

# Azure IoT Academy

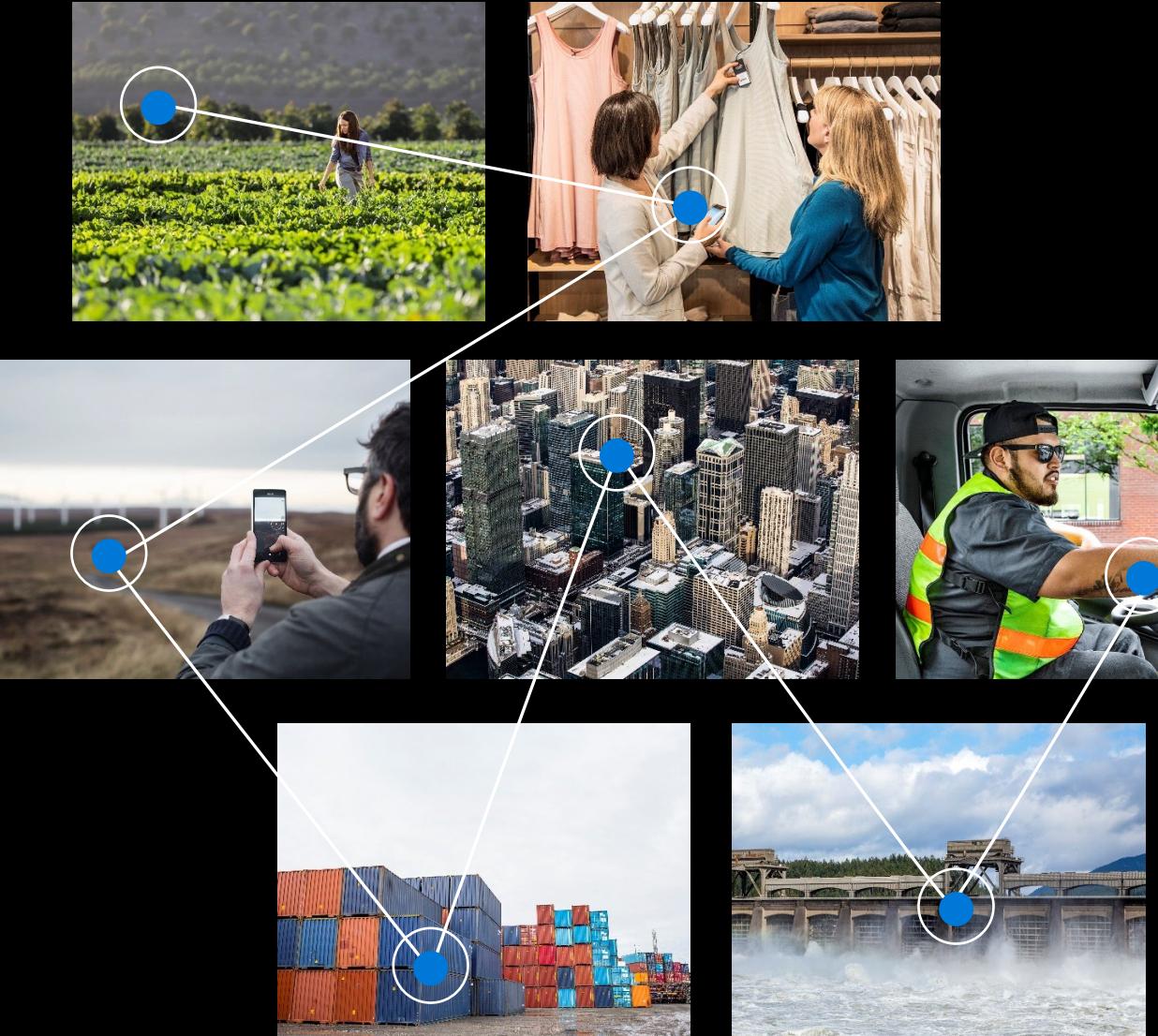
## Transforming your business

Month 2, Day 1: IoT Edge

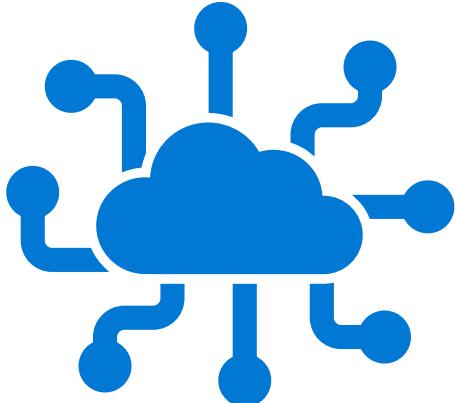
Rebekah Midkiff  
Technical Specialist  
Microsoft

Alan Blythe  
Sr. Technical Specialist  
Microsoft

Eric Johnston  
Sr. Technical Specialist  
Microsoft



# IoT Academy Expectations



- *We have a very large audience, so please keep yourself on mute except when called.*
- *Please raise your hand and wait for acknowledgement before unmuting to ask a question.*
- *Use Teams reactions to ease interactions of a large audience*
- *We want this to be interactive so please don't hesitate to let us know if you have a question (comment in chat or raise hand).*
- *If you're stuck on a hands-on lab, we request that you notify us in chat and raise your hand so we can move you to a breakout meeting for assistance.*

# IoT Academy Journey

## Month 2

- IoT Edge
- EFLOW
- Azure Digital Twins
- Azure Digital Explorer
- Azure Functions
- Partner Showcase

## Month 3

- IoT Security
- Azure Sentinel
- Defender
- Partner Showcase
- Awards Ceremony

# Month One, In Review

- IoT is a movement, a concept, which embraces distributed computing techniques
- Azure IoT Central
  - aPaaS, an extensible platform
  - A single service, many capabilities, built upon many of the core Azure IoT Services
  - Simple pricing
- Azure IoT Hub
  - IoT Hub Provisioning
  - Creating a device. Provisioning with Device Provisioning Service
  - Deploying Modules
  - Stream Analytics and Azure Data Explorer

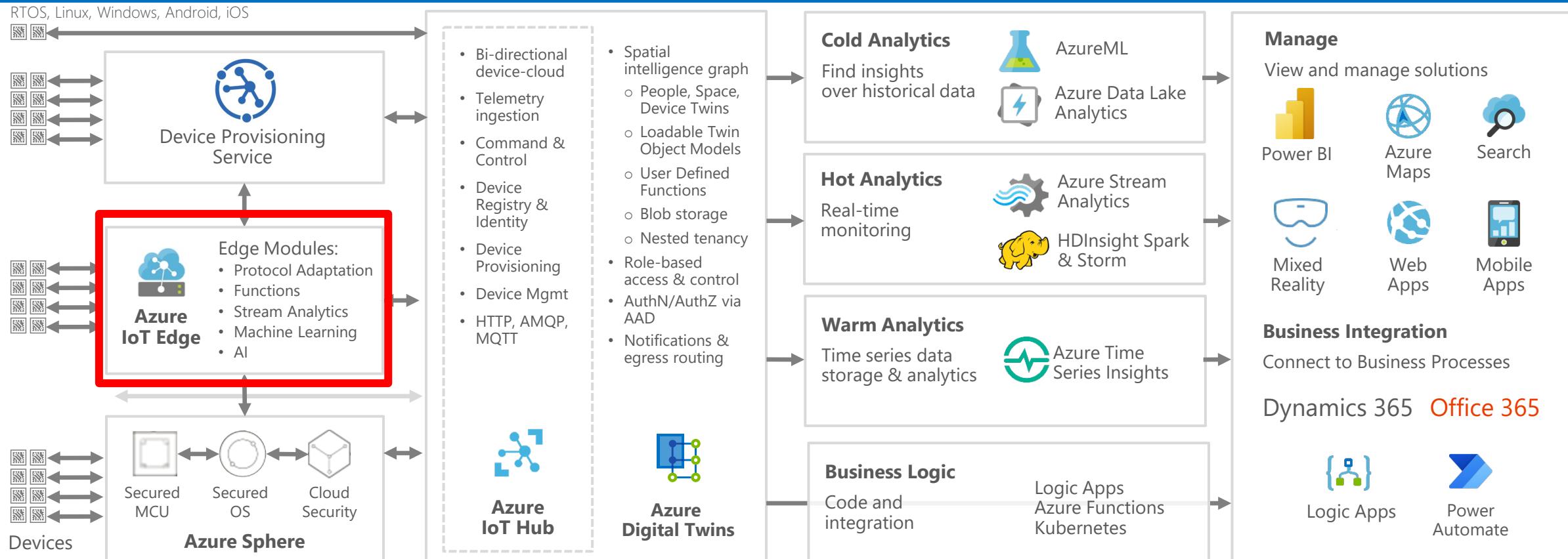
# Day One Agenda (All times are in EST)

- 10:05am – 10:20am Introduction/Expectations Kickoff - Team
- 10:20am- 12:00pm HOL
- 12:00pm – 12:15pm Coffee Break (Flexible timing)
- 12:15pm – 1:00pm HOL
- 1:00pm - 1:45pm Lunch Break
- 1:45pm - 3:15pm HOL
- 3:15pm - 3:30pm Coffee Break (Flexible timing)
- 3:30pm - 4:00pm HOL

# Day Two Agenda (All times are in EST)

- 10:05am – 10:20am Introduction/Expectations Kickoff - Team
- 10:20am- 12:00pm HOL
- 12:00pm – 12:15pm Coffee Break (Flexible timing)
- 12:15pm – 1:00pm HOL
- 1:00pm - 1:45pm Lunch Break
- 1:45pm - 3:15pm HOL
- 3:15pm - 3:30pm Coffee Break (Flexible timing)
- 3:30pm - 4:00pm HOL

# Azure IoT End-to-End Platform

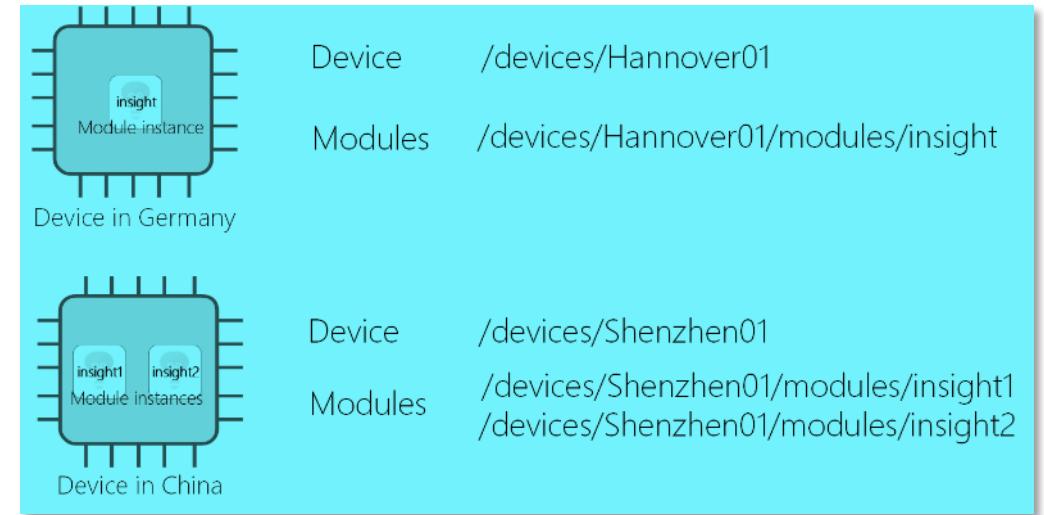
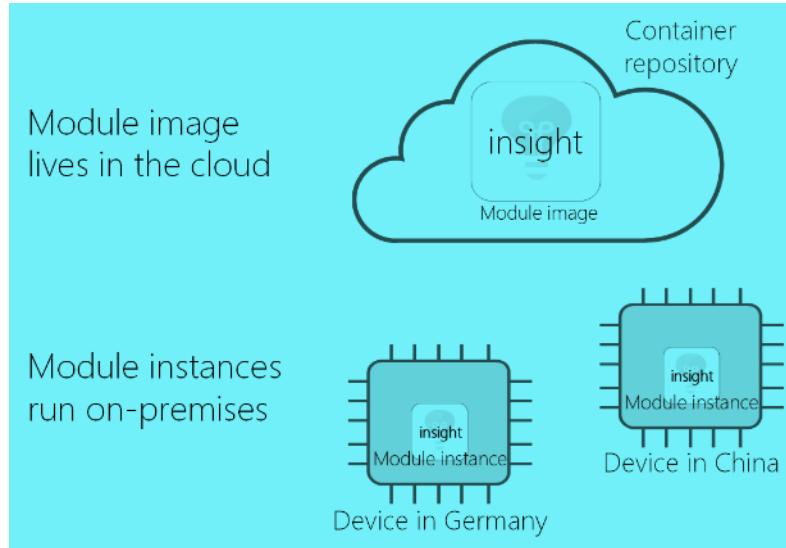


THINGS

INSIGHTS

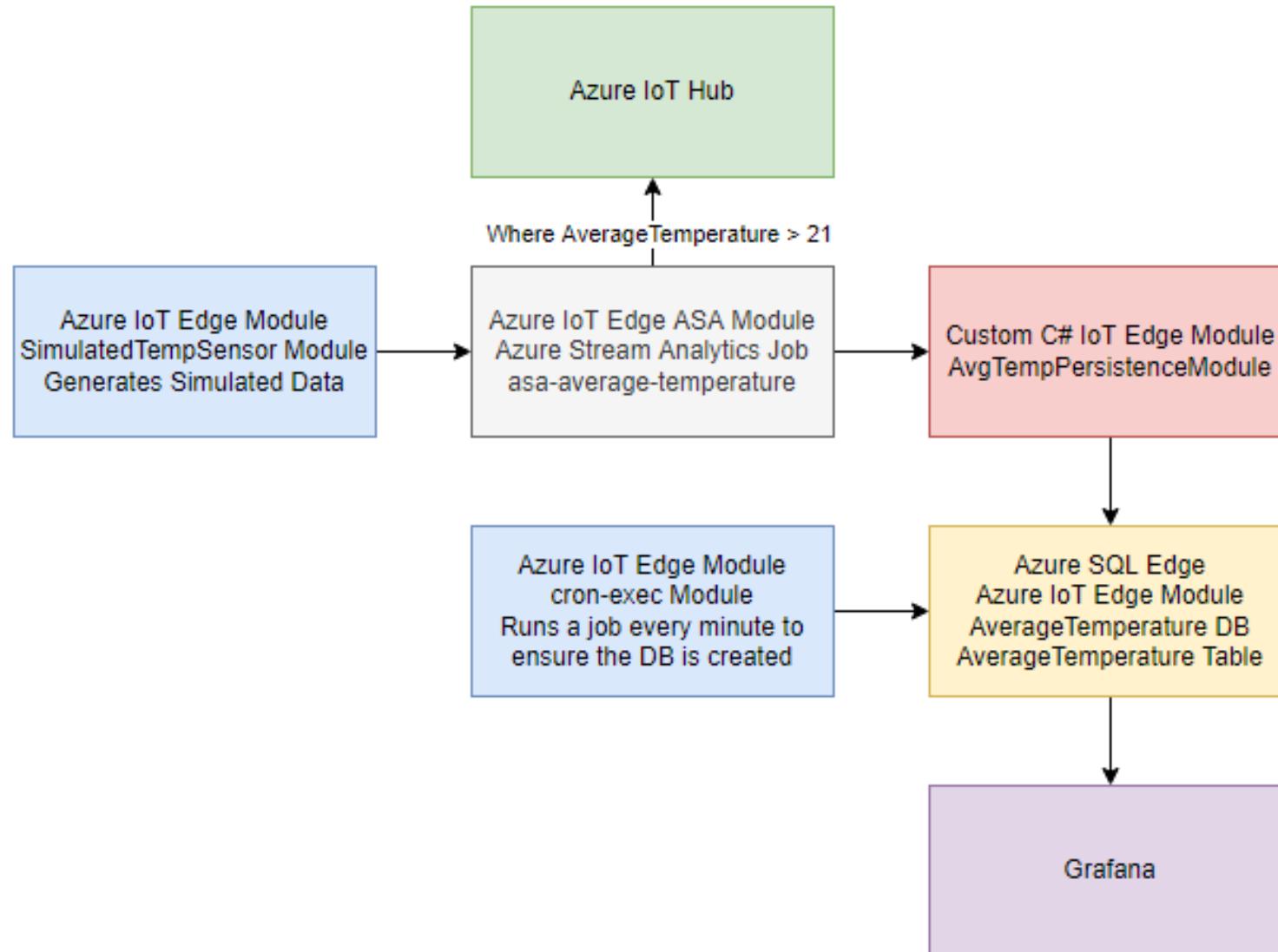
ACTIONS

# 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.

# Edge filtering and aggregation



# Use case: Retail - Smart customer interactions

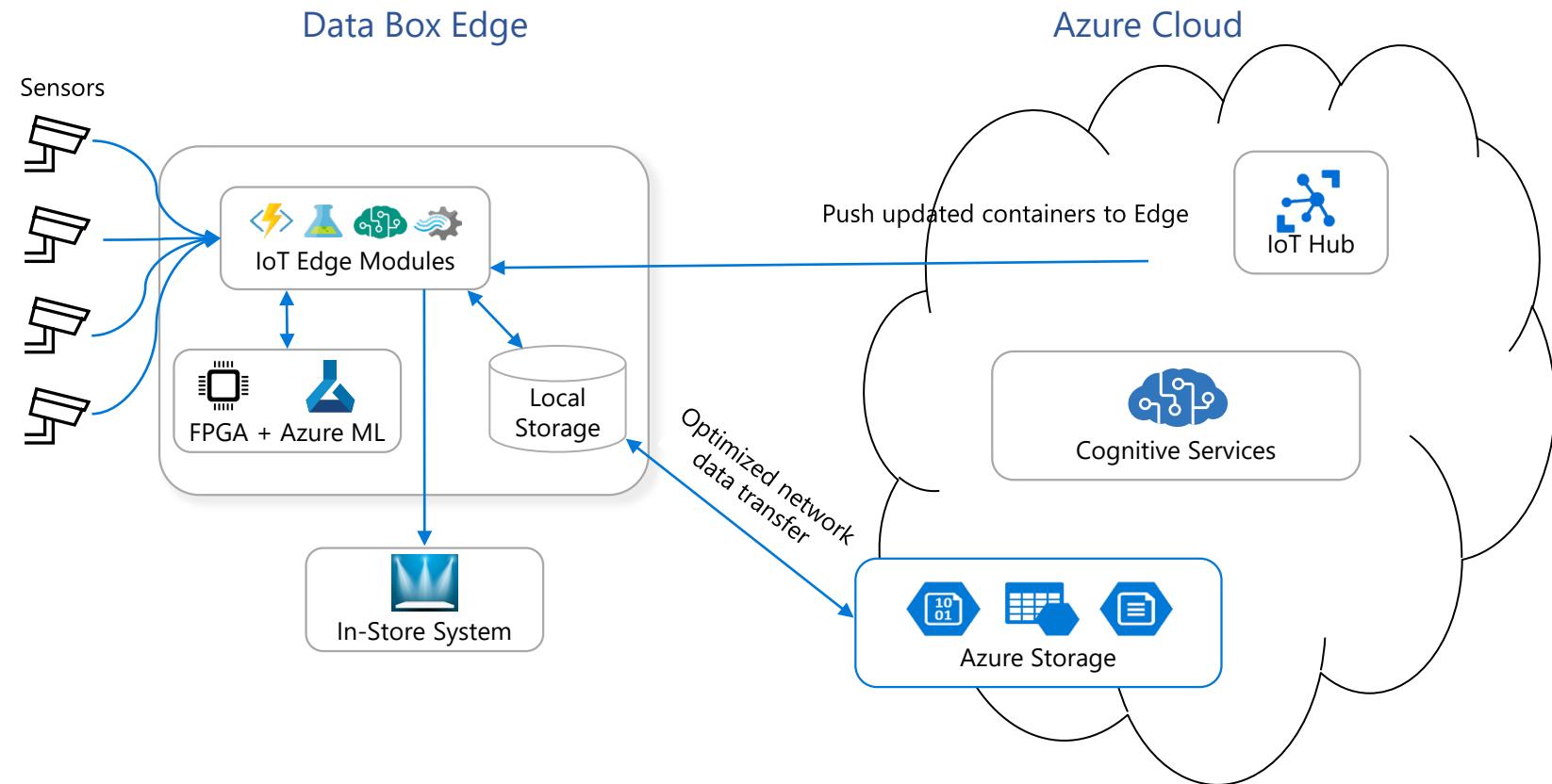
Drive additional purchases with targeted advertising based on demographics

