Université Abdelmalek Essaadi

Ecole Nationale des Sciences Appliquées Al-Hoceima

MODULE : Internet of Things (IoT) and Cloud Computing

Project Report:

IoT Clock with Real-Time Weather Display (Azure Cloud Only)

Prepared by : MAROUA ALAMI HARRAK - HAJAR ELYAZRI

Supervised by : Pr. H. Routaib

Second Year Digital Transformation
and Artificial Intelligence – TDIA 2

Academic Year : 2025 - 2026

Table of Contents:

I. General Introduction :

Over the last decade, the rapid evolution of digital technologies has transformed the way data is collected, processed, and consumed. The Internet of Things (IoT) has become a key enabler of real-time services by connecting physical devices to cloud platforms capable of storing, analyzing, and visualizing data at scale. Weather monitoring and time synchronization are among the most common IoT use cases as they require accurate, continuously updated information.

This project, *"IoT Clock with Real-Time Weather Display (Azure Cloud Only)"*, aims to design and implement a complete IoT pipeline capable of collecting real-time weather information, transmitting it to the cloud, processing it, and displaying it on a dynamic dashboard. The solution relies entirely on Microsoft Azure services, taking advantage of their scalability, availability, and built-in IoT capabilities.

Problem Statement

The project answers the following questions:

- How can an IoT device acquire and send real-time weather data to the cloud?
- How can Microsoft Azure process, transform, and visualize these streaming data flows?
- How can we build a reliable end-to-end architecture for real-time weather and clock display?

Objectives

- Simulate an IoT device using Python and a weather API (OpenWeather).
- Create an Azure IoT Hub to receive streaming telemetry data.
- Configure an Azure Stream Analytics Job to process the data.
- Build a real-time dashboard in Power BI showing time, temperature, humidity, and weather status.

Impact of the Project

This work provides hands-on experience with modern IoT and Cloud Computing technologies, while developing essential engineering skills such as:

- Designing IoT architectures
- Understanding real-time data processing
- Using Azure IoT services
- Creating real-time dashboards with Power BI

The project also demonstrates the practical integration between IoT, cloud platforms, and data visualization tools.

II. IoT Concepts :

1. Introduction to IoT

The Internet of Things (IoT) refers to a network of physical devices—known as “smart objects”—that are equipped with sensors, actuators, and communication interfaces. These devices collect data from their environment, communicate with cloud platforms, and enable intelligent decision-making.

❖ IoT System Components

A typical IoT system includes:

- **Sensors** (temperature, humidity, GPS, etc.)
- **Connectivity layer** (Wi-Fi, MQTT, HTTP, LoRa, etc.)
- **IoT Hub / IoT Edge /Cloud gateway**
- **Data processing layer** (Stream Analytics, Functions, etc.)
- **Applications & dashboards** (Power BI, Web Apps, etc.)

IoT follows the general flow:
Device → **Data Transmission** → **Cloud Processing** → **Visualization / Action**

2. IoT Protocols and Communication

2.1 TCP/IP Model (Reminder)

IoT communication relies on the traditional TCP/IP model:

- **Application layer**: MQTT, HTTP, CoAP
- **Transport layer**: TCP, UDP
- **Network layer**: IP
- **Link layer**: Wi-Fi, Ethernet, Bluetooth

IoT often uses lightweight protocols like MQTT, but for this project, Python uses HTTP (API call) + MQTT/AMQP (when sending to IoT Hub).

2.2 NTP Protocol (Network Time Protocol)

➢ Definition

NTP is a networking protocol used to synchronize clocks across computer systems with high precision. It allows devices to maintain accurate and consistent time.

➢ Role in Real-Time Systems

For IoT applications requiring real-time displays (like clocks), NTP ensures:

- Accurate timestamps
- Synchronization between devices and servers
- Ordering of events and telemetry

➢ Importance for an IoT Clock

In this project:

- The weather data includes timestamps.
- Power BI requires synchronized timestamps for accurate real-time streaming.
- The real-time clock displayed on the dashboard must remain precise.

Therefore, **NTP (Network Time Protocol)** is crucial to ensure consistency between the device time, the cloud time, and the dashboard time.

2.3 MQTT

➢ **Definition**

MQTT (Message Queuing Telemetry Transport) is a lightweight publish–subscribe network protocol used to transport messages between devices. It is designed for remote or resource-constrained environments, where low bandwidth and minimal power consumption are required.

➢ **Role in IoT and in This Project**

In this project, MQTT plays a central role within the Azure IoT communication pipeline:

- As shown in the architecture, data transmission uses MQTT at the edge level (from the Docker container) and for ingestion into Azure IoT Hub.
- It enables efficient and low-power communication for telemetry such as temperature, wind speed, and humidity.
- Azure IoT Hub relies on MQTT as one of its primary protocols for secure and reliable device-to-cloud and cloud-to-device messaging.

3. Real-Time Data Processing in IoT

IoT applications such as weather monitoring rely on continuous real-time data streams. This requires:

- Low-latency communication between the device and Azure IoT Hub
- Streaming analytics capable of processing incoming data instantly
- Immediate visualization through a real-time dashboard (Power BI)

In this project:

- The Python weather module simulates a sensor that periodically sends telemetry data.
- Azure Stream Analytics processes this data in real time as soon as it reaches IoT Hub.
- Power BI updates the dashboard instantly, displaying live metrics such as temperature, wind speed, and humidity.

4. IoT Security

Security is a fundamental part of IoT because devices are exposed to network threats.

❖ Key Security Components

- **Device authentication** using keys or certificates
- **Secure communication** with encrypted channels
- **Cloud-level identity control** (Azure IoT Hub SAS Tokens)
- **Integrity of transmitted messages**

❖ Security in This Project

Azure IoT Hub ensures:

- Device authentication via **Primary Connection String**
- **Encrypted HTTPS communication** for telemetry messages
- Integrity and confidentiality of transmitted data

Even though the device is simulated, it uses **real security mechanisms** used in industrial IoT solutions. NTP is used for **accurate time synchronization**, ensuring timestamps in the telemetry data are reliable.

III. CLOUD COMPUTING CONCEPTS & AZURE

1. Definition of Cloud Computing

Cloud Computing is the delivery of computing services—including servers, storage, databases, networking, software, analytics, and intelligence—over the Internet (“the cloud”) to offer faster innovation, flexible resources, and economies of scale.

Key Features

- **Scalability:** Ability to increase or decrease resources based on demand.
- **Pay-as-you-go:** Pay only for what you use.
- **High availability:** Services are accessible globally at any time.

- **Managed services:** Cloud providers handle maintenance and updates.

In this project, Cloud Computing enables **real-time processing and visualization of IoT data** without needing physical infrastructure.

2. Service Models in Relation to the Project

Cloud computing provides three main service models:

Service Model	Example in Project	Role in Project
IaaS (Infrastructure as a Service)	**Azure Virtual Machine**	Provides computing and storage resources
PaaS (Platform as a Service)	**Azure IoT Hub, Azure Stream Analytics, edge azure**	Allows deploying IoT solutions and streaming analytics without managing infrastructure
SaaS (Software as a Service)	**Power BI**	Provides ready-to-use analytics and dashboards

Project relevance: In this project, **IoT Hub** and **Stream Analytics** handle the IoT data pipeline, and **Power BI** visualizes the results.

3. Deployment Models

Cloud platforms can be deployed in three main ways:

- **Public Cloud:** Services are delivered over the internet and shared across organizations. *(Used in this project – Microsoft Azure)*

- **Private Cloud:** Services are exclusive to one organization.

- **Hybrid Cloud:** Combines private and public clouds for flexibility.

Project choice: Public Cloud (Azure) allows fast deployment, low cost, and access to ready-to-use IoT and analytics services.

4. Advantages of Cloud Computing for IoT

Cloud Computing offers several benefits for IoT projects:

- **Real-time processing:** Handles continuous streaming data from devices.
- **High security:** Built-in encryption and identity management.
- **Scalability:** Supports multiple devices and large data volumes.
- **Reliability and availability:** Data processing and dashboards are always accessible.
- **Integration with analytics:** Easily connected to Power BI for visualization.

In this project, Azure Cloud ensures that simulated IoT devices send weather data to a scalable platform, which can process and visualize the data in real-time.

5. Azure Services Used in the Project

Here we connect theory to practice:

5.1 Azure IoT Hub

- **Definition:** IoT Hub is a managed service that acts as a central message hub for bi-directional communication between IoT applications and devices.

- **Role in the project:** Receives real-time data from the simulated weather device and forwards it to IoT Edge and Stream Analytics.

5.2 Azure IoT Edge

- **Definition:** Extends cloud intelligence to local IoT devices. Supports running containerized workloads close to the device.

- **Role in the project:** Processes data at the edge before sending it to the cloud, reducing latency and bandwidth usage.

5.3 Azure Container Registry (ACR)

- **Definition:** Azure Container Registry (ACR) is a private registry service where you can store, manage, and deploy Docker images or containerized artifacts.
- **Role in the project:** ACR stores the Docker image of the IoT Edge module used to process or transform data before sending it to the cloud. It enables easy deployment of Edge module updates from Azure to the device, ensuring centralized, secure, and scalable management of the containers used in the IoT workflow.

5.4 Azure Stream Analytics

- **Definition:** A PaaS service to analyze and process real-time data streams using SQL-like queries.

- **Role in the project:** Transforms incoming IoT data, calculates metrics, and sends processed data to Power BI for visualization.

5.5 Azure Power BI

- **Definition:** A SaaS solution for creating interactive dashboards and reports.
- **Role in the project:** Displays the real-time clock, temperature, humidity, and weather conditions. The dashboard updates dynamically as new data arrives.

Project relevance:

The pipeline is:
Python Device → IoT Hub → IoT Edge → Stream Analytics → Power BI
IoT Edge ensures real-time processing and efficient data management.

6. Technologies and Prerequisites

- **Python 3**: To simulate IoT device and weather API calls
- **OpenWeatherMap API key**: Provides real-time weather data
- **Azure Subscription**: Access to IoT Hub, IoT Edge, and Stream Analytics
- **Azure IoT Hub Device Connection String**: For secure device-to-cloud communication
- **Python libraries**: requests, azure-iot-device, json….

These technologies are essential to implement the end-to-end IoT + Cloud solution.

IV . Project Architecture

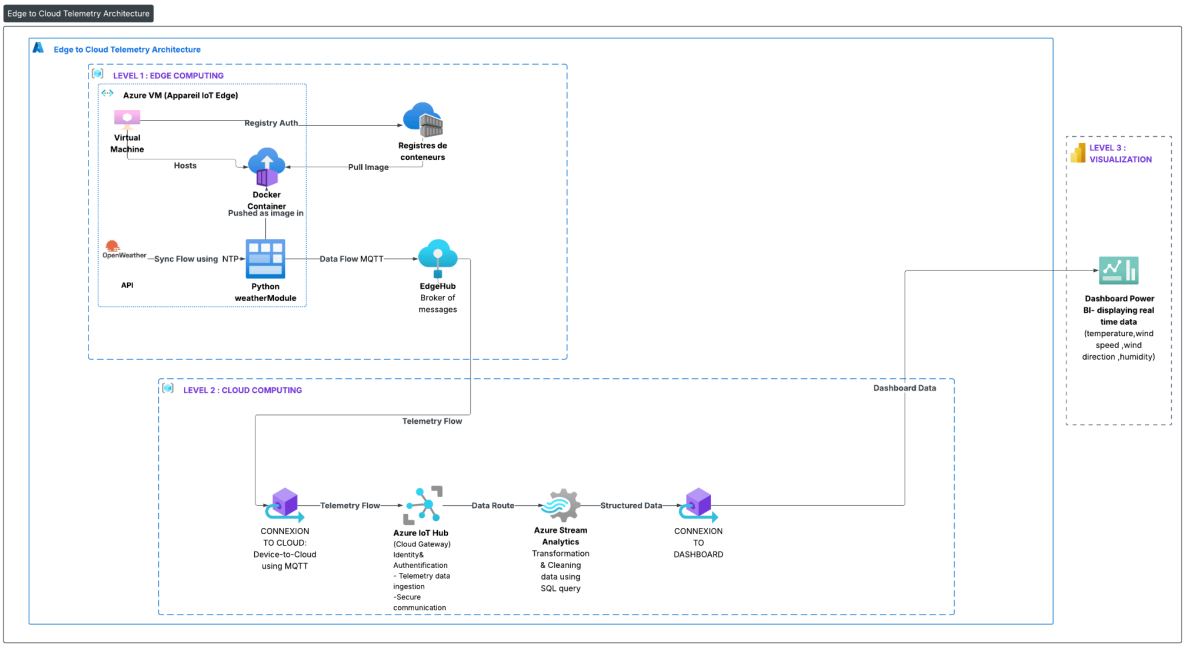
1. Global Architecture of the Project

The project relies on a complete IoT architecture that enables the collection, processing, and real-time visualization of weather data and time.
This architecture follows a structured pipeline composed of five main layers:

Python Weather Simulation → Azure IoT Hub → Azure IoT Edge → Stream Analytics → Power BI Dashboard

This pipeline ensures reliable data ingestion, local pre-processing, real-time SQL processing, and instant visualization in Power BI.

1.1. Architecture Diagram



2. Component Description

2.1. Python Script (Simulated IoT Device)

The Python script acts as a virtual IoT sensor.
Its main functions are:

- retrieving weather data from the **OpenWeatherMap API**,
- structuring the data in **JSON** format: temperature, humidity, weather condition, timestamp,
- sending these data to **Azure IoT Hub** using the connection string.

It simulates a real IoT device without needing physical hardware like ESP32 or Raspberry Pi.

2.2. Azure IoT Hub (Cloud Gateway)

Azure IoT Hub is the central cloud gateway of the IoT system.
In this project, it performs the following roles:

- authenticating the Python device using the **Primary Connection String**,
- receiving JSON telemetry,
- managing and securing device-to-cloud communication,
- distributing data to **IoT Edge** and **Stream Analytics**.

2.3. Azure IoT Edge (Edge Processing Layer)

Azure IoT Edge moves part of the processing logic closer to the device (edge).
Its roles in the project include:

- local data pre-processing,
- filtering or transforming data before sending to the cloud,
- reducing bandwidth consumption,
- optimizing data flow.

It acts as an intelligent intermediary between **IoT Hub** and **Stream Analytics**.

2.4. Azure Stream Analytics (Real-Time Processing)

Stream Analytics provides real-time analysis of the continuous data stream.
It allows:

- continuous ingestion of data coming from IoT Hub/Edge,
- applying SQL queries to filter, select, or enrich the data,
- producing a cleaned and structured stream for Power BI.

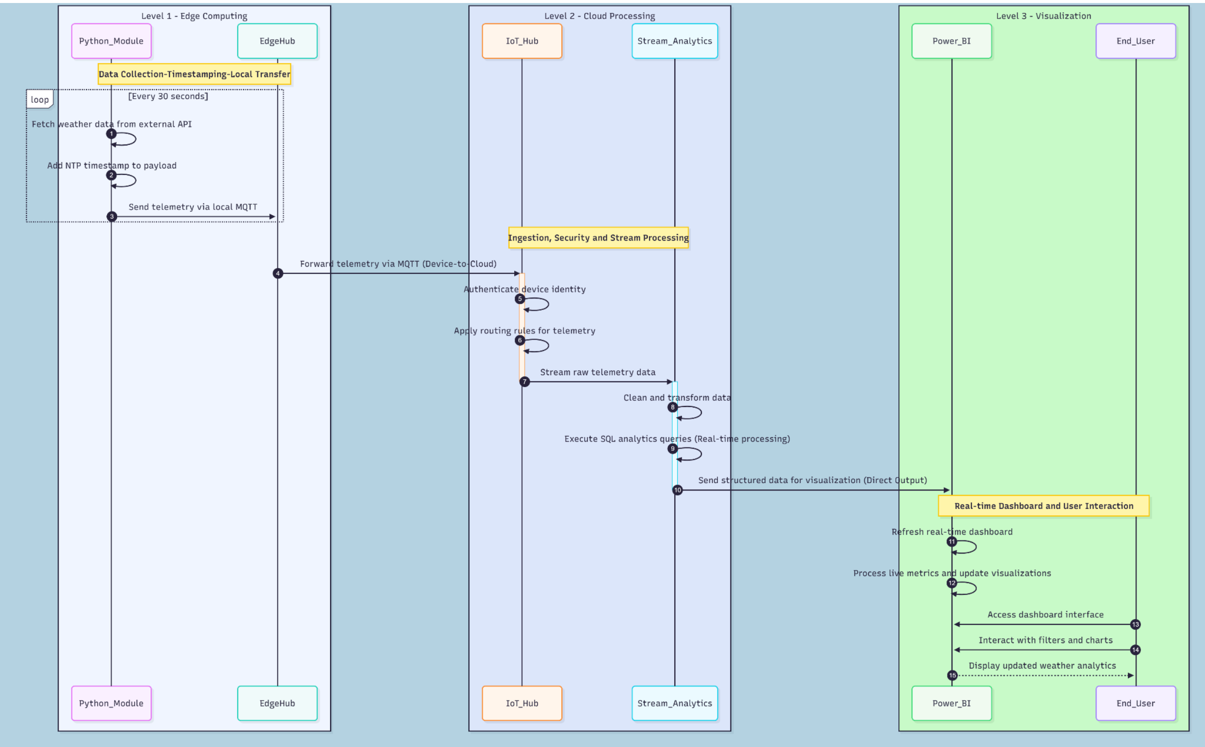
2.5. Power BI Dashboard (Visualization Layer)

Power BI represents the final layer of the system, dedicated to visualization.
The dashboard displays:

- temperature,
- humidity,
- wind speed &wind direction ,
- synchronized time.

