

IoT Edge overcomes the problems of **latency** and **response times**.

**Machine learning models** can be **trained in the cloud** and then **deployed on edge devices**.

**What is IoT Edge?**

**Azure IoT Edge** is a combination of a **cloud service running** in the cloud and a **runtime that runs on the device**.

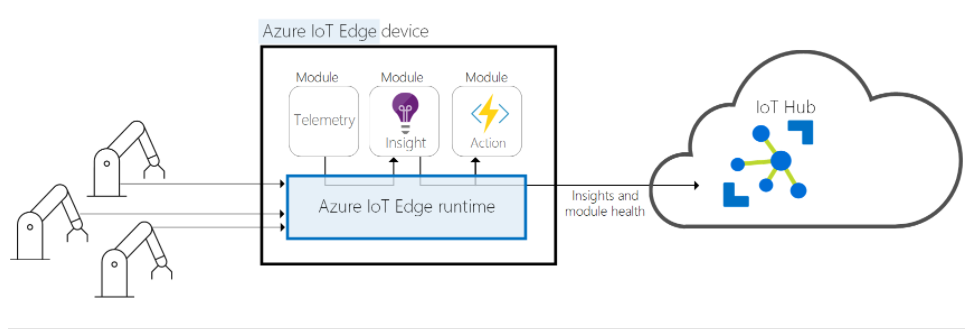
**How IoT Edge works**

Azure IoT Edge comprises three components:

-**IoT Edge modules** are units of execution implemented as **Docker compatible containers**.

**IoT Edge runtime** runs on each IoT Edge device and manages the runtime and communication for the modules deployed to each device. IoT Edge runtime ensures that the modules are always running and report module health to the cloud.

**IoT Edge cloud interface** enables you to monitor and manage IoT Edge devices remotely.



* Module
* Runtime
* Cloud interface

LAB

1. **Verify that the lab prerequisites are met (that you have the required Azure resources)**

|  |  |
| --- | --- |
| Resource Group | rg-az220 |
| IoT Hub | iot-az220-training-001 |

1. **Deploy an Azure IoT Edge Enabled Linux VM**

deploy an **Ubuntu Server VM**

* Resource group: **rg-az220vm**
* **VM name: vm-az220-training-edge0001-00001**
* **Select the region where your Azure IoT Hub is provisioned.**
* Leave Availability options set to No infrastructure redundancy required.
* **Image: Ubuntu Server 18.04 LTS - Gen1**
* Leave **Azure Spot instance** field unchecked.
* **VM Size**, click **Standard\_B1ms**
* **Administrator account/Password**
* **Inbound port rules** are configured to enable inbound **SSH** access to the VM

1. **Create an IoT Edge Device Identity in IoT Hub using Azure CLI**

az extension update --name azure-iot

Create an IoT Edge device identity in your IoT hub:

az iot hub device-identity create --hub-name iot-az220-training-001 --device-id sensor-th-0067001 --edge-enabled

Get connection String:

az iot hub device-identity connection-string show --device-id sensor-th-0067001 --hub-name iot-az220-training-001

This connection string will be used to configure the IoT Edge device to connect to IoT Hub.

HostName=iot-az220-training-001.azure-devices.net;DeviceId=sensor-th-0067001;SharedAccessKey=s9p0vX3nvvTBFQKULRT4H73LEnfCFZPSUOKG0EwUsks=

### **Install IotEdge and Connect IoT Edge Device to IoT Hub**

IoT Edge runtime has three components:

The IoT Edge runtime is deployed on all IoT Edge devices. It has three components. The **IoT Edge security daemon** starts each time an IoT Edge device boots and bootstraps the device by starting the IoT Edge agent. The **IoT Edge agent** facilitates deployment and monitoring of modules on the IoT Edge device, including the IoT Edge hub. The **IoT Edge hub** manages communications between modules on the IoT Edge device, and between the device and IoT Hub.

* 1. Connect to the VM from Cloud Shell:

ssh [username@52.170.205.79](mailto:username@52.170.205.79)

* 1. Add the Microsoft installation packages to the package manager

To configure the VM to access the Microsoft installation packages, run the following command:

Enable root access to VM: **sudo su -**

sudo curl https://packages.microsoft.com/config/ubuntu/18.04/multiarch/prod.list > ./microsoft-prod.list

To add the downloaded package list to the package manager, run the following command:

sudo cp ./microsoft-prod.list /etc/apt/sources.list.d/

To install the packages, the Microsoft GPG public key must be installed. Run the following commands:

curl https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor > microsoft.gpg

sudo cp ./microsoft.gpg /etc/apt/trusted.gpg.d/

* 1. Install a container engine: Moby engine

To update the package lists on the device, run the following command:

sudo apt-get update

To install the **Moby** engine, run the following command:

sudo apt-get install moby-engine

* 1. Install IoT Edge runtime

The IoT Edge security daemon provides and maintains security standards on the IoT Edge device. The daemon starts on every boot and bootstraps the device by starting the rest of the IoT Edge runtime.

