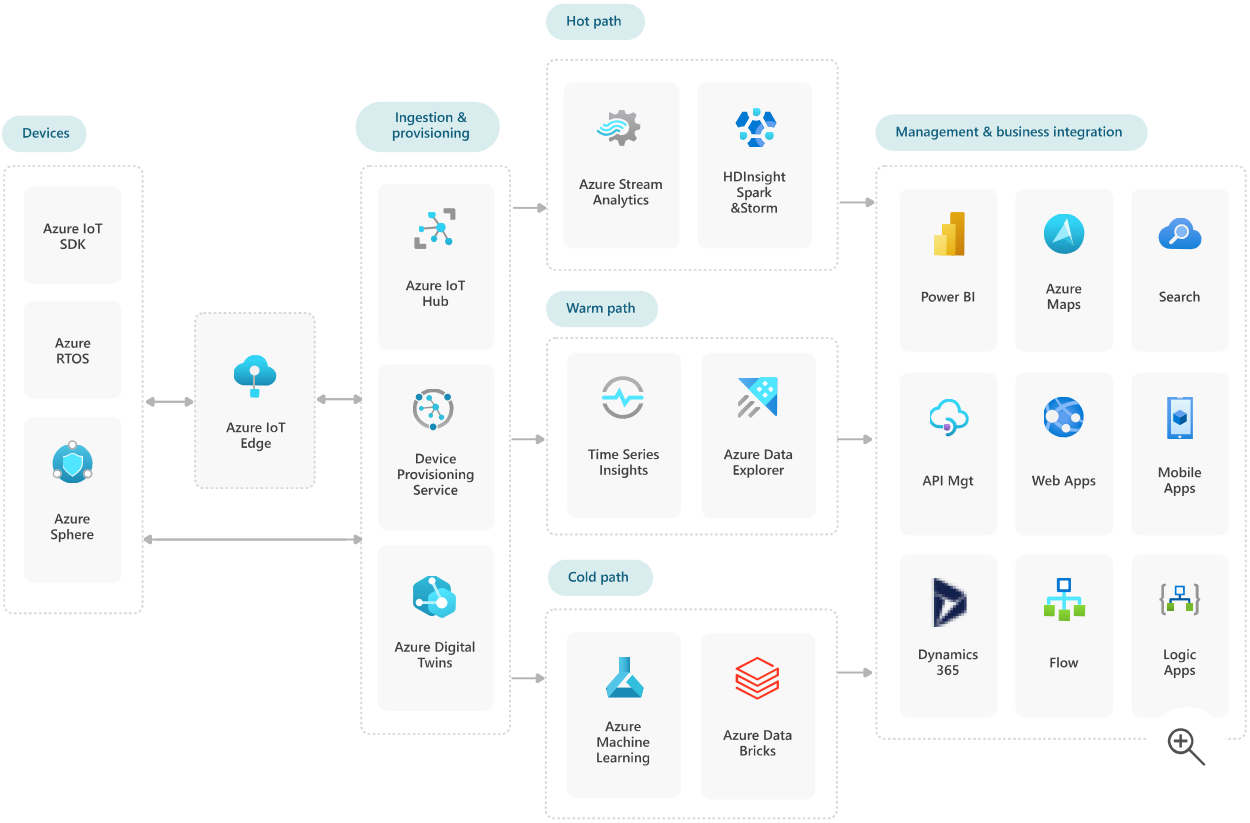
# Azure IoT reference architecture

Functions IoT Hub IoT Device Provisioning Service Stream Analytics Digital Twins

This article discusses a recommended architecture for IoT applications using Azure PaaS (platform-as-a-service) components. The following diagram reflects different Azure components that can be used to architect an IoT solution. The diagram shows, and the article highlights, most of the commonly used services, but no solution requires all of them.




This reference architecture uses Azure PaaS (platform-as-a-service) components. Microsoft recommends getting started with Azure IoT Central, which is an aPaaS (application platform-as-a-service) IoT solution platform. It is designed to simplify and accelerate IoT solution assembly and operations by preassembling, scaling, and managing many of the same PaaS services described in this reference architecture. The result is an out-of-the-box and ready to use UX and API surface area complete with the capabilities needed to connect, manage, and operate fleets of devices at scale. Learn more about how to compare IoT Central (aPaaS) to a PaaS solution approach based on your solution needs.

Azure IoT solutions involve **things** (typically **devices**) that generate data, **insights** that you form about the data, and **actions** that you take based on the insights. Consider a motor that sends temperature data. This data is used to evaluate whether the motor is performing as expected. The insight about its performance is used to prioritize a maintenance schedule for the motor.

If you want to see IoT reference architectures that address solutions that are specific to industry verticals, you can start here:

Industry specific IoT reference architectures

# Devices

Azure IoT supports a large range of devices, from microcontrollers running Azure RTOS and Azure Sphere to developer boards like MX Chip and Raspberry Pi. Azure IoT also supports smart server gateways capable of running custom code. Devices might perform some local processing through a service such as **Azure IoT Edge**, or just connect directly to Azure so that they can send data to and receive data from the IoT solution.