The data is updated automatically thanks to the real-time connection from Stream Analytics.

3. Data Flow Description


Level 1 - Edge Computing
Python_Module
EdgeHub
Data Collection-Timestamping-Local Transfer
loop
[Every 30 seconds]
Fetch weather data from external API
1
Add NTP timestamp to payload
2
Send telemetry via local MQTT
3
Level 2 - Cloud Processing
IoT_Hub
Stream_Analytics
Ingestion, Security and Stream Processing
Forward telemetry via MQTT (Device-to-Cloud)
4
Authenticate device identity
5
Apply routing rules for telemetry
6
Stream raw telemetry data
7
Clean and transform data
8
Execute SQL analytics queries (Real-time processing)
9
Send structured data for visualization (Direct Output)
10
Level 3 - Visualization
Power_BI
End_User
Real-time Dashboard and User Interaction
Refresh real-time dashboard
11
Process live metrics and update visualizations
12
Access dashboard interface
13
Interact with filters and charts
14
Display updated weather analytics
15
Python_Module
EdgeHub
IoT_Hub
Stream_Analytics
Power_BI
End_User

3.1. Python Device → IoT Hub

- Python-based edge device generates weather telemetry with accurate timestamps.
- Uses secure device connection strings to connect to Azure IoT Hub.
- Ensures authenticated, encrypted communication with the cloud.
- Handles reliable data collection and transfer of raw sensor data.
- Maintains data integrity and follows security protocols during transmission.

3.2. IoT Hub → IoT Edge

- IoT Hub receives the telemetry and routes it to IoT Edge for preprocessing.

- IoT Edge acts as a local processing layer to reduce latency.
- Performs filtering, validation, and early aggregation of incoming data.
- Removes redundant or unnecessary values to optimize bandwidth usage.
- Ensures only relevant and validated data continues to cloud-level processing.

3.3. IoT Edge → Stream Analytics

- IoT Edge sends refined telemetry to Azure Stream Analytics.
- Stream Analytics ingests continuous data in real time.
- Applies SQL-based queries for complex event processing and transformations.
- Performs data normalization, outlier detection, and metric calculations.
- Converts raw telemetry into structured, analytics-ready datasets.

3.4. Stream Analytics → Power BI

- Cleaned and enriched data is forwarded to Power BI.
- Power BI dashboards receive updates automatically in real time.
- Visuals refresh dynamically as new telemetry arrives.
- Provides stakeholders with instant access to actionable weather insights.

End-to-End Summary

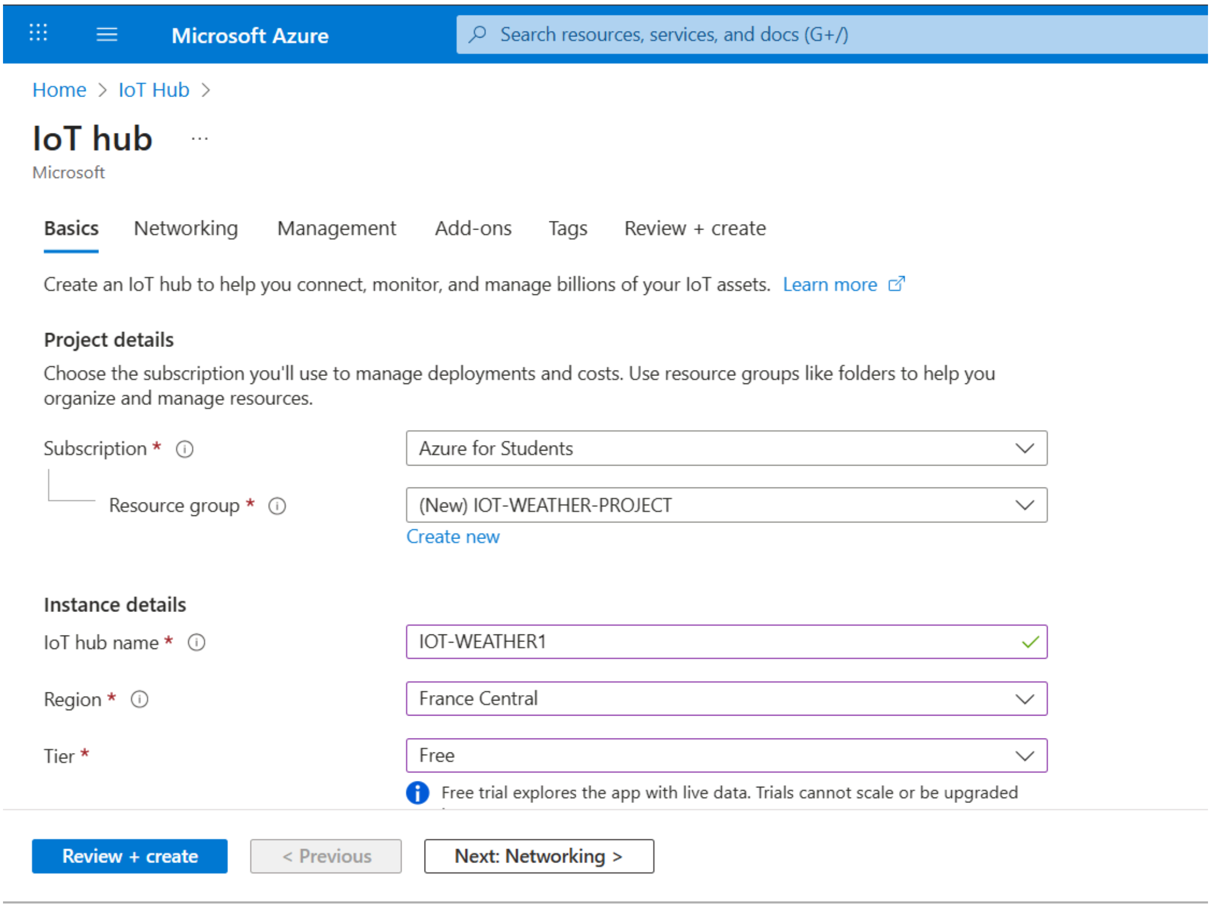
- Ensures seamless data flow from generation → preprocessing → analytics → visualization.

- Maintains security, efficiency, and real-time responsiveness throughout the architecture.

- Scales reliably across the entire IoT and cloud ecosystem.

V – Implementation :

Implementation: Set up an Azure resource group, created an IoT Hub, and configured a connected IoT device.



IoT hub ···
Microsoft

Create an IoT hub to help you connect, monitor, and manage billions of your IoT assets. Learn more

Project details

Choose the subscription you'll use to manage deployments and costs. Use resource groups like folders to help you organize and manage resources.

Subscription * : Azure for Students

Resource group * : (New) IOT-WEATHER-PROJECT

Create new

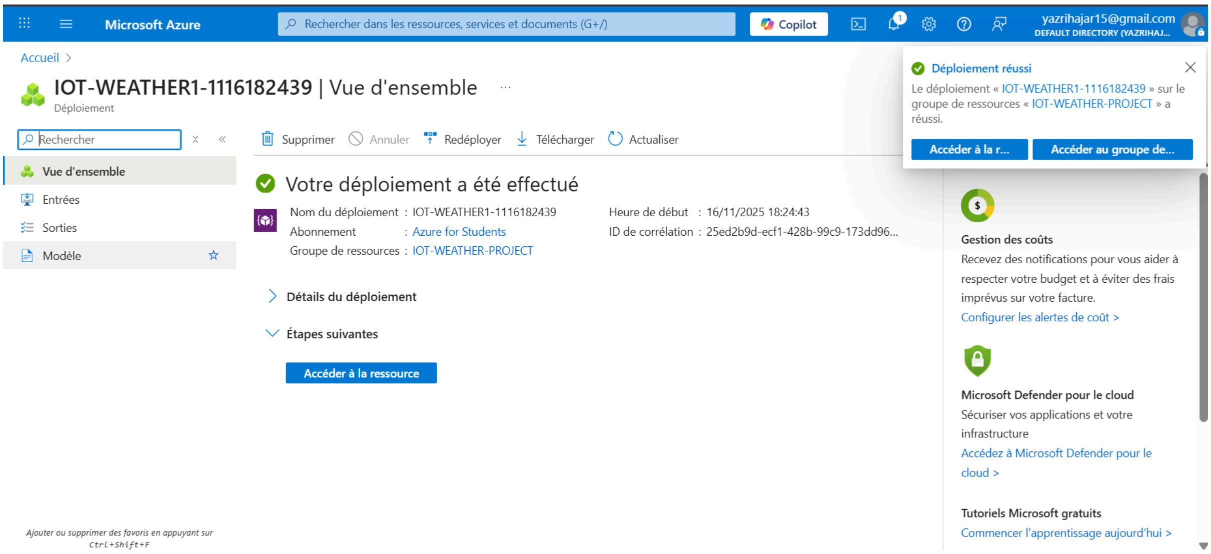
Instance details

IoT hub name * : IOT-WEATHER1 ✓

Region * : France Central

Tier * : Free

Free trial explores the app with live data. Trials cannot scale or be upgraded



Microsoft Azure — Rechercher dans les ressources, services et documents (G+/) — Copilot — yazrihajar15@gmail.com DEFAULT DIRECTORY (YAZRIHAJ...

Accueil >

IOT-WEATHER1-1116182439 | Vue d'ensemble ...
Déploiement

Rechercher
- Vue d'ensemble
- Entrées
- Sorties
- Modèle ☆

Supprimer | Annuler | Redéployer | Télécharger | Actualiser

Votre déploiement a été effectué

Nom du déploiement : IOT-WEATHER1-1116182439
Abonnement : Azure for Students
Groupe de ressources : IOT-WEATHER-PROJECT
Heure de début : 16/11/2025 18:24:43
ID de corrélation : 25ed2b9d-ecf1-428b-99c9-173dd96...

> Détails du déploiement

∨ Étapes suivantes

Accéder à la ressource

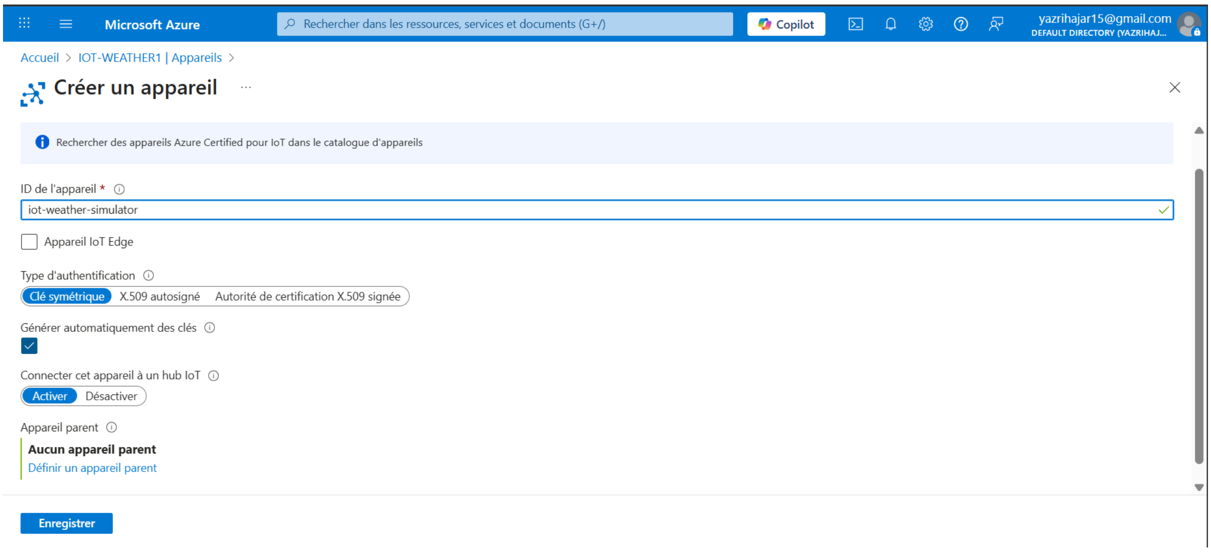
Déploiement réussi
Le déploiement « IOT-WEATHER1-1116182439 » sur le groupe de ressources « IOT-WEATHER-PROJECT » a réussi.
Accéder à la r... | Accéder au groupe de...

Gestion des coûts
Recevez des notifications pour vous aider à respecter votre budget et à éviter des frais imprévus sur votre facture.
Configurer les alertes de coût >

Microsoft Defender pour le cloud
Sécuriser vos applications et votre infrastructure
Accédez à Microsoft Defender pour le cloud >

Tutoriels Microsoft gratuits
Commencer l'apprentissage aujourd'hui >

Ajouter ou supprimer des favoris en appuyant sur Ctrl+Shift+F



Microsoft Azure — Rechercher dans les ressources, services et documents (G+/) — Copilot — yazrihajar15@gmail.com DEFAULT DIRECTORY (YAZRIHAJ...

Accueil > IOT-WEATHER1 | Appareils >

Créer un appareil ...

Rechercher des appareils Azure Certified pour IoT dans le catalogue d'appareils

ID de l'appareil *
iot-weather-simulator

□ Appareil IoT Edge

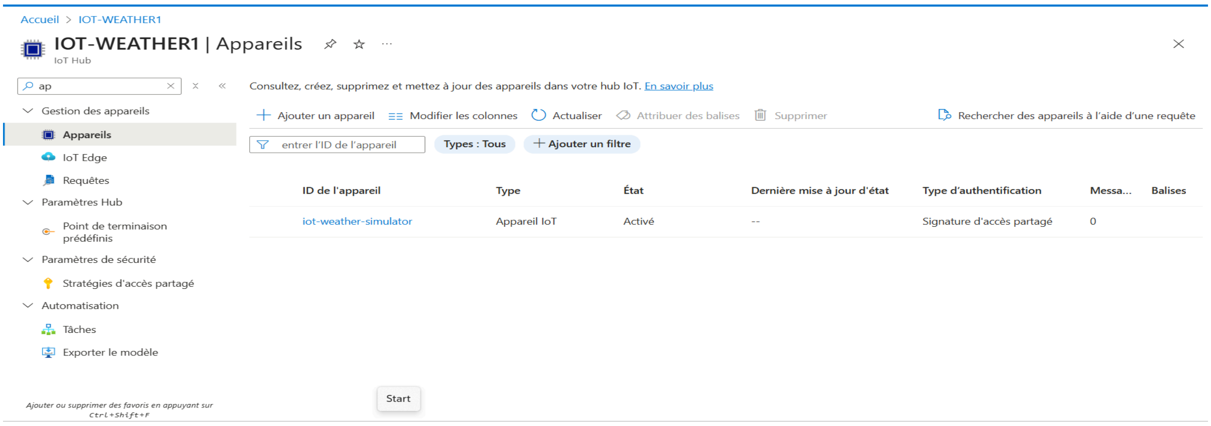
Type d'authentification
[✓ Clé symétrique] X.509 autosigné | Autorité de certification X.509 signée

Générer automatiquement des clés
☑

Connecter cet appareil à un hub IoT
[✓ Activer] Désactiver

Appareil parent
Aucun appareil parent
Définir un appareil parent

Enregistrer



Accueil > IOT-WEATHER1

IOT-WEATHER1 | Appareils
IoT Hub

ap

∨ Gestion des appareils
- Appareils
- IoT Edge
- Requêtes

∨ Paramètres Hub
- Point de terminaison prédéfinis

∨ Paramètres de sécurité
- Stratégies d'accès partagé

∨ Automatisation
- Tâches
- Exporter le modèle

Consultez, créez, supprimez et mettez à jour des appareils dans votre hub IoT. En savoir plus

+ Ajouter un appareil | Modifier les colonnes | Actualiser | Attribuer des balises | Supprimer | Rechercher des appareils à l'aide d'une requête

entrer l'ID de l'appareil | Types : Tous | + Ajouter un filtre

ID de l'appareil	Type	État	Dernière mise à jour d'état	Type d'authentification	Messa...	Balises
iot-weather-simulator	Appareil IoT	Activé	--	Signature d'accès partagé	0	

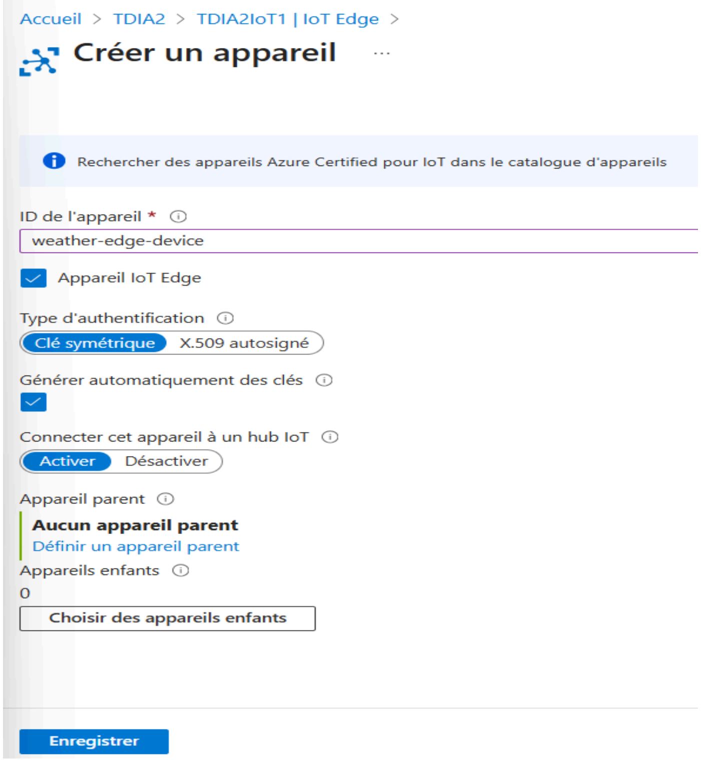
Ajouter ou supprimer des favoris en appuyant sur Ctrl+Shift+F

Start

Implementation: Deployed Azure IoT Edge on an Azure Virtual Machine (VM).

