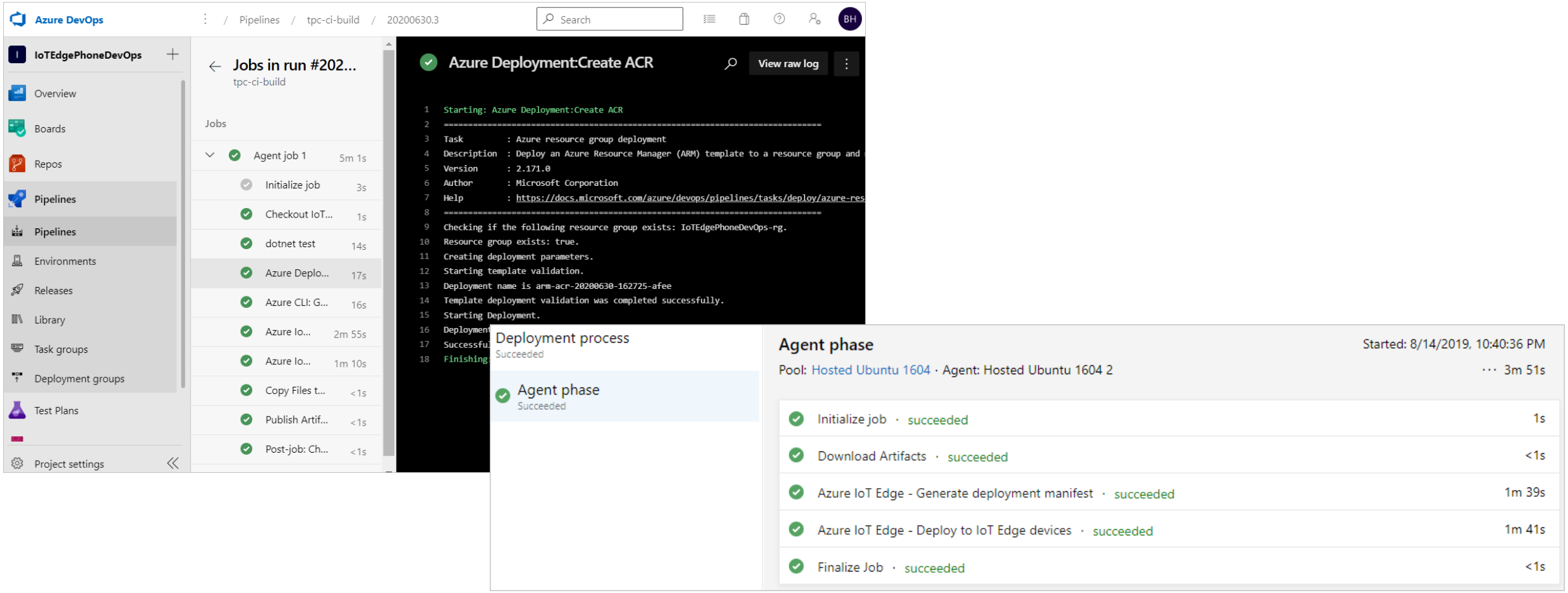




Implement CI/CD for IoT Edge:
https://docs.microsoft.com/en-us/azure/iot-edge/how-to-continuous-integration-continuous-deployment-classic?view=iotedge-1.4

Microsoft

Implement CI/CD for IoT Edge

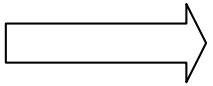


Implement CI/CD for IoT Edge:
https://docs.microsoft.com/en-us/azure/iot-edge/how-to-continuous-integration-continuous-deployment-classic?view=iotedge-1.4

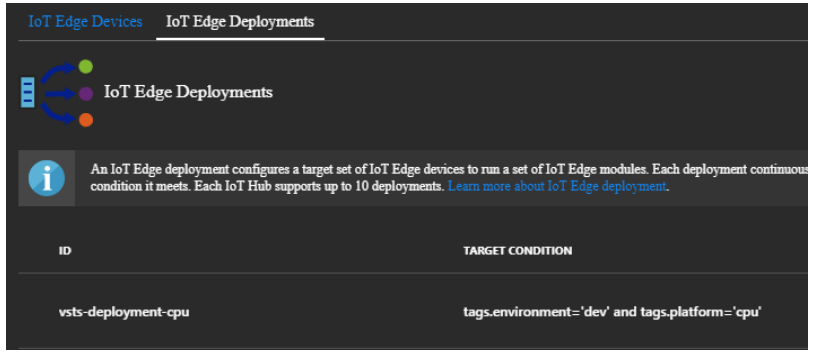
Microsoft

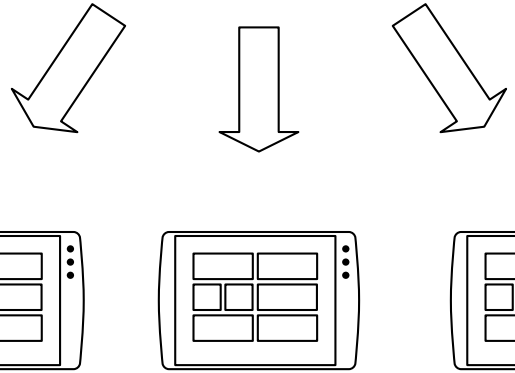
IoT Edge Deployment Process

Publish Deployment to IoT Hub for targeted devices

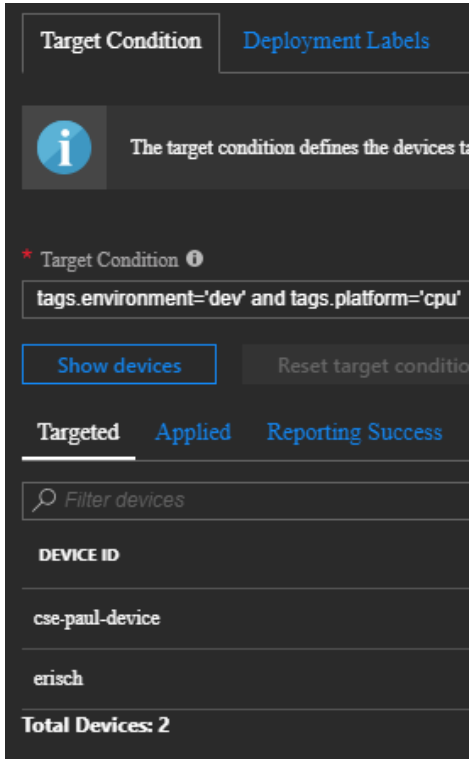


Iot Edge Deployment created





Iot Edge Devices matching target criteria in respective DeviceTwin receive updated deployment configuration