## What problem does this solve?

- Increase customer connections
- Target advertising to drive additional sales
- Add a new monetization opportunity

## What technology does this use?

- IP cameras send data over the local network to Data Box Edge
- Containerized application uses Azure Cognitive Services to process images and determine age and gender
- An in-store system displays ads based on resulting demographic data
- Optional: transfer data to Azure for additional analytics



# Use case: Smart City - Intelligent Crosswalks

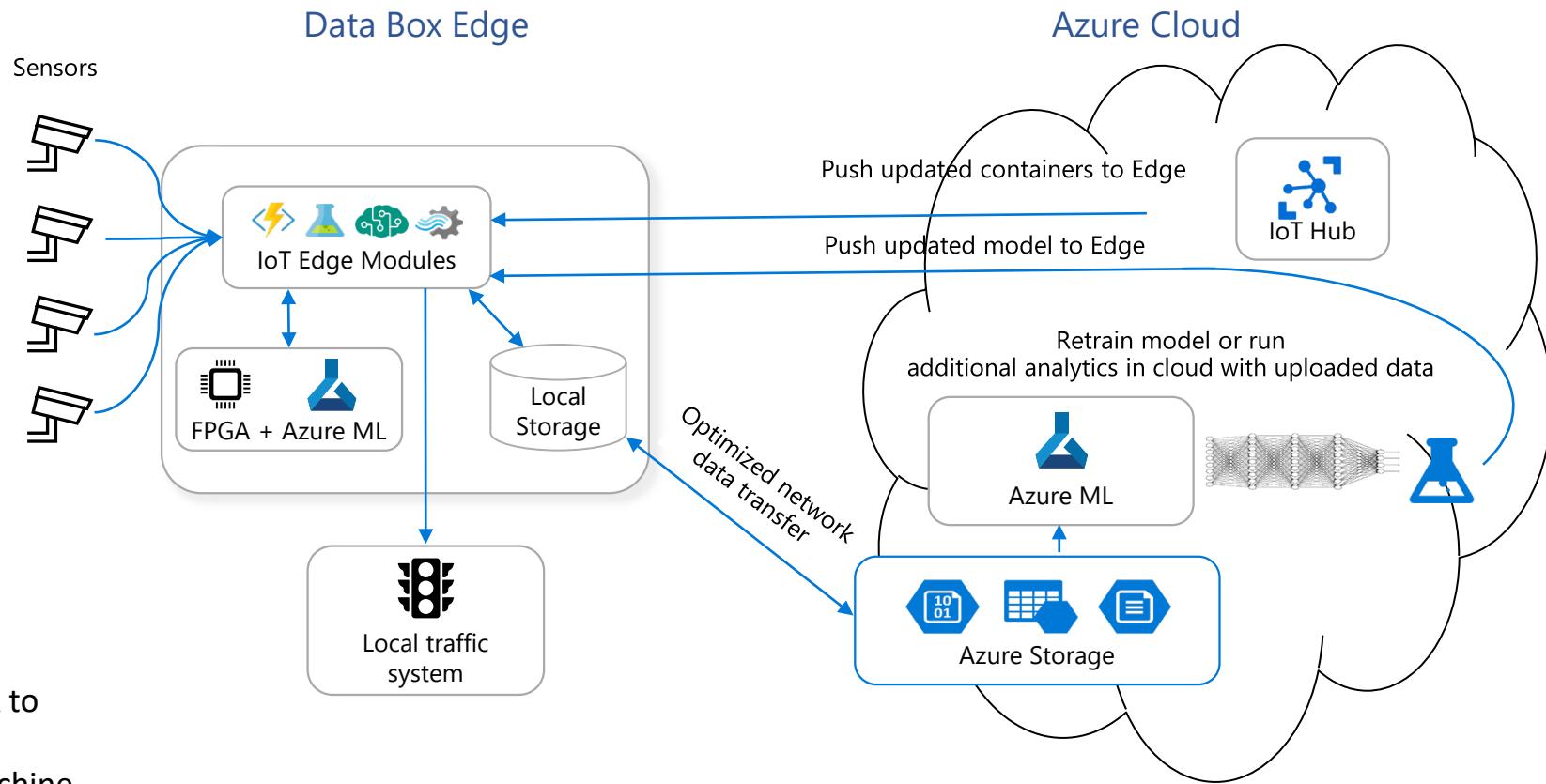
Improve crosswalk safety by extending crossing time as needed

## What problem does this solve?

- Reduce risk of harm to pedestrians from insufficient time to cross
- Reduce risk of vehicle accidents with improved traffic light management

## What technology does this use?

- IP cameras send data over the local network to Data Box Edge
- Containerized application uses a custom machine learning model to identify pedestrians in need of additional crossing time
- Optional: transfer data to Azure for retrain and improve your ML model or for additional analytics



# Use case: Energy – Oil/Gas Anomaly Detection

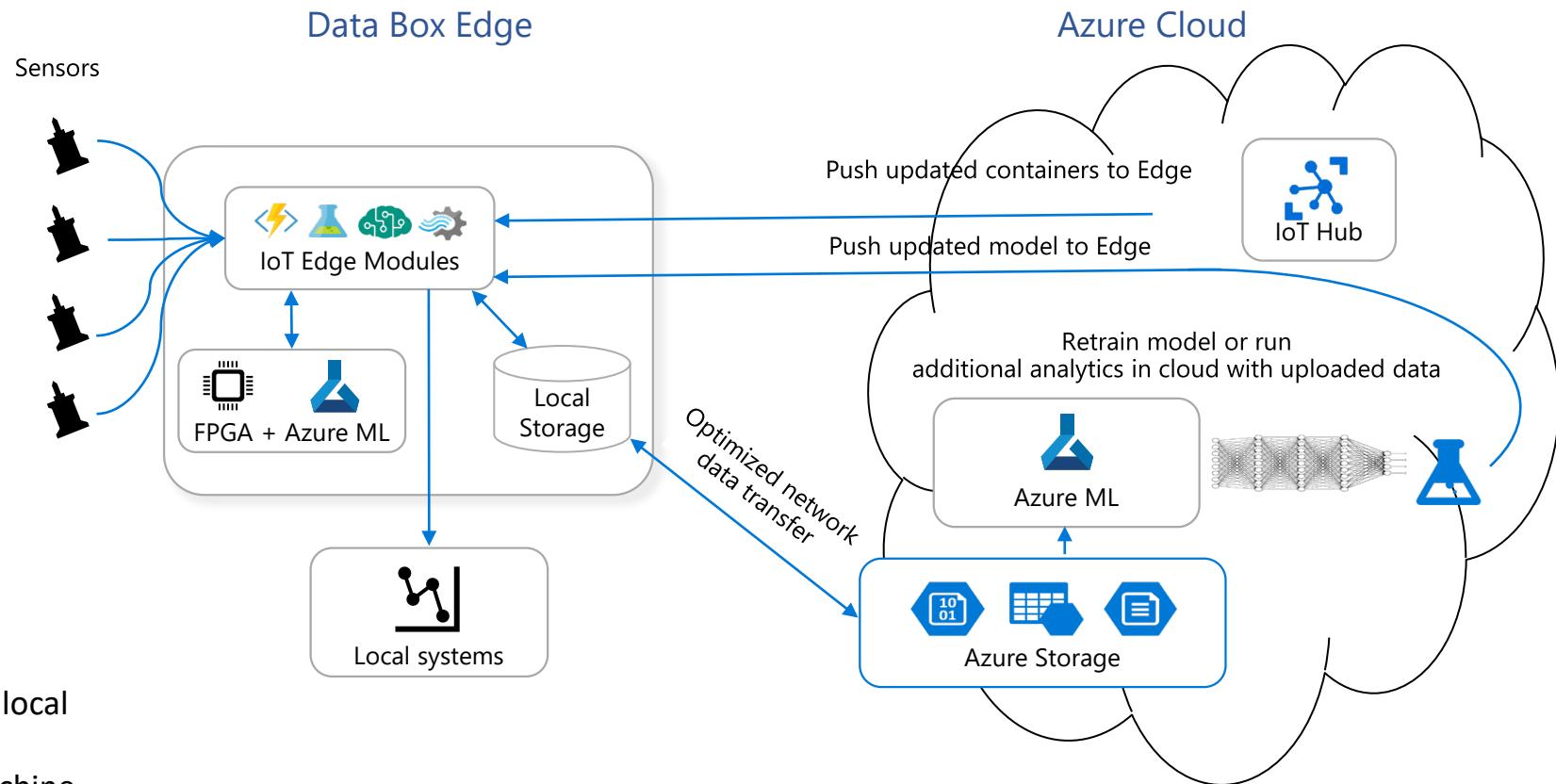
Improve efficiency of oil and gas exploration and harvesting by reducing time to detect anomalies

## What problem does this solve?

- Reduce time to detect well and pipeline anomalies
- Identify interesting data in massive data sets for more effective human analysis

## What technology does this use?

- Seismic sensors send acoustic data over the local network to Data Box Edge
- Containerized application uses a custom machine learning model to detect seismic anomalies
- Optional: transfer interesting data to Azure to retrain and improve your ML model or for additional human or machine analysis



# Use case: Remove data for compliance reasons

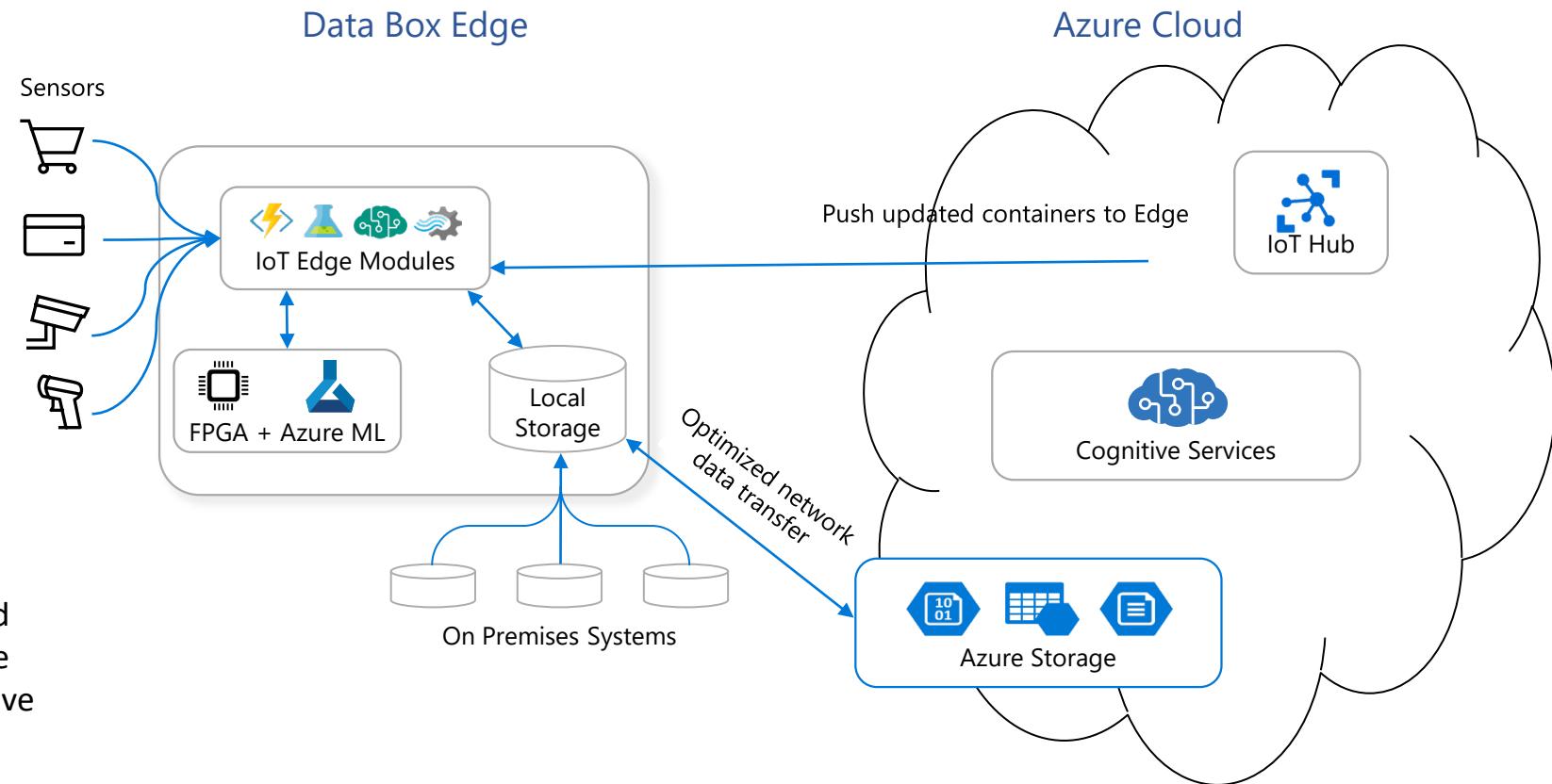
Utilize and store high-value business data while remaining compliant with regulations

## What problem does this solve?

- Mitigate risk of collecting, using, and storing many types of high-value business data

## What technology does this use?

- Various sensors or on-premises systems send data over the local network to Data Box Edge
- Containerized application uses Azure Cognitive Services to detect and remove objects from images or text
- Optional: transfer data to Azure for storage or additional analytics

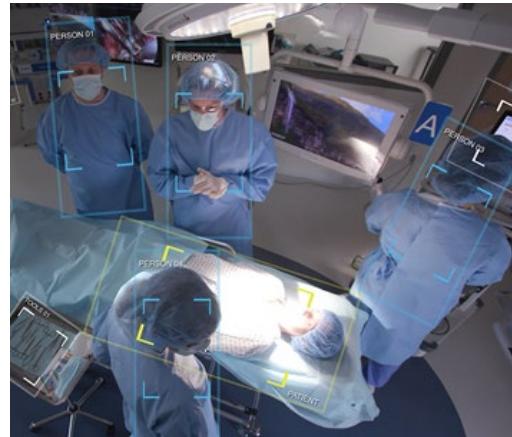


# What do these scenarios have in common?

Retail out of stock detection



Operating room  
doctor/patient tracking



## When viewed from the intelligent edge, they're the same...

Video camera  
captures data



AI analyzes the  
data



Custom business  
logic provides  
analytics, alerting,  
reporting

# Azure Bicep

SPARK



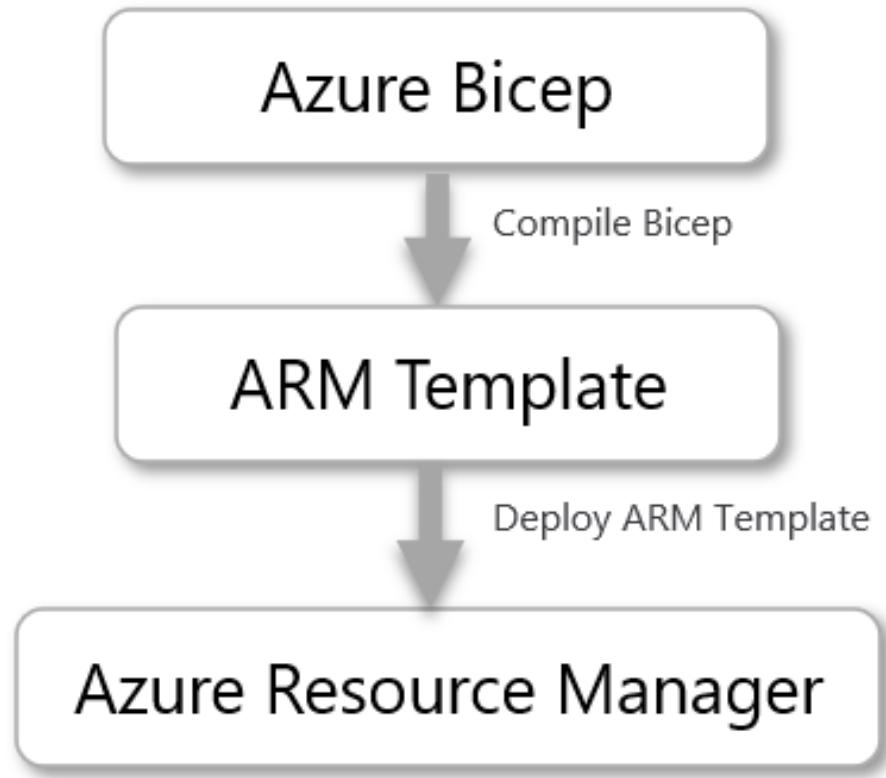
# Declarative vs Imperative – Infrastructure as Code (IaC)



# What are ARM Templates

Azure Resource Manager Templates implement Infrastructure as Code for the Azure Solutions. The template is a JSON file that defines the infrastructure.