1. Creation of the Edge Infrastructure

IoT Edge device registered and linked to the Azure IoT Hub



Accueil > TDIA2 > TDIA2IoT1 | IoT Edge >

Créer un appareil ...

Rechercher des appareils Azure Certified pour IoT dans le catalogue d'appareils

ID de l'appareil *: weather-edge-device

☑ Appareil IoT Edge

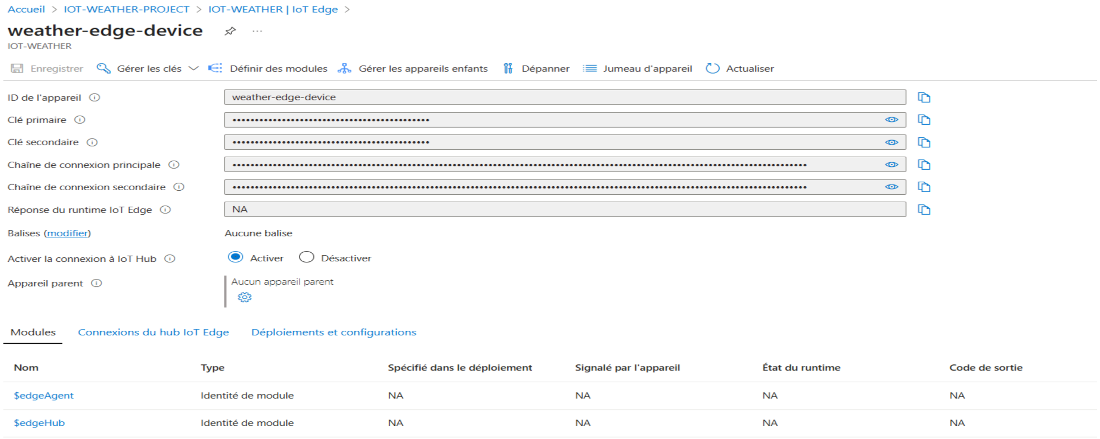
Type d'authentification: [✓ Clé symétrique] X.509 autosigné

Générer automatiquement des clés: ☑

Connecter cet appareil à un hub IoT: [✓ Activer] Désactiver

Appareil parent: **Aucun appareil parent** Définir un appareil parent

Appareils enfants: 0



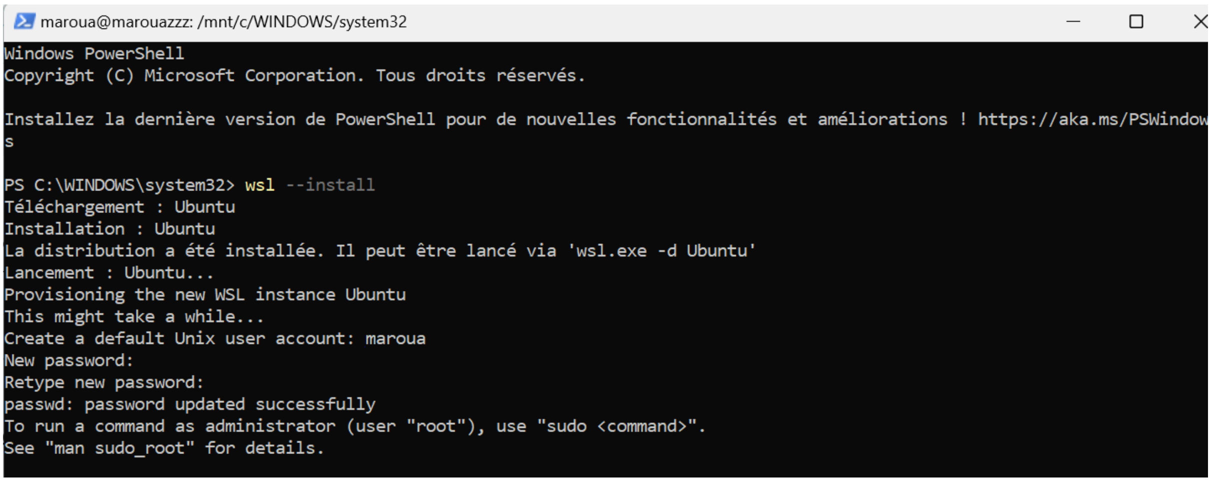
Accueil > IOT-WEATHER-PROJECT > IOT-WEATHER | IoT Edge >

weather-edge-device
IOT-WEATHER

ID de l'appareil	weather-edge-device
Clé primaire	••
Clé secondaire	••
Chaîne de connexion principale	••
Chaîne de connexion secondaire	••
Réponse du runtime IoT Edge	NA
Balises (modifier)	Aucune balise
Activer la connexion à IoT Hub	● Activer ○ Désactiver
Appareil parent	Aucun appareil parent

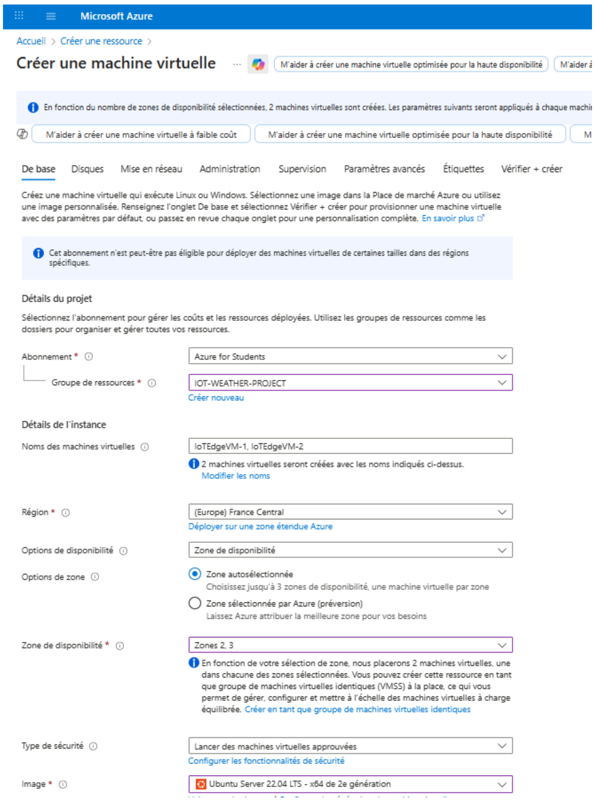
Nom	Type	Spécifié dans le déploiement	Signalé par l'appareil	État du runtime	Code de sortie
$edgeAgent	Identité de module	NA	NA	NA	NA
$edgeHub	Identité de module	NA	NA	NA	NA

2. Installation of the IoT Edge Runtime



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindow
s

PS C:\WINDOWS\system32> wsl --install
Téléchargement : Ubuntu
Installation : Ubuntu
La distribution a été installée. Il peut être lancé via 'wsl.exe -d Ubuntu'
Lancement : Ubuntu...
Provisioning the new WSL instance Ubuntu
This might take a while...
Create a default Unix user account: maroua
New password:
Retype new password:
passwd: password updated successfully
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

- **Azure Virtual Machine deployed and configured with Ubuntu 20.04 LTS.**
- **SSH key generated to ensure secure remote connection.**



Microsoft Azure
Accueil > Créer une ressource >

Créer une machine virtuelle

M'aider à créer une machine virtuelle optimisée pour la haute disponibilité
M'aider à

En fonction du nombre de zones de disponibilité sélectionnées, 2 machines virtuelles sont créées. Les paramètres suivants seront appliqués à chaque machi

M'aider à créer une machine virtuelle à faible coût
M'aider à créer une machine virtuelle optimisée pour la haute disponibilité
M
De base
Disques
Mise en réseau
Administration
Supervision
Paramètres avancés
Étiquettes
Vérifier + créer

Créez une machine virtuelle qui exécute Linux ou Windows. Sélectionnez une image dans la Place de marché Azure ou utilisez une image personnalisée. Renseignez l'onglet De base et sélectionnez Vérifier + créer pour provisionner une machine virtuelle avec des paramètres par défaut, ou passez en revue chaque onglet pour une personnalisation complète. En savoir plus

Cet abonnement n'est peut-être pas éligible pour déployer des machines virtuelles de certaines tailles dans des régions spécifiques.

Détails du projet

Sélectionnez l'abonnement pour gérer les coûts et les ressources déployées. Utilisez les groupes de ressources comme les dossiers pour organiser et gérer toutes vos ressources.

Abonnement * : Azure for Students

Groupe de ressources * : IOT-WEATHER-PROJECT

Créer nouveau

Détails de l'instance

Noms des machines virtuelles : IoTEdgeVM-1, IoTEdgeVM-2

2 machines virtuelles seront créées avec les noms indiqués ci-dessus. Modifier les noms

Région * : (Europe) France Central

Déployer sur une zone étendue Azure

Options de disponibilité : Zone de disponibilité

Options de zone :

● Zone autosélectionnée
Choisissez jusqu'à 3 zones de disponibilité, une machine virtuelle par zone

○ Zone sélectionnée par Azure (préversion)
Laissez Azure attribuer la meilleure zone pour vos besoins

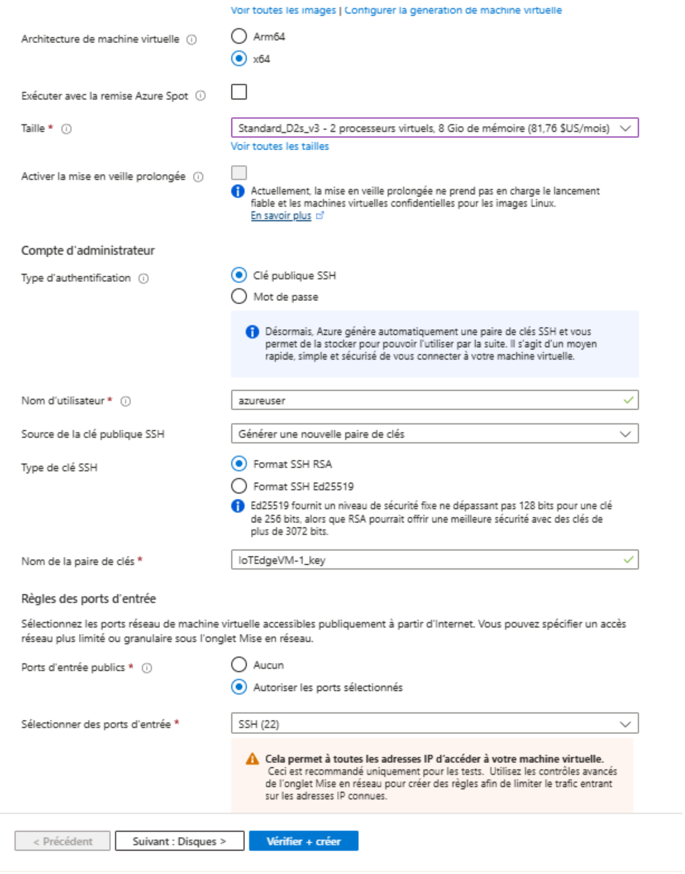
Zone de disponibilité * : Zones 2, 3

En fonction de votre sélection de zone, nous placerons 2 machines virtuelles, une dans chacune des zones sélectionnées. Vous pouvez créer cette ressource en tant que groupe de machines virtuelles identiques (VMSS) à la place, ce qui vous permet de gérer, configurer et mettre à l'échelle des machines virtuelles à charge équilibrée. Créer en tant que groupe de machines virtuelles identiques

Type de sécurité : Lancer des machines virtuelles approuvées

Configurer les fonctionnalités de sécurité

Image * : Ubuntu Server 22.04 LTS - x64 de 2e génération



Voir toutes les images | Configurer la generation de machine virtuelle

Architecture de machine virtuelle ○ Arm64 ● x64

Exécuter avec la remise Azure Spot □

Taille * Standard_D2s_v3 - 2 processeurs virtuels, 8 Gio de mémoire (81,76 $US/mois)

Voir toutes les tailles

Activer la mise en veille prolongée □

Actuellement, la mise en veille prolongée ne prend pas en charge le lancement fiable et les machines virtuelles confidentielles pour les images Linux. En savoir plus

Compte d'administrateur

Type d'authentification ● Clé publique SSH ○ Mot de passe

Désormais, Azure génère automatiquement une paire de clés SSH et vous permet de la stocker pour pouvoir l'utiliser par la suite. Il s'agit d'un moyen rapide, simple et sécurisé de vous connecter à votre machine virtuelle.

Nom d'utilisateur * azureuser

Source de la clé publique SSH Générer une nouvelle paire de clés

Type de clé SSH ● Format SSH RSA ○ Format SSH Ed25519

Ed25519 fournit un niveau de sécurité fixe ne dépassant pas 128 bits pour une clé de 256 bits, alors que RSA pourrait offrir une meilleure sécurité avec des clés de plus de 3072 bits.

Nom de la paire de clés * IoTEdgeVM-1_key

Règles des ports d'entrée

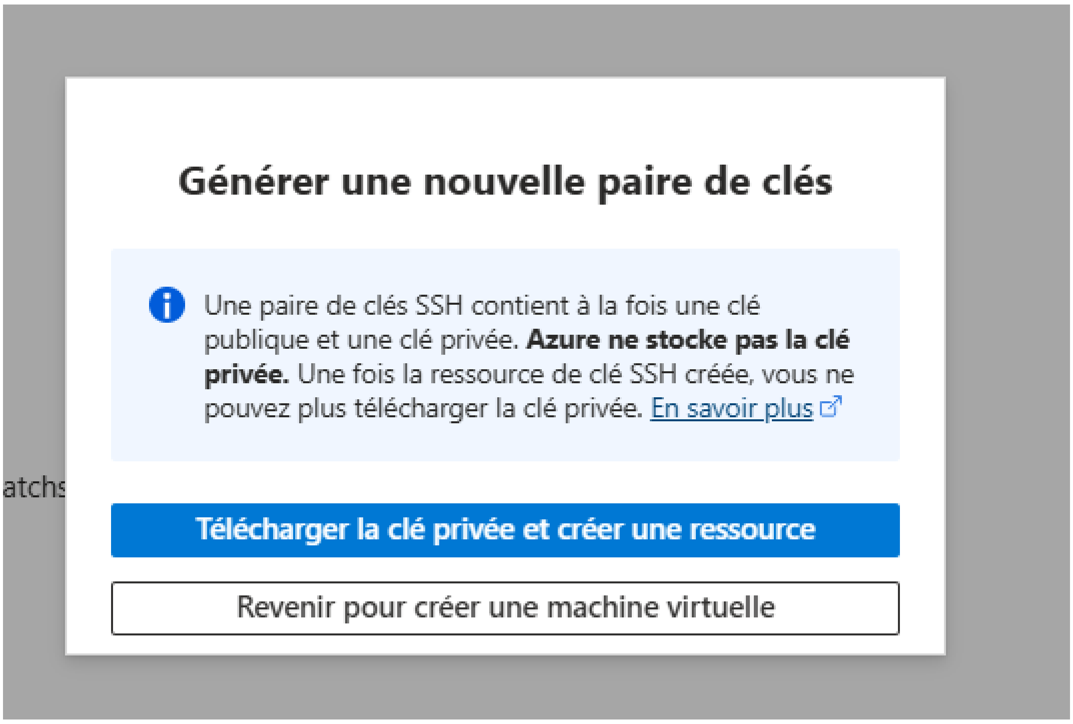
Sélectionnez les ports réseau de machine virtuelle accessibles publiquement à partir d'Internet. Vous pouvez spécifier un accès réseau plus limité ou granulaire sous l'onglet Mise en réseau.