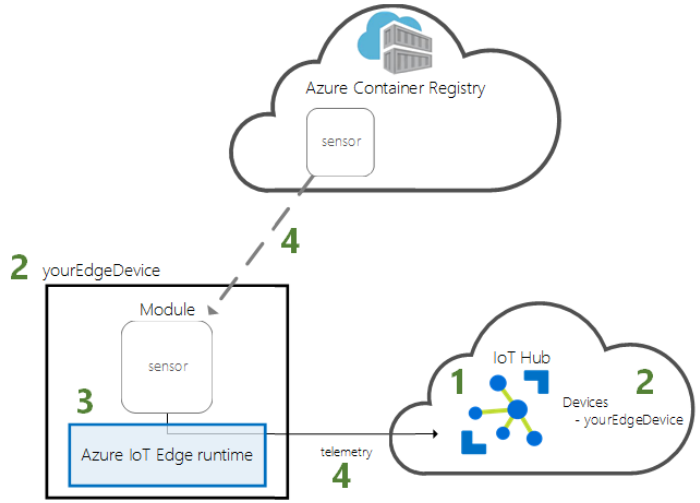


Microsoft

System Modules

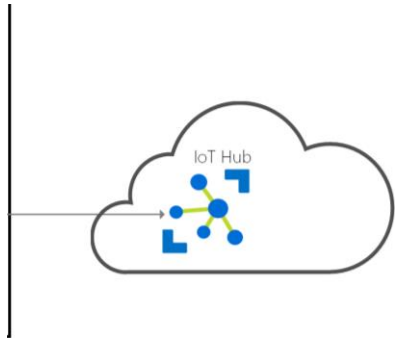
edge-agent:

Deployment & Container orchestration
Ensures module uptime



edge-hub:

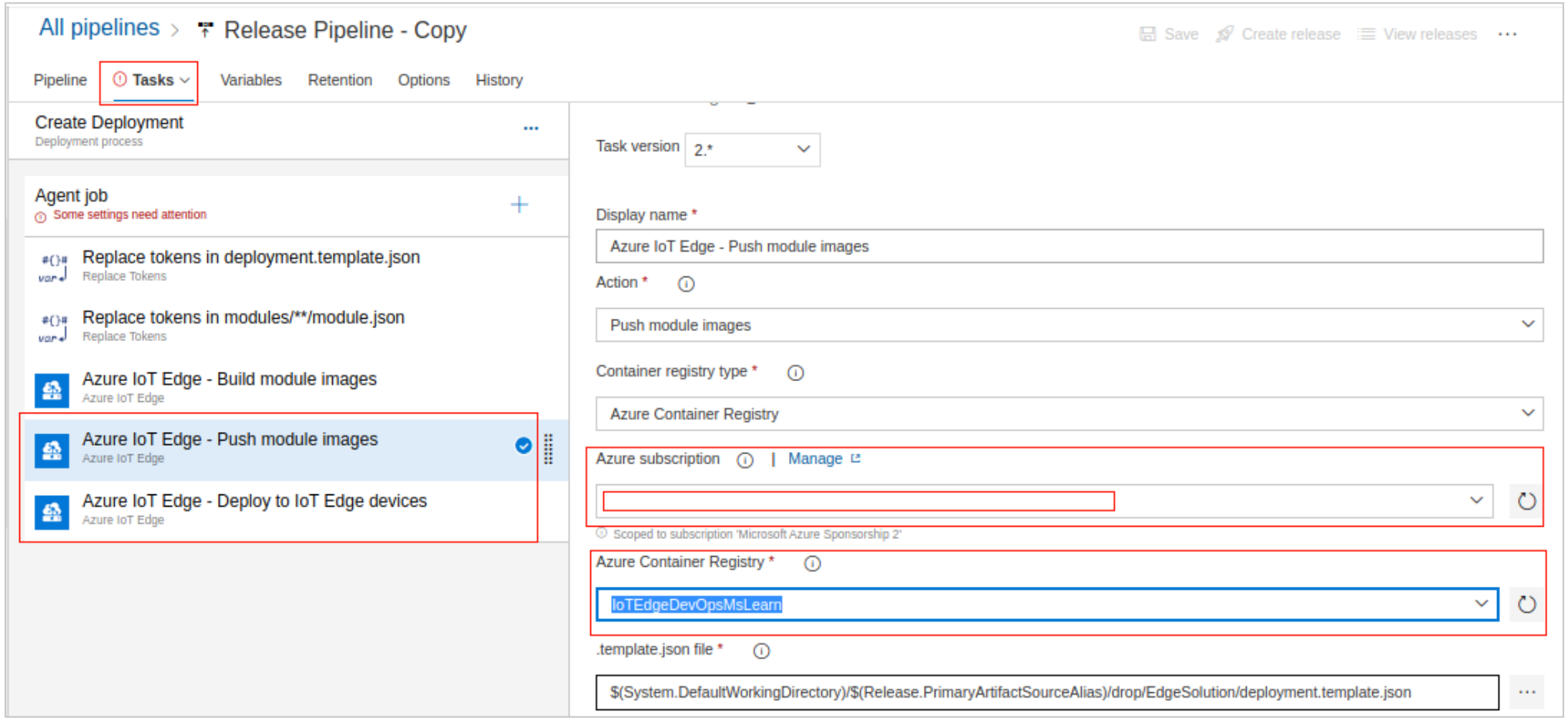
Communication to/from Azure IoT Hub
Inter-module communication



azure-pipelines-tasks/Tasks/AzureIoTEdgeV2/
https://github.com/microsoft/azure-pipelines-tasks/tree/master/Tasks/AzureIoTEdgeV2

