



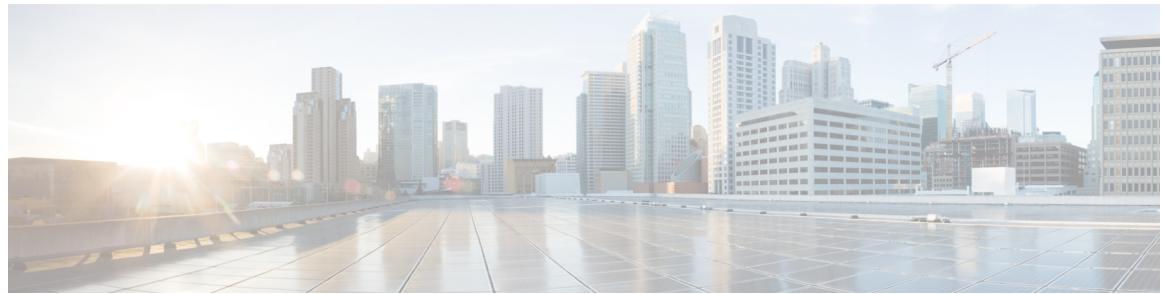
## **Cisco Edge Intelligence User Guide, Release 2.0**

**First Published:** 2025-04-07

### **Americas Headquarters**

Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
USA  
<http://www.cisco.com>  
Tel: 408 526-4000  
800 553-NETS (6387)  
Fax: 408 527-0883

© 2025 Cisco Systems, Inc. All rights reserved.



## CONTENTS

---

---

### CHAPTER 1

#### **Cisco Edge Intelligence 1**

- Edge to multi-cloud data flows 1
- Creating Cisco Edge Intelligence pipelines 2
- Manage pipelines at scale 3

---

### CHAPTER 2

#### **Install and access Cisco Edge Intelligence agents 5**

- Download Cisco Edge Intelligence software 5
- Install the Cisco Edge Intelligence agent using IOx Local Manager 5
  - Static NAT configuration 6
    - Get Cisco Edge Intelligence (container service) IP address for NAT configuration 6
    - Configure static NAT using device GUI 7
    - Configure static NAT using device CLI 7
  - Secure remote access to assets using Cisco Secure Endpoint Access 8
  - Log in to Cisco Edge Intelligence Local Manager 9
    - Reset password 11
- Install Cisco Edge Intelligence application using Cisco Catalyst SDWAN Manager 13
  - Get the Cisco Edge Intelligence image 13
  - Cisco Catalyst SDWAN Manager process overview 13
- Remove a Cisco Edge Intelligence Agent 15
- Port opening mechanism for gateways 15

---

### CHAPTER 3

#### **Cisco Edge Intelligence Local Manager 17**

- How to create pipelines 18
- Data sources 19
  - Configure MQTT asset type 21
  - Configure Modbus-Serial asset type 23

Adding Asset Types for Modbus TCP Connection Type	26
Configure OPC-UA asset type	29
Configure Serial asset type	32
RSU asset type settings	34
NTCIP1202, NTCIP1203, NTCIP1204 asset type settings	36
Add data destinations	40
Add a Microsoft Azure IoT Hub destination	40
Add an MQTT Server destination	42
Add an AWS Server destination	45
About Data Policies	47
Create a Data Logic	47
Create a Data Rule	50
Deploy or undeploy pipelines	51
View health status	52



## CHAPTER 1

# Cisco Edge Intelligence

Cisco Edge Intelligence is an edge-to-multi-cloud data orchestration software that processes the data from connected edge assets. This software operates on Cisco industrial routers, switches and compute gateways for simple out-of-the-box deployment.

Cisco Edge Intelligence is available as an independent IOx application, without dependency on IoT Operations Dashboard cloud platform. This guide covers the standalone solution, while the documentation for Cisco Edge Intelligence deployed using cloud infrastructure (IoT Operations Dashboard) is available [here](#).

### Supported devices

The Cisco Edge Intelligence agent runs on Cisco network devices as a Cisco IOx app on these devices:

- Cisco 829 Industrial Integrated Services Routers (Cisco IR829)
- Cisco Catalyst IR1101 Rugged Series Routers
- Cisco Catalyst IR1800 Rugged Series Routers
- Cisco Catalyst IE3400 Rugged Series Switches
- Cisco IC3000 Industrial Compute Gateway

Cisco Edge Intelligence also provides the flexibility to integrate with multiple applications in multiple clouds. Cisco Edge Intelligence offers native integrations for Microsoft Azure IoT Hub, AWS server, and MQTT applications.

- [Edge to multi-cloud data flows, on page 1](#)
- [Creating Cisco Edge Intelligence pipelines, on page 2](#)

## Edge to multi-cloud data flows

Cisco Edge Intelligence helps you take control of your data throughout key aspects of its lifecycle and helps you to simplify the processes from start to finish.

Figure 1: Data lifecycle



You configure the Cisco Edge Intelligence agent to address the following processes:

- **Extract:** You can automatically ingest data from any edge sensor using Cisco Edge Intelligence hosted on Cisco network equipment. Cisco Edge Intelligence has built-in industry-standard connectors such as OPC Unified Architecture (OPC-UA), Modbus (TCP/IP and Serial), and MQ Telemetry Transport (MQTT) that allow data to be extracted from various dissimilar sources. The data is then converted to industry-standard formats to enable its full use.
- **Transform:** Once the data is extracted, Cisco Edge Intelligence enables real-time processing to filter, compress, or analyze data in a uniquely simple way. You can then create code to define how the extracted data is processed. Using an in-app editor, developers can create, test, and deploy code without having to leave the Cisco Edge Intelligence portal.
- **Govern:** Cisco Edge Intelligence provides a central point for the creation and deployment of policies that govern how edge data is processed and delivered.
- **Deliver:** The extraction, transformation, and governance processes provide you with data from multiple aggregated sources to gain actionable insights for the best decision making. You can then choose what data is sent to which destination, and send the data to multiple destinations or applications.

## Creating Cisco Edge Intelligence pipelines

Creating an edge-to-multicloud data policy is a multistage process that can be completed in the Cisco Edge Intelligence Local Manager.