# Introducing Azure Bicep



# Azure Bicep is Declarative, what does it mean ?

```
Pizza marg 'pizza-medium' = {  
    Ingredients = [  
        Mozarella  
        Tomatoes  
    ]  
}
```

```
resource stg 'Microsoft.Storage/storageAccounts@2019-06-01' = {  
    name : 'spark2022'  
    ...  
}
```

# Bicep : A sample

```
targetScope = 'resourceGroup' } targetScope
param storageAccountName string
param location string = resourceGroup().location } Parameters

var sku = 'Standard_LRS' } Variables

resource sa 'Microsoft.Storage/storageAccounts@2019-06-01' = {
    name: storageAccountName
    location: location
    sku: {
        name: sku
        tier: 'Standard'
    }
    kind: 'StorageV2'
    properties: {
        accessTier: 'Hot'
    }
}
} } Resource to deploy

output storageName string = sa.name } Output
```



# Azure Bicep Benefits

- **Simple Syntax**

Bicep Files are easy to read and more concise.

- **Support all API Versions**

Bicep is a Resource Manager striker which means that we don't need the tool to be updated to get the new services

- **Modularity**

Bicep introduces the "Modules" to enable the reuse of the code and to simplify the deployment. It also helps to break the code into several parts.

- **Only Bicep files**

Using Bicep does not require a state file to manage

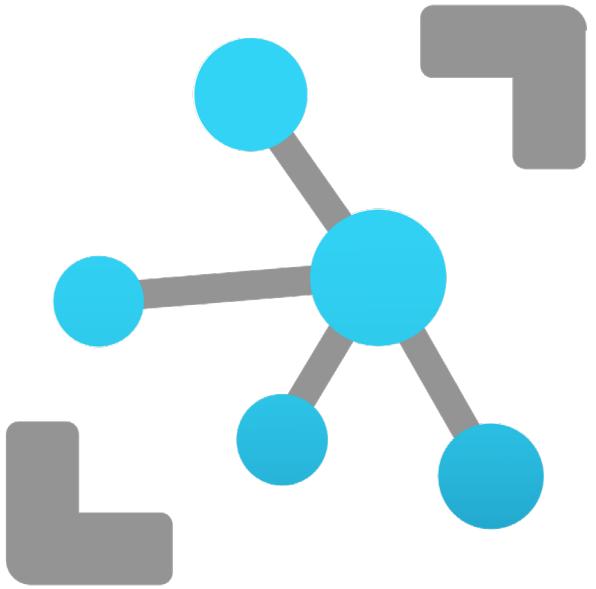
- **Open-source tool**

Bicep is a free open-source tool



# Review Custom App

## Containerized C# Core App





# Azure IoT Edge for Linux on Windows

# Bring Cloud Native Innovation and Linux Workloads to Windows IoT

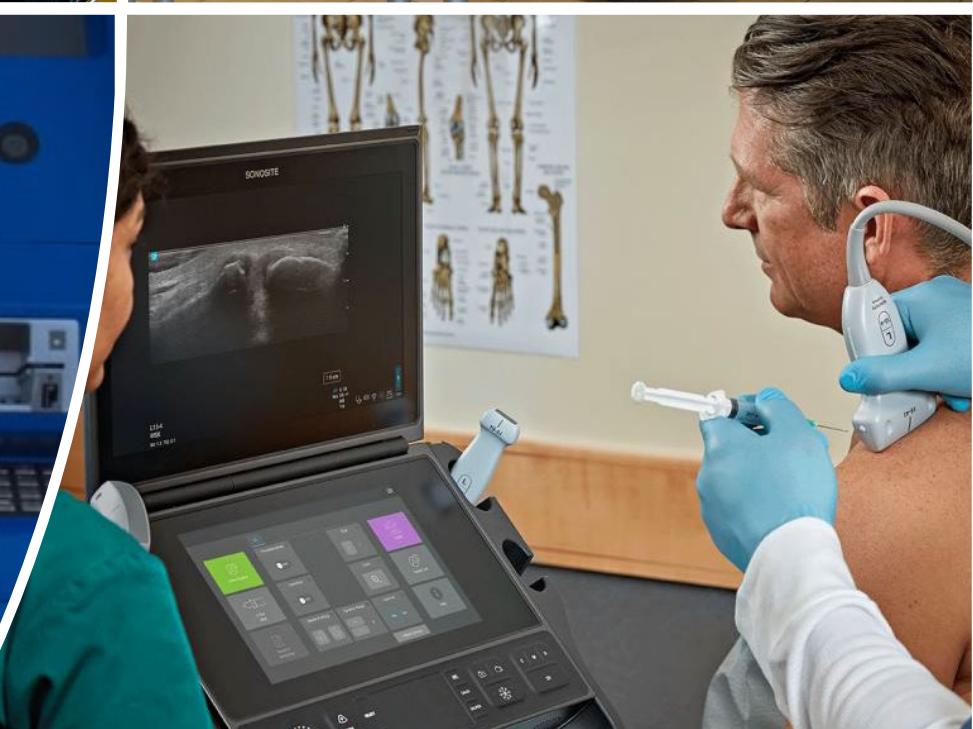
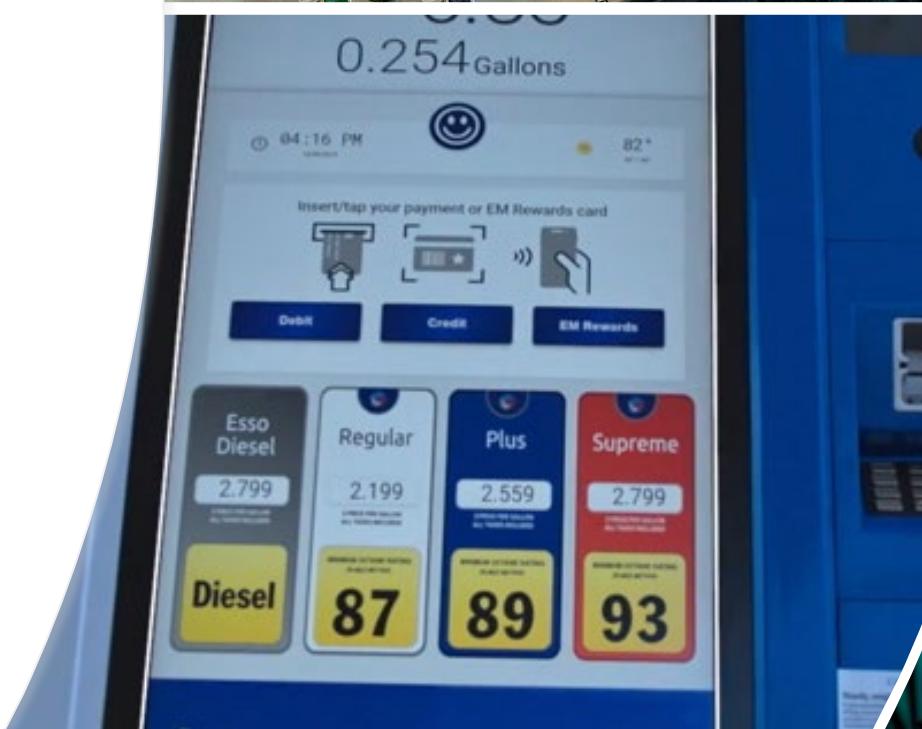
## The best of Windows IoT...

- >80M Devices
- Rich UI & media stacks
- Large installed base of apps
- Trusted in enterprise
- Highly secure, 10-year support



## Plus the best of Linux...

- Cloud native innovation
- Vast ecosystem of AI solutions



# Combining the Best of Linux and Windows

## Windows IoT Strengths

Apps with Interactive UI & natural input  
Win32 app ecosystem  
World-class long-term servicing  
Enterprise-grade device mgt & security  
Out-of-the box solution

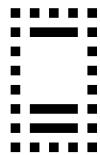
+

## Linux Strengths

Flexibility / Customizability  
Low Cost of Entry  
AI workloads  
Cloud Native Programming Models  
• Containerized Microservices

| If your organization...  | EFLOW provides...   |
|--|---|
| is a Microsoft shop and does not want to introduce a different OS                              | Familiar windows management tools for deploying Linux workloads                     |
| does not have Linux knowledge in house or on location  |   |
| is using Linux today but would prefer Windows to perform the same function                     | Ability to use existing Linux modules but manage & secure your devices with Windows |
| <b>has both Windows and Linux devices and wants to reduce overhead</b>                         | Linux + Windows capabilities and interoperability on a single machine               |
| <b>wants to keep the cost of hardware assets in check</b>                                      |   |
| <b>is sometimes deploying solutions in remote, less accessible locations</b>                   | <b>Deployment of cloud-native workloads to the edge</b>                             |
| <b>has network latency, outages and/or delays that prohibits running workload in the cloud</b> |   |
| is developing IoT Edge solutions and selling those in the IoT Edge marketplace                 | Linux IoT Edge modules run on Windows devices                                       |

# Hybrid Compute Environment



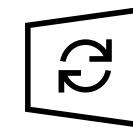
## Curated Linux VM

VM based on CBL Mariner Linux, a Microsoft maintained Linux distribution, optimized for hosting Azure IoT Edge workloads.



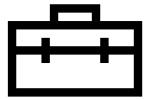
## Linux workload ready

Tier 1 Azure IoT Edge support built-in with a simplified deployment experience and ready for your cloud workloads.



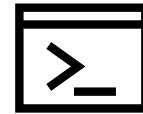
## Always up to date

Keep Linux and Azure IoT Edge binaries current with Windows Update and existing management infrastructure.



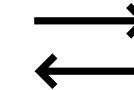
## Windows Admin Center

Windows Admin Center provides a click through deployment wizard and remote management experience.



## Flexible Scripting

PowerShell modules provide the ability to fully script deployments.

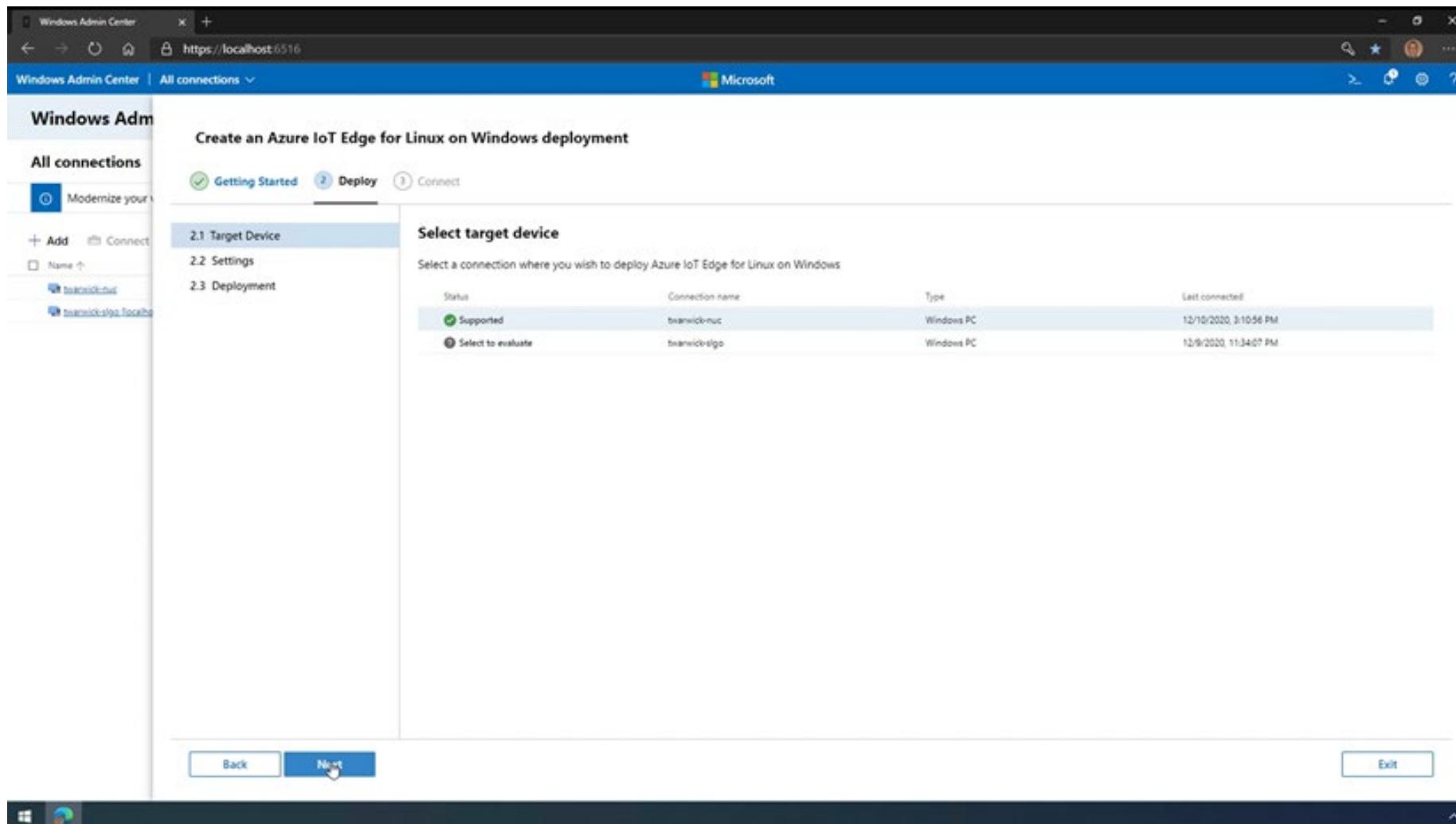


## Interoperability

Utilize the best of both worlds through interop between Windows and Linux modules.

**Sample Code:** <https://aka.ms/AzEFLW-Samples>

# Windows Admin Center – Windows Server



## Settings

User

-  Account
-  Personalization

Language / Region

### Windows Admin Center Updates

PREVIEW ⓘ

 Up to date

Version (1.3.2101.23001)

Automatically update Windows Admin Center

 Suggestions

On

Development

 Advanced Performance Profile

Gateway

 Diagnostic & feedback Updates Extensions Azure Internet Access

## Settings

User

- Account
- Language / Region
- Personalization
- Suggestions

Development

- Advanced
- Performance Profile
- Gateway
- Azure
- Diagnostic & feedback
- Extensions
- Internet Access
- Proxy
- Updates

## Extensions

Windows Admin Center might restart after installing an extension, temporarily affecting anyone using this instance of Windows Admin Center.

Automatically update extensions



On

[Available extensions](#) [Installed extensions](#) [Feeds](#)

Install

39 items

Search



| Name ↑                               | Version  | Created by                   | Package feed              | Status                  |
|--------------------------------------|----------|------------------------------|---------------------------|-------------------------|
| Active Directory                     | 0.86.0   | Microsoft                    | Windows Admin Center Feed | Available               |
| Azure Cloud Shell (Preview)          | 1.10.0   | Microsoft                    | Windows Admin Center Feed | Available               |
| Azure Extended Network (Preview)     | 0.24.1   | Microsoft                    | Windows Admin Center Feed | Available               |
| Azure File Sync                      | 2.35.5   | Microsoft                    | Windows Admin Center Feed | Newer version installed |
| BiiOps Changes                       | 2.0.24   | BiiOps                       | Windows Admin Center Feed | Available               |
| Cluster Creation                     | 1.509.0  | Microsoft                    | Windows Admin Center Feed | Newer version installed |
| Cluster Manager                      | 1.435.15 | Microsoft                    | Windows Admin Center Feed | Newer version installed |
| Configuration Manager Client (Pr...) | 1.2.1    | Ken Wygant (Microsoft CE)    | Windows Admin Center Feed | Available               |
| Containers                           | 1.142.0  | Microsoft                    | Windows Admin Center Feed | Available               |
| DataON MUST Visibility, Monitor...   | 3.3.2    | DataON                       | Windows Admin Center Feed | Available               |
| Dell EMC OpenManage Integrati...     | 2.0.0    | Dell EMC                     | Windows Admin Center Feed | Available               |
| Dell EMC PowerPath                   | 1.0.0    | Dell EMC                     | Windows Admin Center Feed | Available               |
| DHCP (Preview)                       | 0.9.3    | Microsoft                    | Windows Admin Center Feed | Available               |
| DNS (Preview)                        | 0.9.5    | Microsoft                    | Windows Admin Center Feed | Available               |
| Failover cluster tools               | 1.243.0  | Microsoft                    | Windows Admin Center Feed | Newer version installed |
| Files & file sharing                 | 1.214.0  | Microsoft                    | Windows Admin Center Feed | Newer version installed |
| Fujitsu Infrastructure Manager (P... | 0.9.1    | Fujitsu Technology Solutions | Windows Admin Center Feed | Available               |

Details

# Enabling Linux-based Edge Modules on Windows



## Curated Linux VM

VM based on CBL Mariner Linux, a Microsoft maintained Linux distribution, optimized for hosting Azure IoT Edge workloads.



## Linux Edge workload ready

Tier 1 Azure IoT Edge support built-in with a simplified deployment experience and ready for your cloud workloads.



## Always up to date

Keep Linux and Azure IoT Edge binaries current with Windows Update and existing management infrastructure.



