

OpenBlocks IoT Family Azure IoT Edge Set-up Guide



Ver.3.2.0

Plat'Home Co., Ltd.

■ About trademarks

- Linux is a trademark or registered trademark of Linus Torvalds in the United States and/or other countries.
- Company and product names mentioned in this Set-up Guide may be trademarks or registered trademarks of their respective companies.
- Product names and other proper nouns in this Set-up Guide are trademarks or registered trademarks of their respective companies.
- Microsoft, .NET, Windows and Microsoft Azure are registered trademarks of the Microsoft Corporation in the United States and/or other countries.
- Docker and Docker logo are trademarks or registered trademarks of Docker, Inc. in the United States and/or other countries. Docker, Inc. and other parties may also have trademark rights in other terms used herein.

■ Before using this product

- No reproduction of this material is allowed without written permission of Plat'Home Co., Ltd.
- Content and information contained within this material may be changed or updated without prior notice.
- We consistently aim to keep the content in this material as precise as possible. However, should any errors in descriptions, etc. be noticed, please contact Plat'Home Co., Ltd. The latest version of this material can be downloaded from our website.
- While using this product, please be aware that it is not designed or assumed for use in fields where there is a risk to life.
- Regardless of the aforementioned, in no event will Plat'Home be liable for any special, incidental, indirect or consequential damage arising out of use of this product, including but not limited to damage to profits or loss.

Table of contents

Chapter 1 General	4
Chapter 2 Azure IoT Edge setup	4
2-1. Installing Azure IoT Edge, including WEB UI	4
2-2. [Preparations in advance] IoT Hub setup	5
2-3. Azure IoT Edge setup from WEB UI	5
2-4. Azure IoT Edge setup	6
2-5. Azure IoT Edge operation	8
2-6. Environment variables of Azure IoT Edge modules	10
Chapter 3 Others	11
3-1. Conjunction with Docker for WEB UI	11
3-2. Compatibility with Azure IoT Edge public beta	11
3-3. Add containers of Azure IoT Edge	11

Chapter 1 General

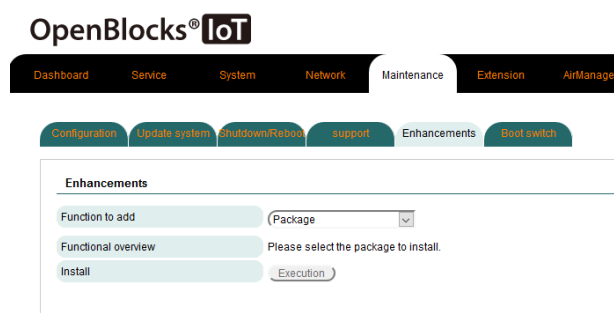
This manual describes how to use Azure IoT Edge that can be installed in the OpenBlocks IoT Family, including web user interface (hereinafter referred to as "WEB UI").

* Azure IoT Edge allows you to run the analysis and custom business logic that you were doing in the cloud on the device side.

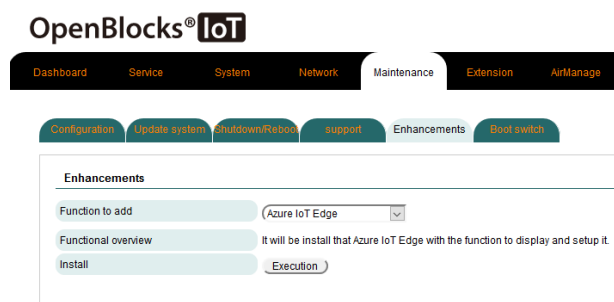
Chapter 2 Azure IoT Edge setup

2-1. Installing Azure IoT Edge, including WEB UI

At the time of shipment from our factory, Docker, Azure IoT Edge module and WEB UI for Azure IoT Edge are not installed in this product. To install WEB UI and Azure IoT Edge, using the **[Maintenance]-[Enhancements]** tab.



When choosing the **[Maintenance]** tab of WEB UI and clicking on the **[Enhancements]** tab, it is possible to choose a package for extensions.



From the pull-down menu showing a list of packages to be installed, choose "Azure IoT Edge".

Press the Execution button to install the program.

After completing installation, the unit will require rebooting to make the installation effective. Choose the Shutdown/Reboot tab from the Maintenance tab to reboot the unit.

When installing Azure IoT Edge, if Docker DAEMON does not exist, Docker DAEMON will also be installed.

While installing Docker or Moby, certain drivers modules will be compiled, which takes additional time. A button to check the installation status will appear. Press this button to check on the progress of installation.