### Summary

In the Cisco Edge Intelligence Local Manager, you create pipelines to define the progression of data. Data management begins with an extraction of the data from different sources. The gathered data is transformed using data policy configurations and the data policies are then deployed to a wide range of destinations.

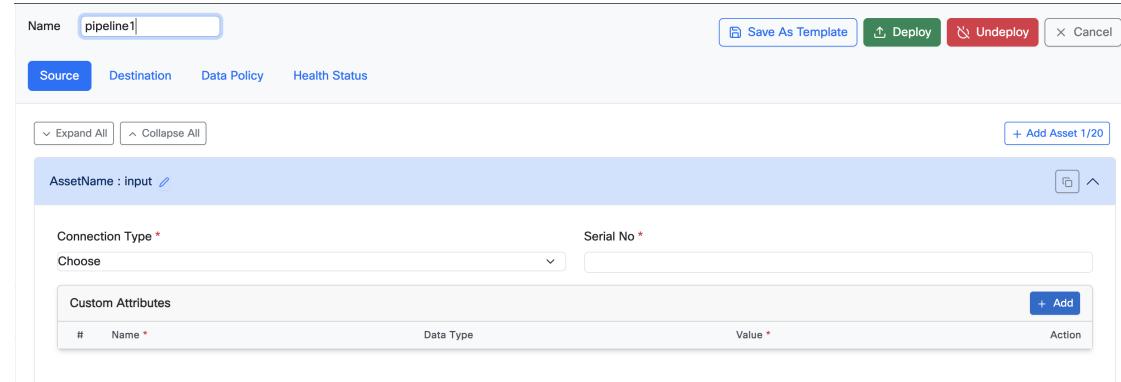
Creating a Cisco Edge Intelligence pipeline comprises the following steps:

- **Data sources:** Define assets or data source types based on the communication protocols they use. Each protocol then allows further configurations to define the data sources.
- You can add up to 20 data sources in a pipeline.
- **Add data destinations:** Add data destinations such as MQTT servers, Microsoft Azure IoT Hub, or AWS IoT Core.
- **Create a data policy:** Define a data policy to define how data is sent from data sources to destinations. A data policy can comprise one of the following methods:

- **Data logic:** Data logic involves using JavaScript, developed using an in-app code editor, to transform data before it is sent to a destination (if local processing of data is required).
- **Data rule:** Data rule allows data flow from defined sources to defined destinations, without any data transformation.

## Workflow

Figure 2: Pipeline creation page in Cisco Edge Intelligence Local Manager



The screenshot shows the 'pipeline1' configuration page. The 'Source' tab is active. The 'AssetName : input' section includes a 'Connection Type' dropdown set to 'Choose', a 'Serial No.' input field, and a 'Custom Attributes' table with one row: # (empty), Name \* (empty), Data Type (empty), Value \* (empty), and Action (empty). Buttons for 'Save As Template', 'Deploy', 'Undeploy', and 'Cancel' are at the top right.

## Manage pipelines at scale

To manage the deployment of Cisco Edge Intelligence pipelines at a scale across devices and agents, you have these options:

- Use the Cisco Edge Intelligence Release 2.0 API.
- Using CLI: See [Cisco Edge Intelligence CLI Utility Tool](#) for an example of pipeline management using CLI.





## CHAPTER 2

# Install and access Cisco Edge Intelligence agents

- Download Cisco Edge Intelligence software, on page 5
- Install the Cisco Edge Intelligence agent using IOx Local Manager, on page 5
- Install Cisco Edge Intelligence application using Cisco Catalyst SDWAN Manager, on page 13
- Remove a Cisco Edge Intelligence Agent, on page 15
- Port opening mechanism for gateways, on page 15

## Download Cisco Edge Intelligence software

Two Cisco Edge Intelligence software packages are available on [software.cisco.com](http://software.cisco.com), based on how you want to install the agent on your devices:

- Using the IOx Local Manager.
- Using the Cisco Catalyst SD-WAN Manager.

## Install the Cisco Edge Intelligence agent using IOx Local Manager

### Before you begin

[Download](#) the Cisco Edge Intelligence software package.

### Procedure

---

- Step 1** Log in to the network device Web UI.
- Step 2** From the menu, choose **Configuration > Services > IOx**.
- Step 3** Log in to the Cisco IOx Local Manager.
- Step 4** In the **Applications** tab, click **Add New**.
- Step 5** In the **Deploy Application** dialog box:

**Static NAT configuration**

- a. In the **Application Id** field, add a name for the Cisco Edge Intelligence application. This field supports only alphanumeric characters and the underscore character, and accepts up to 40 characters.
- b. Click **Choose File**, and select the downloaded software package.

After the upload is complete, the **Applications** tab displays the Cisco Edge Intelligence application listing.

**Step 6**

On the Cisco Edge Intelligence listing, click **Activate**.

**Step 7**

The **Resources** page is displayed because a peripheral configuration is required for application activation.

- a. To edit the peripheral configuration in the **Peripheral Configuration** area, click **Edit**.
- b. The following details are automatically populated:
  1. Device Type: Serial (non-editable)
  2. Name: A name is auto-populated, and you can edit the value.
  3. Label: A label is auto-populated, and you can edit the value.
- c. Click **OK**. The **Peripheral Configuration** area now displays the status **Present**.

**Note**

Make sure to check **Network** and **Resource Profile** configuration details are available under **Resources** tab.

**Step 8**

To activate the Cisco Edge Intelligence application, click **Activate App** at the top of the **Resources** page.

## Static NAT configuration

To allow inbound traffic to reach the Cisco Edge Intelligence UI or API, you must configure static NAT for a TCP service.

Configuring static NAT offers the following advantages:

- Control access to internal resources by allowing only designated services to be exposed to the outside network.
- The service is always accessible through the same public IP address and port, ensuring consistency and simplified access.

Static NAT configuration includes the following steps:

1. Get the inside IP address of the Cisco Edge Intelligence application. This is an IPv4 address.
2. Configure static NAT using the network device's GUI or CLI.