Ports d'entrée publics * ○ Aucun ● Autoriser les ports sélectionnés

Sélectionner des ports d'entrée * SSH (22)

Cela permet à toutes les adresses IP d'accéder à votre machine virtuelle. Ceci est recommandé uniquement pour les tests. Utilisez les contrôles avancés de l'onglet Mise en réseau pour créer des règles afin de limiter le trafic entrant sur les adresses IP connues.

< Précédent
Suivant : Disques >
Vérifier + créer

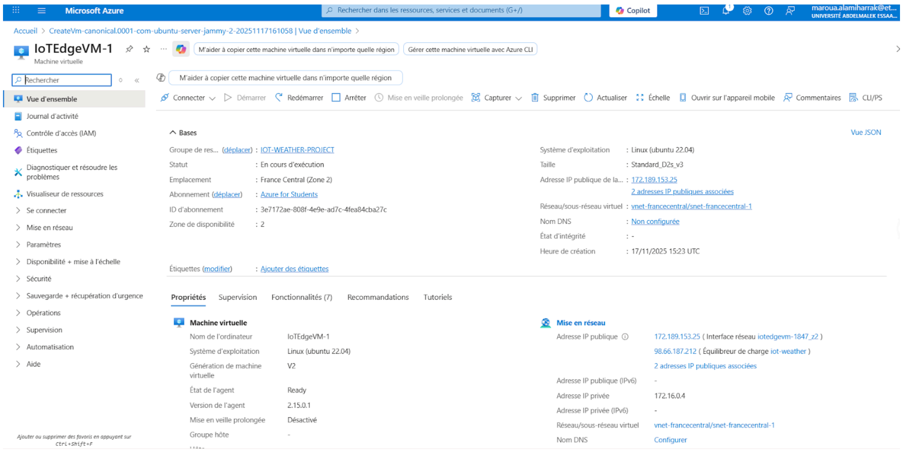


Générer une nouvelle paire de clés

Une paire de clés SSH contient à la fois une clé publique et une clé privée. **Azure ne stocke pas la clé privée.** Une fois la ressource de clé SSH créée, vous ne pouvez plus télécharger la clé privée. En savoir plus

atchs
Télécharger la clé privée et créer une ressource
Revenir pour créer une machine virtuelle

Once the VM is deployed, it must be accessed from our local machine (PC).


Microsoft Azure
Rechercher dans les ressources, services et documents (G+/)
Copilot
maroua.alamiharrak@et...
UNIVERSITÉ ABDELMALEK ESSAA...
Accueil > CreateVm-canonical.0001-com-ubuntu-server-jammy-2-20251117161058 | Vue d'ensemble >
IoTEdgeVM-1
Machine virtuelle
M'aider à copier cette machine virtuelle dans n'importe quelle région
Gérer cette machine virtuelle avec Azure CLI
Rechercher
Vue d'ensemble
Journal d'activité
Contrôle d'accès (IAM)
Étiquettes
Diagnostiquer et résoudre les problèmes
Visualiseur de ressources
Se connecter
Mise en réseau
Paramètres
Disponibilité + mise à l'échelle
Sécurité
Sauvegarde + récupération d'urgence
Opérations
Supervision
Automatisation
Aide
Ajouter ou supprimer des favoris en appuyant sur Ctrl+Shift+F
Connecter
Démarrer
Redémarrer
Arrêter
Mise en veille prolongée
Capturer
Supprimer
Actualiser
Échelle
Ouvrir sur l'appareil mobile
Commentaires
CLI/PS
Vue JSON
Propriétés
Supervision
Fonctionnalités (7)
Recommandations
Tutoriels

Bases

Groupe de res... (déplacer) : IOT-WEATHER-PROJECT
Statut : En cours d'exécution
Emplacement : France Central (Zone 2)
Abonnement (déplacer) : Azure for Students
ID d'abonnement : 3e7172ae-808f-4e9e-ad7c-4fea84cba27c
Zone de disponibilité : 2
Étiquettes (modifier) : Ajouter des étiquettes

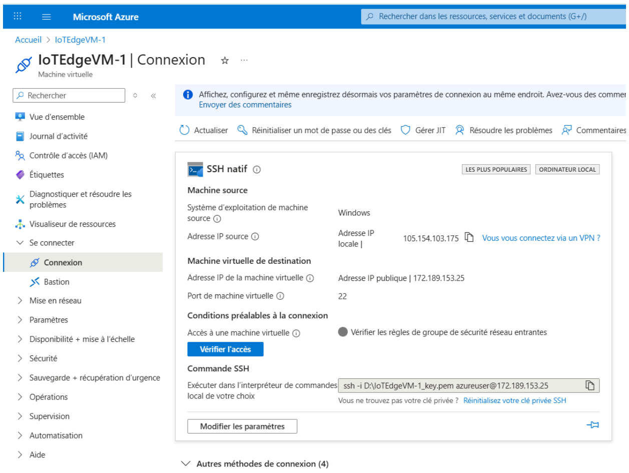
Système d'exploitation : Linux (ubuntu 22.04)
Taille : Standard_D2s_v3
Adresse IP publique de la... : 172.189.153.25
2 adresses IP publiques associées
Réseau/sous-réseau virtuel : vnet-francecentral/snet-francecentral-1
Nom DNS : Non configurée
État d'intégrité : -
Heure de création : 17/11/2025 15:23 UTC

Machine virtuelle

Nom de l'ordinateur IoTEdgeVM-1
Système d'exploitation Linux (ubuntu 22.04)
Génération de machine virtuelle V2
État de l'agent Ready
Version de l'agent 2.15.0.1
Mise en veille prolongée Désactivé
Groupe hôte -

Mise en réseau

Adresse IP publique 172.189.153.25 (Interface réseau iotedgevm-1847_z2)
98.66.187.212 (Équilibreur de charge iot-weather)
2 adresses IP publiques associées
Adresse IP publique (IPv6) -
Adresse IP privée 172.16.0.4
Adresse IP privée (IPv6) -
Réseau/sous-réseau virtuel vnet-francecentral/snet-francecentral-1
Nom DNS Configurer



Affichez, configurez et même enregistrez désormais vos paramètres de connexion au même endroit. Avez-vous des commer
Envoyer des commentaires

SSH natif

Machine source

Système d'exploitation de machine source Windows
Adresse IP source Adresse IP locale | 105.154.103.175 Vous vous connectez via un VPN ?

Machine virtuelle de destination

Adresse IP de la machine virtuelle Adresse IP publique | 172.189.153.25
Port de machine virtuelle 22

Conditions préalables à la connexion

Accès à une machine virtuelle ● Vérifier les règles de groupe de sécurité réseau entrantes

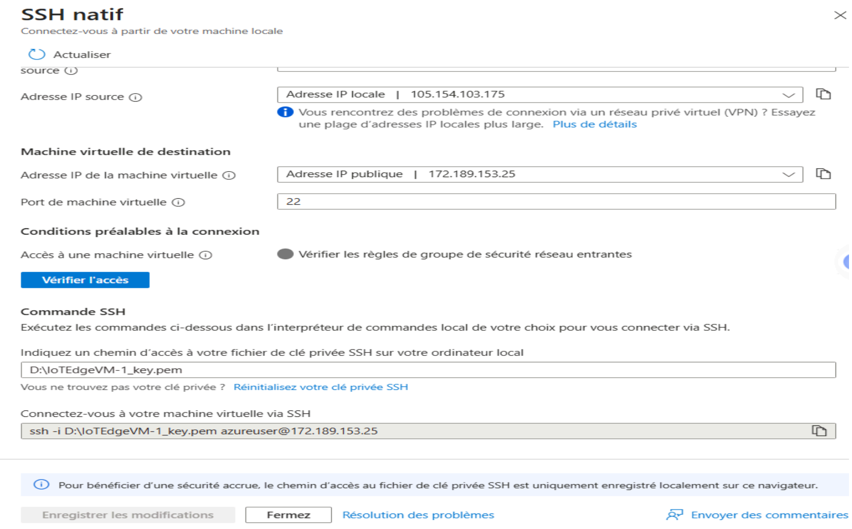
Commande SSH

Exécuter dans l'interpréteur de commandes local de votre choix

```
ssh -i D:\IoTEdgeVM-1_key.pem azureuser@172.189.153.25
```

Vous ne trouvez pas votre clé privée ? Réinitialisez votre clé privée SSH

We use the SSH address on our local machine to establish the connection.



SSH natif

Connectez-vous à partir de votre machine locale

source

Adresse IP source Adresse IP locale | 105.154.103.175

Vous rencontrez des problèmes de connexion via un réseau privé virtuel (VPN) ? Essayez une plage d'adresses IP locales plus large. Plus de détails

Machine virtuelle de destination

Adresse IP de la machine virtuelle Adresse IP publique | 172.189.153.25
Port de machine virtuelle 22

Conditions préalables à la connexion

Accès à une machine virtuelle ● Vérifier les règles de groupe de sécurité réseau entrantes

Commande SSH

Exécutez les commandes ci-dessous dans l'interpréteur de commandes local de votre choix pour vous connecter via SSH.

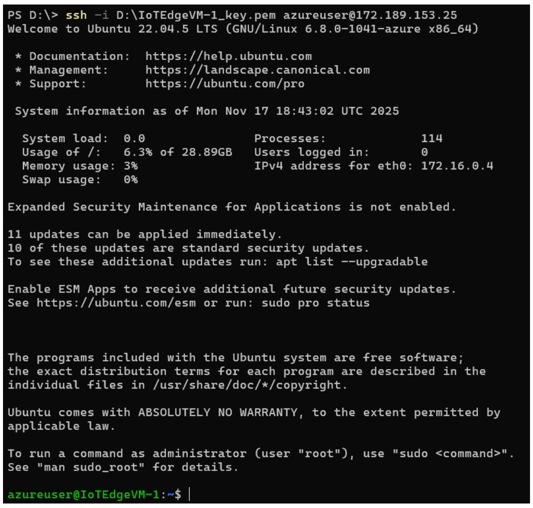
Indiquez un chemin d'accès à votre fichier de clé privée SSH sur votre ordinateur local
D:\IoTEdgeVM-1_key.pem
Vous ne trouvez pas votre clé privée ? Réinitialisez votre clé privée SSH

Connectez-vous à votre machine virtuelle via SSH

```
ssh -i D:\IoTEdgeVM-1_key.pem azureuser@172.189.153.25
```

Pour bénéficier d'une sécurité accrue, le chemin d'accès au fichier de clé privée SSH est uniquement enregistré localement sur ce navigateur.

Establishing a connection to the VM

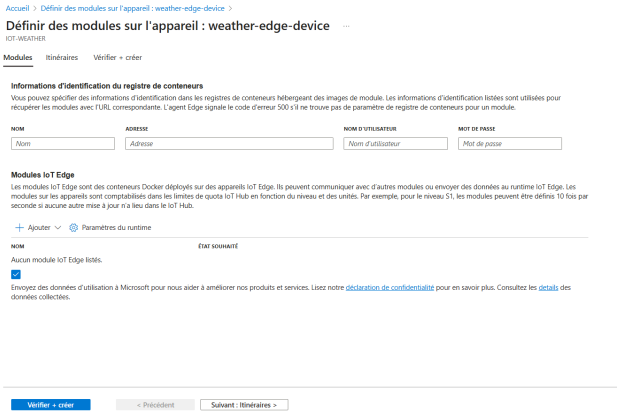


```
PS D:\> ssh -i D:\IoTEdgeVM-1_key.pem azureuser@172.189.153.25
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1041-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Mon Nov 17 18:43:02 UTC 2025

  System load:  0.0               Processes:             114
  Usage of /:   6.3% of 28.89GB   Users logged in:       0
  Memory usage: 3%                IPv4 address for eth0: 172.16.0.4
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

11 updates can be applied immediately.
10 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

azureuser@IoTEdgeVM-1:~$ 
```

Installation of IoT Edge on the VM:

After deploying the VM, we installed IoT Edge by adding Microsoft repositories and keys, then updating the package list. This allowed us to build a Docker container for our module and push it to the **Azure Container Registry** for deployment.

❖ Creation of the Edge Device Module in Azure



Définir des modules sur l'appareil : weather-edge-device

IOT-WEATHER

Informations d'identification du registre de conteneurs

Vous pouvez spécifier des informations d'identification dans les registres de conteneurs hébergeant des images de module. Les informations d'identification listées sont utilisées pour récupérer les modules avec l'URL correspondante. L'agent Edge signale le code d'erreur 500 s'il ne trouve pas de paramètre de registre de conteneurs pour un module.

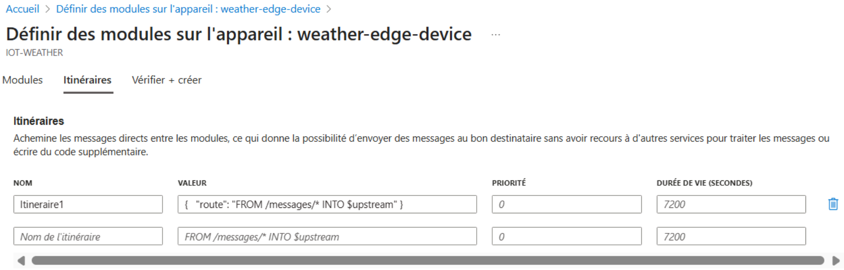
NOM	ADRESSE	NOM D'UTILISATEUR	MOT DE PASSE
Nom	Adresse	Nom d'utilisateur	Mot de passe

Modules IoT Edge

Les modules IoT Edge sont des conteneurs Docker déployés sur des appareils IoT Edge. Ils peuvent communiquer avec d'autres modules ou envoyer des données au runtime IoT Edge. Les modules sur les appareils sont comptabilisés dans les limites de quota IoT Hub en fonction du niveau et des unités. Par exemple, pour le niveau S1, les modules peuvent être définis 10 fois par seconde si aucune autre mise à jour n'a lieu dans le IoT Hub.

NOM	ÉTAT SOUHAITÉ
Aucun module IoT Edge listés.	

☑ Envoyez des données d'utilisation à Microsoft pour nous aider à améliorer nos produits et services. Lisez notre déclaration de confidentialité pour en savoir plus. Consultez les details des données collectées.



Accueil > Définir des modules sur l'appareil : weather-edge-device >
Modules
Itinéraires
Vérifier + créer

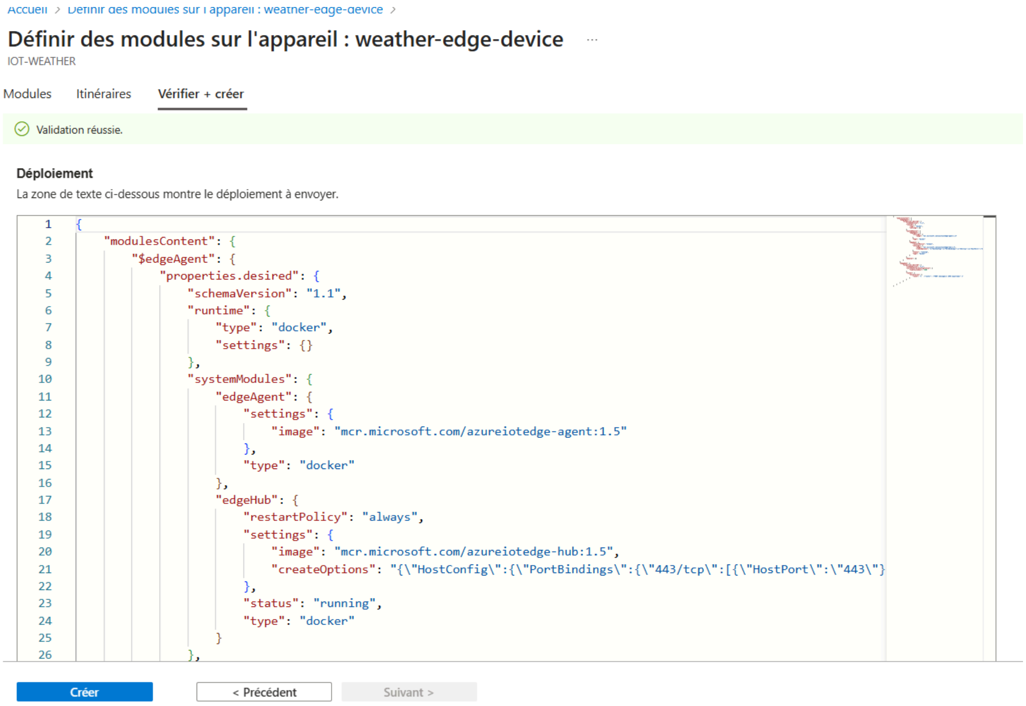
Définir des modules sur l'appareil : weather-edge-device

IOT-WEATHER

Itinéraires

Achemine les messages directs entre les modules, ce qui donne la possibilité d'envoyer des messages au bon destinataire sans avoir recours à d'autres services pour traiter les messages ou écrire du code supplémentaire.