Microsoft

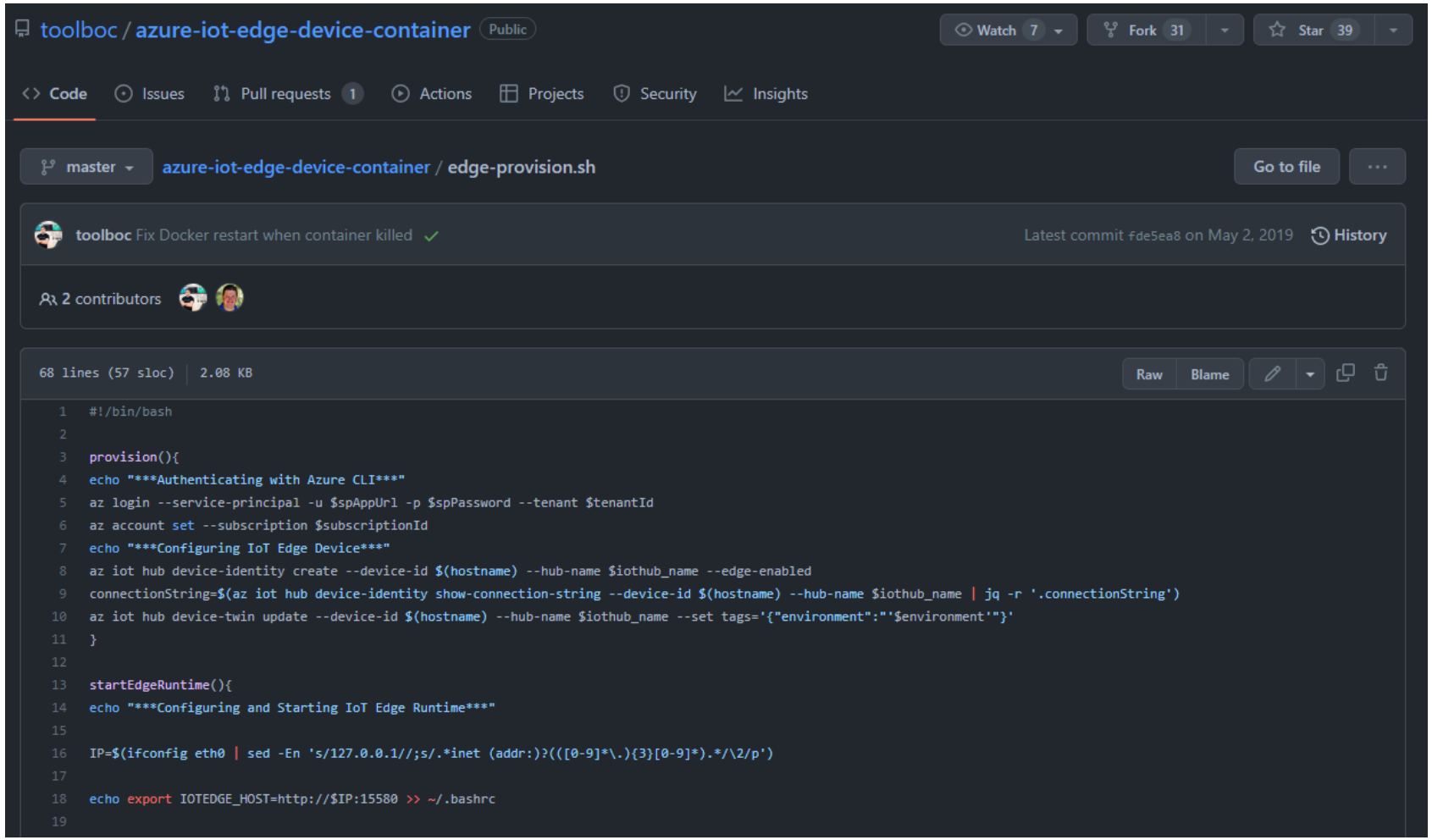
Implement CI/CD for IoT Edge – smoke tests



Exercise - Create a CD release pipeline for IoT Edge with a smoke test:
https://docs.microsoft.com/en-us/training/modules/implement-cicd-iot-edge/6-exercise-create-release-pipeline-iot-edge

Microsoft

Implement CI/CD for IoT Edge – device container



azure-iot-edge-device-container
https://github.com/toolboc/azure-iot-edge-device-container

Microsoft

Implement CI/CD for IoT Edge – smoke tests design

Designing an automated smoke-testing strategy provides many benefits.

•Deploying to virtual IoT devices is cheaper than deploying to physical IoT devices.

•A successful smoke test ensures that the program is ready for further testing.

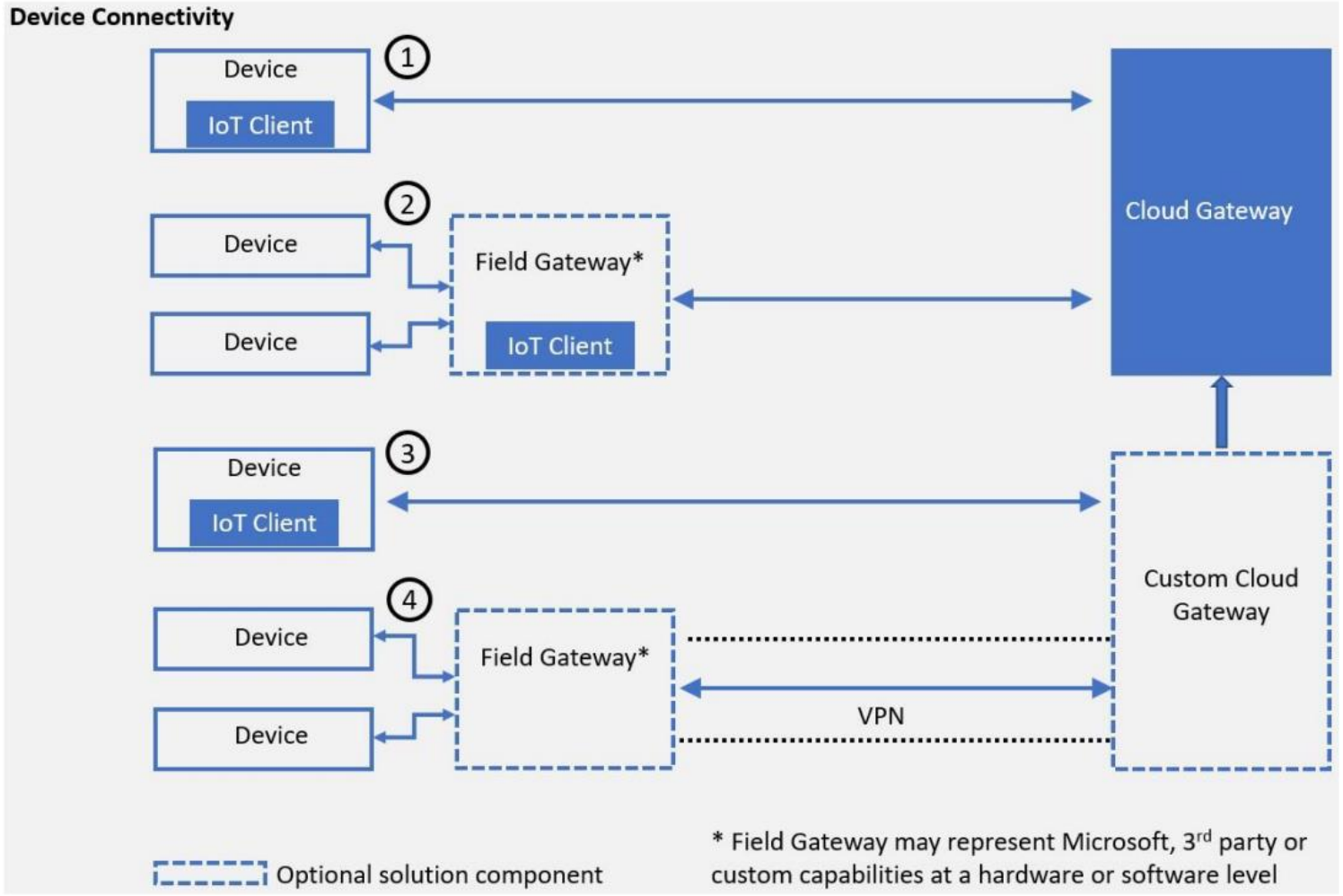
•Automated smoke testing avoids expensive rework for more sophisticated tests.

•Optimizing the time for your developer and the test teams.

To implement this solution, you'll create a smoke test as a subset of your main functionality. The test should be self-scoring, automated, and able to run on virtual IoT devices.