## Get Cisco Edge Intelligence (container service) IP address for NAT configuration

### Before you begin

The Cisco Edge Intelligence application must be active.

### Procedure

---

**Step 1** From the device GUI menu, choose **Configuration > Services > IOx**.

**Step 2** In the **Applications** page, on the Cisco Edge Intelligence Local Manager listing, click **Manage**.

**Step 3** In the **App-info** tab, in the **Network information** area, click **eth0**.

**Step 4** The details of the interface configuration are displayed, including the IPv4 address. Copy the IPv4 address for static NAT configuration.

---

## Configure static NAT using device GUI

### Procedure

---

**Step 1** From the menu, choose **Configuration > Security > NAT**.

**Step 2** Click **Add**.

**Step 3** From the **Static Mode** drop-down menu, choose **TCP**.

**Step 4** For **NAT direction**, choose **Inside**.

**Step 5** Enter the local IP.

**Step 6** In the **Local Port** field, enter **8008**.

**Step 7** In the **Global IP** field, enter the external IP address that you want to use.

**Step 8** In the **Global Port** field, enter **8008**.

**Step 9** Click **Apply to Device**.

---

## Configure static NAT using device CLI

To configure static NAT for a TCP service, use the following command:

```
ip nat inside source static tcp inside-local-ip-address inside-port-number
inside-global-ip-address outside-port-number extendable
```

The components of the command are:

- *inside-local-ip-address*: IPv4 address of the Cisco Edge Intelligence application.
- *inside-port-number*: Cisco Edge Intelligence application uses port 8008.
- *inside-global-ip-address*: Translated (public or external-facing) IP address.
- *outside-port-number*: The external port that maps to internal service.
- **extendable**: Defines that multiple NAT entries can be created for same internal IP.

## Procedure

---

**Step 1** Define an interface with an IP address and as a NAT inside interface, using the **ip nat inside** command.

```
Router#configure terminal
Router(config)#eth0 <number>
Router(config-if)#ip address <inside-local-IP-address>
Router(config-if)#ip nat inside
Router(config-if)#exit
```

**Step 2** Define an interface with an IP address and as a NAT outside interface, using the **ip nat outside** command.

```
Router#configure terminal
Router(config)##eth0 <number>
Router(config-if)#ip address <outside-global-IP-address>
Router(config-if)#ip nat outside
Router(config-if)#exit
```

**Step 3** Configure static NAT for a TCP service with the following command.

```
Router#configure terminal
Router(config)#ip nat inside source static tcp <inside-local-ip-address> <inside-port-number>
<inside-global-ip-address> <outside-port-number> extendable
```

**Note**

Make sure to enter `enable` command before you configure terminal.

---

## Secure remote access to assets using Cisco Secure Endpoint Access

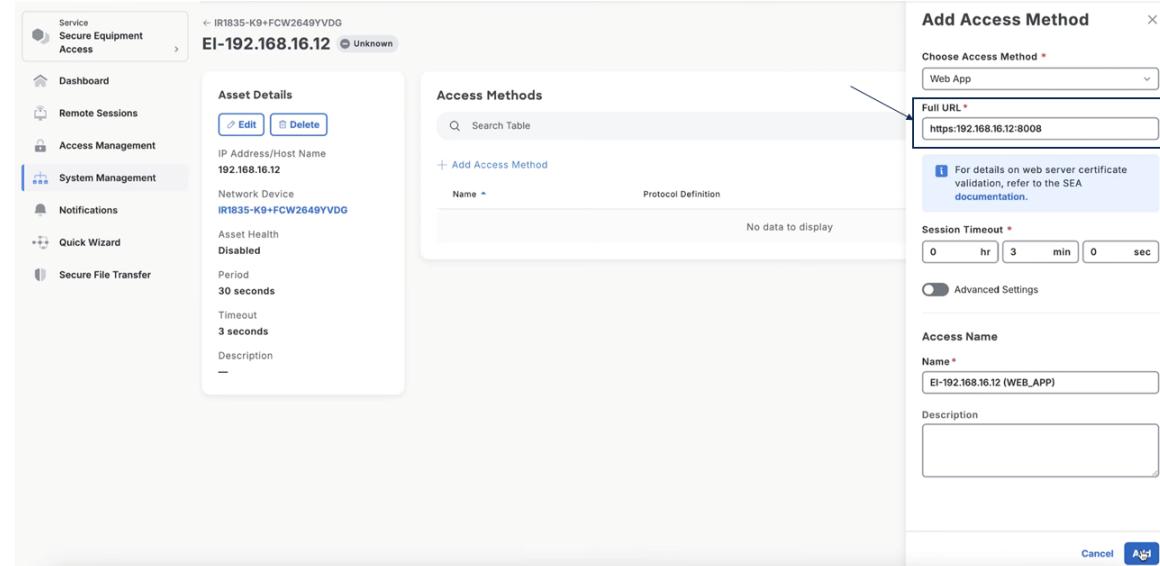
With Secure Equipment Access (SEA), Cisco is solving the challenges of deploying secure remote access to operational assets at scale. It embeds the Zero Trust Network Access (ZTNA) gateway function into Cisco industrial switches and routers, making secure remote access capabilities very simple to deploy at scale.

Cisco Secure Equipment Access comes with a cloud portal that centralizes gateway management and configuration of remote access policies.

To know more about how to get access to a remote session, see the [Request access to a remote session](#).

**Note**

While creating **Access Method**, make sure to enter the appropriate IP address with 8008-port number on **Full URL** field.



## Log in to Cisco Edge Intelligence Local Manager

### Procedure

**Step 1** From the network device menu, choose **Configuration > Services > IOx**.

**Step 2** On the Cisco Edge Intelligence application listing, click **Start**.

**Note**

Make sure that all field values are available and highlighted with green.