To update the package lists on the device

sudo apt-get update

To list the versions of **IoT Edge runtime** that are available

apt list -a iotedge

To install the latest version of the **IoT Edge runtime**, run the following command:

sudo apt-get install iotedge

sudo apt-get install iotedge=1.0.9-1 libiothsm-std=1.0.9-1

To confirm that the Azure IoT Edge Runtime is installed on the VM, run the following command:

iotedge version

* 1. Configure connection to IoT Hub

To ensure that you are able to configure Azure IoT Edge, enter the following command:

sudo chmod a+w /etc/iotedge/config.yaml

To configure Azure IoT Edge, the **/etc/iotedge/config.yaml** configuration file needs to be modified to contain the full path to the certificate and key files on the IoT Edge Device. Before the file can be edited, you must be sure that the **config.yaml** file is not read-only. The command above sets the **config.yaml** file to be writable.

sudo vi /etc/iotedge/config.yaml

# Manual provisioning configuration using a connection string

provisioning:

source: "manual"

device\_connection\_string: "<ADD DEVICE CONNECTION STRING HERE>"

dynamic\_reprovisioning: false

sudo systemctl restart iotedge

sudo systemctl status iotedge

To verify the IoT Edge runtime has connected, run the following command:

sudo iotedge check

1. **Add an Edge Module to the Edge Device**

In this exercise, you will **add a Simulated Temperature Sensor** as a **custom IoT Edge Module**, and **deploy it** to run on the IoT Edge Device.

1. **Portal > Search “**Simulated Temperature Sensor” **Create** to the IoT Hub and IoT Dege Device before.

Route: $upstream

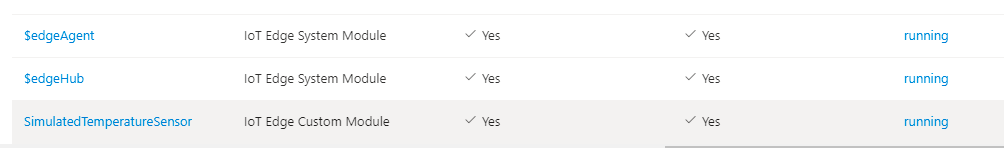
{

"routes": {

"route": "FROM /messages/\* INTO $upstream"

}

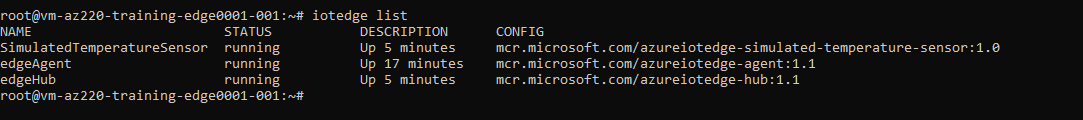
}



**Test**

**Iotedge list**

View the messages being sent from the temperature sensor module using the command sudo **iotedge logs SimulatedTemperatureSensor -f.**



1. **Deploy Azure Stream Analytics as an IoT Edge Module**

#### **Task 1: Create Azure Storage Account**

Resource Group: rg-az220

Name: az220storage001

Location == Iot Hub

#### **Task 2: Create an Azure Stream Analytics job**

**Hosting environment** field to **Edge**.

#### **Task 3: Configure Azure Stream Analytics Job**

**Add stream input**, and then click **Edge Hub**.

**Input alias** field, enter **temperature**

**Event serialization format** dropdown, ensure that **JSON** is selected.

**Outputs** pane, click **+ Add**, and then click **Edge Hub**.

**Output alias** field, enter **alert**

**Event serialization format** dropdown, ensure that **JSON** is selected.

**Format** dropdown, ensure that **Line separated** is selected.

**Encoding** dropdown, ensure that **UTF-8** is selected.

Query:

SELECT

'reset' AS command

INTO

alert

FROM

temperature TIMESTAMP BY timeCreated

GROUP BY TumblingWindow(second,15)

HAVING Avg(machine.temperature) > 25

#### **Task 4: Configure Storage Account Settings**

**Stream Analytics job > Configure > Storage account settings**

**Add storage account**.

**jobdefinition** as the name of the container

#### **Task 5: Deploy the Stream Analytics Job**

Iot Hub > IoT Edge >Device ID > sensor-th0067001

**IoT Edge Modules**, click **Add**, and then click **Azure Stream Analytics Module**.

1. On the **Set modules on device: sensor-th-0067** pane, click **Next: Routes >**.

Notice that the existing routing is displayed.

1. Replace the default route defined with the following three routes:
   * Route 1
     + NAME: **telemetryToCloud**
     + VALUE: FROM /messages/modules/tempsensor/\* INTO $upstream
   * Route 2
     + NAME: **alertsToReset**
     + VALUE: FROM /messages/modules/asa-az220-training-{your-id}/\* INTO BrokeredEndpoint("/modules/tempsensor/inputs/control")
   * Route 3
     + NAME: **telemetryToAsa**
     + VALUE: FROM /messages/modules/tempsensor/\* INTO BrokeredEndpoint("/modules/asa-az220-training-{your-id}/inputs/temperature")

**Note**: Be sure to replace the asa-az220-training-{your-id} placeholder with the name of your Azure Stream Analytics job module. You can click **Previous** to view the list of modules and their names, then click **Next** to come back to this step.

The routes being defined are as follows:

* + The **telemetryToCloud** route sends the all messages from the tempsensor module output to Azure IoT Hub.
  + The **alertsToReset** route sends all alert messages from the Stream Analytics module output to the input of the **tempsensor** module.
  + The **telemetryToAsa** route sends all messages from the tempsensor module output to the Stream Analytics module input.

iotedge logs tempsensor