Please note that Azure IoT Edge uses Docker internally. To check the status of containers, etc. to be used by Azure IoT Edge, please also consider using the WEB UI (Docker) function.

In addition, data between the host machine (OpenBlocks IoT series) and sensors or other devices can be transmitted to Azure IoT Edge. In this case, please consider using the IoT data control function.

2-2. [Preparations in advance] IoT Hub setup

Before using Azure IoT Edge, it is necessary to create an IoT Hub and to register IoT Edge devices in advance.

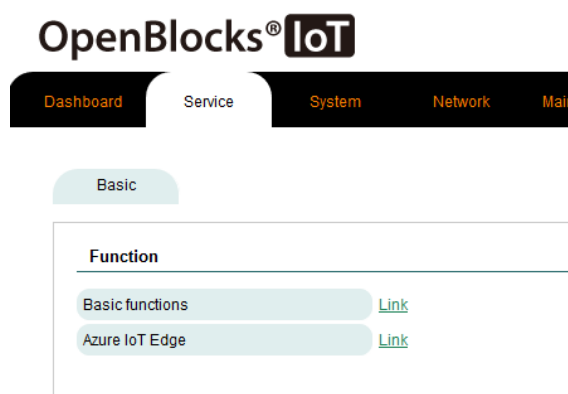
For these procedures, please refer to "Create an IoT Hub" and "Register an IoT Edge device" at the URL below and carry them out.

<https://docs.microsoft.com/en-us/azure/iot-edge/quickstart-linux>

Make a note of the connection string of each registered IoT Edge device (primary or secondary key).

2-3. Azure IoT Edge setup from WEB UI

When the Azure IoT Edge package has been installed, link of Azure IoT Edge function will be displayed in the **[Service]-[Basic]** tab.



Choose the Service tab and click on the **Azure IoT Edge** link in the **[Basic]** tab. The root tab will be switched to display the **[Dashboard]**, **[Basic]** and **[Azure IoT Edge]** tabs for service.

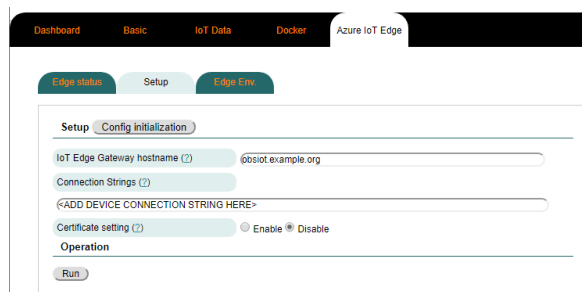
*Links and root tabs for service that appear in the **[Service]-[Basic]** tab may differ, depending on the packages installed using the **[Maintenance]-[Enhancements]** tab.

2-4. Azure IoT Edge setup

To perform operations, etc. from this chapter and below, setup described in Chapter 2-2 "[Preparation in advance] IoT Hub setup" must be completed in advance.

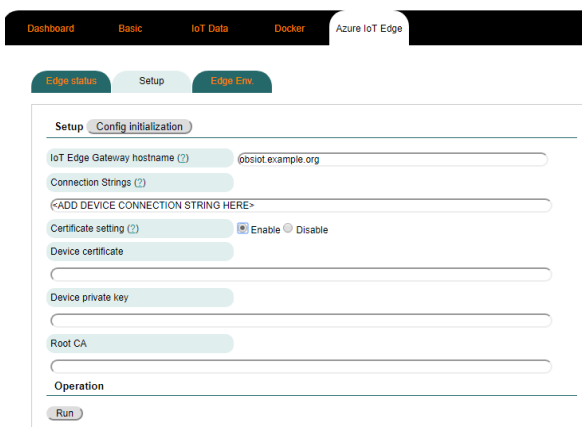
Choose the **[Azure IoT Edge]-[Setup]** tab, and use the connection string taken note of in the IoT Hub setup to set up Azure IoT Edge.

*Not set X.509 Certificate



The screenshot shows the 'Setup' tab of the Azure IoT Edge configuration interface. The 'Config initialization' section is active. The 'IoT Edge Gateway hostname' is set to 'pbsiot.example.org'. The 'Connection Strings' field is empty, with a placeholder '<ADD DEVICE CONNECTION STRING HERE>'. The 'Certificate setting' is set to 'Disable' (radio button selected). The 'Operation' section is at the bottom with a 'Run' button.