**Step 3** From a different browser, launch the Cisco Edge Intelligence Local Manager, using port 8008. The login url is <https://<device ip>:8008/login>

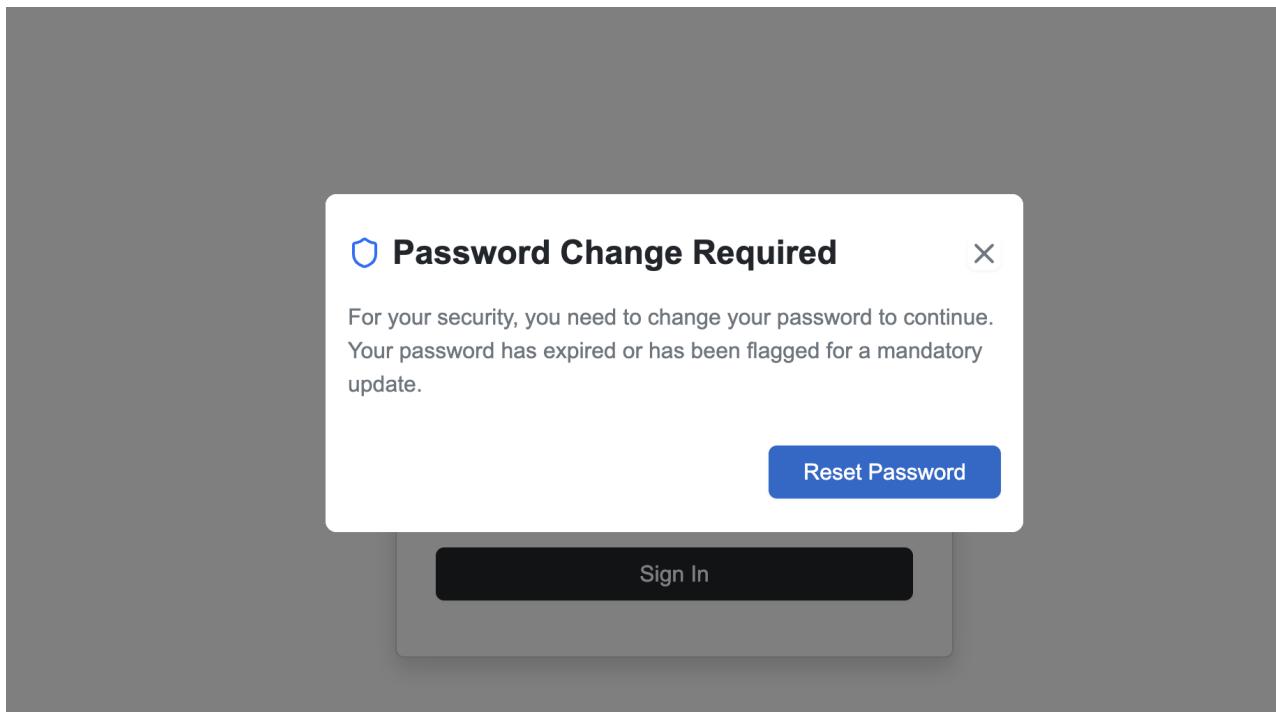
**Step 4** At first login, the login credentials are:

- Username: admin
- Password: eiUser@123

**Note**

All Cisco Edge Intelligence login passwords, default or otherwise, expire in 30 days and must be updated.

**Step 5** After you log in, you are prompted to immediately reset your password.



**Step 6** You must reset your password at first login for security.



Edge Intelligence

Protect Your Account: Update Your Password

Username

Current Password

New Password

Confirm Password

**Change Password**

Password Policy:

- Password Maximum Length: 32 Characters
- Password Minimum Length: 8 Characters
- Password history size: 4
- Max unsuccessful login attempts: 5
- Password Expiry Interval (days): 90
- Password cannot be same as username or reverse of username
- Password cannot contain cisco or reverse of cisco
- No Character can be repeated more than 3 times consecutively in the password
- Must contain at least one character from all the characters sets: upper-case, lower-case, digits, special Characters

## Reset password

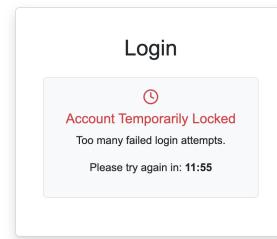
You can reset your password for any security reasons or reset it if you forgot it.

If you do multiple wrong attempts or forgot the password, then you can proceed to reset the password to the default password.

**Note**

If you do 5 consecutive attempts, then the account will be locked temporarily.

For every unsuccessful attempt, the account locks for some time duration. After that time the login window reappears.



## Procedure

**Step 1** To reset the password, connect to the IOx application session using the application ID.

**Step 2** Navigate to the Local Manager configuration directory.

**Step 3** Reset the credentials by replacing the current credentials file with the factory default version.

```
app-hosting connect appid <appid> session
cd /opt/cisco/config/dslinks/go-localmgr
cp credentials.factory.json credentials.json
pkill localmgr
```

**Step 4** Terminate the Local Manager process to force it to restart with the default credentials.

**Step 5** Password is reset to the default password.

**Note**

For a default password and to set up a new password, refer to the chapter [Log in to Cisco Edge Intelligence Local Manager, on page 9](#).

# Install Cisco Edge Intelligence application using Cisco Catalyst SDWAN Manager

Cisco Edge Intelligence is enabled by installing the EI Agent software on your Cisco network devices. The EI Agent is a Cisco IOx app that runs on Cisco network devices such as IR829, IR1101, IR1800, IE3400, and IC3000.

## Get the Cisco Edge Intelligence image

### Procedure

---

**Step 1** From Cisco Software, download the Cisco Edge Intelligence image file for SDWAN-managed devices.

**Step 2** Upload the image file to a remote file server.

---

## Cisco Catalyst SDWAN Manager process overview

Here, we assume your familiarity with Cisco Catalyst SDWAN Manager. The Cisco Edge Intelligence application is installed on network devices as a custom application using configuration groups.

For information on onboarding network devices and Day 0 configurations, see the [Cisco Catalyst SD-WAN Getting Started Guide](#).