NOM	VALEUR	PRIORITÉ	DURÉE DE VIE (SECONDES)
Itineraire1	{ "route": "FROM /messages/* INTO $upstream" }	0	7200
Nom de l'itinéraire	FROM /messages/* INTO $upstream	0	7200



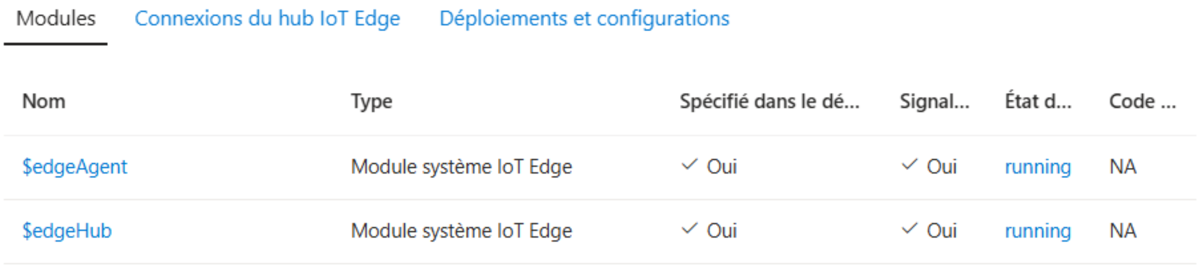
Accueil > Définir des modules sur l'appareil : weather-edge-device >
Définir des modules sur l'appareil : weather-edge-device ...
IOT-WEATHER
Modules
Itinéraires
Vérifier + créer
Validation réussie.

Déploiement

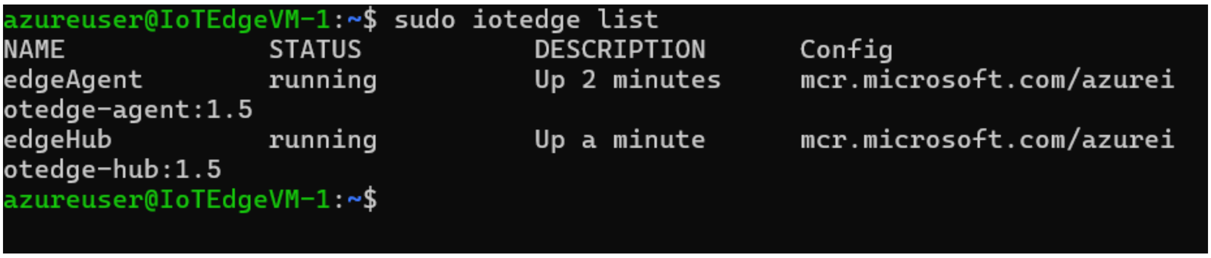
La zone de texte ci-dessous montre le déploiement à envoyer.

```
{
    "modulesContent": {
        "$edgeAgent": {
            "properties.desired": {
                "schemaVersion": "1.1",
                "runtime": {
                    "type": "docker",
                    "settings": {}
                },
                "systemModules": {
                    "edgeAgent": {
                        "settings": {
                            "image": "mcr.microsoft.com/azureiotedge-agent:1.5"
                        },
                        "type": "docker"
                    },
                    "edgeHub": {
                        "restartPolicy": "always",
                        "settings": {
                            "image": "mcr.microsoft.com/azureiotedge-hub:1.5",
                            "createOptions": "{\"HostConfig\":{\"PortBindings\":{\"443/tcp\":[{\"HostPort\":\"443\"}
                        },
                        "status": "running",
                        "type": "docker"
                    }
                },
```

After creating routes and deploying, the Edge Agent and Edge Hub are finally running! They can be monitored either through Azure or directly on the VM.

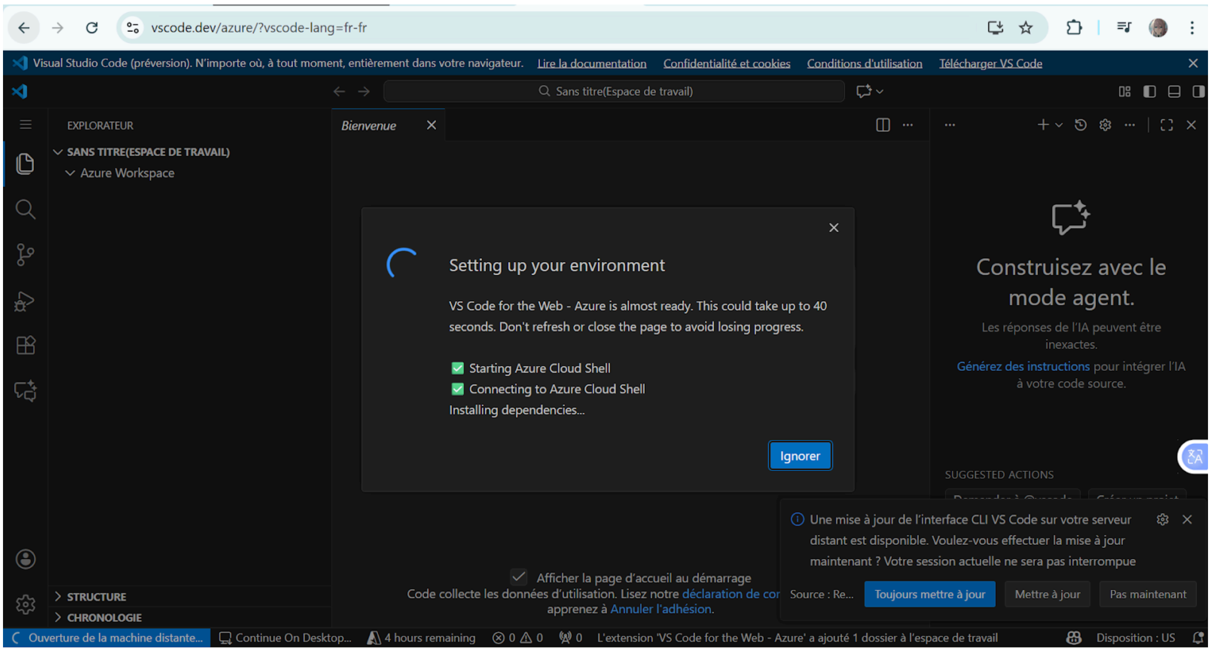


Nom	Type	Spécifié dans le dé...	Signal...	État d...	Code ...
$edgeAgent	Module système IoT Edge	✓ Oui	✓ Oui	running	NA
$edgeHub	Module système IoT Edge	✓ Oui	✓ Oui	running	NA



```
azureuser@IoTEdgeVM-1:~$ sudo iotedge list
NAME             STATUS           DESCRIPTION      Config
edgeAgent        running          Up 2 minutes     mcr.microsoft.com/azurei
otedge-agent:1.5
edgeHub          running          Up a minute      mcr.microsoft.com/azurei
otedge-hub:1.5
azureuser@IoTEdgeVM-1:~$
```

❖ Creation of the Python IoT Edge Module: weatherModule



Visual Studio Code (préversion). N'importe où, à tout moment, entièrement dans votre navigateur.

Setting up your environment

VS Code for the Web - Azure is almost ready. This could take up to 40 seconds. Don't refresh or close the page to avoid losing progress.

☑ Starting Azure Cloud Shell
☑ Connecting to Azure Cloud Shell
Installing dependencies...

Construisez avec le mode agent.

Les réponses de l'IA peuvent être inexactes.

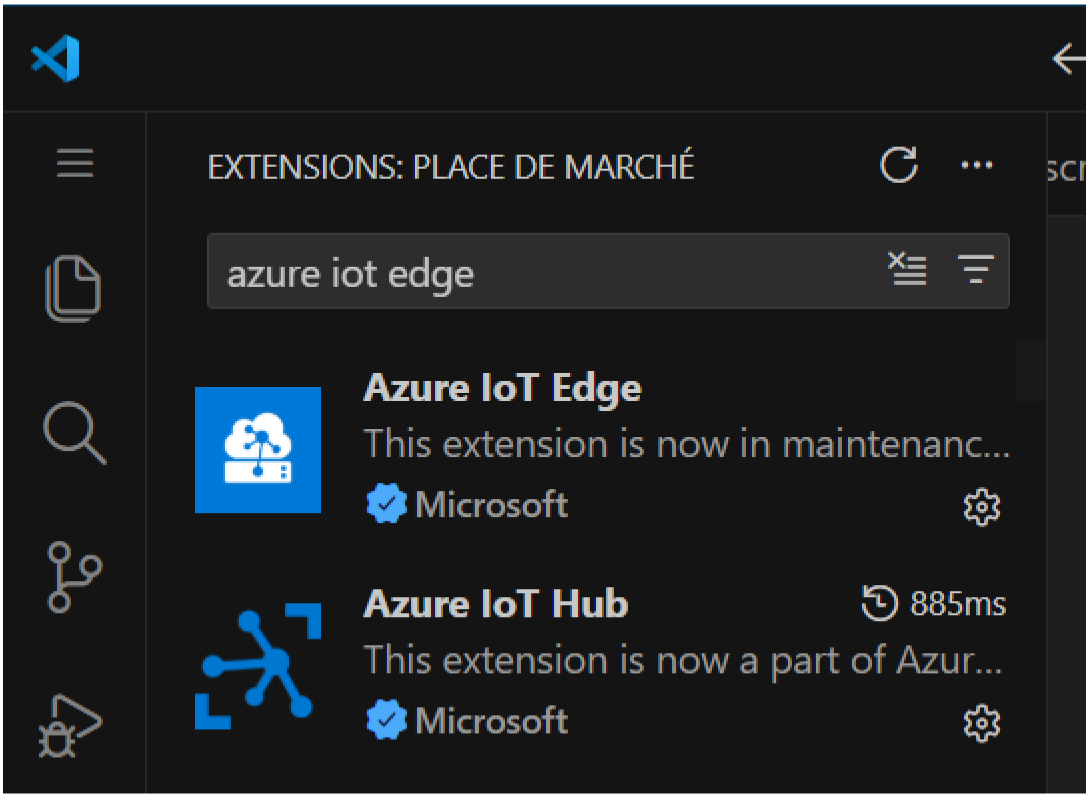
Générez des instructions pour intégrer l'IA à votre code source.

Une mise à jour de l'interface CLI VS Code sur votre serveur distant est disponible. Voulez-vous effectuer la mise à jour maintenant ? Votre session actuelle ne sera pas interrompue

✓ Afficher la page d'accueil au démarrage

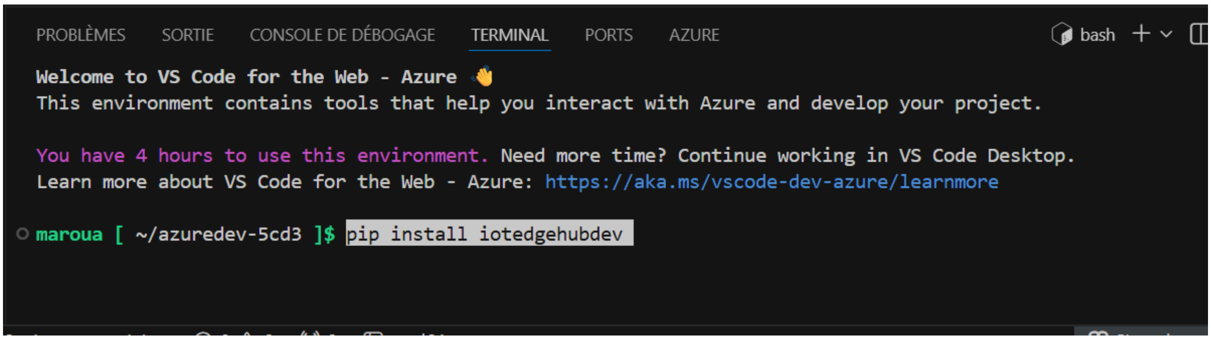
Code collecte les données d'utilisation. Lisez notre déclaration de cor apprenez à Annuler l'adhésion.

L'extension 'VS Code for the Web - Azure' a ajouté 1 dossier à l'espace de travail



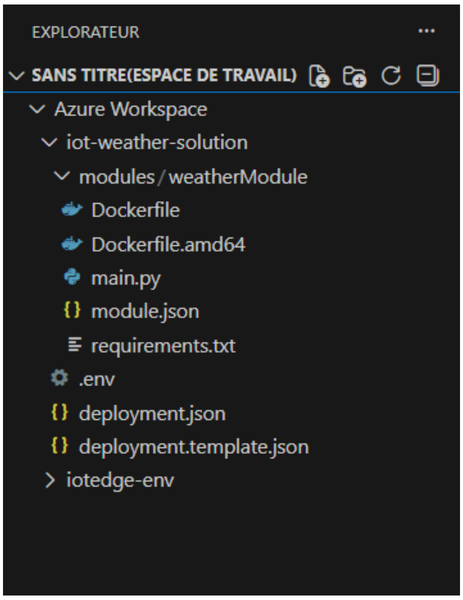
This extension is now in maintenanc...

This extension is now a part of Azur...



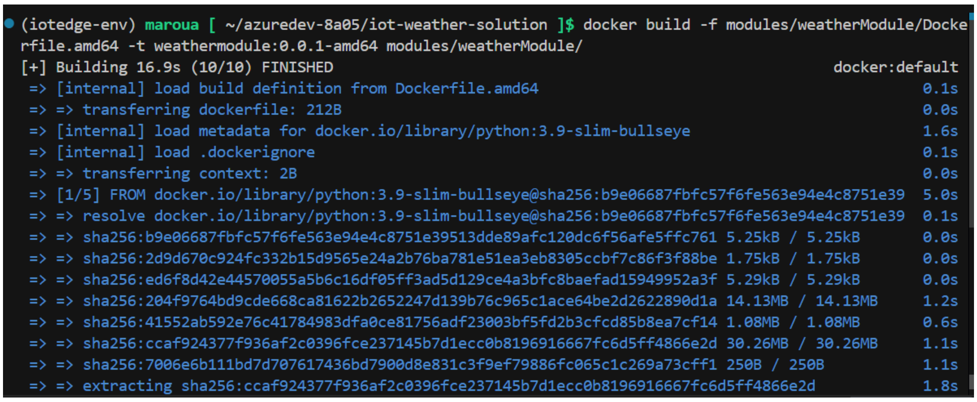
```
Welcome to VS Code for the Web - Azure 👋
This environment contains tools that help you interact with Azure and develop your project.

You have 4 hours to use this environment. Need more time? Continue working in VS Code Desktop.
Learn more about VS Code for the Web - Azure: https://aka.ms/vscode-dev-azure/learnmore

○ maroua [ ~/azuredev-5cd3 ]$ pip install iotedgehubdev
```

Based on these extensions the following results were obtained:



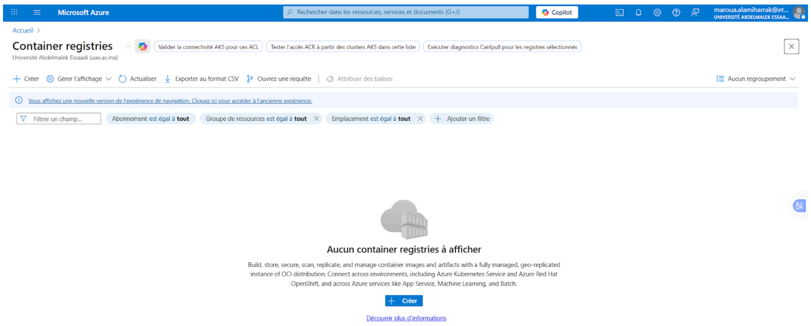
```
∨ Azure Workspace
  ∨ iot-weather-solution
    ∨ modules / weatherModule
        Dockerfile
        Dockerfile.amd64
        main.py
        module.json
        requirements.txt
      .env
      deployment.json
      deployment.template.json
  > iotedge-env
```