Exercise - Create a CD release pipeline for IoT Edge with a smoke test:
https://docs.microsoft.com/en-us/training/modules/implement-cicd-iot-edge/6-exercise-create-release-pipeline-iot-edge

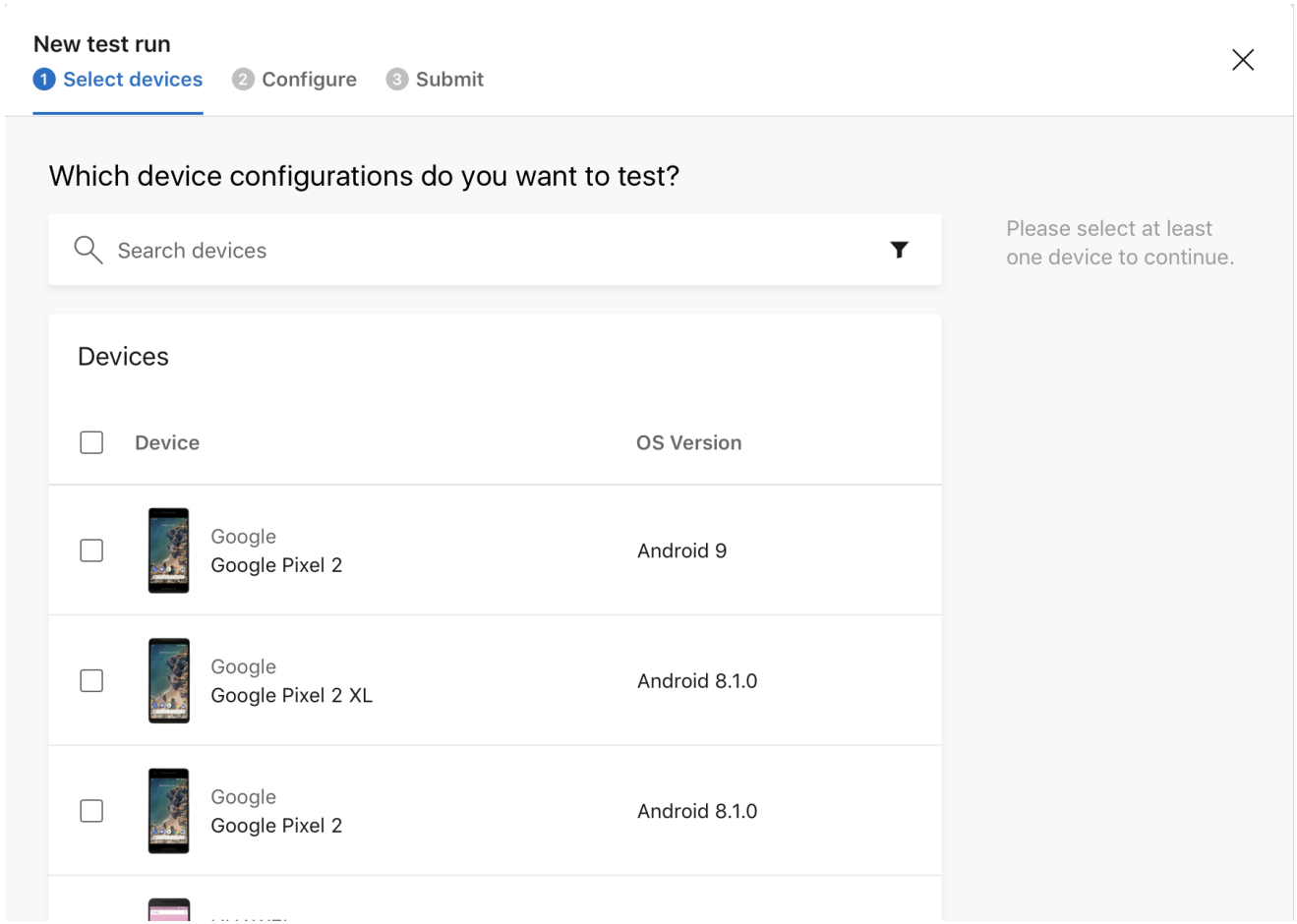
Device Connectivity



Microsoft Azure IoT Reference Architecture V 2.1.1 (2021)
https://azure.microsoft.com/en-us/resources/microsoft-azure-iot-reference-architecture/

Microsoft

DevOps for Mobile Apps and Devices



App Center Test - Testing Builds on Real Devices - https://docs.microsoft.com/en-us/appcenter/build/build-test-integration
DevOps for mobile apps challenges and best practices- https://developer.ibm.com/articles/mo-bestdevops-mobileapps/