---

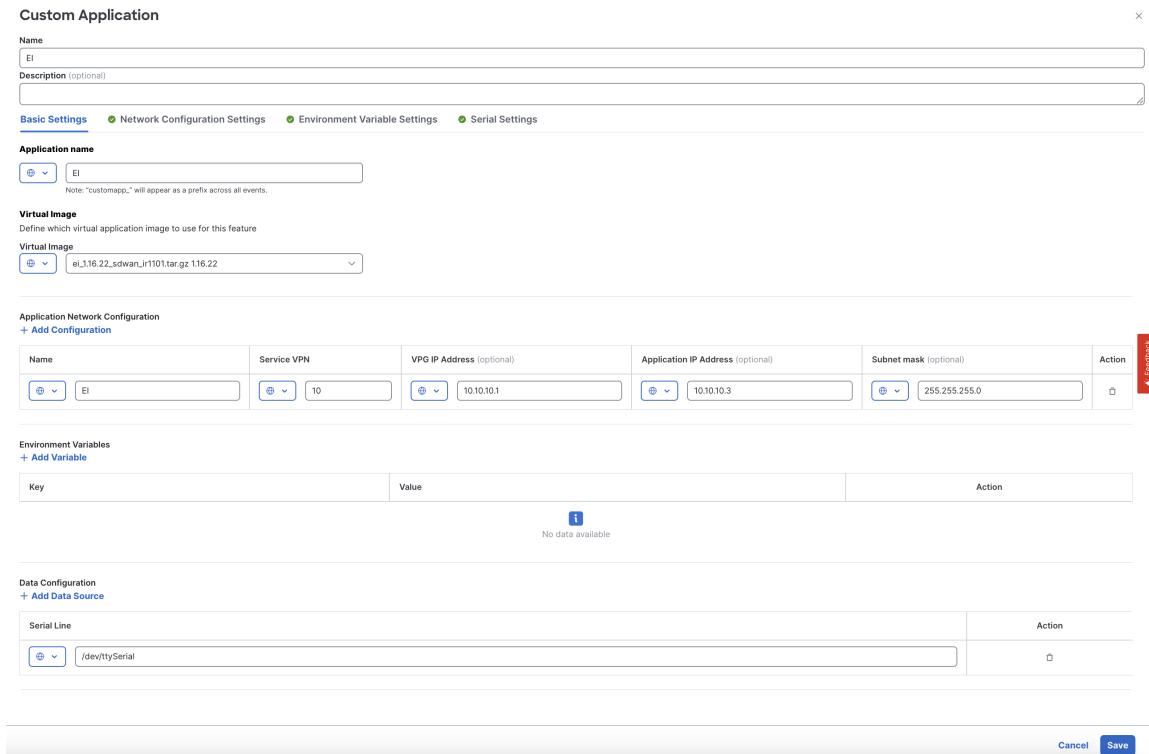
**Note** Ensure that the Cisco Edge Intelligence application and the assets that the application must reach are in the same VPN.

---

[Task 1: Register the remote file server](#)

[Task 2: Add a software image to the repository using the remote server method](#)

[Task 3: Add a custom application profile to a configuration group](#)



1. In Network Configuration Settings, define the **VPG IP Address** and **Application IP Address** as device-specific configurations instead of global configurations.
2. Configure serial interface. Here's an example of a serial interface configuration:

```

iox
!
!
interface Async0/2/0
no shutdown
encapsulation relay-line
vrf forwarding 10
!
relay line 0/2/0 0/0/0
!
!
ip http authentication local
ip http server
!

```

#### Task 4: Deploy a configuration group with a custom application



##### Note

To access the Cisco Edge Intelligence GUI, you must use the application IP address.

After you deploy Cisco Edge Intelligence as a custom application to the target network devices, use the url: <Application IP address>:8008/login.

# Remove a Cisco Edge Intelligence Agent



**Note** Removing a Cisco Edge Intelligence Agent deletes all the existing data that are related to the Cisco Edge Intelligence Agent and cannot be undone.

You might want to remove a Cisco Edge Intelligence Agent that was used for testing, or to decommission a running instance. A Cisco Edge Intelligence Agent can be removed from the system only when its status is **Not Reachable**.

Remove the Cisco Edge Intelligence Agent from the network device. When the decommissioned Cisco Edge Intelligence Agent's status is updated to **Not Reachable**.

You can also disconnect the network device from the network by choosing the IOx Local Manager and clicking **Delete**.

## Port opening mechanism for gateways

Certain NTCIP and RSU functions require for the opening of specific ports.

- Port 5001 is designated for RSU
- Port 5002 for NTCIP (streaming modes like J2735 or trafficware)
- Port 1162 for trap notifications (For example, wrong way detection)

### Procedure

**Step 1** Use the console access to open the ports for IR1101.

**Step 2** Connect to the device via **ssh** or **telnet**.

**Step 3** Use the command **show app-hosting detail** to find the internal IP of the device:

Network interfaces

```
eth0:  
  MAC address      : 52:54:dd:81:2e:49  
  IPv4 address     : 172.16.16.3 - Internal IP  
  IPv6 address     : ::  
  Network name     : VPG0  
  Multicast        : No  
  Mirroring        : No
```

**Step 4** Use the command **config t**

**Step 5** Open a required port with below given command:

```
ip nat inside source static udp 192.168.1.100 5002 interface GigabitEthernet0/1 5002
```

where 5002 - can be any port which you must open;

- Internal IP - you can find using the command **command show app-hosting detail**
- Global IP - IP of gateway (IOx Local Manager access IP)

**Note**

Above command is used for dynamic global IP. If you required a static global IP, use the command

```
ip nat inside source static udp 192.168.1.100 5001 interface GigabitEthernet0/1 5001
```

**Step 6** Use the command **exit**

**Step 7** Verify that new rules are added with **show ip nat translations** command.

**Step 8** Open ports for other gateways with IOx Local Manager.

**Step 9** Connect to IOx Local Manager.

**Step 10** Click **Activate**.

**Step 11** Select the network which configuration you want to change, then click **edit** and it navigates you to **Port Mapping** page.