❖ Build the Docker Module Image



```
● (iotedge-env) maroua [ ~/azuredev-8a05/iot-weather-solution ]$ docker build -f modules/weatherModule/Docke
  rfile.amd64 -t weathermodule:0.0.1-amd64 modules/weatherModule/
  [+] Building 16.9s (10/10) FINISHED                                                          docker:default
   => [internal] load build definition from Dockerfile.amd64                                             0.1s
   => => transferring dockerfile: 212B                                                                   0.0s
   => [internal] load metadata for docker.io/library/python:3.9-slim-bullseye                           1.6s
   => [internal] load .dockerignore                                                                      0.1s
   => => transferring context: 2B                                                                        0.0s
   => [1/5] FROM docker.io/library/python:3.9-slim-bullseye@sha256:b9e06687fbfc57f6fe563e94e4c8751e39  5.0s
   => => resolve docker.io/library/python:3.9-slim-bullseye@sha256:b9e06687fbfc57f6fe563e94e4c8751e39  0.1s
   => => sha256:b9e06687fbfc57f6fe563e94e4c8751e39513dde89afc120dc6f56afe5ffc761 5.25kB / 5.25kB       0.0s
   => => sha256:2d9d670c924fc332b15d9565e24a2b76ba781e51ea3eb8305ccbf7c86f3f88be 1.75kB / 1.75kB       0.0s
   => => sha256:ed6f8d42e44570055a5b6c16df05ff3ad5d129ce4a3bfc8baefad15949952a3f 5.29kB / 5.29kB       0.0s
   => => sha256:204f9764bd9cde668ca81622b2652247d139b76c965c1ace64be2d2622890d1a 14.13MB / 14.13MB     1.2s
   => => sha256:41552ab592e76c41784983dfa0ce81756adf23003bf5fd2b3cfcd85b8ea7cf14 1.08MB / 1.08MB       0.6s
   => => sha256:ccaf924377f936af2c0396fce237145b7d1ecc0b8196916667fc6d5ff4866e2d 30.26MB / 30.26MB     1.1s
   => => sha256:7006e6b111bd7d707617436bd7900d8e831c3f9ef79886fc065c1c269a73cff1 250B / 250B           1.1s
   => => extracting sha256:ccaf924377f936af2c0396fce237145b7d1ecc0b8196916667fc6d5ff4866e2d             1.8s
```

❖ Azure Container Registry Configuration


Microsoft Azure
Rechercher dans les ressources, services et documents (G+/)
Copilot
maroua.alamiharrak@et...
UNIVERSITÉ ABDELMALEK ESSAA...
Accueil

Container registries

Université Abdelmalek Essaadi (uae.ac.ma)

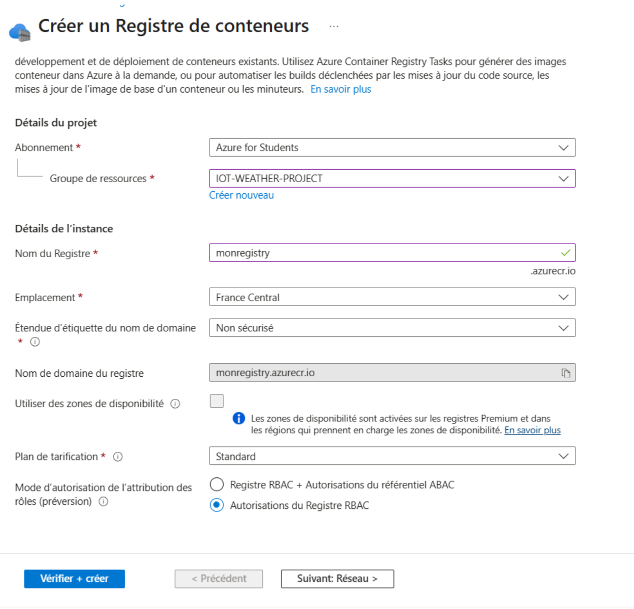
Valider la connectivité AKS pour ces ACL
Tester l'accès ACR à partir des clusters AKS dans cette liste
Exécuter diagnostics Canlpull pour les registres sélectionnés
Créer
Gérer l'affichage
Actualiser
Exporter au format CSV
Ouvrez une requête
Attribuer des balises
Aucun regroupement

Vous affichez une nouvelle version de l'expérience de navigation. Cliquez ici pour accéder à l'ancienne expérience.

Filtrer un champ...
Abonnement est égal à **tout**
Groupe de ressources est égal à **tout**
Emplacement est égal à **tout**
Ajouter un filtre

Aucun container registries à afficher

Build, store, secure, scan, replicate, and manage container images and artifacts with a fully managed, geo-replicated instance of OCI distribution. Connect across environments, including Azure Kubernetes Service and Azure Red Hat OpenShift, and across Azure services like App Service, Machine Learning, and Batch.



Créer un Registre de conteneurs

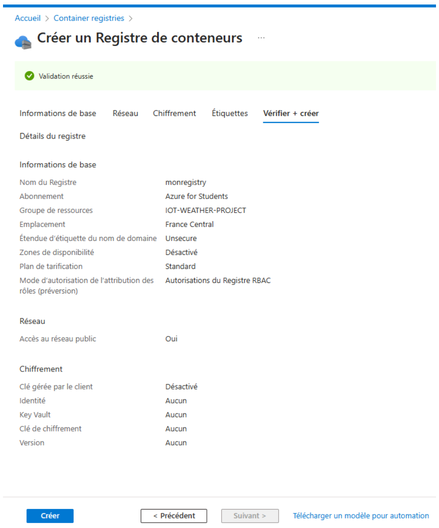
développement et de déploiement de conteneurs existants. Utilisez Azure Container Registry Tasks pour générer des images conteneur dans Azure à la demande, ou pour automatiser les builds déclenchées par les mises à jour du code source, les mises à jour de l'image de base d'un conteneur ou les minuteurs. En savoir plus

Détails du projet

Abonnement *	Azure for Students
Groupe de ressources *	IOT-WEATHER-PROJECT Créer nouveau

Détails de l'instance

Nom du Registre *	monregistry ✓ .azurecr.io
Emplacement *	France Central
Étendue d'étiquette du nom de domaine *	Non sécurisé
Nom de domaine du registre	monregistry.azurecr.io
Utiliser des zones de disponibilité	□ Les zones de disponibilité sont activées sur les registres Premium et dans les régions qui prennent en charge les zones de disponibilité. En savoir plus
Plan de tarification *	Standard
Mode d'autorisation de l'attribution des rôles (préversion)	○ Registre RBAC + Autorisations du référentiel ABAC ● Autorisations du Registre RBAC



Créer un Registre de conteneurs

Validation réussie

Détails du registre

Informations de base

Nom du Registre	monregistry
Abonnement	Azure for Students
Groupe de ressources	IOT-WEATHER-PROJECT
Emplacement	France Central
Étendue d'étiquette du nom de domaine	Unsecure
Zones de disponibilité	Désactivé
Plan de tarification	Standard
Mode d'autorisation de l'attribution des rôles (préversion)	Autorisations du Registre RBAC

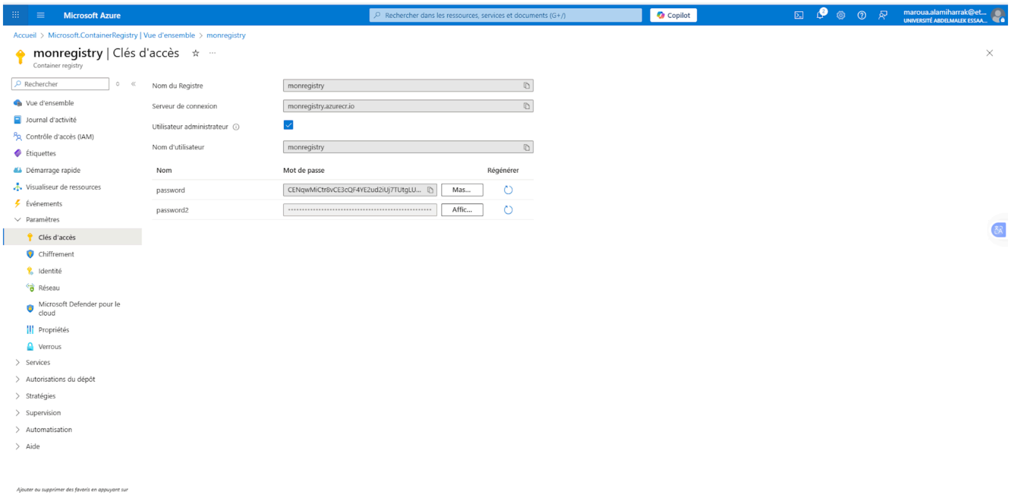
Réseau

Accès au réseau public	Oui

Chiffrement

Clé gérée par le client	Désactivé
Identité	Aucun
Key Vault	Aucun
Clé de chiffrement	Aucun
Version	Aucun

Grant access and retrieve the password, as we will need it later.

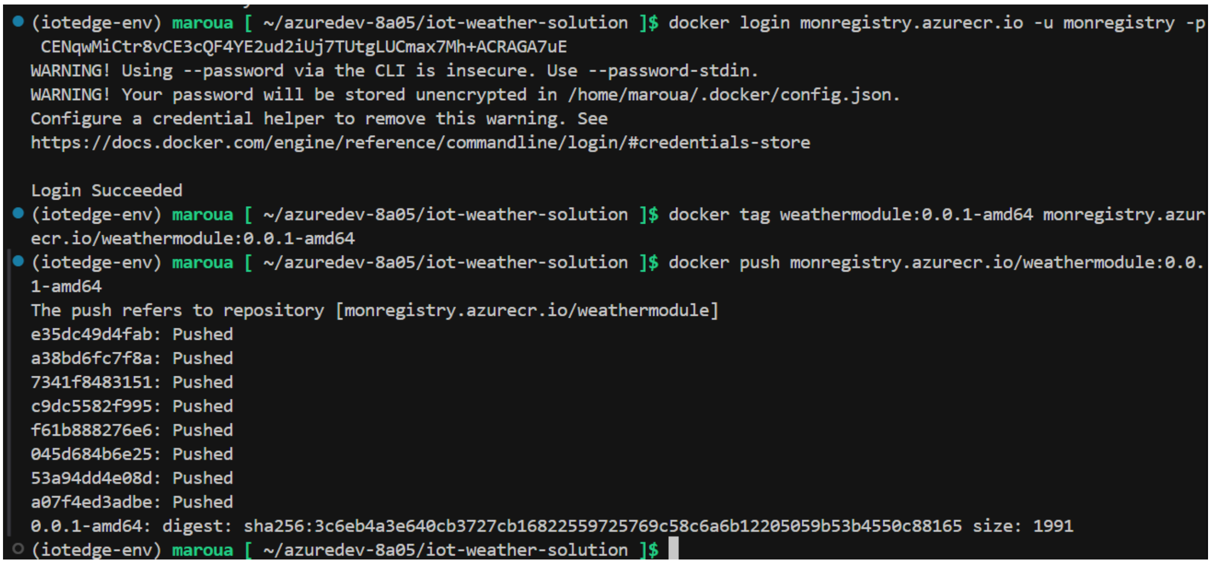


monregistry | Clés d'accès
Container registry

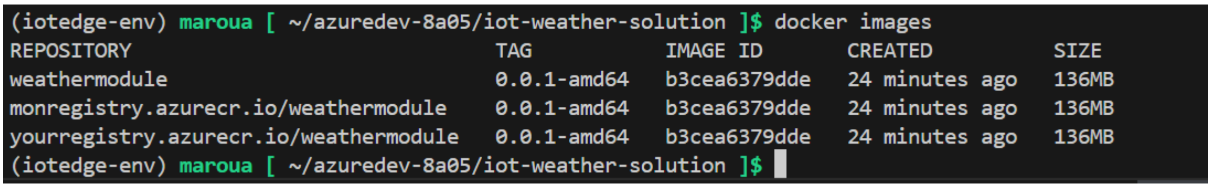
Nom du Registre: monregistry
Serveur de connexion: monregistry.azurecr.io
Utilisateur administrateur: ☑
Nom d'utilisateur: monregistry

Nom	Mot de passe		Régénérer
password	CENqwMiCtr8vCE3cQF4YE2ud2iUj7TUtgLU...	Mas...	↻
password2	··	Affic...	↻

❖ Push the Image to Azure Container Registry

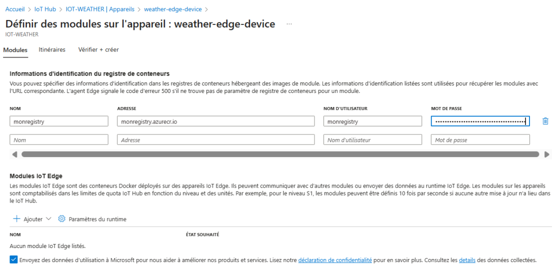


```
● (iotedge-env) maroua [ ~/azuredev-8a05/iot-weather-solution ]$ docker login monregistry.azurecr.io -u monregistry -p
   CENqwMiCtr8vCE3cQF4YE2ud2iUj7TUtgLUCmax7Mh+ACRAGA7uE
  WARNING! Using --password via the CLI is insecure. Use --password-stdin.
  WARNING! Your password will be stored unencrypted in /home/maroua/.docker/config.json.
  Configure a credential helper to remove this warning. See
  https://docs.docker.com/engine/reference/commandline/login/#credentials-store

  Login Succeeded
● (iotedge-env) maroua [ ~/azuredev-8a05/iot-weather-solution ]$ docker tag weathermodule:0.0.1-amd64 monregistry.azur
  ecr.io/weathermodule:0.0.1-amd64
● (iotedge-env) maroua [ ~/azuredev-8a05/iot-weather-solution ]$ docker push monregistry.azurecr.io/weathermodule:0.0.
  1-amd64
  The push refers to repository [monregistry.azurecr.io/weathermodule]
  e35dc49d4fab: Pushed
  a38bd6fc7f8a: Pushed
  7341f8483151: Pushed
  c9dc5582f995: Pushed
  f61b888276e6: Pushed
  045d684b6e25: Pushed
  53a94dd4e08d: Pushed
  a07f4ed3adbe: Pushed
  0.0.1-amd64: digest: sha256:3c6eb4a3e640cb3727cb16822559725769c58c6a6b12205059b53b4550c88165 size: 1991
○ (iotedge-env) maroua [ ~/azuredev-8a05/iot-weather-solution ]$ 
```



```
(iotedge-env) maroua [ ~/azuredev-8a05/iot-weather-solution ]$ docker images
REPOSITORY                               TAG           IMAGE ID       CREATED          SIZE
weathermodule                            0.0.1-amd64   b3cea6379dde   24 minutes ago   136MB
monregistry.azurecr.io/weathermodule     0.0.1-amd64   b3cea6379dde   24 minutes ago   136MB
yourregistry.azurecr.io/weathermodule    0.0.1-amd64   b3cea6379dde   24 minutes ago   136MB
(iotedge-env) maroua [ ~/azuredev-8a05/iot-weather-solution ]$ 
```

❖ Deploy to Your IoT Edge


Accueil > IoT Hub > IOT-WEATHER | Appareils > weather-edge-device >
Modules
Itinéraires
Vérifier + créer
+ Ajouter
Paramètres du runtime

Définir des modules sur l'appareil : weather-edge-device
IOT-WEATHER

Informations d'identification du registre de conteneurs

Vous pouvez spécifier des informations d'identification dans les registres de conteneurs hébergeant des images de module. Les informations d'identification listées sont utilisées pour récupérer les modules avec l'URL correspondante. L'agent Edge signale le code d'erreur 500 s'il ne trouve pas de paramètre de registre de conteneurs pour un module.