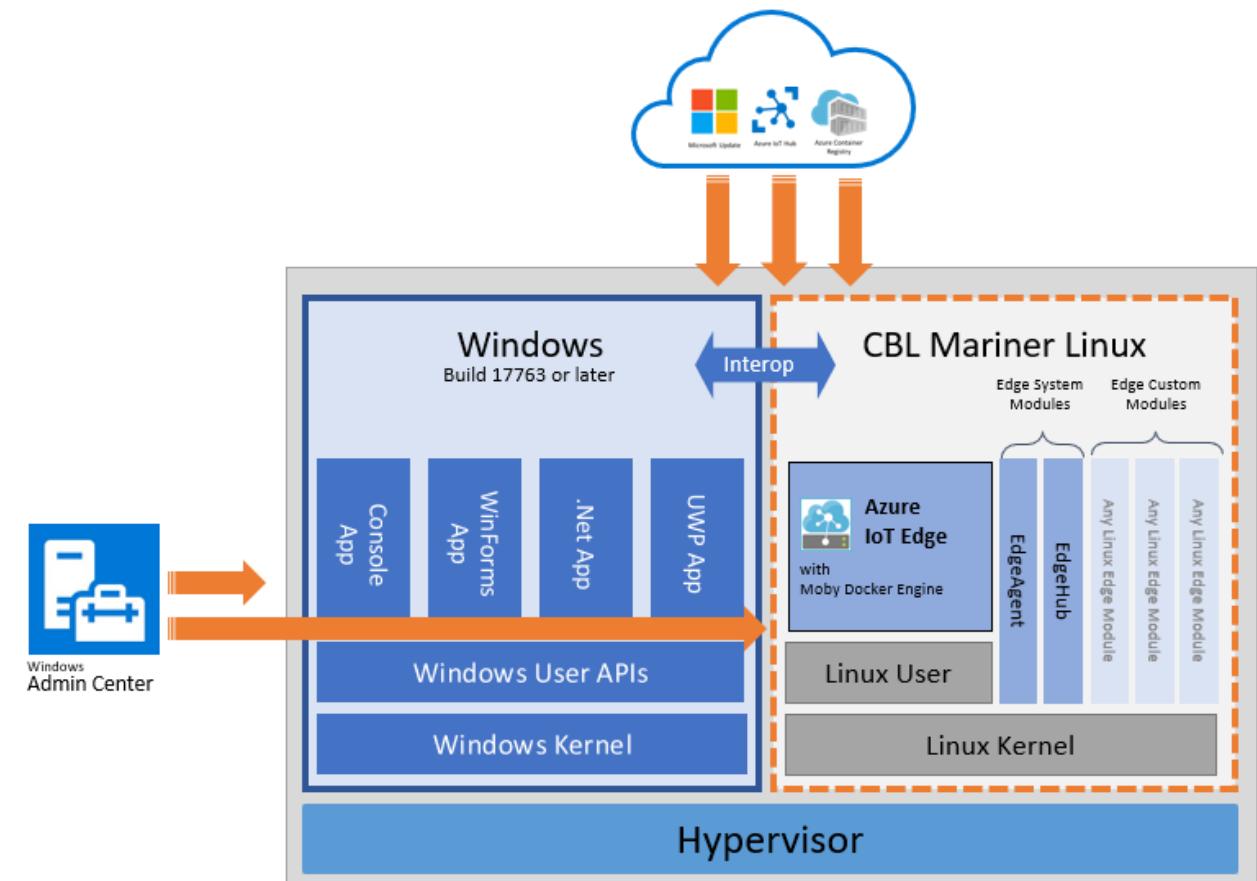
## Flexible Scripting

PowerShell modules provide the ability to fully script deployments.



## Interoperability

Utilize the best of both worlds through interop between Windows and Linux modules.



Learn More: <https://aka.ms/AzEFLOW-docs>

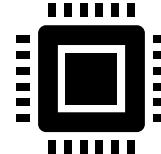
## Hardware Access



### TPM 2.0

Trusted Environment  
for DPS Provisioning

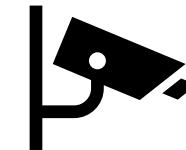
Learn More:  
[aka.ms/AzEFLow-TPM](https://aka.ms/AzEFLow-TPM)



### GPU

HW Accelerated  
Inferencing

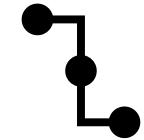
Learn More:  
[aka.ms/AzEFLow-GPU](https://aka.ms/AzEFLow-GPU)



### Camera

AI / ML with USB  
attached Camera

Learn More:  
[aka.ms/AzEFLow-USBCameraToRTSP](https://aka.ms/AzEFLow-USBCameraToRTSP)



### Serial

Data ingestion from  
serial attached devices

Learn More:  
[aka.ms/AzEFLow-SerialPassthrough](https://aka.ms/AzEFLow-SerialPassthrough)

# Resources

## Blogs & Videos

### General Availability

Blog: [aka.ms/AzEFLOW-blog](https://aka.ms/AzEFLOW-blog)

IoT Show: [aka.ms/AzEFLOW-show](https://aka.ms/AzEFLOW-show)

### EFLOW and WSL in Edge Dev

Blog: [aka.ms/AzEFLOW-wslblog](https://aka.ms/AzEFLOW-wslblog)

IoT Show: [aka.ms/AzEFLOW-wslshow](https://aka.ms/AzEFLOW-wslshow)

## Product

### Documentation

Overview: [aka.ms/AzEFLOW-docs](https://aka.ms/AzEFLOW-docs)

Quickstart: [aka.ms/AzEFLOW-quickstart](https://aka.ms/AzEFLOW-quickstart)

Install: [aka.ms/AxEFLOW-install](https://aka.ms/AxEFLOW-install)

PowerShell: [aka.ms/AzEFLOW-powershell](https://aka.ms/AzEFLOW-powershell)

Modules: [aka.ms/AzEFLOW-modules](https://aka.ms/AzEFLOW-modules)

## Github

### GitHub

Home: [aka.ms/AzEFLOW-github](https://aka.ms/AzEFLOW-github)

Wiki: [aka.ms/AzEFLOW-wiki](https://aka.ms/AzEFLOW-wiki)

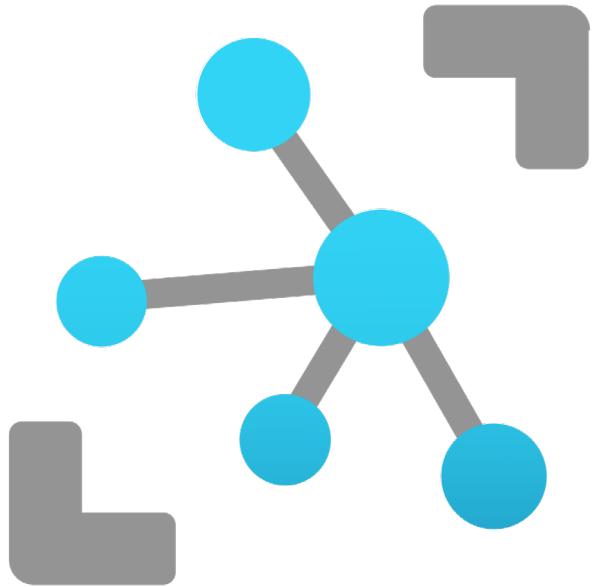
Issues: [aka.ms/AzEFLOW-issues](https://aka.ms/AzEFLOW-issues)

Samples: [aka.ms/AzEFLOW-samples](https://aka.ms/AzEFLOW-samples)

Releases: [aka.ms/AzEFLOW-releases](https://aka.ms/AzEFLOW-releases)

# Hands-On Lab

Bicep, Azure VM, and EFLOW

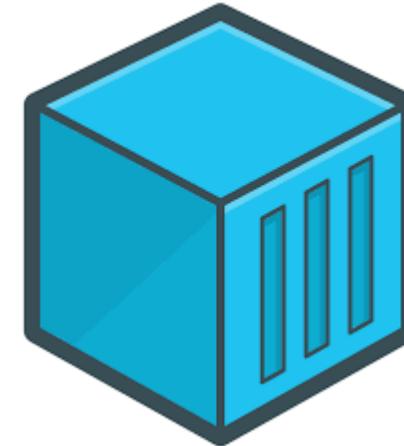


# Container definition

A container is an isolated area of an OS with some limits on how many resources it can use.

A container is like a fast lightweight VM

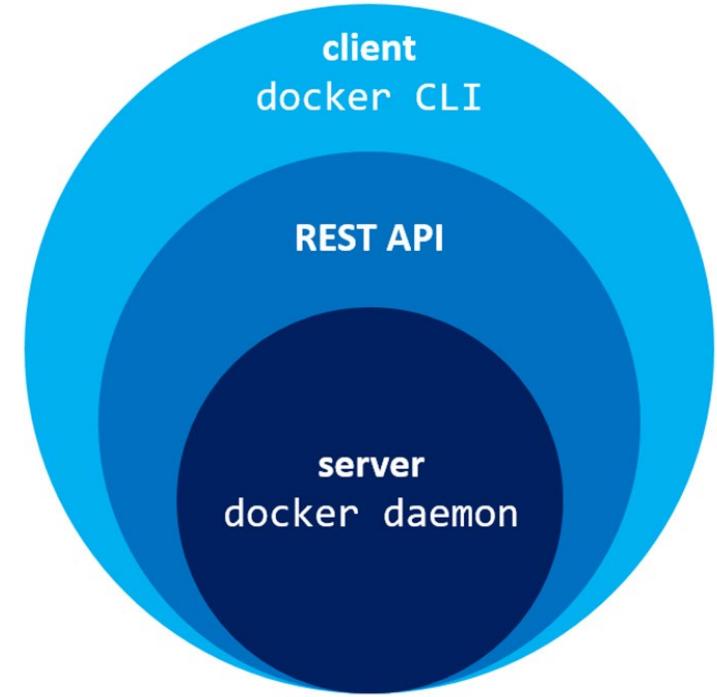
- A VM slices up HW resources
  - CPU
  - Memory
  - Etc.
- A Container slices up OS resources
  - Process namespace
  - Network namespace
  - File system namespace
  - Etc.



# Docker components

Installing docker gets you 2 components:

- Docker client
  - CLI interface
  - Commands are interpreted and transformed into API calls to the daemon
- Docker daemon (server)
  - Implements the docker remote API
  - Interfaces with the OS to create and manage containers
  - Long-running background process



```
PS C:\Users\andavr> docker version
Client: Docker Engine - Community
 Cloud integration: 1.0.2
 Version:          20.10.0-rc1
 API version:     1.41
 Go version:      go1.13.15
 Git commit:       5cc2396
 Built:           Tue Nov 17 22:49:28 2020
 OS/Arch:         windows/amd64
 Context:          default
 Experimental:    true

Server: Docker Engine - Community
Engine:
 Version:          20.10.0-rc1
 API version:     1.41 (minimum version 1.12)
 Go version:      go1.13.15
 Git commit:       131bf7e
 Built:           Tue Nov 17 22:52:57 2020
 OS/Arch:         linux/amd64
 Experimental:    false
```

# Docker Hello World

```
docker container run hello-world
```

- Image was not found locally so it was pulled from the Docker Hub
  - After running the host now has
    - a local copy of the image in the local registry
    - A container in the exited/stopped state

```
docker container ps -a
```

- lists all containers

# docker images

- lists images

```
PS C:\Users\andavr> docker container run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
0e03bdcc26d7: Pull complete
Digest: sha256:e7c70bb24b462baa86c102610182e3efcb12a04854e8c582838d92970a09f323
Status: Downloaded newer image for hello-world:latest

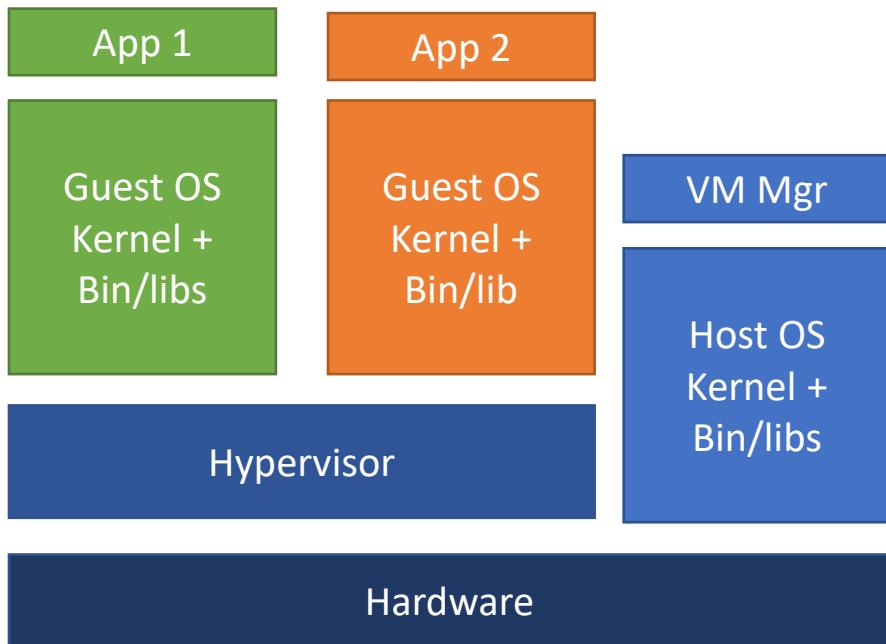
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.
```

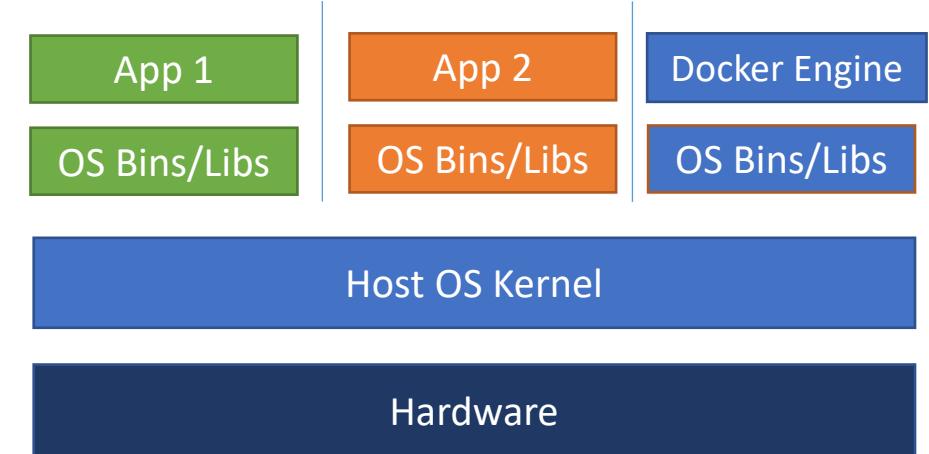
```
PS C:\Users\andavr> docker container ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
d4fa5b5a9db2        hello-world        "/hello"          5 minutes ago     Exited (0) 5 minutes ago   dreamy_rhodes
```

| REPOSITORY                         | TAG    | IMAGE ID     | CREATED       | SIZE   |
|------------------------------------|--------|--------------|---------------|--------|
| multistage                         | latest | de0b01d721d6 | 13 days ago   | 5.59MB |
| gittest                            | latest | d5e9613b66ce | 13 days ago   | 88.8MB |
| capp                               | latest | 1289337188f7 | 13 days ago   | 213MB  |
| docker/desktop-storage-provisioner | v1.1   | e704287ce753 | 8 months ago  | 41.8MB |
| docker/desktop-vpnkit-controller   | v1.0   | 79da37e5a3aa | 8 months ago  | 36.6MB |
| k8s.gcr.io/pause                   | 3.2    | 80d28bedfe5d | 9 months ago  | 683kB  |
| postgres                           | <none> | ed5a45034282 | 9 months ago  | 251MB  |
| hello-world                        | latest | bf756fb1ae65 | 10 months ago | 13.3kB |

# VM vs container



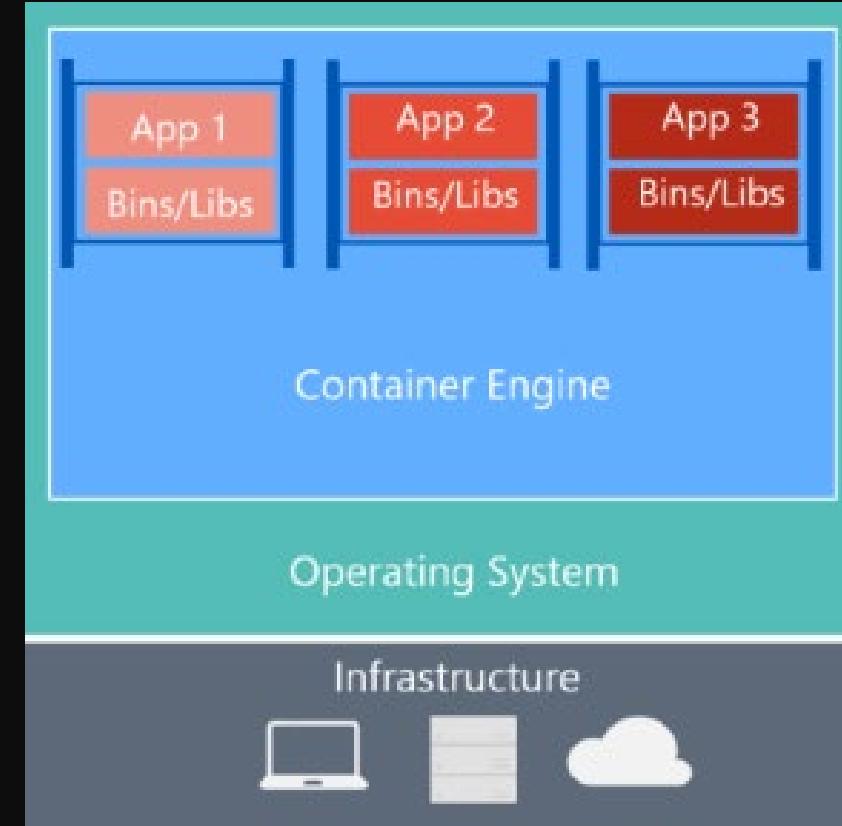
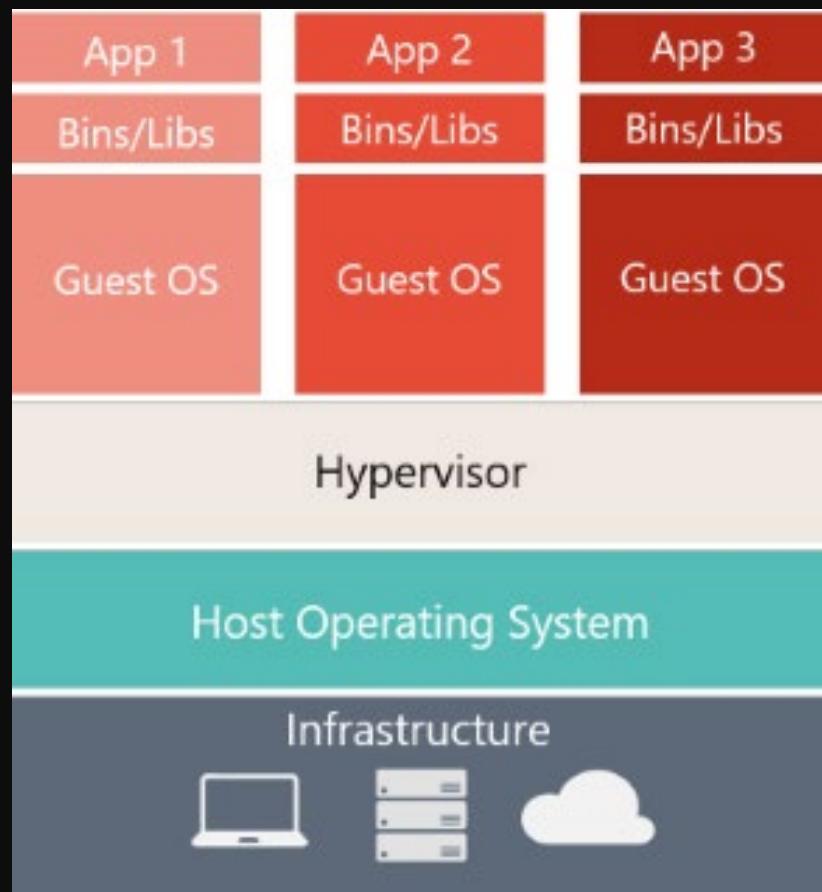
VM