**Step 12** Add all required **TCP Port Mappings** and **UDP Port Mappings** and click **OK**.

**Step 13** Click **OK** on main page.

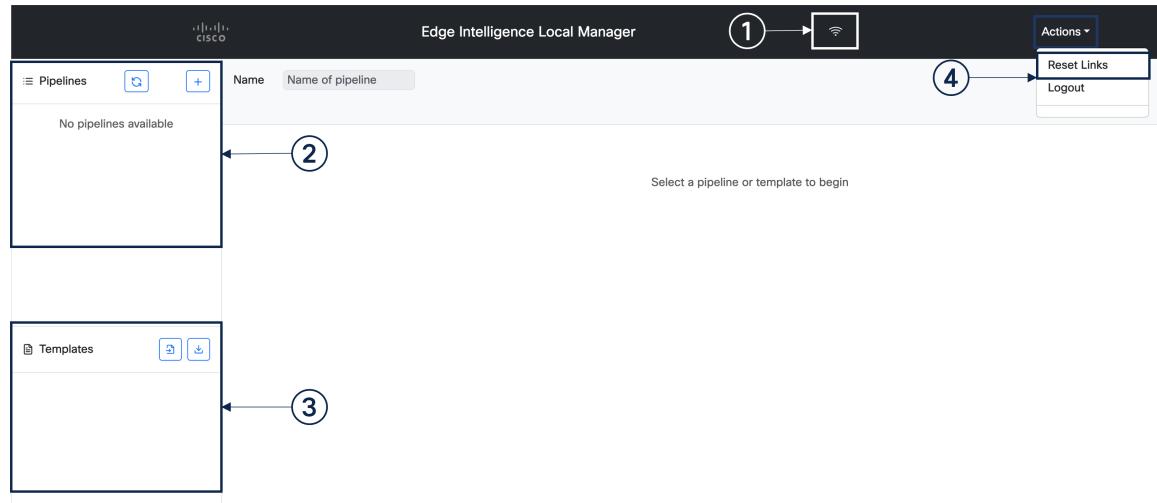
---



## Cisco Edge Intelligence Local Manager

In Cisco Edge Intelligence Local Manager dashboard, we create pipelines to define the data progression. Along with pipeline creation, the following features are available in the dashboard.

**Figure 3: Cisco Edge Intelligence Local Manager dashboard**



**1. Check Cisco Edge Intelligence connection status:**

Hover over the network connection icon (Wi-Fi icon) in the top banner to view whether Cisco Edge Intelligence is online or offline, and for agent details such as version and ID.

**2. Create and view deployed pipelines:**

- The pipelines area provides a quick view of the deployed pipelines and their respective statuses.
- Click an existing pipeline to view its health status details, or to edit the pipeline's configurations.
- To create a new pipeline, click the plus (+) icon.

**3. View, import, or export templates:**

The templates area displays all the templates that are available in the Cisco Edge Intelligence Local Manager.

- Click the import icon to upload template files from your local system.

- Click the download icon to download one or all the templates to your local system, in JSON format. You can then import the templates into other agents for deployment.

#### 4. Delete all pipelines:

To delete all the pipelines in your Cisco Edge Intelligence Local Manager, choose **Actions > Reset Links** from the top banner.



**Caution** Once deleted, a pipeline cannot be retrieved.

Templates are browser-specific and access is restricted to your user credentials. However, deployed pipelines do not have similar restrictions. Multiple users can check the health status of a pipeline and edit any existing pipeline configurations.

- [How to create pipelines, on page 18](#)
- [Data sources, on page 19](#)
- [Add data destinations, on page 40](#)
- [About Data Policies, on page 47](#)
- [Deploy or undeploy pipelines, on page 51](#)
- [View health status, on page 52](#)

## How to create pipelines

Creating a Cisco Edge Intelligence pipeline involves defining:

- One/multiple source asset type
- One data destination
- A data transformation method, in the form of a data rule or a data logic

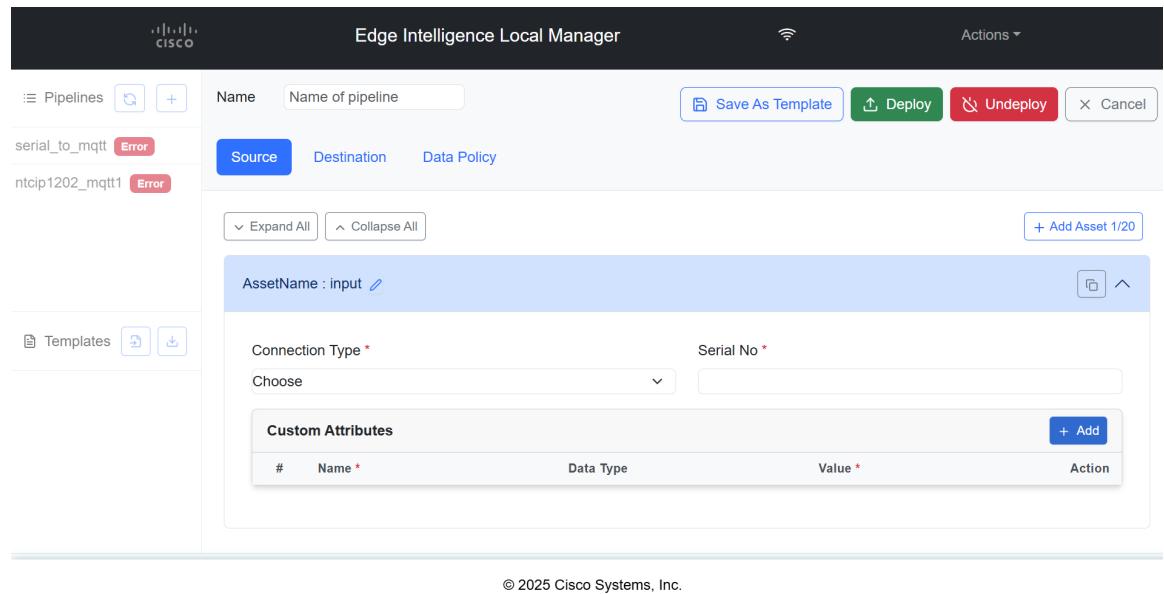
To create a pipeline in the Cisco Edge Intelligence Local Manager GUI, in the **Pipelines** area of the left pane, click the plus (+) sign button. At the top of the page, enter a name for the pipeline. Ensure that each pipeline name for the Cisco Edge Intelligence must be unique, in compliance with the following recommendations.