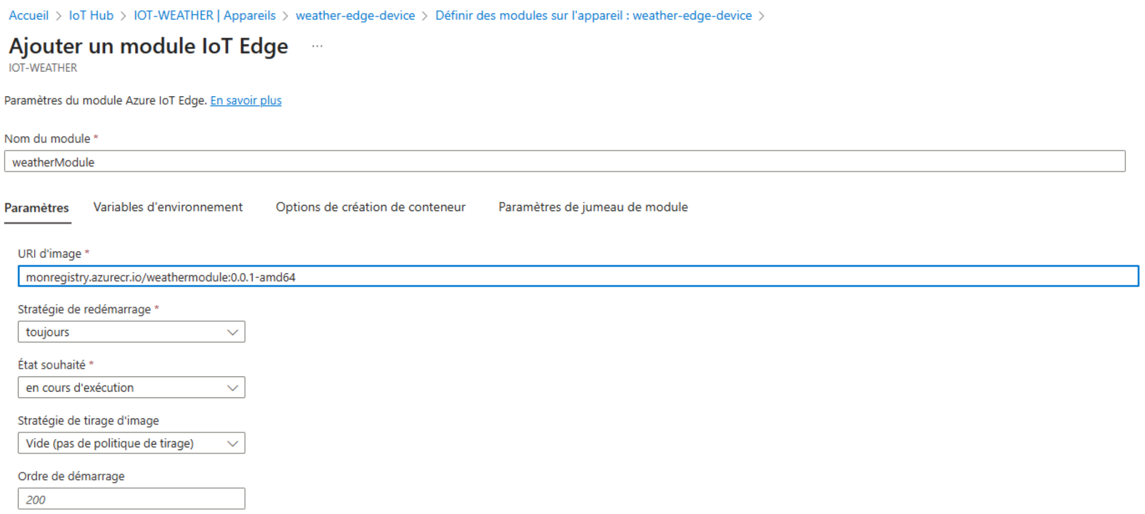
NOM	ADRESSE	NOM D'UTILISATEUR	MOT DE PASSE
monregistry	monregistry.azurecr.io	monregistry	····································
Nom	Adresse	Nom d'utilisateur	Mot de passe

Modules IoT Edge

Les modules IoT Edge sont des conteneurs Docker déployés sur des appareils IoT Edge. Ils peuvent communiquer avec d'autres modules ou envoyer des données au runtime IoT Edge. Les modules sur les appareils sont comptabilisés dans les limites de quota IoT Hub en fonction du niveau et des unités. Par exemple, pour le niveau S1, les modules peuvent être définis 10 fois par seconde si aucune autre mise à jour n'a lieu dans le IoT Hub.

NOM	ÉTAT SOUHAITÉ
Aucun module IoT Edge listés.	

☑ Envoyez des données d'utilisation à Microsoft pour nous aider à améliorer nos produits et services. Lisez notre déclaration de confidentialité pour en savoir plus. Consultez les details des données collectées.


Accueil > IoT Hub > IOT-WEATHER | Appareils > weather-edge-device > Définir des modules sur l'appareil : weather-edge-device >

Ajouter un module IoT Edge ···

IOT-WEATHER

Paramètres du module Azure IoT Edge. En savoir plus

Nom du module *: weatherModule

Paramètres
Variables d'environnement
Options de création de conteneur
Paramètres de jumeau de module

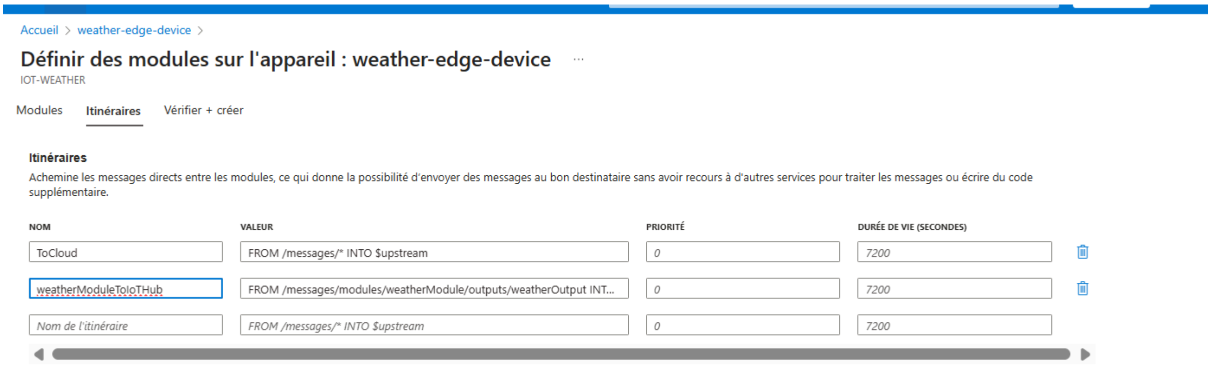
URI d'image *: monregistry.azurecr.io/weathermodule:0.0.1-amd64

Stratégie de redémarrage *: toujours

État souhaité *: en cours d'exécution

Stratégie de tirage d'image: Vide (pas de politique de tirage)

Ordre de démarrage: 200


Accueil > weather-edge-device >

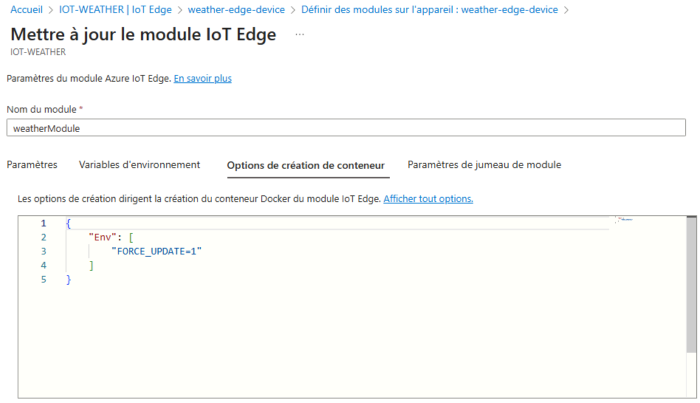
Définir des modules sur l'appareil : weather-edge-device ···

IOT-WEATHER

Itinéraires

Achemine les messages directs entre les modules, ce qui donne la possibilité d'envoyer des messages au bon destinataire sans avoir recours à d'autres services pour traiter les messages ou écrire du code supplémentaire.

NOM	VALEUR	PRIORITÉ	DURÉE DE VIE (SECONDES)
ToCloud	FROM /messages/* INTO $upstream	0	7200
weatherModuleToIoTHub	FROM /messages/modules/weatherModule/outputs/weatherOutput INT...	0	7200
Nom de l'itinéraire	FROM /messages/* INTO $upstream	0	7200


Accueil > IOT-WEATHER | IoT Edge > weather-edge-device > Définir des modules sur l'appareil : weather-edge-device >

Mettre à jour le module IoT Edge ···

IOT-WEATHER

Paramètres du module Azure IoT Edge. En savoir plus

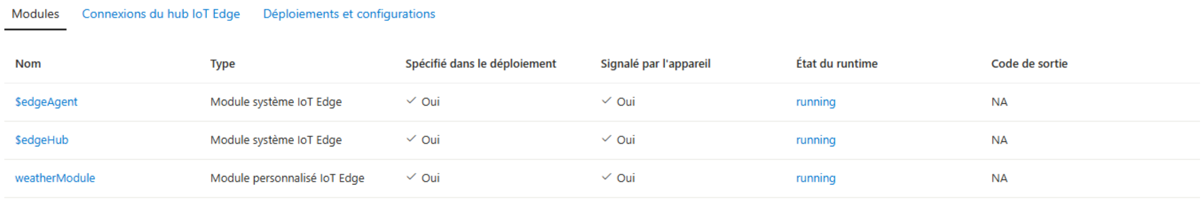
Nom du module *: weatherModule

Paramètres
Variables d'environnement
Options de création de conteneur
Paramètres de jumeau de module

Les options de création dirigent la création du conteneur Docker du module IoT Edge. Afficher tout options.

```
{
    "Env": [
        "FORCE_UPDATE=1"
    ]
}
```

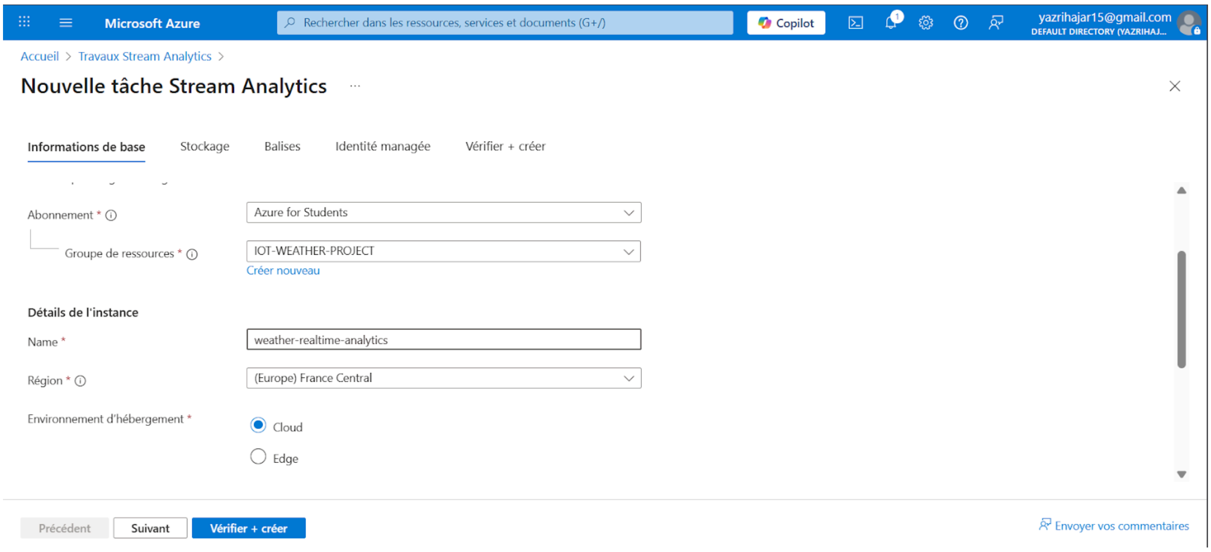
❖ verify That the Module Is Successfully Deployed



Modules | Connexions du hub IoT Edge | Déploiements et configurations

Nom	Type	Spécifié dans le déploiement	Signalé par l'appareil	État du runtime	Code de sortie
$edgeAgent	Module système IoT Edge	✓ Oui	✓ Oui	running	NA
$edgeHub	Module système IoT Edge	✓ Oui	✓ Oui	running	NA
weatherModule	Module personnalisé IoT Edge	✓ Oui	✓ Oui	running	NA

Implementation: Created a Stream Analytics job with IoT Hub as the input and Power BI as the output.



Microsoft Azure
Rechercher dans les ressources, services et documents (G+/)
Copilot
yazrihajar15@gmail.com
DEFAULT DIRECTORY (YAZRIHAJ...
Accueil > Travaux Stream Analytics >

Nouvelle tâche Stream Analytics

Abonnement * : Azure for Students

Groupe de ressources * : IOT-WEATHER-PROJECT

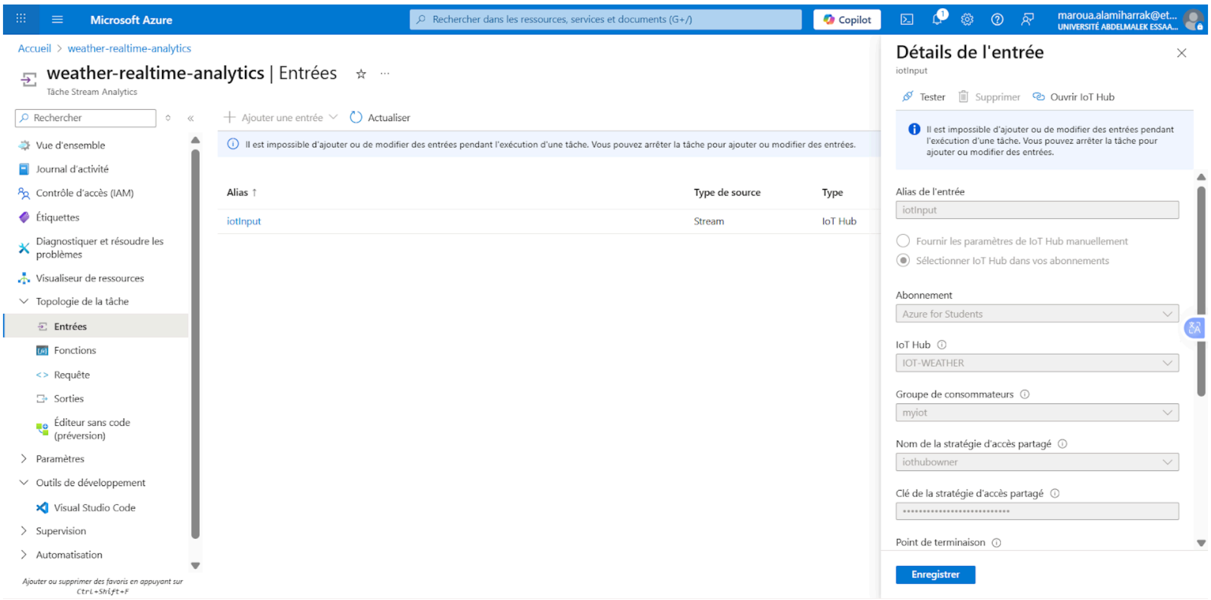
Créer nouveau

Détails de l'instance

Name * : weather-realtime-analytics

Région * : (Europe) France Central

Environnement d'hébergement * : ● Cloud ○ Edge



Microsoft Azure
Rechercher dans les ressources, services et documents (G+/)
Copilot
maroua.alamiharrak@et...
UNIVERSITÉ ABDELMALEK ESSAA...
Accueil > weather-realtime-analytics

weather-realtime-analytics | Entrées

Tâche Stream Analytics

Il est impossible d'ajouter ou de modifier des entrées pendant l'exécution d'une tâche. Vous pouvez arrêter la tâche pour ajouter ou modifier des entrées.

Alias ↑	Type de source	Type
iotInput	Stream	IoT Hub

Détails de l'entrée

iotInput

Il est impossible d'ajouter ou de modifier des entrées pendant l'exécution d'une tâche. Vous pouvez arrêter la tâche pour ajouter ou modifier des entrées.

Alias de l'entrée : iotInput

○ Fournir les paramètres de IoT Hub manuellement
● Sélectionner IoT Hub dans vos abonnements

Abonnement : Azure for Students

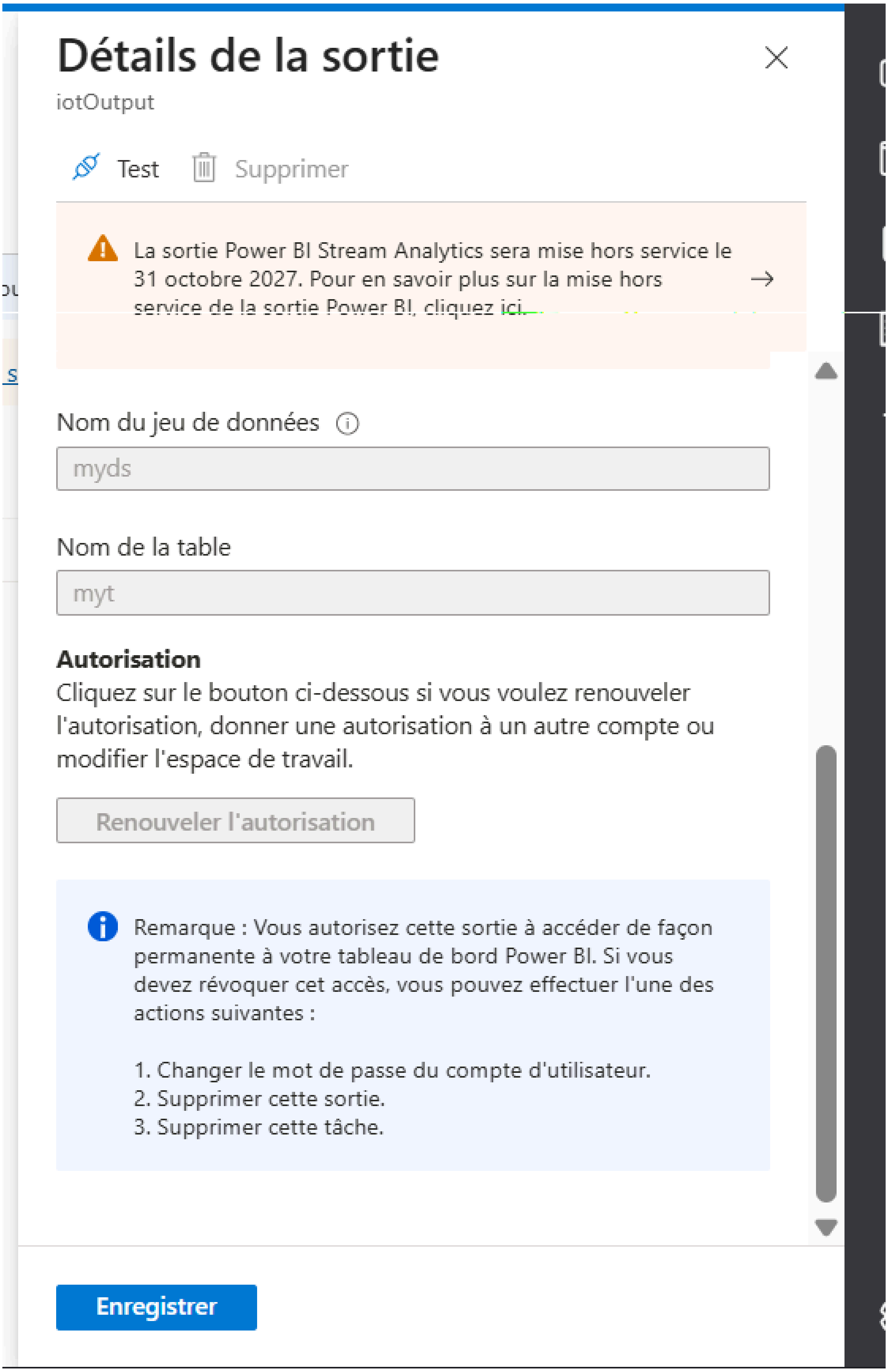
IoT Hub : IOT-WEATHER

Groupe de consommateurs : myiot

Nom de la stratégie d'accès partagé : iothubowner

Clé de la stratégie d'accès partagé : ••••••••••••••••••••••••••

Point de terminaison



Détails de la sortie

iotOutput

⚠ La sortie Power BI Stream Analytics sera mise hors service le 31 octobre 2027. Pour en savoir plus sur la mise hors service de la sortie Power BI, cliquez ici.