*Set X.509 Certificate



The screenshot shows the 'Setup' tab of the Azure IoT Edge configuration interface. The 'Config initialization' section is active. The 'IoT Edge Gateway hostname' is set to 'pbsiot.example.org'. The 'Connection Strings' field is empty, with a placeholder '<ADD DEVICE CONNECTION STRING HERE>'. The 'Certificate setting' is set to 'Enable' (radio button selected). Below this, there are input fields for 'Device certificate', 'Device private key', and 'Root CA'. The 'Operation' section is at the bottom with a 'Run' button.

Setup

Initialization of the configured configuration can be done by **Config initialization** button.

IoT Edge Gateway hostname:

The FQDN of this unit is input by default.

*There is no need to change.

Connection Strings:

Specify the connection string of the created IoT Edge.

Certificate setting:

To set X.509 certificate, choose "Enable." If not, choose "Disable." If chose "Disable", a certificate will be created internally.

*In addition, it is not recommended to actually operate the internally generated certificate.

Device certificate:

Specify the file pathname of the X.509 device certificate.

Device private key:

Specify the file pathname of the X.509 device private key.

Root CA:

Specify the file pathname of the X.509 root certificate.

Upload the certificate and private key using the **[System]-[File Management]** tab.

As the IoT Edge daemon checks the authority of the file and file storage directory of each certificate, it is necessary to grant read permission to the file, execution authority and read authority to the directory.

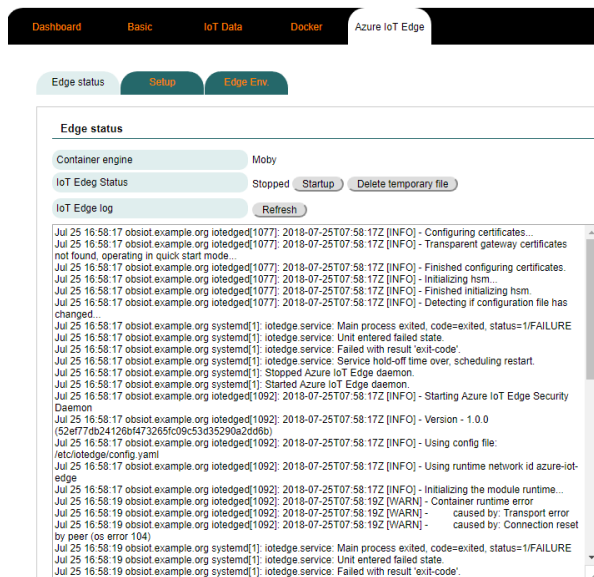
After making the necessary entries, press the **Run** button.

Normally, after completing this process, the screen will move to the **Edge status** tab.

2-5. Azure IoT Edge operation

In the [Azure IoT Edge]-[Edge status] tab, it is possible to carry out the following actions:

- Show which container engine is being used.
- Show status of IoT Edge daemon.
- Control IoT Edge daemon to start up/stop.
- Deleting temporary file for IoT Edge.
- Show log of IoT Edge daemon.



Edge status

Container engine:

Show which container engine is being used, Docker or Moby. In addition, Moby is recommended for the container engine used in IoT Edge, please confirm that you are Moby if you get support on IoT Edge.

IoT Edge status:

Show status of IoT Edge daemon.

A processing button corresponding to the situation is displayed.

- **Stopped**

Startup: start up IoT Edge daemon.

Delete temporary file: Deletes files internally generated by the IoT Edge daemon. If you change the configuration information of the setup, we recommend that you execute this processing.

- **Running**

Stop: stop IoT Edge daemon.

Reboot: stop and start up IoT Edge daemon.

IoT Log:

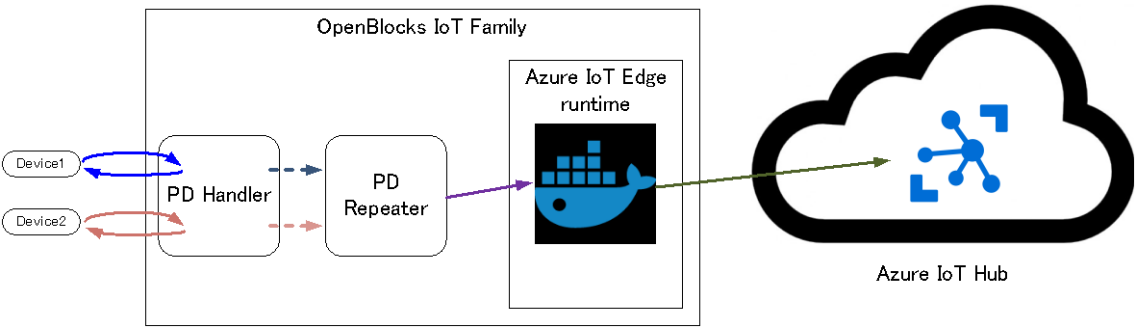
Show log of IoT Edge daemon.