When devices are connected to the cloud, there are several services that assist with ingesting data. **Azure IoT Hub** is a cloud gateway service that can securely connect and manage devices. **IoT Hub Device Provisioning Service (DPS)** enables zero-touch, just-in-time provisioning that helps to register a large number of devices in a secure and scalable manner. **Azure Digital Twins** enables virtual models of real world systems.

# Insights

Once devices have been connected in the cloud, their data can be processed and explored to gain custom insights about their environment. At a high level, there are three ways to process data — hot path, warm path, and cold path. The difference between them has to do with requirements for latency and data access.

- The **hot path** analyzes data in near-real-time as it arrives. In the hot path, telemetry must be processed with very low latency. The hot path is typically implemented using a stream processing engine. Consider using services such as **Azure Stream Analytics** or **HDInsight**. The output may trigger an alert, or be written to a structured format that can be queried using analytical tools.
- The **warm path** analyzes data that can accommodate longer delays for more detailed processing. Consider **Azure Data Explorer** or **Azure Time Series Insights** for storing and analyzing large volumes of data.

- The **cold path** performs batch processing at longer intervals (hourly or daily). The cold path typically operates over large volumes of data which can be stored in **Azure Data Lake**, and the results don't need to be as timely as the hot or warm paths. Consider using **Azure Machine Learning** or **Azure Databricks** to analyze cold data.

# Actions

You can use the insights gathered about your data to manage and control your environment. Business integration actions might include storing informational messages, raising alarms, sending email or SMS messages, or integrating with business applications such as CRM and ERP. The following services are available for management and business integration:

- **Power BI** connects to, models, and visualizes your data. Power BI enables you to collaborate on data and use artificial intelligence to make data-driven decisions.
- **Azure Maps** allows you to create location aware web and mobile applications using geospatial services (search, maps, routing, tracking, and traffic), APIs, and SDKs.
- **Azure Cognitive Search** provides a search service over varied types of content. This includes indexing, AI enrichment, and querying capabilities.
- **Azure API Management** provides a single place to manage all of your APIs.
- **Azure Web Apps** enables you to deploy web applications that scale with your organization.
- **Mobile Apps** allows you to build cross platform and native apps for iOs, Android, Windows, or Mac.
- **Dynamics 365** combines CRM (customer relationship management) and ERP (enterprise resource planning) in the cloud.
- **Microsoft Flow** is a SaaS offering for automating workflows across applications and other SaaS services.
- **Azure Logic Apps** is a cloud-based PaaS offering used to create and automate workflows that integrate your apps, data, services, and systems.

There are also several services provided by Azure to help you monitor your entire IoT solution and keep it secure. Diagnostic services include **Azure Monitor**. Security services such as **Azure Active Directory** and **Microsoft Defender for IoT** help you control, view, and manage your security settings, threat detection and response.

# Digital Twins

Customers are exploring Digital Twins as a mechanism to control and monitor connected environments. A digital twin is a virtual model of a real-world environment that is driven with data from business systems and IoT devices. It is used to enable insights and actions for a business or organization. Developers and architects are looking to digital twins as the solution that enables intelligent and connected environments such as the following:

- Predictive maintenance in manufacturing
- Supply chain visibility
- Smart shelves for real-time inventory
- Connected homes and smart buildings

# Deployment at scale

Build your solution to deploy at global scale. For optimal scalability, build your IoT application as discrete services that can scale independently. This section contains scalability considerations for various Azure services.

**Functions**. When reading from the Event Hubs endpoint, there is a maximum of function instance per event hub partition. The maximum processing rate is determined by how fast one function instance can process the events from a single partition. The function should process messages in batches.

**IoT Hub**. For IoT Hub, consider the following scale factors:

- The maximum daily quota of messages into IoT Hub.
- The quota of connected devices in an IoT Hub instance.
- Ingestion throughput — how quickly IoT Hub can ingest messages.
- Processing throughput — how quickly the incoming messages are processed.

Each IoT hub is provisioned with a certain number of units in a specific pricing and scale tier. The tier and number of units determine the maximum daily quota of messages that devices can send to the hub. For more information, see IoT Hub quotas and throttling. You can scale up a hub without interrupting existing operations.

**Stream Analytics**. Stream Analytics jobs scale best if they are parallel at all points in the Stream Analytics pipeline, from input to query to output. A fully parallel job allows Stream Analytics to split the work across multiple compute nodes. For more information, see Leverage query parallelization in Azure Stream Analytics.

IoT Hub automatically partitions device messages based on the device ID. All of the messages from a particular device will always arrive on the same partition, but a single

partition will have messages from multiple devices. Therefore, the unit of parallelization is the partition ID.

# Security

This section contains considerations for building secure solutions.

## Zero Trust security model

Zero Trust is a security model that assumes breaches will happen and treats every access attempt as if it originates from an open network. Zero Trust assumes that you have implemented the basics such as securing identities and limiting access. This includes explicitly verifying users, having visibility into their devices, and being able to make dynamic access decisions using real-time risk detection. After the basics are met, you can shift your focus to the following Zero Trust requirements for IoT solutions:

- Use strong identity to authenticate devices.
- Use least privileged access to mitigate blast radius.
- Monitor device health to gate access or flag devices for remediation.
- Perform updates to keep devices healthy.
- Monitor to detect and respond to emerging threats.