Nom du jeu de données: myds

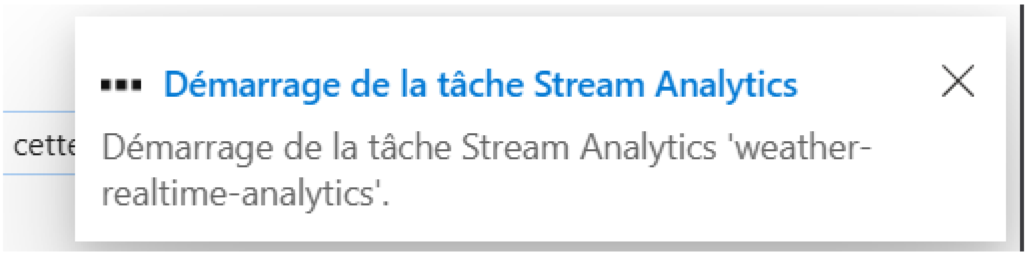
Nom de la table: myt

Autorisation

Cliquez sur le bouton ci-dessous si vous voulez renouveler l'autorisation, donner une autorisation à un autre compte ou modifier l'espace de travail.

Remarque : Vous autorisez cette sortie à accéder de façon permanente à votre tableau de bord Power BI. Si vous devez révoquer cet accès, vous pouvez effectuer l'une des actions suivantes :

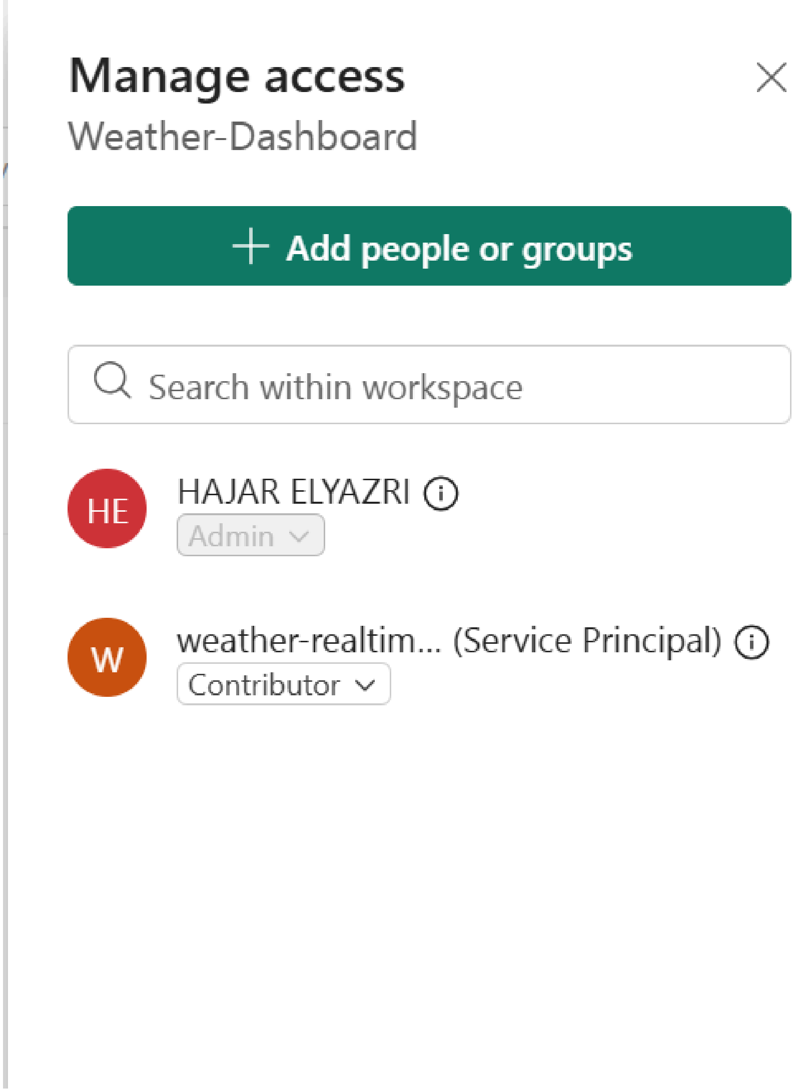
1. Changer le mot de passe du compte d'utilisateur.
2. Supprimer cette sortie.
3. Supprimer cette tâche.



Démarrage de la tâche Stream Analytics

Démarrage de la tâche Stream Analytics 'weather-realtime-analytics'.

Implementation: Provisioned an Azure resource group, IoT Hub, and a connected IoT device. Set up a Stream Analytics job to ingest data from the IoT Hub and stream it to Power BI for real-time visualization in the designated workspace.



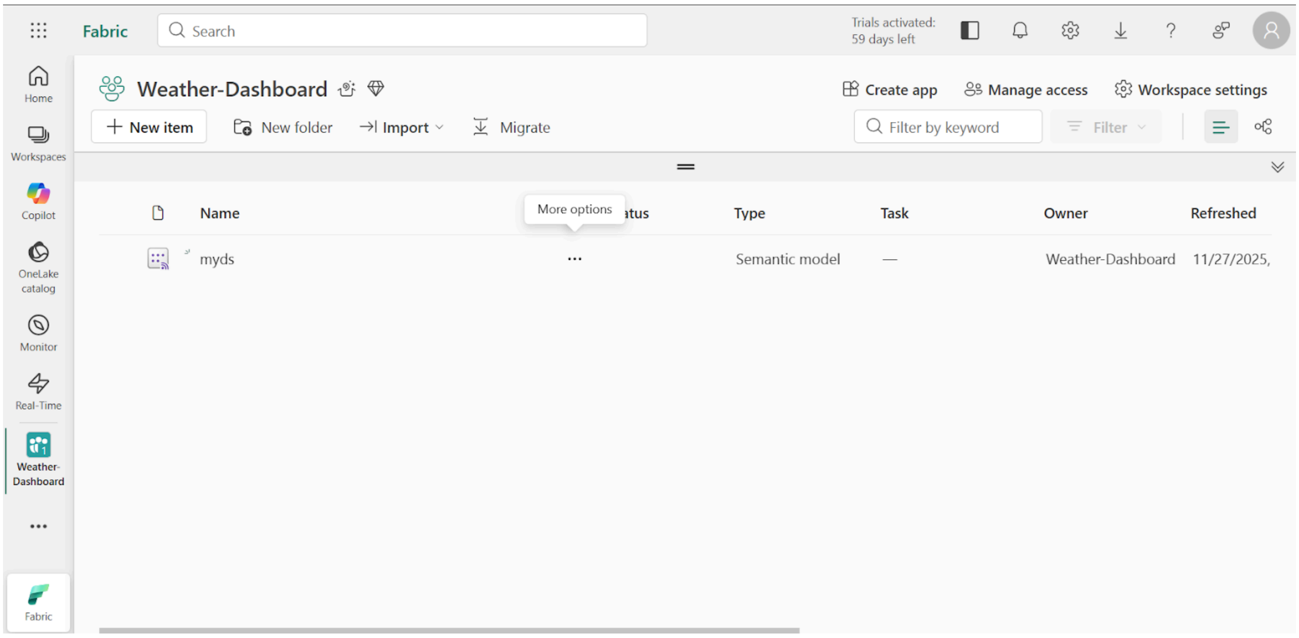
Manage access

Weather-Dashboard

HE HAJAR ELYAZRI — Admin

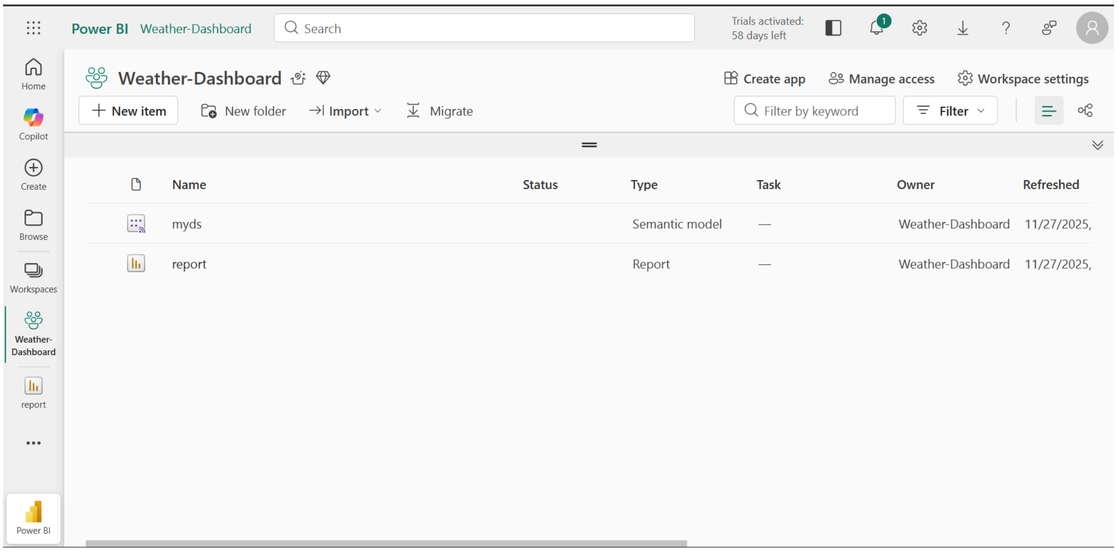
W weather-realtim... (Service Principal) — Contributor

Granted Power BI access to the Stream Analytics job to enable data streaming into the workspace.



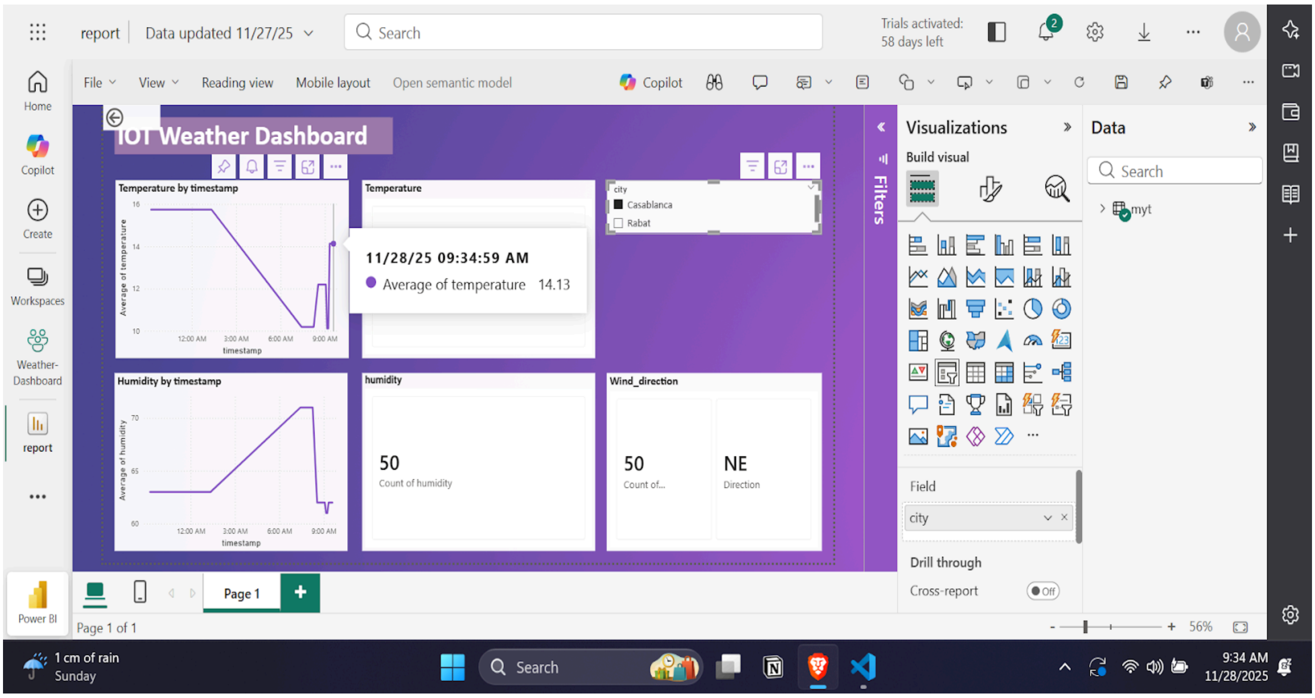
Name	Status	Type	Task	Owner	Refreshed
myds		Semantic model	—	Weather-Dashboard	11/27/2025,

❖ **Created a report in Power BI to visualize and display the ingested IoT data.**



Name	Status	Type	Task	Owner	Refreshed
myds		Semantic model	—	Weather-Dashboard	11/27/2025,
report		Report	—	Weather-Dashboard	11/27/2025,

Final Implementation: Provisioned an Azure resource group, IoT Hub, and a connected IoT device. Configured a Stream Analytics job to ingest data from the IoT Hub and granted Power BI access, enabling real-time streaming of telemetry data. Successfully deployed a live Power BI dashboard for real-time visualization and monitoring of the IoT data.



VII – General Conclusion:

This project demonstrates a complete end-to-end IoT architecture that integrates edge computing, cloud services, real-time analytics, and interactive visualization. By designing and deploying a Python-based weather module running in Docker containers and connected through Azure IoT Edge, the system successfully collects, timestamps, and preprocesses telemetry before securely transmitting it to the cloud via Azure IoT Hub. This approach ensures both data integrity and reliable, low-latency communication while adhering to modern IoT security standards.

The cloud layer, supported by IoT Hub and Stream Analytics, plays a central role in transforming raw telemetry into structured, high-value information. Through SQL-based queries, data cleaning, aggregation, and enrichment, the pipeline achieves efficient real-time processing that enables immediate insight generation. Stream Analytics ensures that only meaningful, high-quality data reaches the visualization tools, optimizing cloud resource consumption and improving overall system performance.

On the visualization layer, Power BI provides a dynamic and intuitive dashboard for monitoring weather trends. Its real-time refresh capabilities, combined with interactive charts and automated KPI tracking, enable stakeholders to access clear, actionable insights at any moment. The platform transforms complex data flows into user-friendly analytics, highlighting the value of integrating IoT data with business intelligence tools.

Overall, this project showcases the strength of a scalable and modular IoT ecosystem, capable of operating seamlessly from local edge processing to cloud-based analytics and visualization. It highlights the importance of reliability, security, and data efficiency in modern IoT solutions and demonstrates how distributed architectures can significantly improve responsiveness and analytical precision. Beyond its technical implementation, the system provides a solid foundation for future enhancements, such as AI-based forecasting, sensor integration, automation, and advanced alerting mechanisms.

This work confirms the relevance and effectiveness of combining IoT Edge, Azure services, and real-time analytics to build intelligent, performant, and future-ready IoT applications.

References / Resources:

- **Azure IoT Hub Documentation:** https://docs.microsoft.com/azure/iot-hub/
- **Azure IoT Edge Documentation:** https://docs.microsoft.com/azure/iot-edge/
- **Azure Stream Analytics Documentation:** https://docs.microsoft.com/azure/stream-analytics/
- **Azure Resource Manager Documentation:** https://docs.microsoft.com/azure/azure-resource-manager/
- **Power BI Documentation:** https://docs.microsoft.com/power-bi/
- **Lucidchart:** https://www.lucidchart.com/ – used for project architecture and sequence diagrams
- **OpenWeather API:** https://openweathermap.org/api – used to fetch weather data
- **GitHub Repository:** https://github.com/marouaa145/azuredev-8a05