Microsoft

Build you own -Mobile Apps Test center

Run apps on a hardware device
https://developer.android.com/studio/run/device

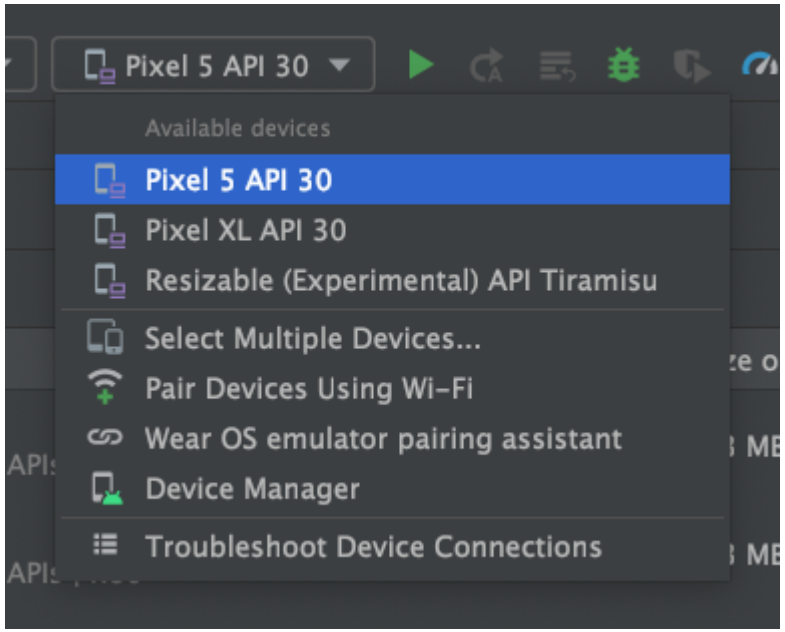
Microsoft

Glimpse of a real - Mobile Apps Test center

Firebase Test Lab
https://firebase.google.com/docs/test-lab/

Microsoft

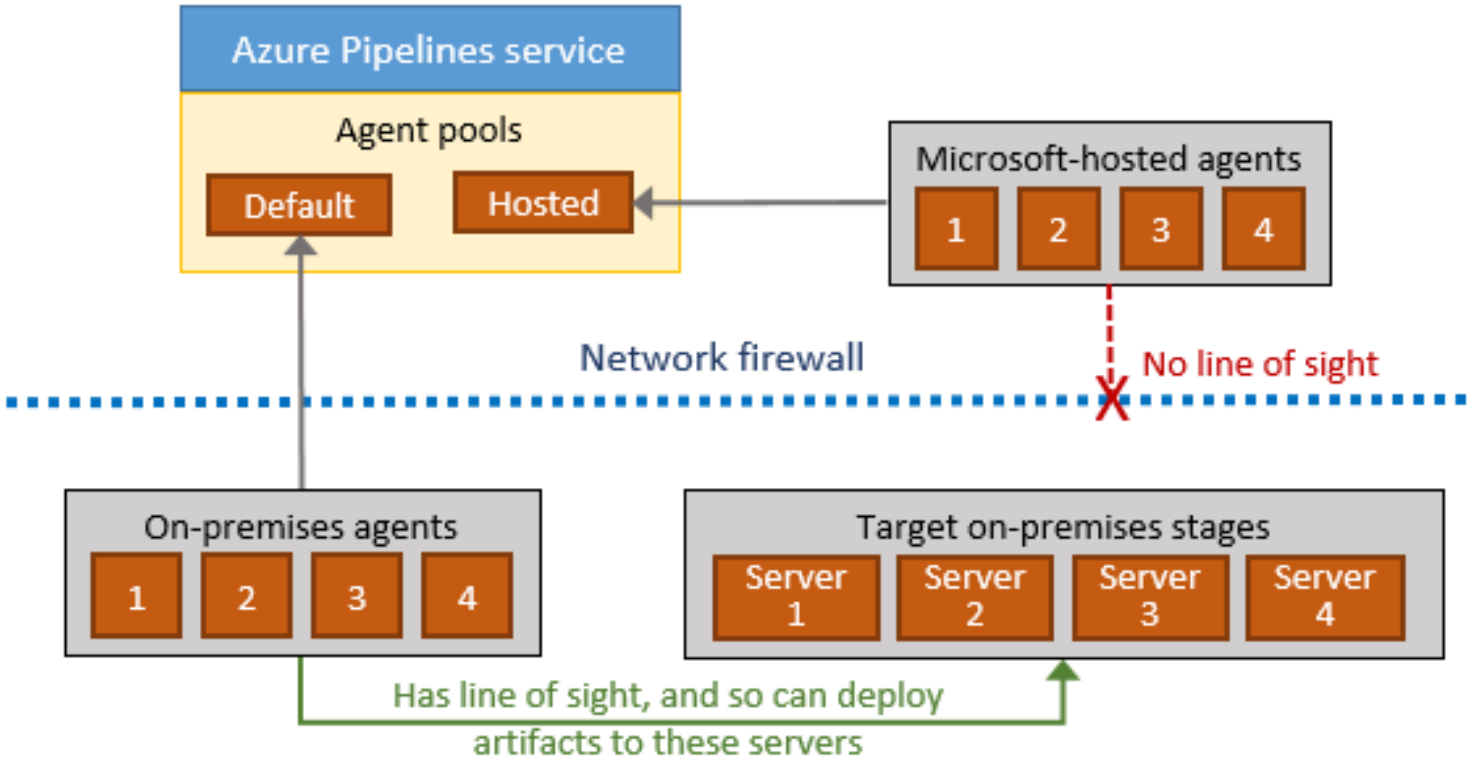
Emulators



Run apps on the Android Emulator
https://developer.android.com/studio/run/emulator

Microsoft

DevOps Self-Hosted build agents



Azure Pipelines agents
https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/agents?view=azure-devops

Microsoft

DevOps – Self Hosted Runner on Pi

GitHub Runner on Raspberry Pi
https://www.linkedin.com/pulse/github-runner-raspberry-pi-alexander-yochev/

Microsoft

Mobile Devices Seamless updates

A/B (Seamless) system updates

Modern Android devices have two copies of each partition (A and B) and can apply an update to the currently unused partition while the system is running but idle. A/B devices do not need space to download the update package because they can apply the update as they read it from the network; this is known as streaming A/B.

Non-A/B system updates

Older Android devices have a dedicated recovery partition containing the software needed to unpack a downloaded update package and apply the update to the other partitions.

OTA Updates
https://source.android.com/docs/core/ota

Device update

OTA

Over-The-Air

FOTA

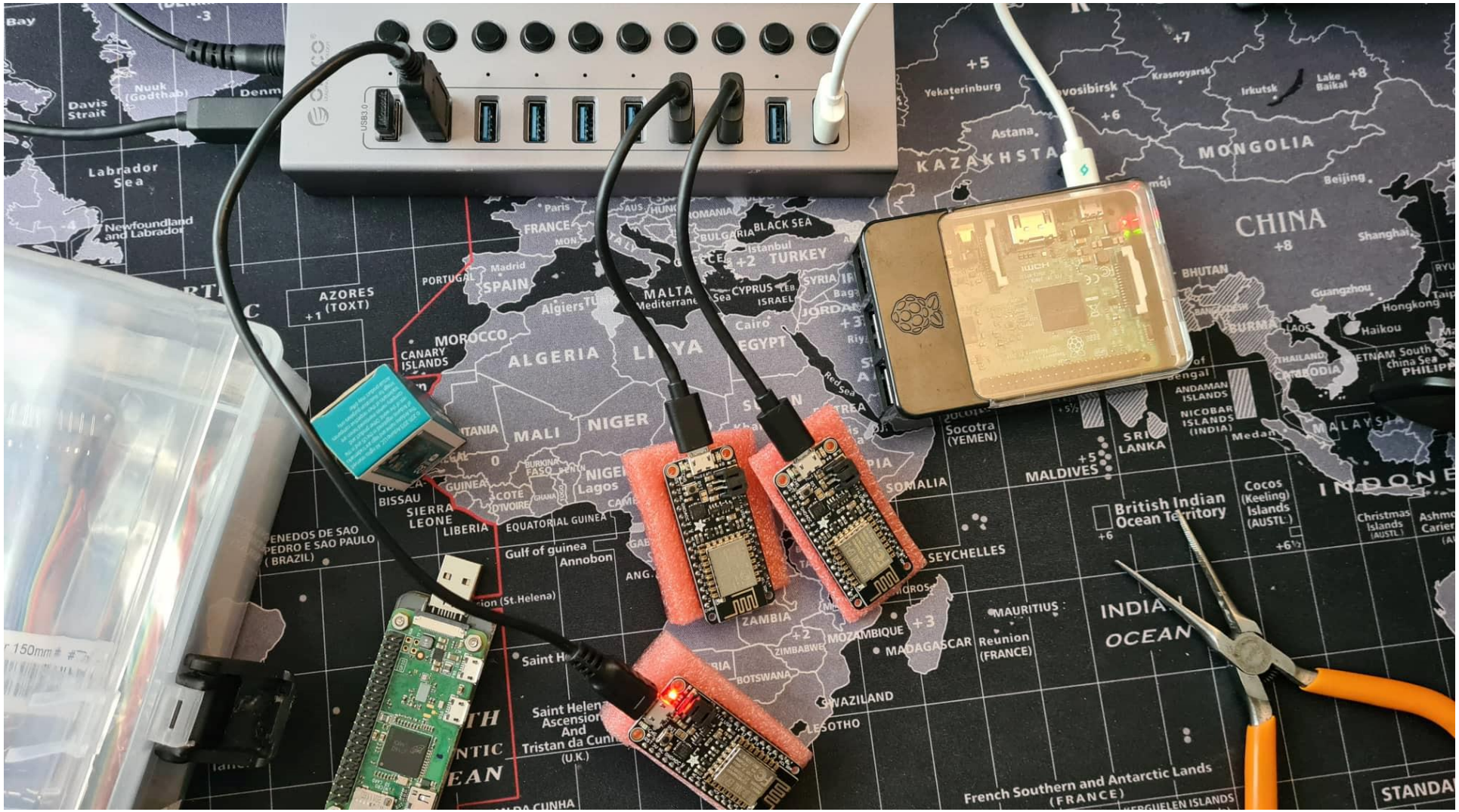
Firmware Updates
Over-the-air

DFU

Device Firmware
Update

Microsoft

Build you own – IoT Test Center ?!