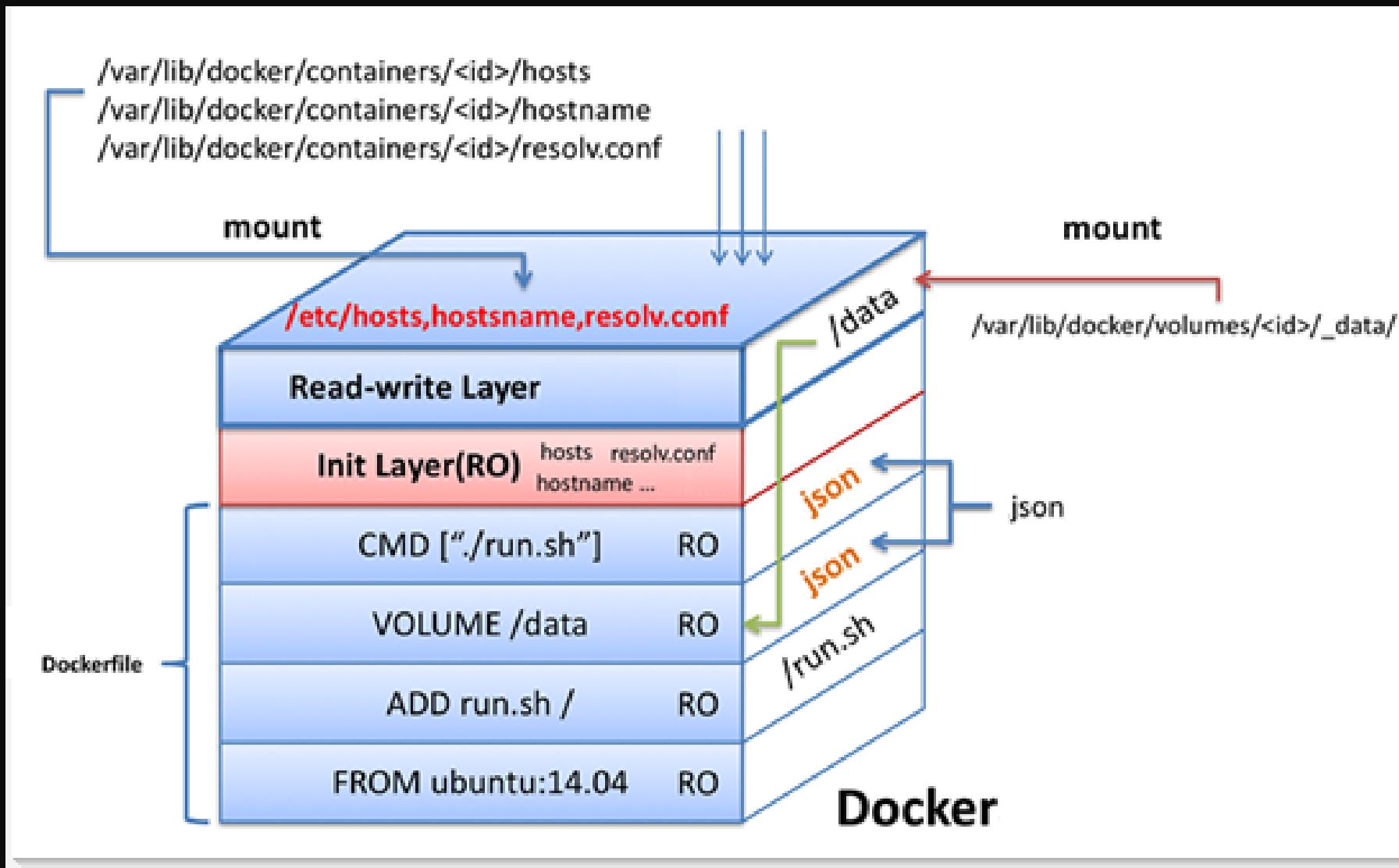


Container

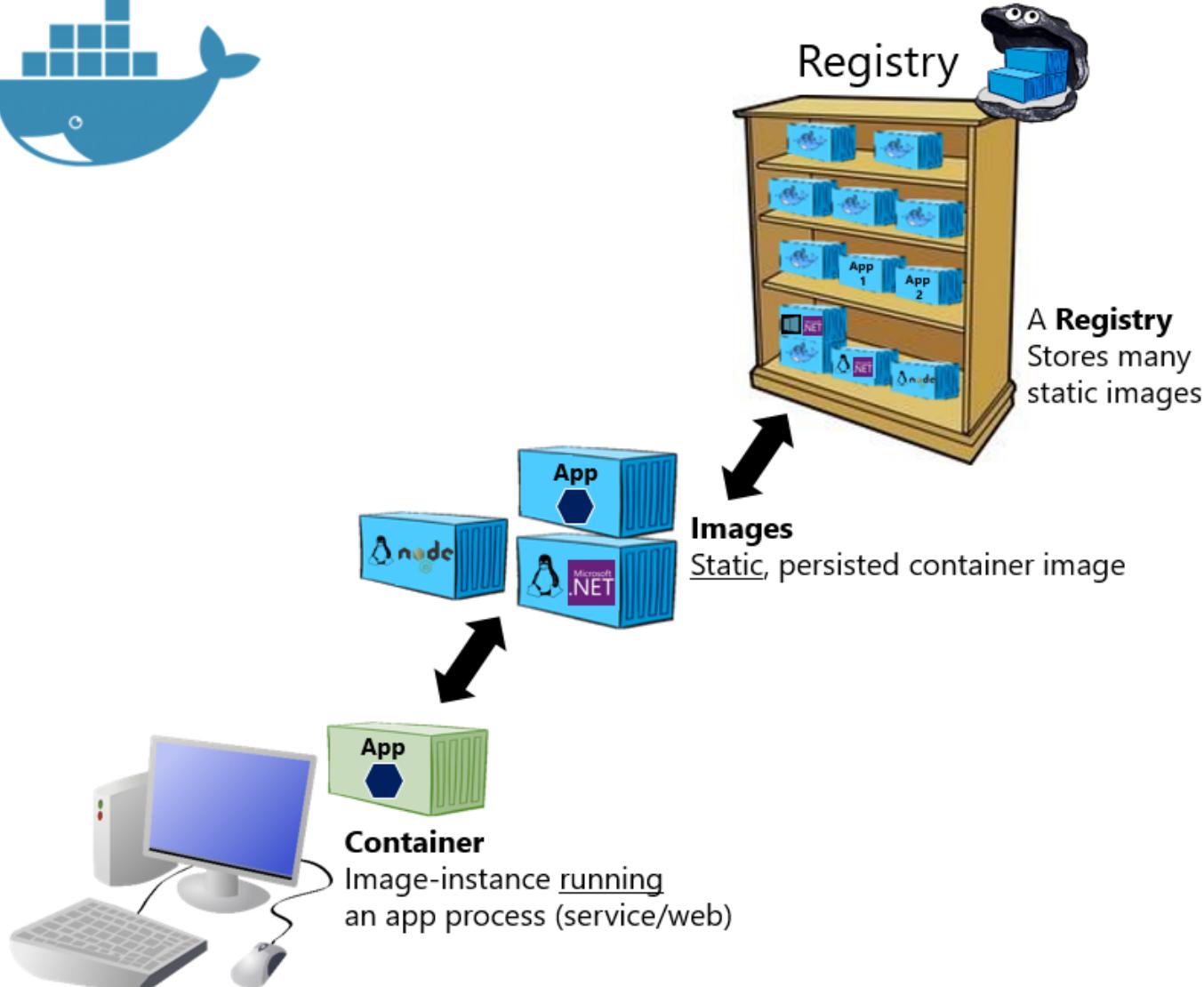
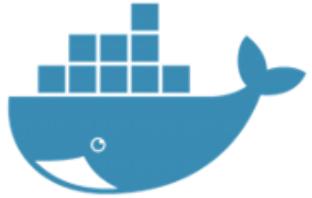
# VM vs container

|                    | Containers   | VMs                         |
|--------------------|--|-----------------------------|
| Security isolation | Sufficient for most 1 <sup>st</sup> party workload | Needed for hostile workload |
| Resource isolation | Resource-dependent                                 | Stricter                    |
| Perf overhead      | Near native perf                                   | Noticeable overhead         |
| Startup Time       | Fast (seconds)                                     | Slow (minutes)              |
| Image Size         | Small  | Large                       |





# Basic taxonomy in Docker



**Hosted Docker Registry**

Docker Trusted Registry on-prem.

**Docker Hub Registry**

Docker Trusted Registry on-cloud

**Azure Container Registry**

AWS Container Registry

Google Container Registry

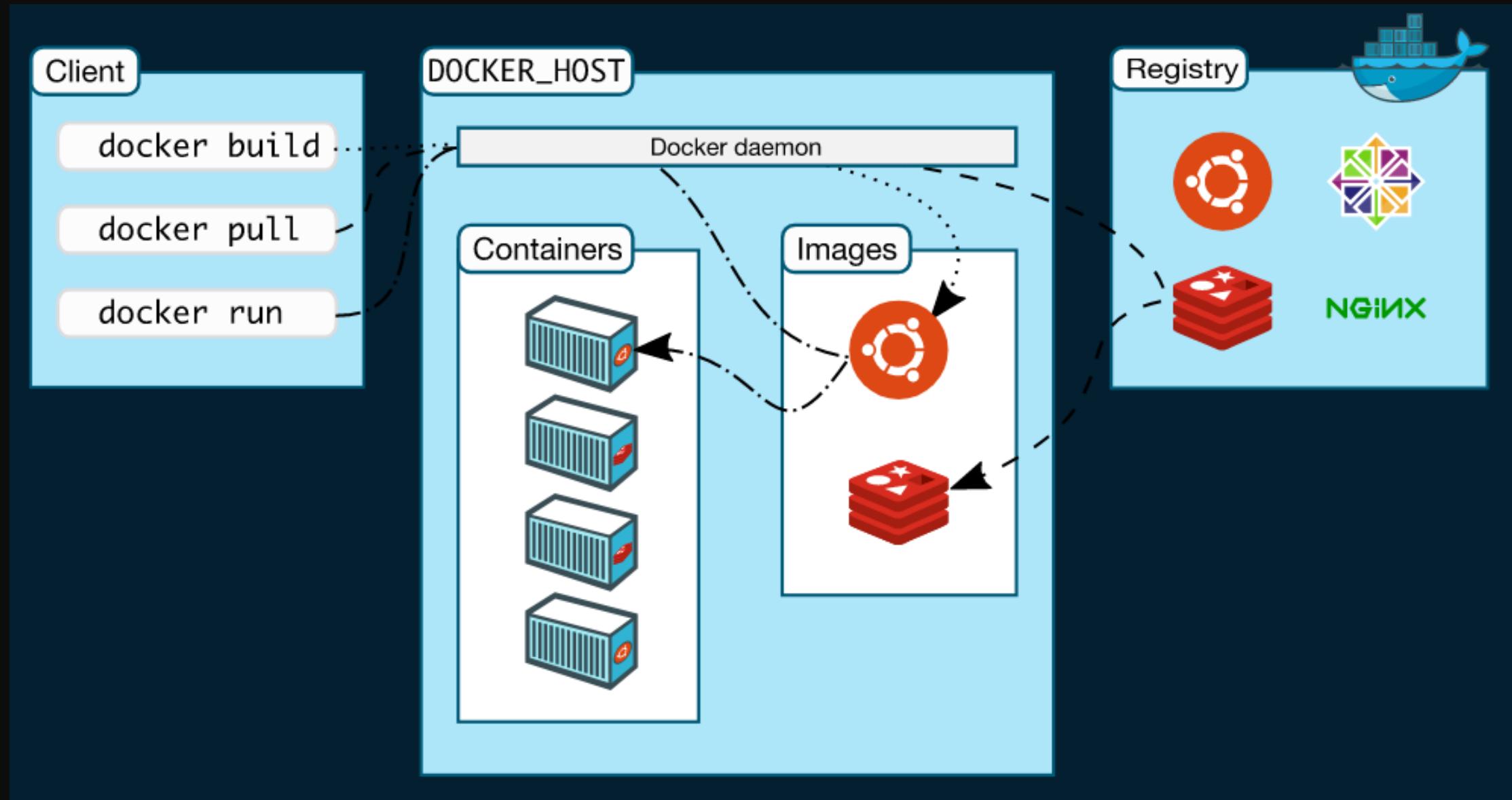
Quay Registry

Other Cloud

**On-premises**

('n' private organizations)

**Public Cloud**  
(specific vendors)



<https://docs.docker.com/get-started/overview/>

# What to expect from container?

- Namespace isolation: separate ps tree, network, user IDs, mounted fs
- Resource limiting/prioritization: CPU, memory, block I/O, network
- Layered filesystem: creation of container images
- Storage: volume mount, container image, network storage

# Motivation for Containerization

## **Utilization and Capacity Agility**

- Hyper-packing without *today's* virtualization overheads
- Flexible resource partitioning with provision for usage spikes
- Easy scale-out and scale-in
- Abstract from the underlying hardware.

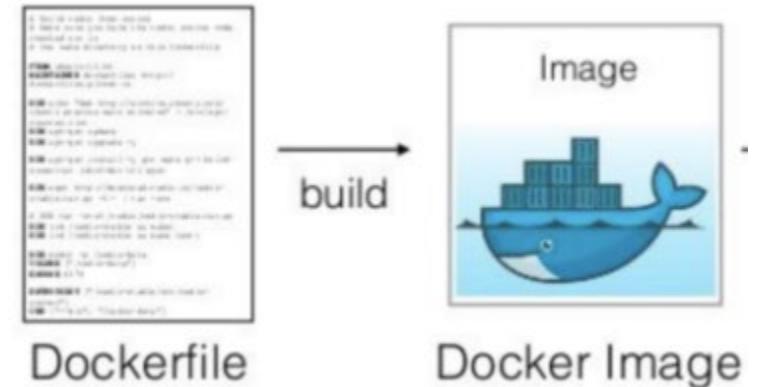
## **Dev Agility**

- Simplified CI/CD through container as the deployment unit.
- Bring your own version of dependencies.
- Seamless upgrades: Side-by-side container deployment and quick rollbacks.
- Reproducibility: Write once, run everywhere.
- Fast startup

# Containerizing an app

App + Dockerfile -> docker image

- Create a “Dockerfile”
  - Usually in the root directory of your app
  - List of instructions on how to build the image
  - Good for documentation as well
  - <INSTRUCTION> <value>
- Instructions
  - FROM alpine:tag // start from a base image
  - LABEL maintainer = “andavr@..” // add labels
  - RUN apk add build-base // run inside the container
  - COPY [location on host] [location in image]
  - WORKDIR /src //sets the working directory
  - EXPOSE 8080 //expose port to host
  - ENTRYPOINT [“entrypoint\_exe”, “params here”]



```
FROM alpine:latest
LABEL maintainer=andavr@microsoft.com
RUN apk add build-base
COPY . /src
WORKDIR /src
RUN gcc -o test test.c
RUN chmod +x test
ENTRYPOINT ["/src/test"]
```

```
docker image build -t capp .
docker run capp //will execute test
```

# Container example

```
docker container run -d --name my-apache-app -p 8080:80  
-v ${PWD}:/usr/local/apache2/htdocs/ httpd:2.4
```

Command line arguments:

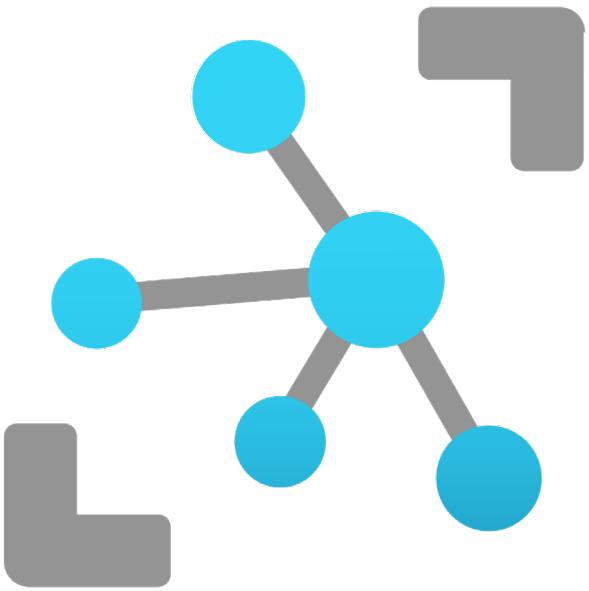
- -d : detached mode (run in background)
- -it : interactive (alternative to detached)
- --name : container name (used for stop, remove etc)
- -p : port mapping – 8080 on the host points to 80 in the container
- -v : volume mapping – current directory will be mapped to /usr/local/apache2/htdocs in the container
- httpd:2.4 : image\_name:tag
  - Official images are referred to by the name directly
  - Unofficial images : image\_repo/image\_name:tag



## Index of /

- [Administrator/](#)
- [Default/](#)
- [Public/](#)
- [QTA-repos\\_2/](#)
- [QTU-os\\_griffin-22/](#)
- [QTU-os\\_griffin-51/](#)
- [QTU-os\\_griffin-53/](#)
- [QTU-os\\_griffin-54/](#)
- [QTU-os\\_griffin-55/](#)
- [QTU-os\\_griffin-56/](#)
- [QTU-os\\_griffin-57/](#)
- [QTU-os\\_griffin-58/](#)
- [QTU-os\\_griffin-59/](#)
- [QTU-os\\_griffin-60/](#)
- [QTU-os\\_griffin-62/](#)
- [QTU-os\\_griffin-63/](#)
- [andavr/](#)
- [desktop.ini](#)