- Do not use special characters for a pipeline name, as special characters are removed during internal processing. We recommend using the CamelCase naming convention.

For example, use `WaterSensorSalinityJ2345` instead of `Water Sensor-Salinity_J2345`.

- Avoid ending a pipeline name with the letter `s` to indicate a plural form, as this name can cause internal naming conflicts.

For example, use `ColdStoragePlc` instead of `ColdStoragePlcs`.



© 2025 Cisco Systems, Inc.

## Data sources

Define assets or data source types based on the communication protocols they use. Each protocol then allows further configurations to define the data sources.

In the **Source** tab, these fields are the required for all type of connection types.

Field	Description
Asset Name	Click the pencil icon to enter a name for the asset. This name gives a significant value to the data logic.
Connection Type	From the drop-down list, choose one of the following protocols: <ul style="list-style-type: none"> <li>MQTT</li> <li>Modbus – Serial</li> <li>Modbus – TCP/IP</li> <li>OPC-UA</li> <li>Serial</li> <li>RSU</li> <li>NTCIP1202</li> <li>NTCIP1203</li> <li>NTCIP1204</li> </ul>
Serial Number	Enter a serial number for the chosen source type.

Field	Description
Custom Attribute Configuration	<p>You can add custom attributes along with each asset type-specific attribute. Add the following details to configure the custom attribute:</p> <ol style="list-style-type: none"> <li>1. Name: Enter a name for the custom attribute.</li> <li>2. Data Type: From the drop-down list, select a data type, string, double, encrypted string, or file.</li> <li>3. Value: Enter a custom attribute value.</li> </ol> <p>If the Data Type is selected as File:</p> <ul style="list-style-type: none"> <li>• There can be only one attribute type File for a given asset type.</li> <li>• The max size of the uploaded file should be 12 KB.</li> <li>• The file can be of any type - ASCII or binary.</li> <li>• If the asset is part of a data rule policy and the destination is configured to send the custom attribute to the northbound destination, the base64 equivalent of the file contents will be sent.</li> <li>• If the asset is part of a data logic policy, a custom attribute value is available in the data logic script as a byte array (UInt8Array), and it can be converted to the original format for access.</li> <li>• For example, if the custom attribute "reference_data" was of File type and the uploaded file was as ASCII file, the following code shows how to convert the value available as UInt8Array into ASCII text.</li> </ul> <pre>function on_update() {     ...     var file_contents = new     TextDecoder().decode(input.reference_data);     // Converts UInt8Array to     ASCII     ... }</pre> <p>Click <b>Add</b> to add more custom attributes to any asset configuration</p>



**Note** Cisco Edge Intelligence Local Manager have the provision to configure the multiple assets under a single pipeline. You can configure upto 20-assets type details and run the deployment.

If you want to duplicate an existing asset details with a different serial number, click copy-paste icon next to the asset name.

## Configure MQTT asset type

The MQTT asset type enables and configures the Cisco Edge Intelligence MQTT Server. You can publish data to the EI Agent from an MQTT client on the following ports after deploying this to an EI Agent:

- Port 8883 for TLS configurations
- Port 1883 for non-TLS configurations

In the **Source** tab, from the **Connection Type** drop-down list, choose **MQTT**. Then, fill out the following fields.

Field	Description
<b>Configuration Details</b>	
<b>Enable TLS</b>	Choose this checkbox to enable TLS. When you choose to enable TLS, additional fields are displayed to allow the upload of certificate and private key files. The MQTT server that is deployed on the EI agent, uses these certificates and private keys to authenticate the MQTT clients connecting to it.
<b>Username</b>	Enter a username. The connecting MQTT Clients (sensors) will use this username for authentication at the MQTT Server which will be spawned on the EI Agent.

Field	Description
<b>Password</b>	Enter the Password. The connecting MQTT Clients (sensors) will use this password for authentication at the MQTT Server which will be spawned on the EI Agent.
<b>Advanced Settings</b>	
<b>Client ID</b>	<p>Add a client ID to publish data. (Topic-based device or sensor identity detection is not supported).</p> <p>The client ID, specified in the MQTT client connection, differentiates various MQTT connections to Cisco Edge Intelligence.</p>
<b>Attribute Definitions</b>	<p>The data model explains how data is represented in the asset, and what MQTT topics the asset should be updated from. To define a data model, add the following details:</p> <ol style="list-style-type: none"> <li><b>Name:</b> Enter a name for the data model</li> <li><b>Label:</b> Enter a label for the data model</li> <li><b>Data Type:</b> From the drop-down menu, select a measurement entity (string, integer, float, long, or boolean) for the MQTT topic.</li> <li><b>Topic:</b> Enter the MQTT topic over which the measurement is sent by the transmitting MQTT client. For example, <code>sensors/tempXY/temp</code>.</li> </ol> <p>Click <b>Add</b> to add more data model attributes to the configuration</p>



**Note** The MQTT topic used to publish sensor data and the data format should match the **data model** JSON file.

#### MQTT data model example:

```
{
  "apiVersion": 1.0,
  "connectionType": "MQTT",
  "fields": {
    "temperature": {
      "category": "TELEMETRY",
      "label": "Temperature",
      "description": "Outside temperature sensor XY | Temperature",
      "datatype": "Float",
      "topic": "sensors/tempXY/temp"
    },
    "humidity": {
      "category": "TELEMETRY",
      "label": "Humidity",
      "description": "Outside temperature sensor XY | Humidity",
      "datatype": "Float"
    }
  }
}
```

```
        "datatype": "Float",
        "topic": "sensors/tempXY/hum"
    },
    "attr1": {
        "category": "ATTRIBUTE",
        "label": "Attribute 1",
        "datatype": "Float",
        "description": "My Attribute 1",
        "required": true,
        "defaultValue": 12.9
    },
    "attr2": {
        "category": "ATTRIBUTE",
        "label": "Attribute 2",
        "datatype": "String",
        "description": "My Attribute 2",
        "required": false,
        "defaultValue": null
    },
    "encrypted_attr3": {
        "category": "ATTRIBUTE",
        "label": "New Attribute",
        "datatype": "EncryptedString",
        "description": "My Attribute 3",
        "required": true,
        "defaultValue": null
    }
}
}
```

#### MQTT topic and sensor data used by MQTT Client example:

MQTT Client Topic: sensors/tempXY/hum  
MQTT Client Data: 50.0

## Configure Modbus-Serial asset type

From the **Device Type** drop-down list, choose **MODBUS-Serial**. Then enter the required details in the following fields.