Refresh button updates the latest log information.

After setup is completed, container downloading is carried out by starting IoT Edge daemon from this page or restarting the main unit. After that, Azure IoT Edge will continue to run.

2-6. Environment variables of Azure IoT Edge modules

As shown in the figure below, in conjunction with the IoT data control function, it is possible to send data and perform a filtering process to Azure IoT Hub via a container downloaded by Azure IoT Edge runtime.



When device data is sent to an IoT Edge module via the IoT data control function, environment variables that the IoT Edge module has must be used.

Such environment variables can be checked using the **[Azure IoT Edge]-[Edge Env.]** tab.

Dashboard		
Basic		
IoT Data		
Docker		
Azure IoT Edge		
Edge status		
Setup		
Edge Env.		
Edge Env.		
SedgeAgent	IOTEDGE_APIVERSION	2018-06-28
	IOTEDGE_AUTHSCHEME	sas Token
	IOTEDGE_DEVICEID	EdgeDeviceID001
	IOTEDGE_IOTHUBHOSTNAME	OpenBlocksIoT001.azure-devices.net
	IOTEDGE_MANAGEMENTURI	unix://var/run/iotedge/mgmt.sock
	IOTEDGE_MODULEGENERATIONID	63665751912928792
SedgeHub	IOTEDGE_MODULEID	SedgeAgent
	IOTEDGE_WORKLOADURI	unix://var/run/iotedge/workload.sock
	EdgeDeviceHostName	obstoc.example.org
	IOTEDGE_APIVERSION	2018-06-28
SedgeHub	IOTEDGE_AUTHSCHEME	sas Token
	IOTEDGE_DEVICEID	EdgeDeviceID001
	IOTEDGE_IOTHUBHOSTNAME	OpenBlocksIoT001.azure-devices.net
	IOTEDGE_MODULEGENERATIONID	63665751912928792
	IOTEDGE_MODULEID	SedgeHub
	IOTEDGE_WORKLOADURI	unix://var/run/iotedge/workload.sock
dummyContainer	IOTEDGE_APIVERSION	2018-06-28
	IOTEDGE_AUTHSCHEME	sas Token
	IOTEDGE_DEVICEID	EdgeDeviceID001
	IOTEDGE_GATEWAYHOSTNAME	obstoc.example.org
	IOTEDGE_IOTHUBHOSTNAME	OpenBlocksIoT001.azure-devices.net
	IOTEDGE_MODULEGENERATIONID	636657544465241038
dummyContainer	IOTEDGE_MODULEID	dummyContainer
	IOTEDGE_WORKLOADURI	unix://var/run/iotedge/workload.sock

Edge Env.

Refers to container information downloaded after activation and shows environment variables for Azure IoT Edge.

Please note that the information necessary by the IoT data control function is **module ID** (IOTEDGE_MODULEID) and **GW Host** (IOTEDGE_GATEWAYHOST). Enter this information in the relevant tab in the IoT data control function.

Chapter 3 Others

3-1. Conjunction with Docker for WEB UI

With this function, total control of Azure IoT edge is possible, but no individual container control is available. If the filtering function in added IoT Edge modules or Azure IoT Edge is used, excessive memory may be consumed, affecting the host machine (OpenBlocks).

If installing Docker for WEB UI, it is possible to check the resource status or set up the control function by installing Docker for WEB UI. We recommend considering this option.

3-2. Compatibility with Azure IoT Edge public beta

With the release of Azure IoT Edge 's GA version, IoT Edge internal system and containers were renewed. As a result, compatibility with the public beta version is gone. Therefore, please delete various data etc. of the public beta version. If you update our package, it will be deleted automatically.

3-3. Add containers of Azure IoT Edge

We may prepare and deploy non-standard containers such as Azure Functions and Azure ML. For these containers, we will refer to and deploy from Azure Container Registry and Docker Hub.

The following URL is a description of cases using Azure Functions as containers and Azure Container Registry. Please refer to here and customize IoT Edge.

<https://docs.microsoft.com/en/azure/iot-edge/tutorial-deploy-function#create-a-container-registry>

Plat'Home Co., Ltd.
NIHON BUILDING KUDANBEKKAN, 3F
4-2-3, Kudankita, Chiyoda-ku, TOKYO 102-0073, JAPAN