# Review Grafana Container



# Edge Analytics

## Local Execution

Stream analytics runs on 'edge devices'

## Unlock the Value of Untapped data

Only ~5% of data in industrial processes is sent to the cloud

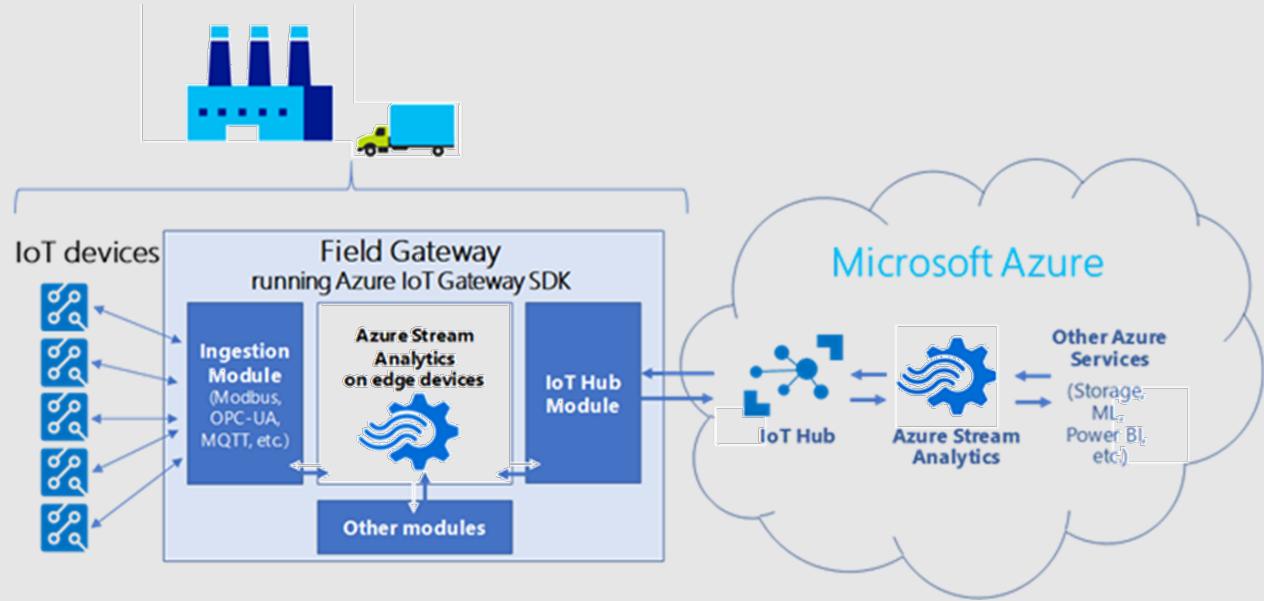
Deploy intelligence near the data to unlock the full value of data

## Seamless development and operations

Stream analytics jobs run in the cloud and on edge devices

## Intelligent actions

Deploy situational awareness, custom code, ML models on the edge



# Edge Analytics Scenarios

Low Latency

Resiliency

Efficient Use of Bandwidth

Compliance

Management at IoT-Scale

# Canonical Use Cases

## Reduction

When you are interested in only parts or significant changes in your operational data

## Aggregation

When business operations need an aggregate view

## Batching

When connectivity is intermittent and cost is high

## Transformation

Converting messages from legacy industrial automation to modern applications formats

## Edge Intelligence

Machine Learning models on the edge

Reference data

# Edge Analytics Platform Requirements

Windows or Linux

Azure IoT Gateway

Memory

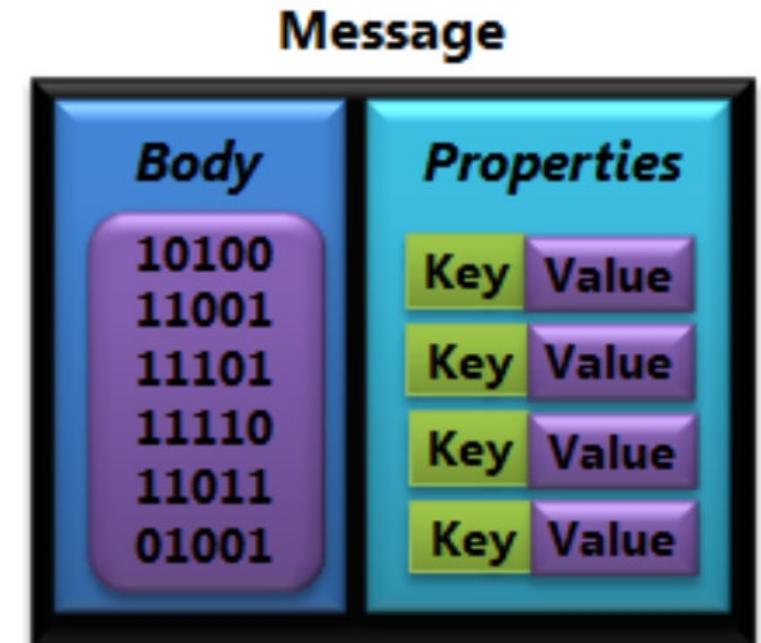
Minified stream analytics engine needs ~2MB of main memory

Storage and additional memory based on the amount of data

Node.js

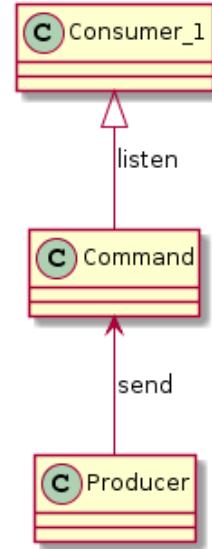
# What exactly is a message?

- Self-contained package of business data, user metadata, and system metadata
- Involves a sender and a receiver
- Unfortunately, the terminology can be confusing...
  - Messaging can take the form of *messages* or *events*



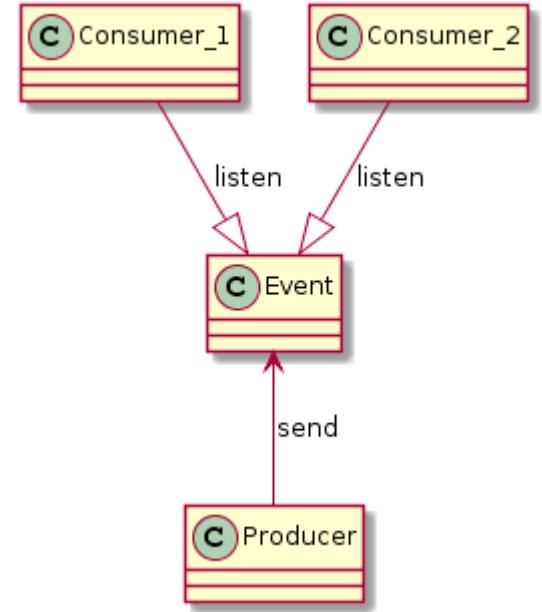
# Messaging

- Expressed as *command* or *query*
- A sender requires a receiver to perform an action (i.e. *future tense*)
- Sender has expectation of receiver and how it must handle the message
- Both sides are aware of a contract that defines the message
- The sender may or may not require a response
- We speak in terms of *sending messages*
- For messaging, we'll look at Storage and Service Bus Queues



# Eventing

- Notification of a state change, that has happened (i.e., *past tense*)
- Publisher has no expectation of how the event will be handled
- 0 – many subscribers can choose to act upon it
- Can be single, discrete event or part of a related series (stream)
- A contract to define the event exists among each participant
- We speak in terms of *publishing events*
- For Eventing, we'll look at Service Bus Topics, Event Grid, and Event Hub



# IoT Edge Offline Operation

# Offline Operation

## Configuration

- IoT Edge devices automatically have offline capabilities enabled
- Declare a parent-child relationship between the devices in IoT Hub

## Sync with IoT Hub

- The IoT Edge device needs to be online to sync with IoT Hub at least once

## Go offline

- IoT Edge device, modules, and children can operate indefinitely offline

## Reconnect and resync with IoT Hub

- IoT Edge device syncs again when reconnected

# Optional offline settings

## Time to live

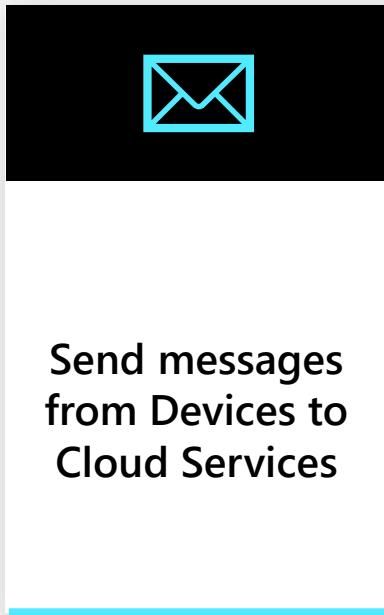
- Amount of time before messages expire
- Desired property of the IoT Edge Hub, stored in the Module Twin
- Default is 7200 seconds
- Configure in the Azure portal or directly in the Deployment Manifest

## Host storage for system modules

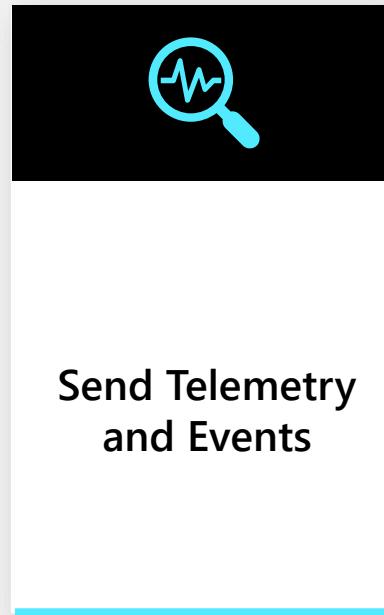
- Add additional space for Message Storage
- Messages and module state information stored in IoT Edge hub's local container filesystem
- Dedicate storage on the host IoT Edge device

# Configuring Message Routing in Azure IoT Hub

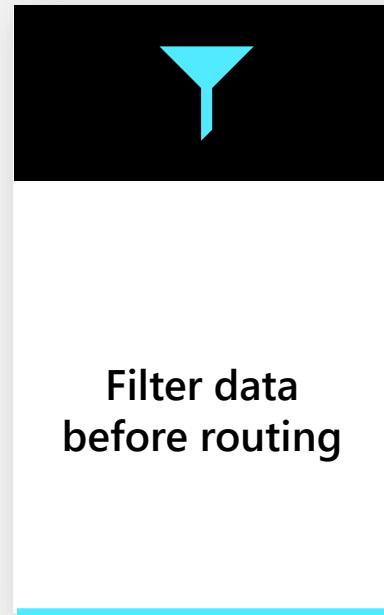
# Message routing in Azure IoT hub



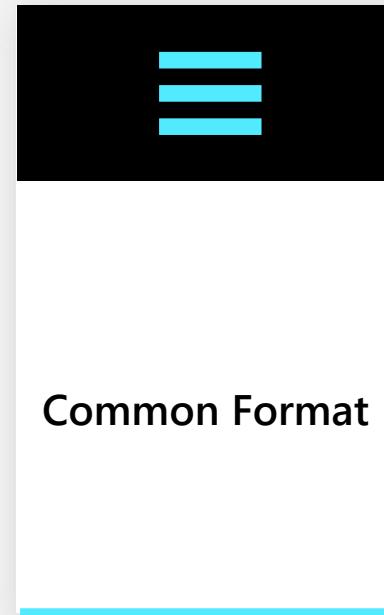
Send messages  
from Devices to  
Cloud Services



Send Telemetry  
and Events



Filter data  
before routing



Common Format

# Routing endpoints

## Built in Endpoint

- Handles Messages and Events
- Compatible with Event Hub

## Messages and Queries

- Routed to all endpoints that match a Query

## Endpoints

- Built-in endpoint
- Azure Storage
- Service Bus Queues and Service Bus Topics
- Event Hubs

# Routing endpoints

## Built in Endpoint

- Receive device-cloud messages
- Custom Routes will stop data flowing to Built in Endpoint

## Azure Storage

- Blob Storage or Data Lake
- Apache Avro (Default) or JSON
- Encoding set when endpoint created

## Service Bus

- Allows Message Routing to other Azure Services
- Must not have sessions or Duplicate Detection
- Endpoint unreachable

## Event Hubs

- Allows Message Routing to other Azure Services

# Fallback Route

Can be enabled if routing is turned on

Sends Messages that don't match routes to built-in endpoint

If no routes to Built-In Endpoint, only non matched messages are sent

Can be enabled and disabled in the Message Routing Blade

Can use an ARM template

# Adding Queries to Message Routes

## Message Properties

- Common format for all device-to-cloud messaging.
- JSON Format
- System Properties identify content of Message
- Can add Application Properties
- Not Case Sensitive
- Use Unique Property Names

## Message Body

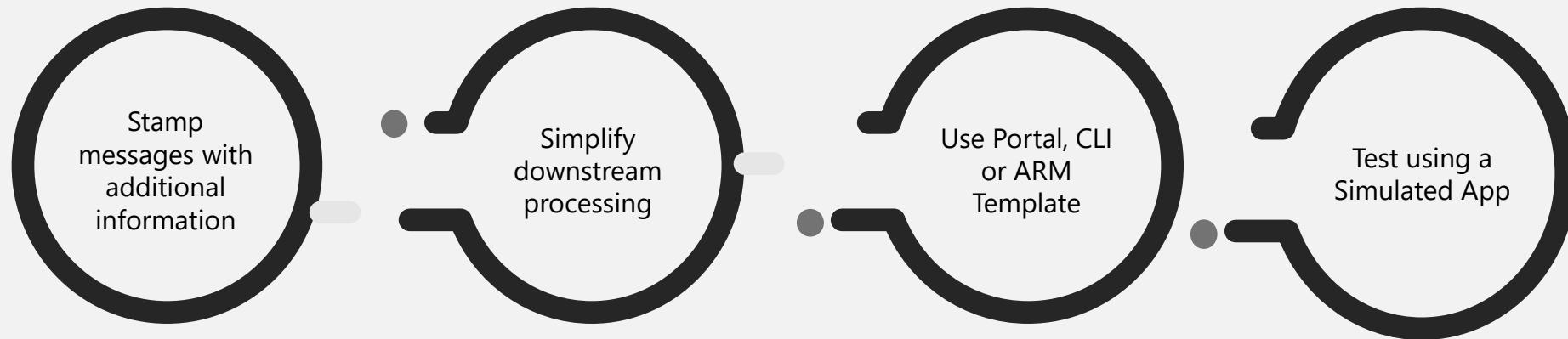
- Body must be JSON encoded
- contentType set to application/JSON
- contentEncoding set to support UTF encoding
- Non-qualifying Messages not evaluated

## Device Twin

- Query on Tags, Desired and Reported Properties
- Can Query on Module Twin

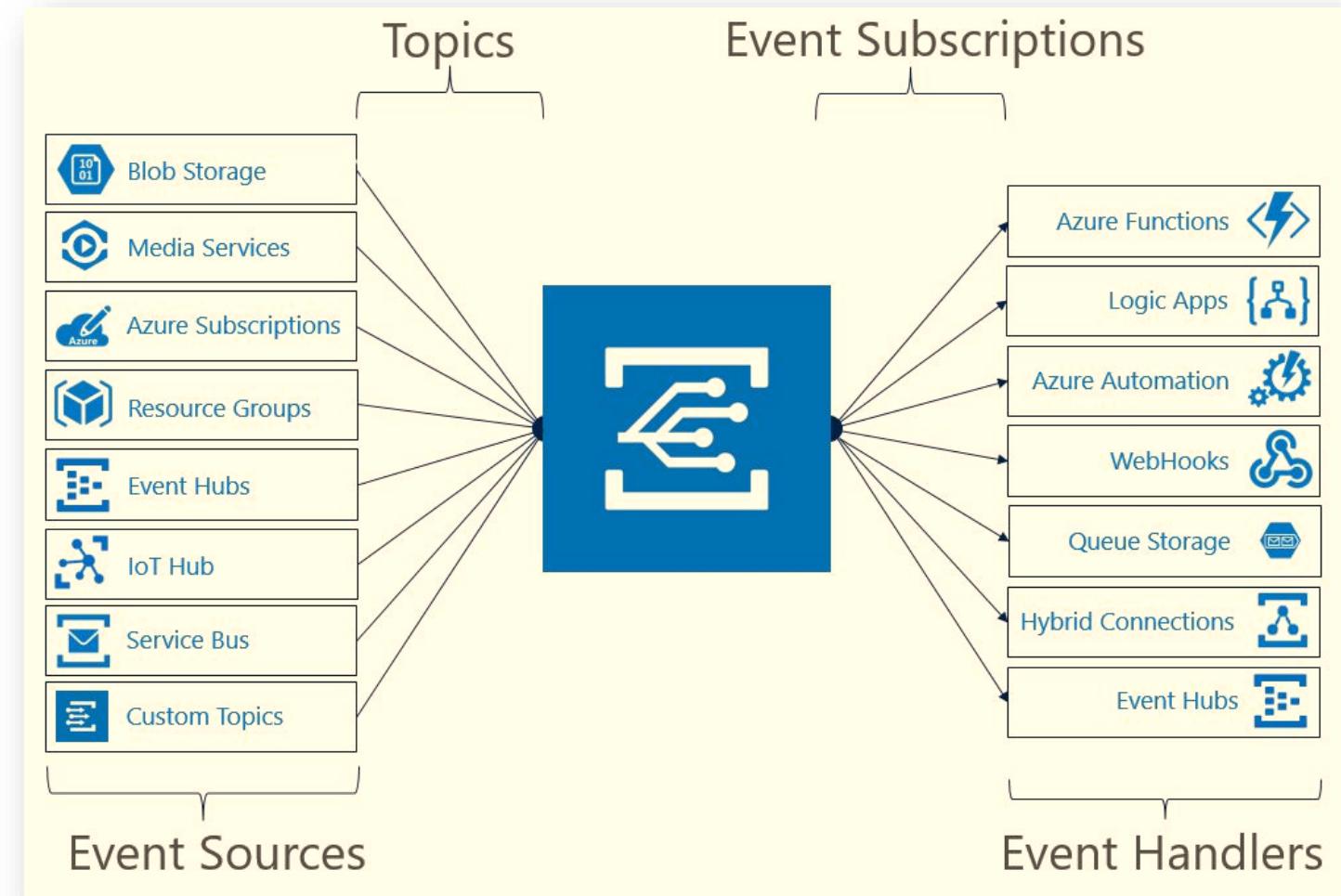
# Implementing Message Enrichments and Integrating with Event Grid