Microsoft

ESP32 OTA

The OTA update mechanism allows a device to update itself based on data received while the normal firmware is running (for example, over Wi-Fi or Bluetooth.)

OTA requires configuring the Partition Table of the device with at least two "OTA app slot" partitions (i.e. ota_0 and ota_1) and an "OTA Data Partition".

The OTA operation functions write a new app firmware image to whichever OTA app slot that is currently not selected for booting. Once the image is verified, the OTA Data partition is updated to specify that this image should be used for the next boot.

Over The Air Updates (OTA)
https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/ota.html#ota-process-overview

Microsoft

ESP32 OTA - Arduino

```ino
#include <esp32fota.h>
#include <WiFi.h>

const char *ssid = "";
const char *password = "";

esp32FOTA esp32FOTA("esp32-fota-http", "1.0.0");

void setup()
{
  esp32FOTA.checkURL = "http://server/fota/fota.json";
  Serial.begin(115200);
  setup_wifi();
}

void setup_wifi()
{
  delay(10);
  Serial.print("Connecting to ");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
}

void loop()
{
  bool updatedNeeded = esp32FOTA.execHTTPcheck();
  if (updatedNeeded)
  {
    esp32FOTA.execOTA();
  }
  delay(2000);
}
```

- Web update (requires web server)
- Batch firmware sync
- Force firmware update
- https support
- Signature check
- https or https
- Signature verification
- Semantic versioning support

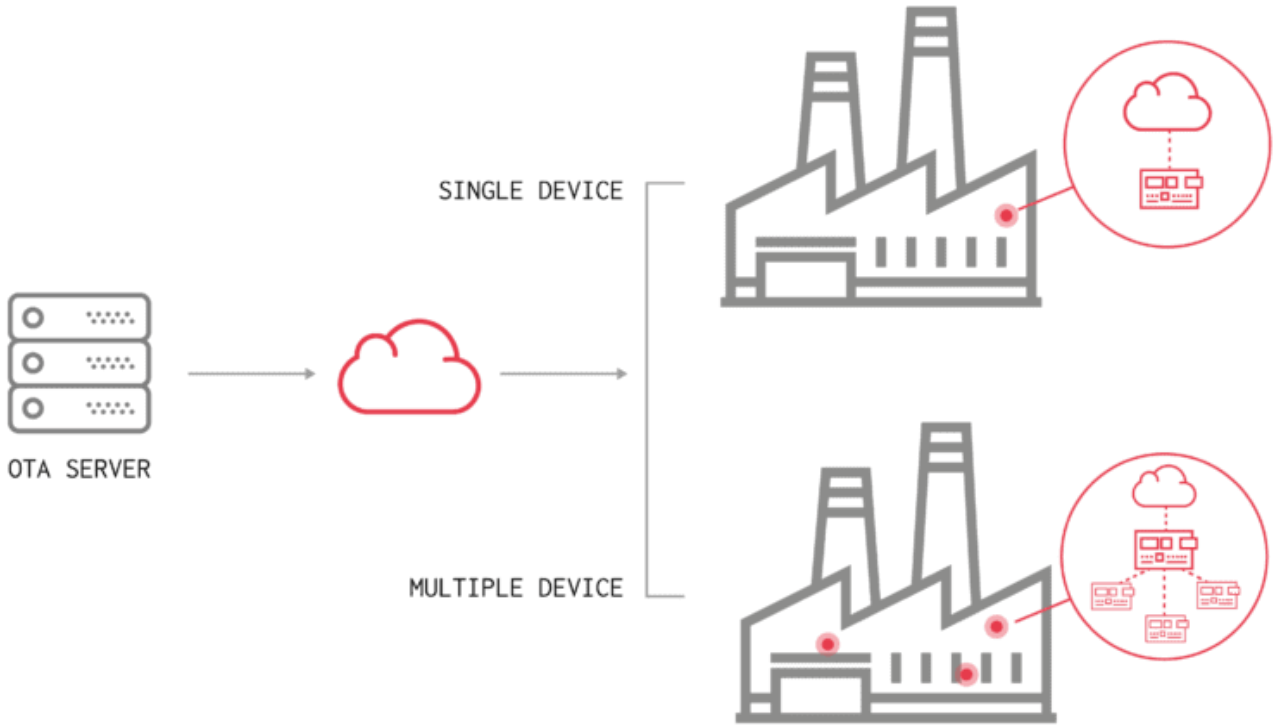
esp32FOTA library for Arduino
https://github.com/chrisjoyce911/esp32FOTA

Microsoft

Emulators and Virtual Devices ?!