Read the Zero Trust Cybersecurity for the Internet of Things whitepaper for full details.

## Trustworthy and secure communication

All information received from and sent to a device must be trustworthy. Unless a device can support the following cryptographic capabilities, it should be constrained to local networks and all internetwork communication should go through a field gateway:

- Data encryption and digital signatures with a provably secure, publicly analyzed, and broadly implemented symmetric-key encryption algorithm.
- Support for either TLS 1.2 for TCP or other stream-based communication paths or DTLS 1.2 for datagram-based communication paths. Support of X.509 certificate handling is optional and can be replaced by the more compute-efficient and wire-efficient pre-shared key mode for TLS, which can be implemented with support for the AES and SHA-2 algorithms.
- Updateable key-store and per-device keys. Each device must have unique key material or tokens that identify it to the system. The devices should store the key securely on the device (for example, using a secure key-store). The device should

be able to update the keys or tokens periodically, or reactively in emergency situations such as a system breach.
- The firmware and application software on the device must allow for updates to enable the repair of discovered security vulnerabilities.

Many devices are too constrained to support these requirements. In that case, a field gateway should be used. Devices connect securely to the field gateway through a local area network, and the gateway enables secure communication to the cloud.

### Physical tamper-proofing

It is strongly recommended that device design incorporates features that defend against physical manipulation attempts, to help ensure the security integrity and trustworthiness of the overall system.

For example:

- Choose microcontrollers/microprocessors or auxiliary hardware that provides secure storage and use of cryptographic key material, such as trusted platform module (TPM) integration.
- Secure boot loader and secure software loading anchored in the TPM.
- Use sensors to detect intrusion attempts and attempts to manipulate the device environment with alerting and potential "digital self-destruction" of the device.

For additional security considerations, see Internet of Things (IoT) security architecture.

## Reliability and performance

A key area of consideration for resilient IoT solutions is business continuity and disaster recovery. Designing for High Availability (HA) and Disaster Recovery (DR) can help you define and achieve required uptime goals for your solution.

Different Azure services offer different options for redundancy and failover to help you achieve the uptime goals that best suit your business objectives. Incorporating any of these HA/DR alternatives into your IoT solution requires a careful evaluation of the trade-offs between the:

- Level of resiliency you require
- Implementation and maintenance complexity
- Cost of Goods Sold (COGS) impact

The article Azure Business Continuity Technical Guidance describes a general framework to help you think about business continuity and disaster recovery. The Disaster recovery

and high availability for Azure applications paper provides architecture design guidance on strategies for Azure applications to achieve High Availability (HA) and Disaster Recovery (DR).

You can also find service-specific performance information in the documentation for each Azure IoT service.

## Cost considerations

In general, use the Azure pricing calculator to estimate costs. Other considerations are described in the Cost section in Microsoft Azure Well-Architected Framework.

## Next steps

For more information about the individual pieces of a solution architecture, see the following topics:

- Azure IoT Edge
- Azure IoT Hub
- Azure IoT Hub Device Provisioning Service (DPS)
- Azure Digital Twins
- Azure Stream Analytics
- Azure HDInsight
- Azure Time Series Insights
- Azure Data Explorer
- Azure Machine Learning
- Azure Databricks
- Power BI
- Azure Maps
- Azure Cognitive Search
- API Management
- Azure App Service
- Azure Mobile Apps
- Dynamics 365
- Microsoft Power Automate (Microsoft Flow)
- Azure Logic Apps

---

## Recommended content

### IoT solution architecture - Azure Example Scenarios

Understand the topological relationship between IoT devices, platform, and applications and learn about IoT gateways, communications protocols, and provisioning.

### Condition monitoring for industrial IoT - Azure Solution Ideas

This example demonstrates how manufacturers can connect their assets to the cloud using OPC UA and the Industrial Components.

### IoT conceptual overview - Azure Example Scenarios

Learn how events, insights, and actions interact in internet-of-things (IoT) solutions.

### Get started with Azure IoT solutions - Azure Reference Architectures

Learn basic IoT concepts, how to get started building an Azure IoT solution, and how to optimize an IoT solution for production.

### Azure industrial IoT guidance - Azure Architecture Center

Examine architectural guidance on Azure industrial IoT (IIoT) analytics using platform as a service (PaaS) components.

### Create smart places by using Azure Digital Twins - Azure Example Scenarios

Use Azure Digital Twins to create models of smart places from IoT device data. View and control products, systems, environments, and experiences.

### Automated guided vehicles fleet control - Azure Example Scenarios

This example architecture shows an end-to-end approach for an automotive original equipment manufacturer (OEM) and includes a reference architecture and several published supporting open-source libraries that can be reused.

### Connected factory hierarchy service - Azure Solution Ideas

Implement a hierarchy service so business stakeholders can centrally define the organization of production assets like machines within factories.

Show more