# Implementing Message Enrichments in IoT Hub



# Integrating with Event Grid

- Send Event Notifications to other Services
- Trigger Downstream Processes
- Built in support for Azure Functions, Logic Apps etc



# Stream Analytics Inputs and Outputs

## Supported stream input types

- IoT Hub
- Edge Hub
- Event Hub

## Supported stream output types

- Event Hub
- SQL Database
- Blob Storage/ADLS Gen2
- Edge Hub

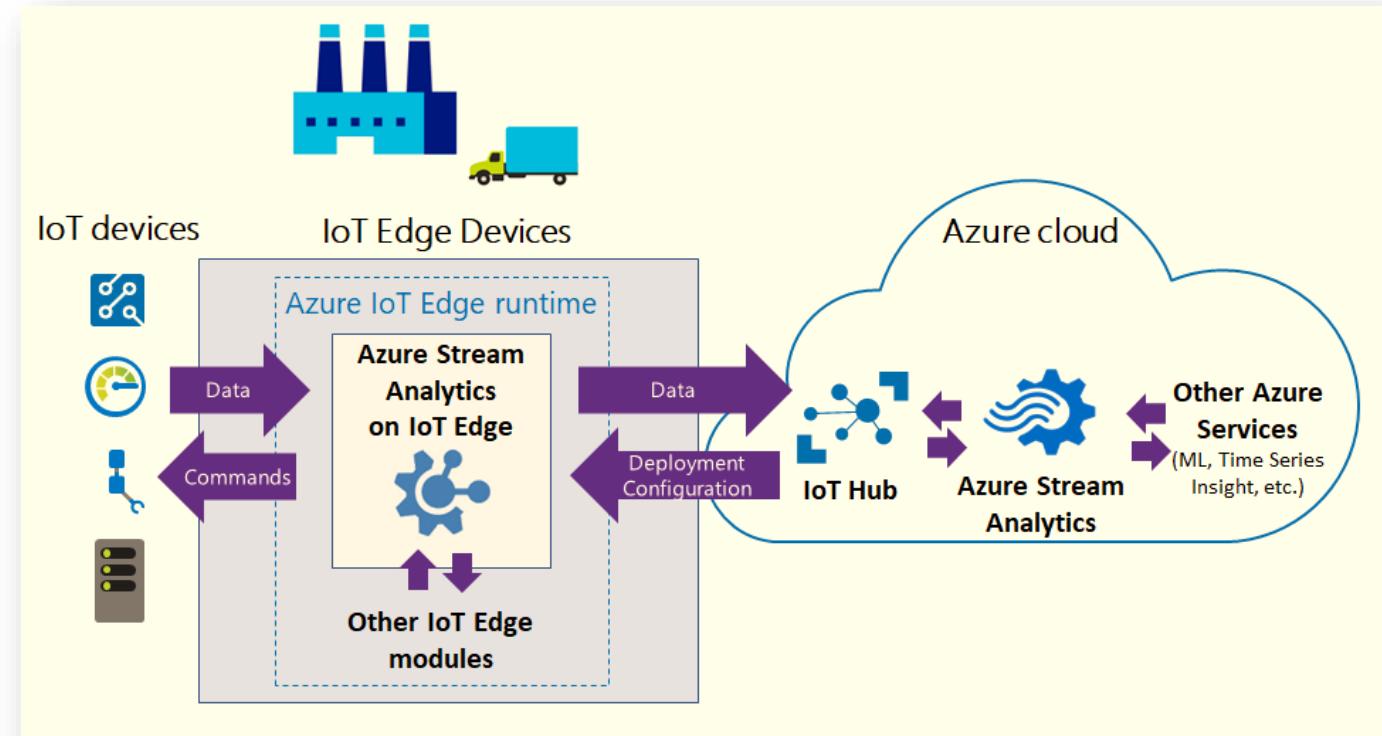
# Azure Stream Analytics on IoT Edge

Low-latency command and control

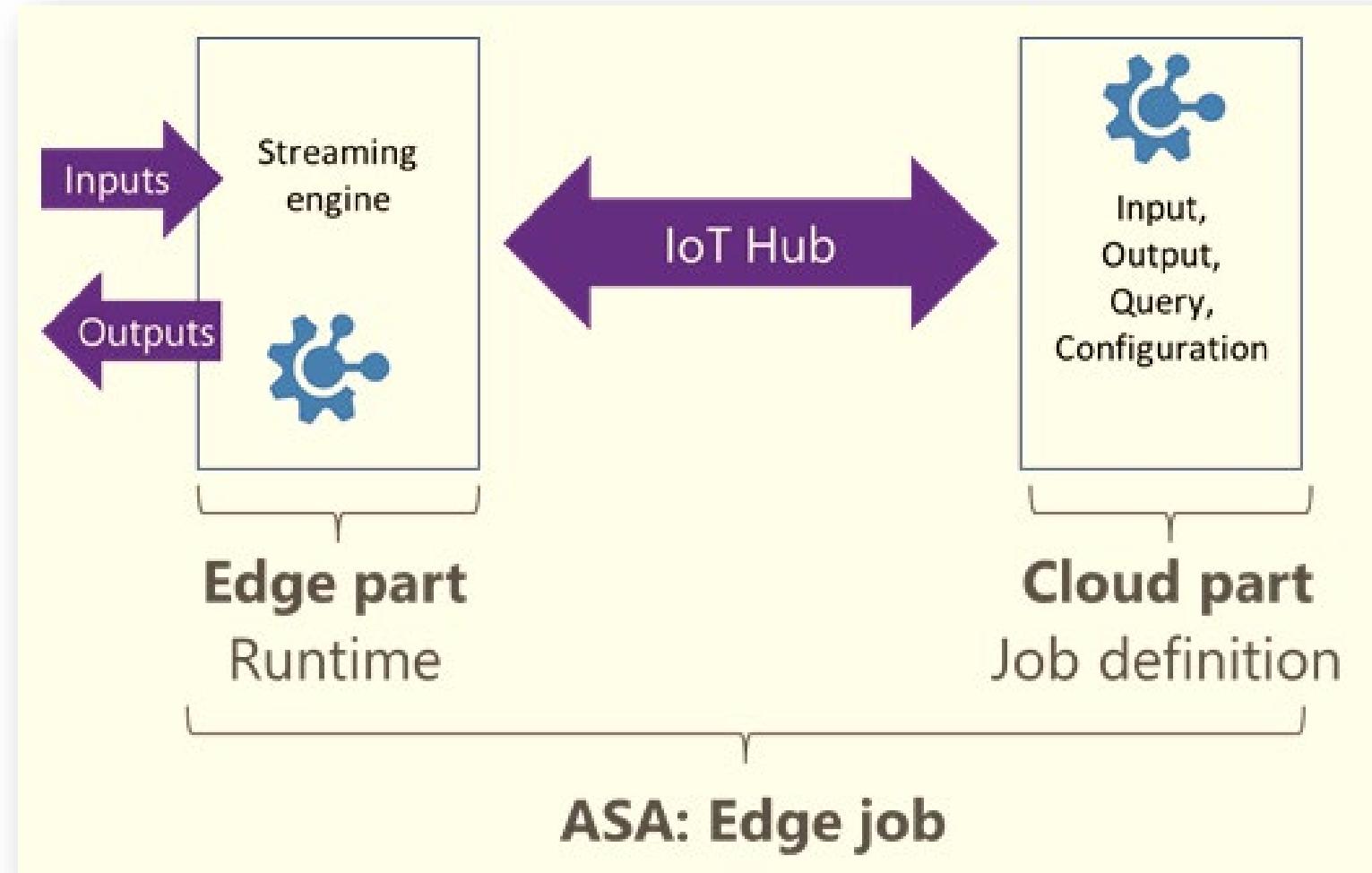
Limited connectivity to the cloud

Limited bandwidth

Compliance



# Edge jobs in Azure Stream Analytics



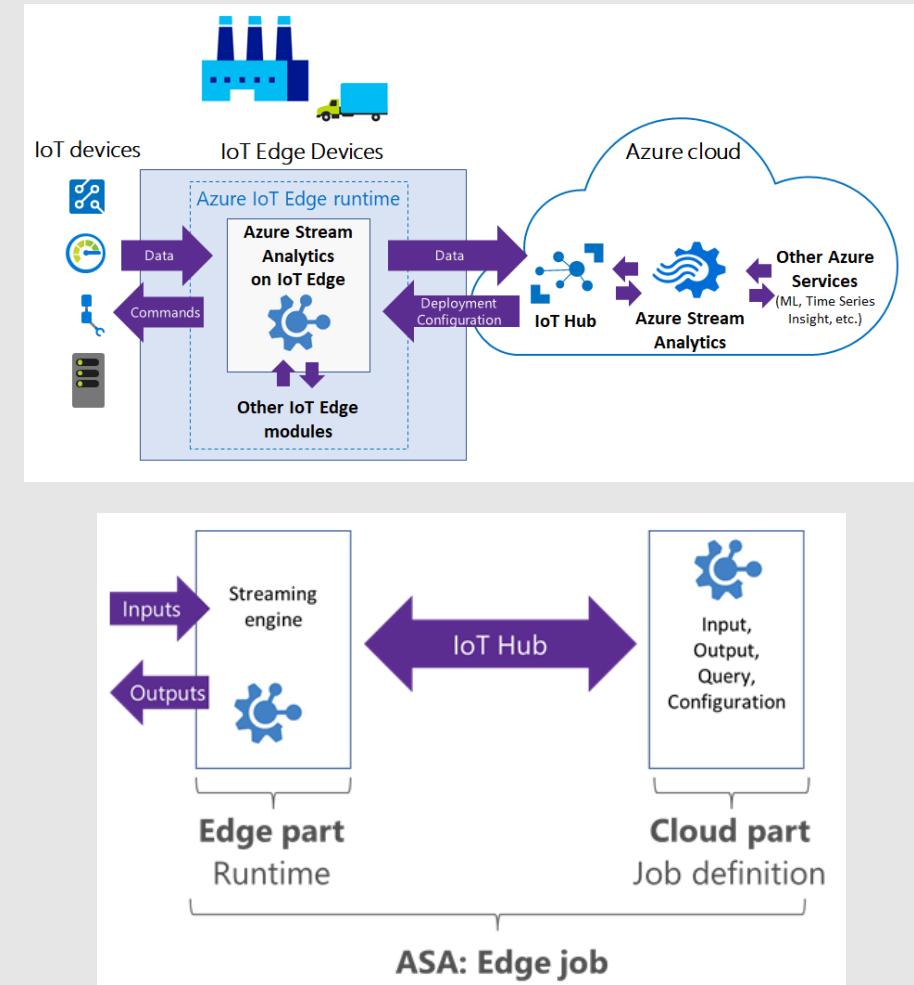
# Azure Stream Analytics on IoT Edge

Azure Stream Analytics (ASA) on IoT Edge empowers developers to deploy near-real-time analytical intelligence closer to IoT devices, unlocking the full value of device-generated data.

Enhancements include:

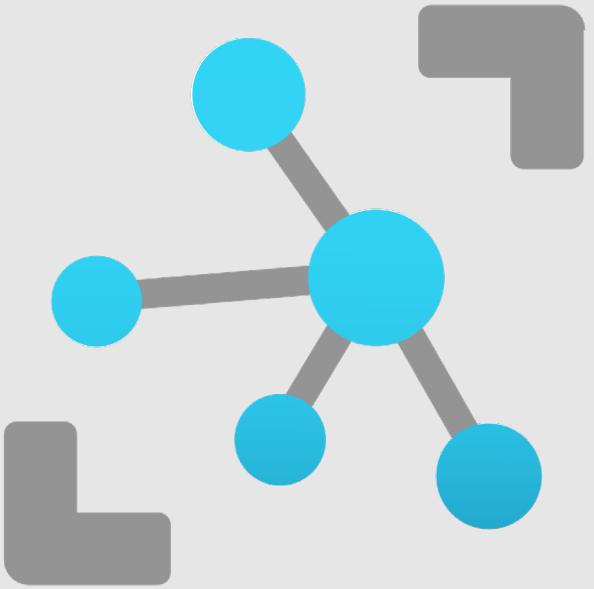
- **Improved support for offline scenarios.** After the initial deployment, ASA on IoT Edge can be restarted without any cloud connection.
- **Improved logging.** Developers can enable verbose debug logs for troubleshooting purposes.
- **Improved monitoring.** New metrics such as input, output, and error count are enabled.
- **Simplified update flow for query logic.** After updating the logic of your job in ASA, you can update your IoT Edge deployment in just a few clicks in the IoT Hub portal.
- **Ability to update reference data without restarting the container.** Updating reference data location can now be done through a new IoT Edge deployment.
- **Programmatic deployments.** ASA on IoT Edge jobs can now be created and packaged using rest APIs, allowing CI/CD.
- **Better parity with cloud jobs.** Multiple options previously only available for cloud jobs (e.g. GZIP compression, JSON format options, etc.) have been added, making it easier to move jobs between cloud and IoT Edge.

Learn more by reading the [announcement blog](#) and checking out our [documentation](#).



# Hands-On Lab

Create an Azure Stream Analytics Job

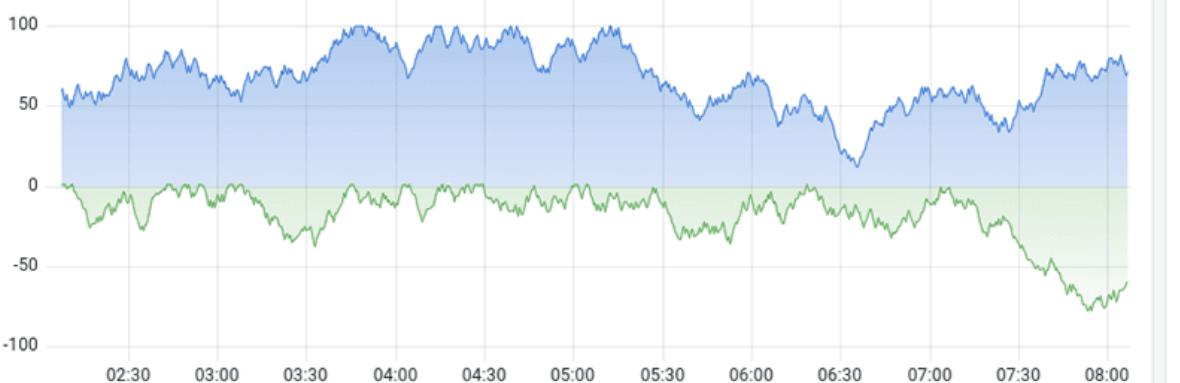


# Grafana

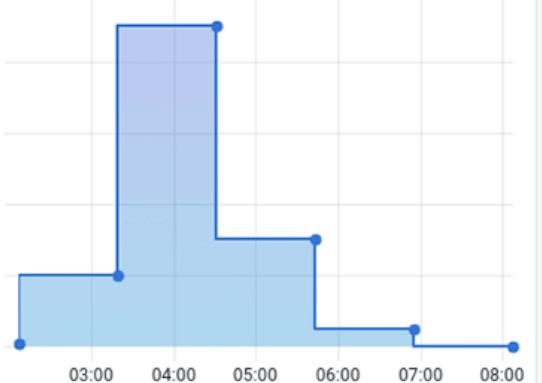
“Grafana allows you to query, visualize, alert on and understand your metrics no matter where they are stored. Create, explore, and share beautiful dashboards with your team and foster a data driven culture.”