Microsoft

Benefits of OTA Updates in IoT

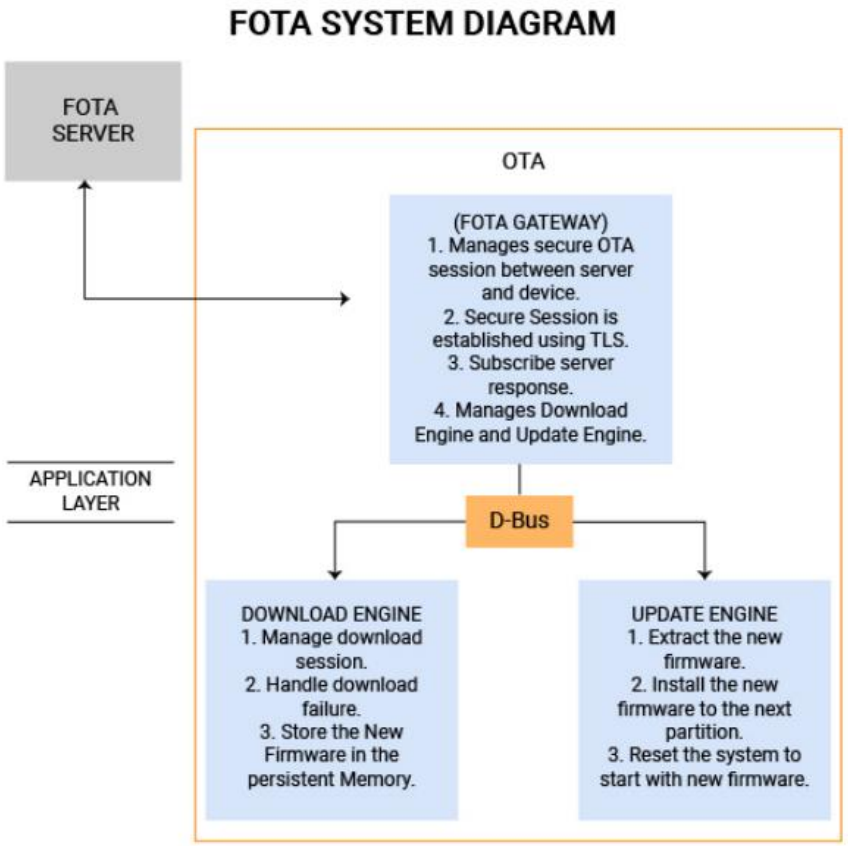


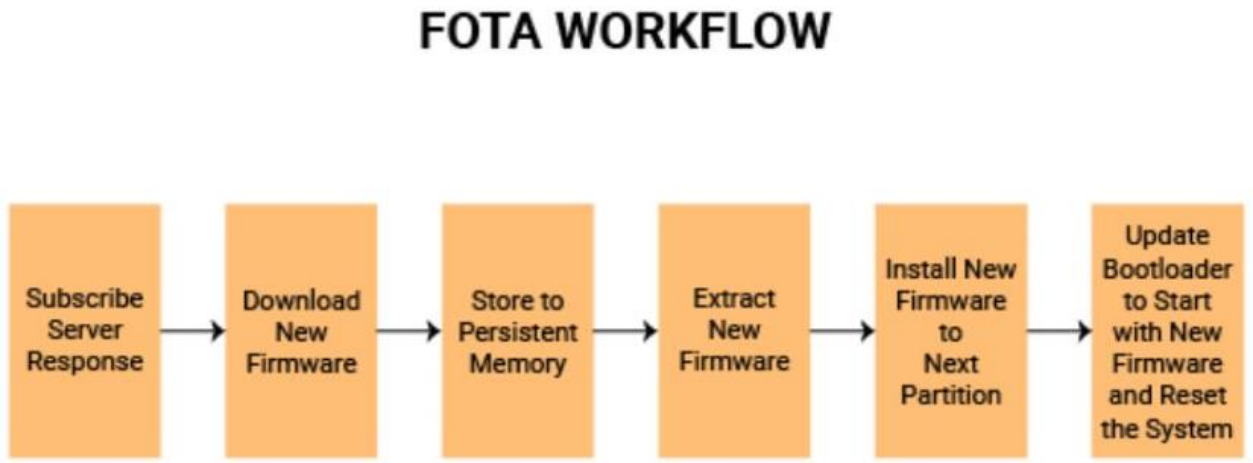
- Improved safety and compliance
- Maintenance cost reduction
- Additional revenue streams
- Reduced operational cost

Benefits of OTA Updates in IoT
https://jfrog.com/connect/post/ota-updates-for-remote-devices-different-methods-explained/

Microsoft

FOTA system modules and flow

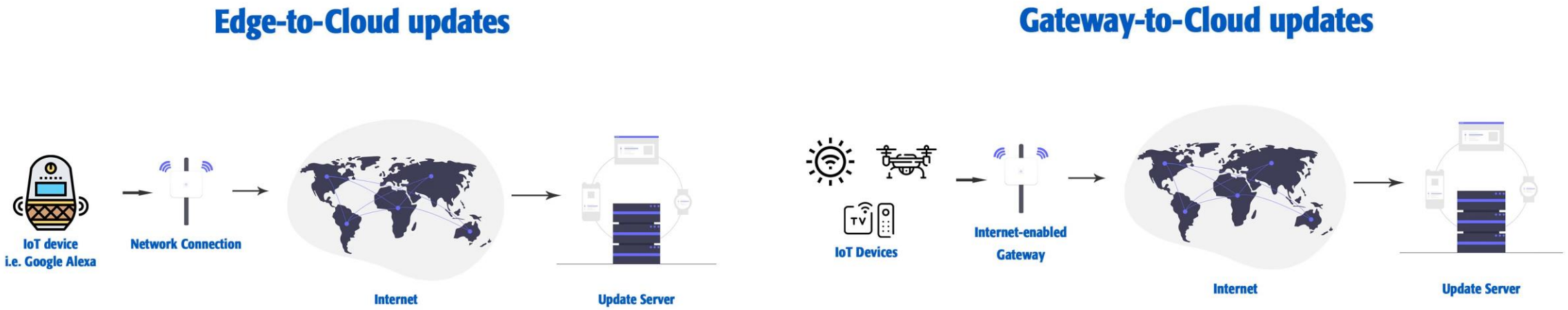




FOTA Update | Firmware Over The Air – For IoT And Automotive Devices
https://www.embitel.com/firmware-over-the-air-fota-updates-for-iot-and-automotive-devices

Microsoft

OTA update methods



1. Edge-to-Cloud OTA updates (E2C)
2. Gateway-to-Cloud OTA updates (G2C)
3. Edge-to-Gateway-to-Cloud OTA updates (E2G2C)

Benefits of OTA Updates in IoT
https://jfrog.com/connect/post/ota-updates-for-remote-devices-different-methods-explained/

Microsoft

Best practices for OTA updates

Incorporating an OTA mechanism from day-one: adding OTA update routines to the firmware greatly improves the chance of introducing feature updates and bug fixes at a later time, during the production and even after deployment.

Progressive development approach: following a progressive development approach, it can not only make the development more reliable, but also streamlines the development of new features.

Prioritized updates: To make sure that the functionality of the running devices is not interrupted.

Atomic updates with rollback: instead of pushing large updates, micro updates

Backing up current configuration

Management platform: using a central management platform, the previous updates must be logged and managed to track the changes up to the date.

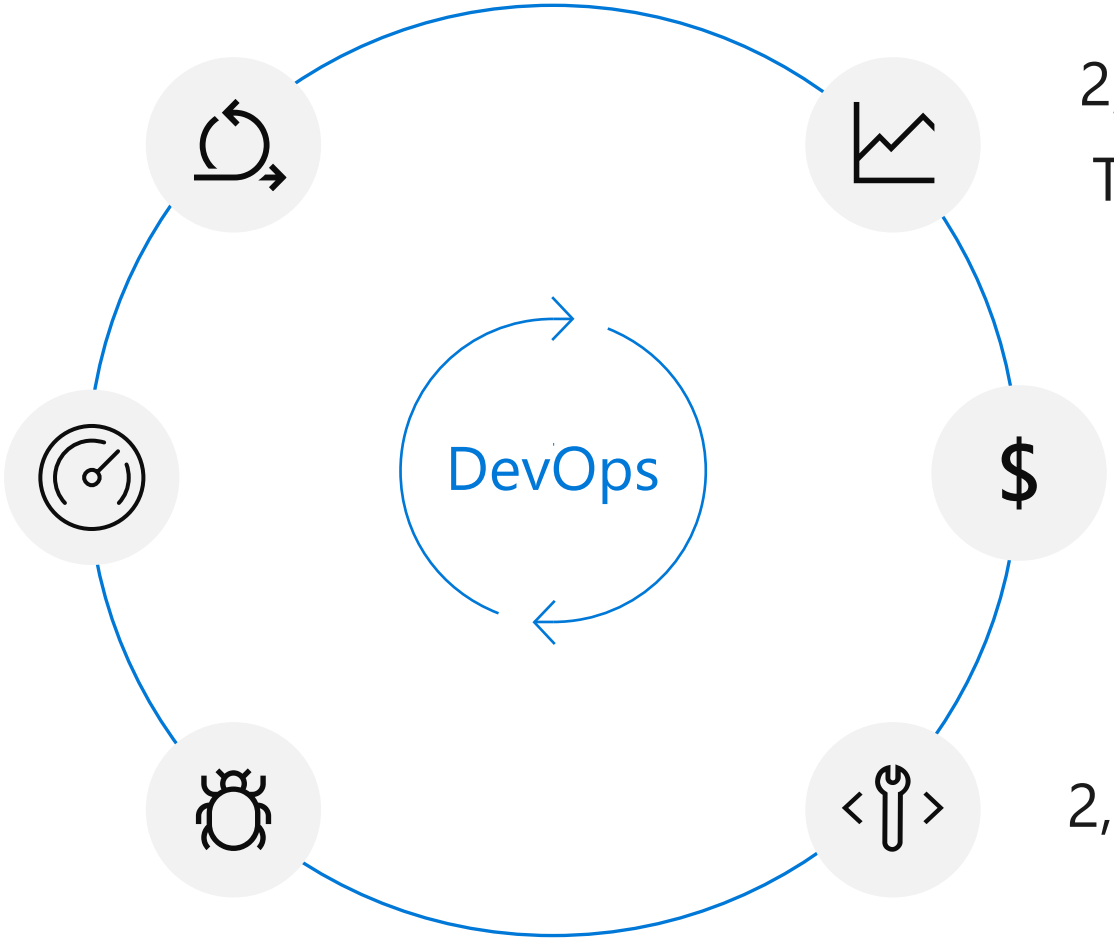
Benefits of OTA Updates in IoT
https://jfrog.com/connect/post/ota-updates-for-remote-devices-different-methods-explained/