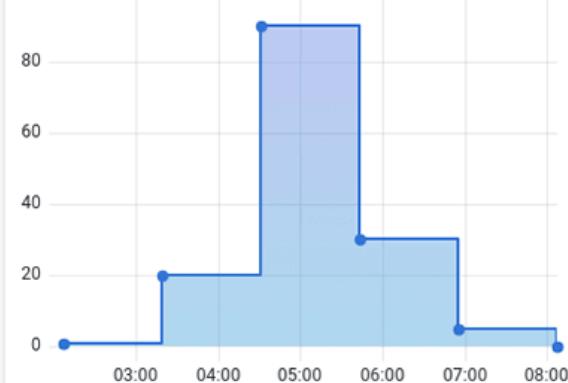
Line graph with opacity area



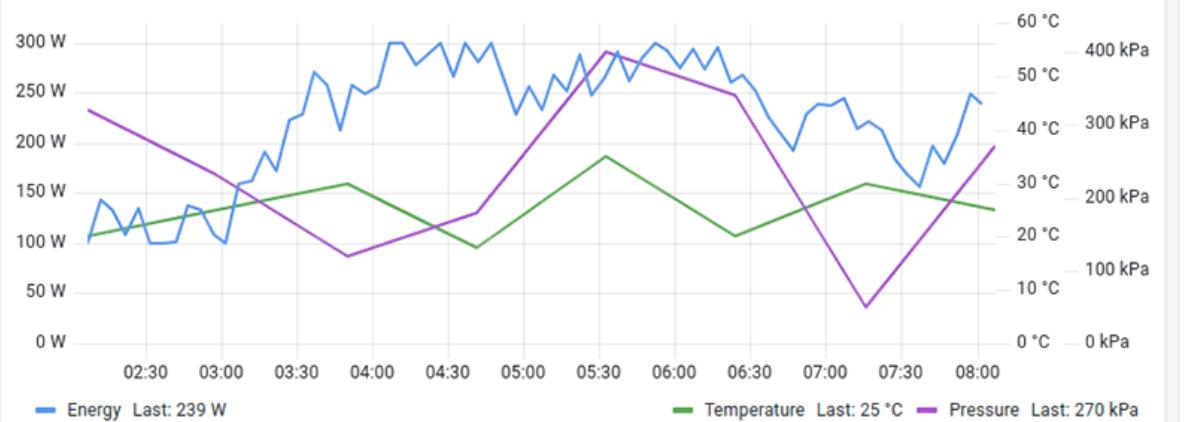
Interpolation mode: Step before



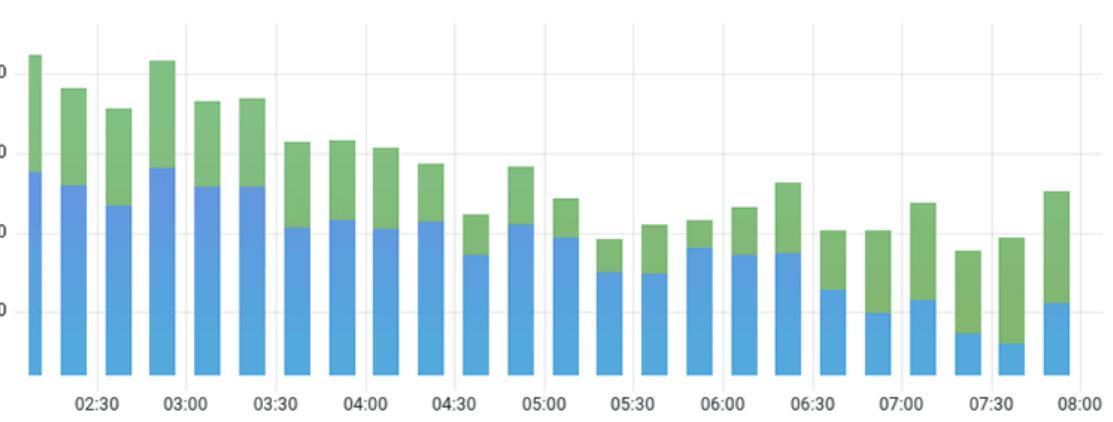
Interpolation mode: Step after



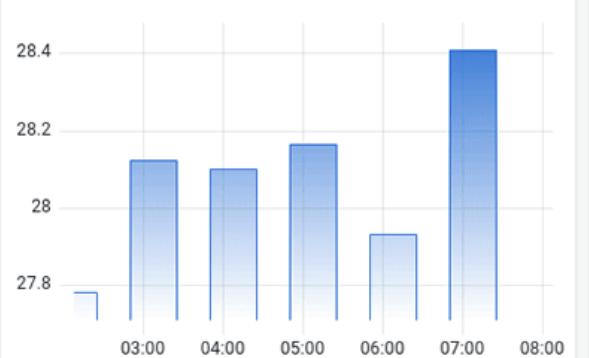
Multiple Y-Axes and units



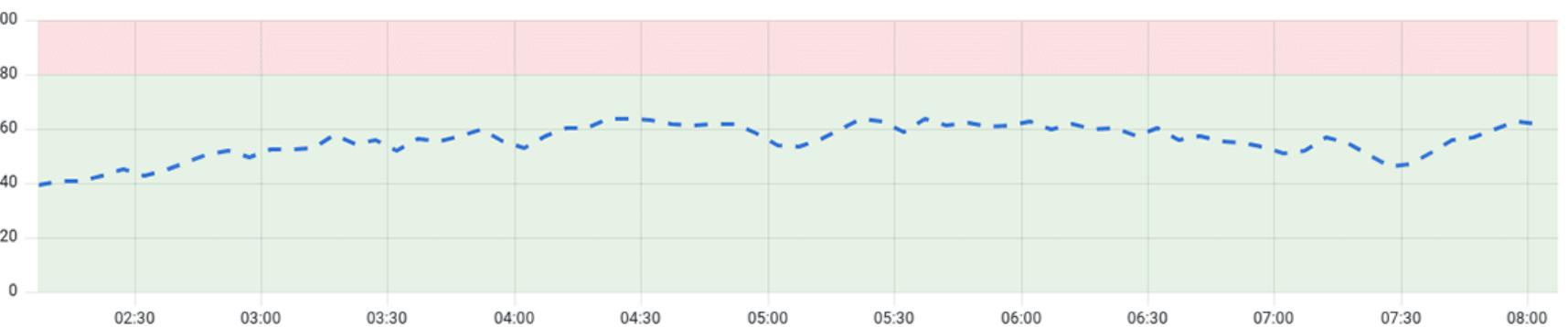
Stacked bars

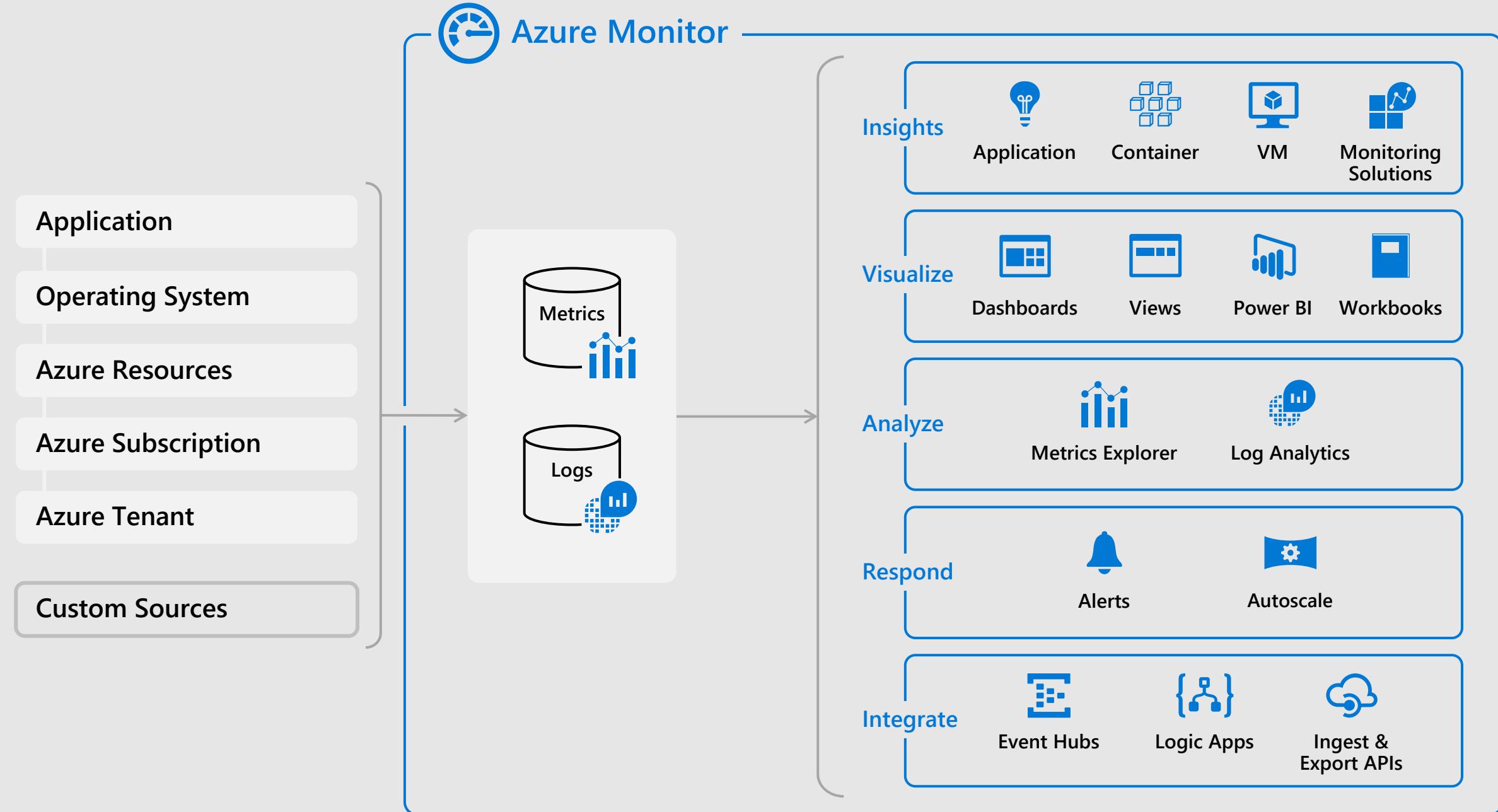


Bars with opacity gradient



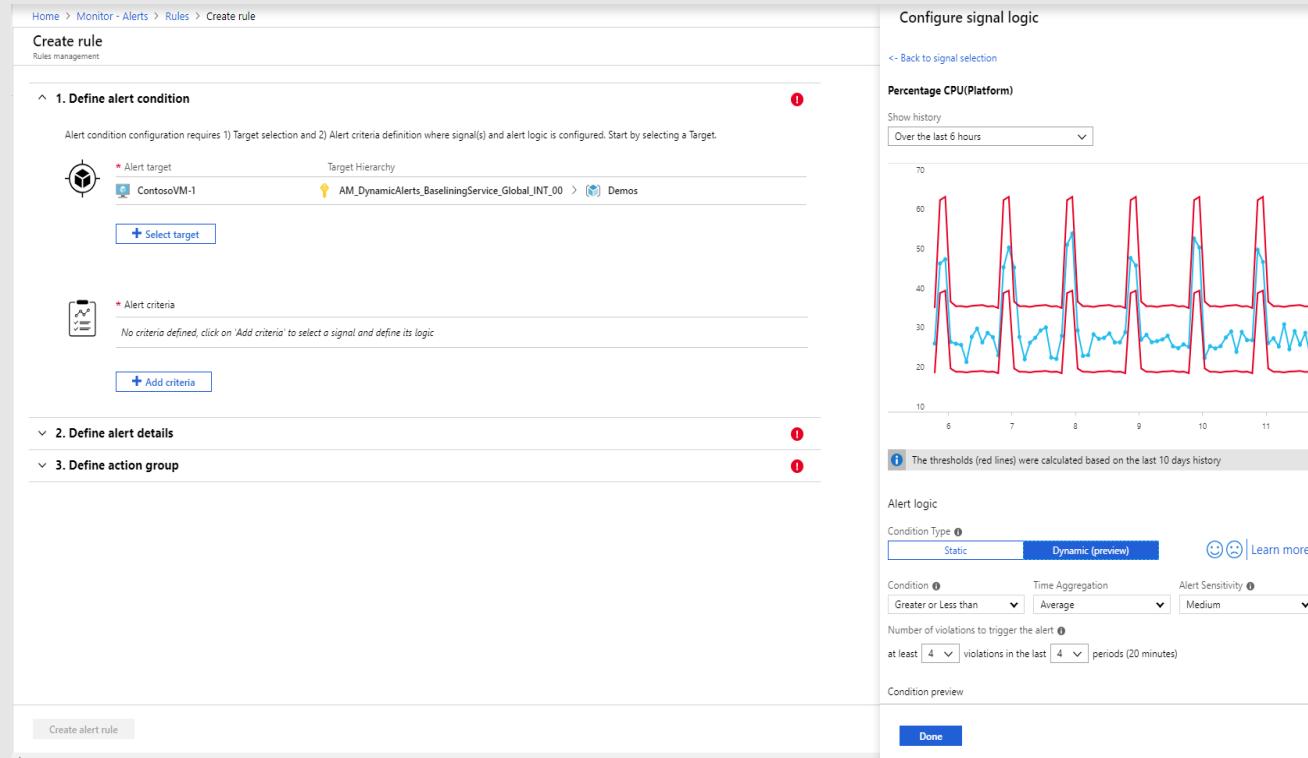
Dashed lines + threshold regions





# Smarter Alerts with Dynamic Thresholds

Intelligent tuning, Scalable authoring, Lower Noise



The screenshot shows the 'Create rule' wizard in the Azure portal. The current step is 'Configure signal logic'. It displays a chart titled 'Percentage CPU(Platform)' over the last 6 hours, showing red spikes indicating signal anomalies. Below the chart, it says 'The thresholds (red lines) were calculated based on the last 10 days history'. At the bottom, there's an 'Alert logic' section with 'Condition Type' set to 'Dynamic (preview)', 'Time Aggregation' to 'Average', and 'Alert Sensitivity' to 'Medium'. The condition is set to 'Greater or Less than' with a threshold of '4'. The 'Number of violations to trigger the alert' is set to 'at least 4 violations in the last 4 periods (20 minutes)'. A 'Done' button is at the bottom right.



Learns signal behavior and auto adjusts thresholds



Learns long-range patterns (holidays, downtimes) and reduces noise

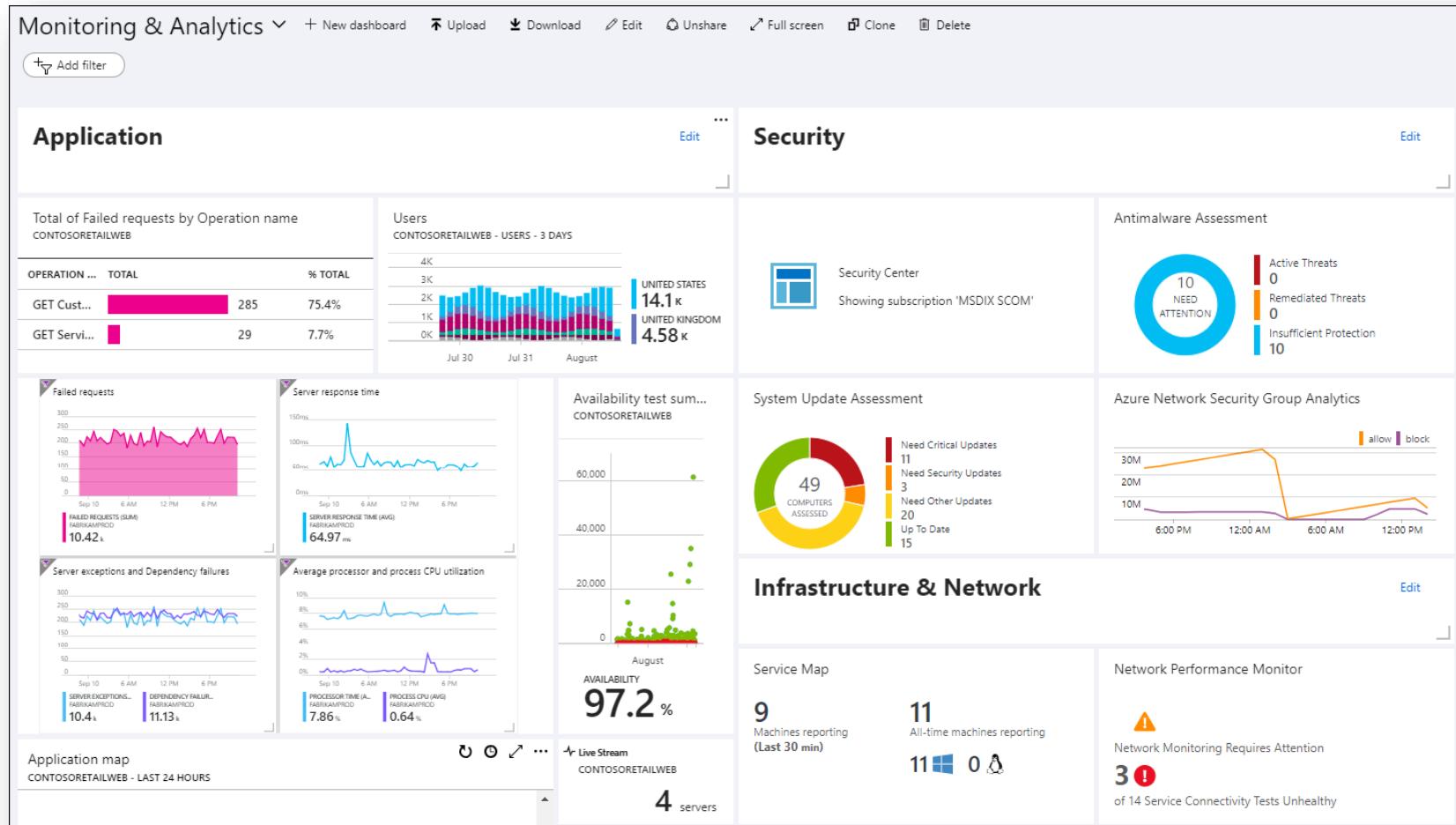


Learns service's performance and adapts thresholds

# Azure dashboards

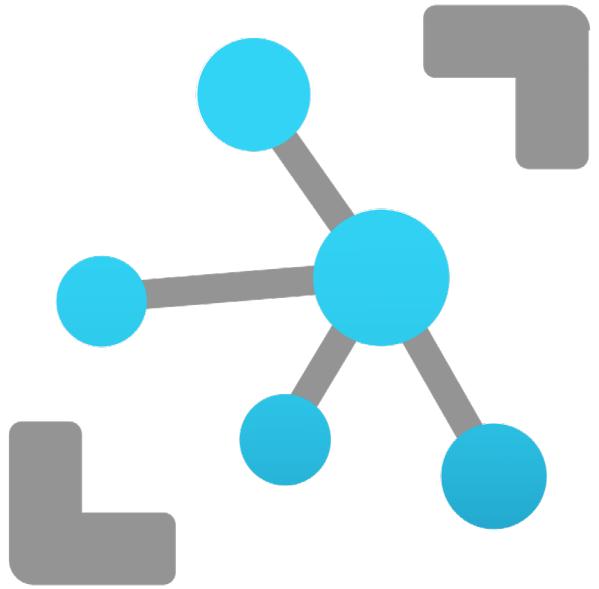
Combine different kinds of data, including both metrics and logs, into a single pane in the Azure portal and optionally share the dashboard.

Elements throughout Azure Monitor can be added: any log query or metrics chart



# Hands-On Lab

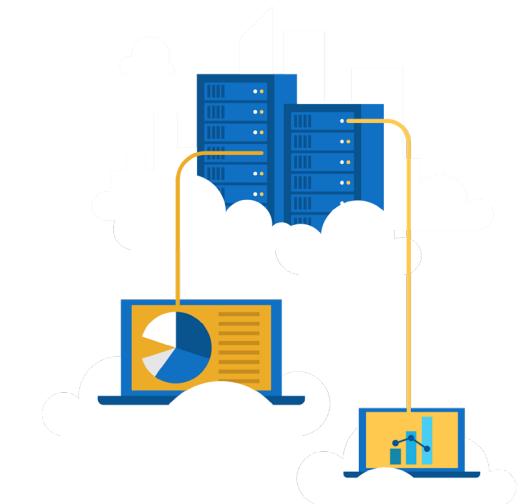
Azure Monitor, Log Analytics, Grafana



**What did we learn?**

**Send questions to [iotacademy@microsoft.com](mailto:iotacademy@microsoft.com)**

**We start tomorrow at 10:10am ET**



# Further Reference

- <https://github.com/Azure/iot-hub-device-update/tree/main/docs/agent-reference>
- <https://github.com/Azure/iot-hub-device-update/blob/main/docs/agent-reference/whats-new.md>
- <https://github.com/Azure/iot-hub-device-update/blob/main/docs/agent-reference/how-to-build-agent-code.md>

# Hands-On Lab



# Break Time!