High Performance DevOps Companies Achieve...

46x Deployment Frequency

Faster Time to Market

7x Lower Change Failure Rate



2,555x Faster Lead Time For Changes

Increased Revenue

2,604x Faster Mean Time to Recover

Microsoft

Source: 2018 Accelerate: State of DevOps: Strategies for a New Economy." N. Forsgren, J. Humble, G. Kim. DevOps Research and Assessment (DORA)

Secure firmware Over-The-Air updates for IoT

In practice, there are **multiple ways of updating an IoT device**, since **different manufacturers** follow **different approaches** depending on their tools, infrastructures, and strategies.

These ways fall in one of these **two categories**:

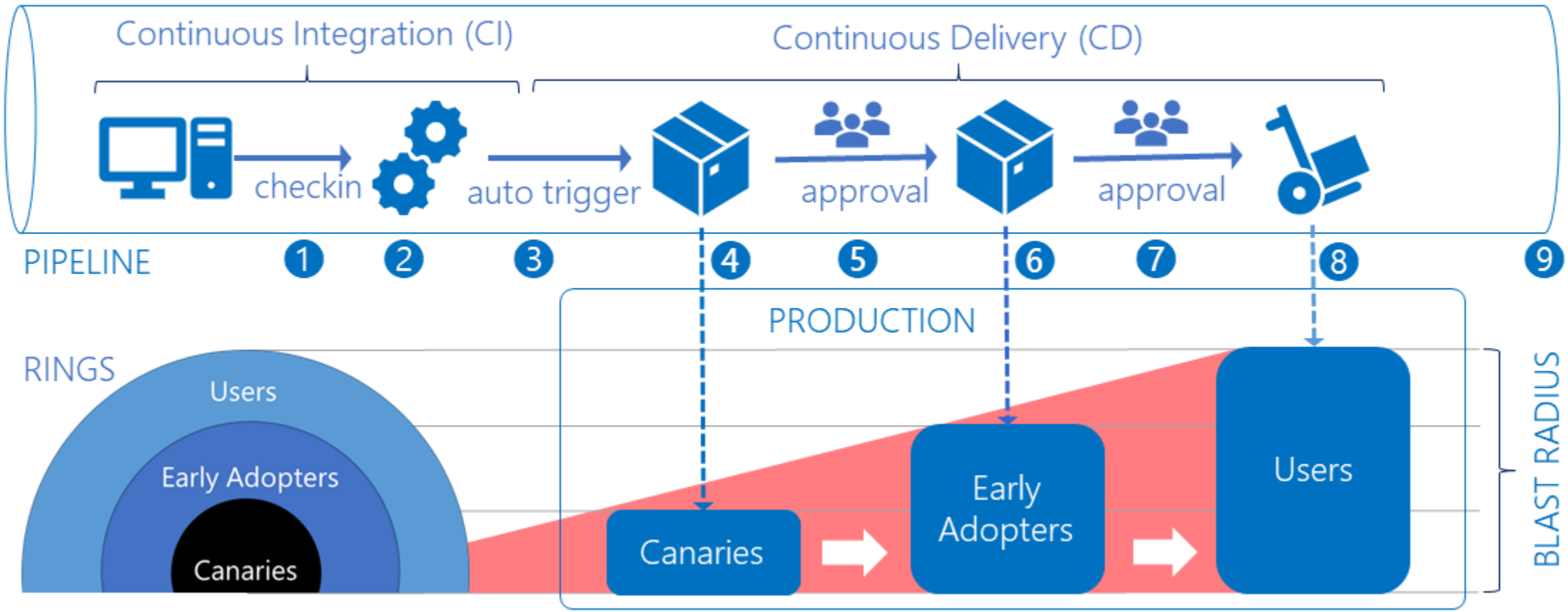
(1) the **wired updates** (e.g., via a USB port) or
(2) the **wireless Over the Air** (OTA) updates.

Also, the update can be either full or partial.

Secure firmware Over-The-Air updates for IoT: Survey, challenges, and discussions
https://www.sciencedirect.com/science/article/abs/pii/S2542660522000142

Microsoft

Final Thoughts



Use deployment rings with extension releases
https://docs.microsoft.com/en-us/azure/devops/migrate/phase-rollout-with-rings?view=azure-devops

Microsoft

~7700 Scholar Articles

Syed, M.H. and Fernandez, E.B., 2016, April. **Cloud ecosystems support for Internet of Things and DevOps using patterns**. In 2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI) (pp. 301-304). IEEE.

Bou Ghantous, G. and Gill, A.Q., 2021. **Evaluating the DevOps Reference Architecture for Multi-cloud IoT-Applications**. SN Computer Science, 2(2), pp.1-35.

Judvaitis, J., Balass, R. and Greitans, M., 2021. **Mobile IoT-Edge-Cloud Continuum Based and DevOps Enabled Software Framework**. Journal of Sensor and Actuator Networks, 10(4), p.62.

Truong, H.L. and Klein, P., 2020. **Devops contract for assuring execution of iot microservices in the edge**. Internet of Things, 9, p.100150.

López-Peña, M.A., Díaz, J., Pérez, J.E. and Humanes, H., 2020. **DevOps for IoT systems: Fast and continuous monitoring feedback of system availability**. IEEE Internet of Things Journal, 7(10), pp.10695-10707.

IoT, DevOps
https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=DevOps+IoT&btnG=

Microsoft

Thank you.