## Configure Modbus-Serial asset type

Connection Type *	Serial No *																				
Modbus - Serial																					
Configuration Details																					
Transport *	Serial Port * 																				
RTU	Parity																				
Baud Rate *	None																				
9600	Data Bits																				
Stop Bits	8																				
<input checked="" type="radio"/> 1 <input type="radio"/> 2																					
Slave ID *																					
1																					
Advanced Settings																					
Zero On Failed Poll	Use Batch Polling																				
<input type="radio"/> True <input checked="" type="radio"/> False	<input checked="" type="radio"/> True <input type="radio"/> False																				
Contiguous Batch Request Only	Use Multiple Write Commands																				
<input checked="" type="radio"/> True <input type="radio"/> False	As Appropriate																				
Timeout (In milliseconds) *	Retries																				
500	2																				
Max Read Bit Count	Max Read Register Count																				
2000	125																				
Max Write Register Count																					
120																					
Attribute Definitions																					
<table border="1"> <thead> <tr> <th>#</th> <th>Name *</th> <th>Label *</th> <th>Data Type *</th> <th>Polling Interval (ms) *</th> <th>Type *</th> <th>Offset *</th> <th>RawType</th> <th>Access Mode</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>1</td> <td></td> <td></td> <td>String</td> <td></td> <td>COIL</td> <td></td> <td>UINT16</td> <td>Read Only</td> <td></td> </tr> </tbody> </table>		#	Name *	Label *	Data Type *	Polling Interval (ms) *	Type *	Offset *	RawType	Access Mode	Action	1			String		COIL		UINT16	Read Only	
#	Name *	Label *	Data Type *	Polling Interval (ms) *	Type *	Offset *	RawType	Access Mode	Action												
1			String		COIL		UINT16	Read Only													
<a href="#">+ Add</a>																					

## Field

## Description

## Configuration Details

<b>Transport</b>	This field is not editable. It has a default value set for RTU.
<b>Serial Port</b>	Enter a port number. For example, /dev/ttyS0.
<b>Baud Rate</b>	Enter a baud rate.
<b>Parity</b>	From the drop-down list, choose <b>None</b> , <b>Odd</b> , or <b>Even</b> .
<b>Stop Bits</b>	This value can be either 1 or 2. Choose the required radio button.
<b>Data Bits</b>	From the drop-down list, choose a value 5–9.
<b>Slave ID</b>	This refers to the unique identifier assigned to a Modbus secondary device (such as a sensor or PLC) on a serial network. Modbus protocol requires each secondary device to have a unique ID (1–247) to distinguish it on a shared bus. This ID ensures that the Modbus primary (such as the Cisco EI agent) sends requests to the correct device and processes its responses.

## Advanced Settings

<b>Zero or Failed Poll</b>	The value if there is no response from the asset. This field is not editable.
<b>Use Batch Polling</b>	To request batch responses from the asset. This field is not editable.

Field	Description
<b>Contiguous Batch Request Only</b>	To request contiguous batch responses from the asset. This field is not editable.
<b>Use Multiple Write Commands</b>	This field is not editable.
<b>Timeout</b>	The time in which to receive the data before it is reset. The default value is 500.
<b>Retries</b>	The number of times the server requests for retransmission of data. This field is not editable.
<b>Max Read Bit Count</b>	The maximum number of bits that the server reads in one read request. This field is not editable.
<b>Max Read Register Count</b>	The maximum number of registers that the server reads in one read request. This field is not editable.
<b>Max Write Register Count</b>	The maximum number of registers that the server writes in one write request. This field is not editable.
<b>Attribute Definitions</b>	<p>A data model explains how data is represented in the asset. You can create a data model using a JSON file. Select one of the following:</p> <ol style="list-style-type: none"> <li><b>Name:</b> Enter a name for the data model.</li> <li><b>Label:</b> Enter a label value.</li> <li><b>Data Type:</b> From the drop-down list, select a measurement entity (string, integer, float, long, or boolean) for the MODBUS serial data.</li> <li><b>Polling Interval (ms):</b> Enter a value to define the fastest rate at which the server must test and debug.</li> <li><b>Type:</b> From the drop-down list, choose a data type for industrial control of factory devices. The available choices are <b>COIL</b>, <b>HOLDING</b>, <b>DISCRETE</b>, and <b>INPUT</b>.</li> <li><b>Offset:</b> Enter a value to reference a specific register within the function.</li> <li><b>RawType:</b> From the drop-down list, choose the value type for storing binary data or byte strings.</li> <li><b>Access Mode:</b> From the drop-down list, choose an access mode. The available choices are <b>Read Only</b>, <b>Read &amp; Write</b>, and <b>Write Only</b>.</li> </ol> <p>Click <b>Add</b> to add more data model attributes to the configuration</p>

**Modbus\_serial data model example:**

## Adding Asset Types for Modbus TCP Connection Type

```
{
  "apiVersion": 1,
  "connectionType": "MODBUS_SERIAL",
  "fields": {
    "pressureInPascal": {
      "label": "MetrLabeModb1",
      "pollingInterval": 50,
      "offset": 12,
      "type": "HOLDING",
      "datatype": "Float",
      "rawType": "VARCHARSTRING",
      "description": "",
      "access": "Write"
    },
    "TemperatureInDegrees": {
      "label": "Temperature",
      "pollingInterval": 41,
      "offset": 56,
      "type": "DISCRETE",
      "datatype": "String",
      "rawType": "INT16",
      "description": "",
      "access": "ReadWrite"
    },
    "HumidityInDegrees": {
      "label": "Humidity",
      "pollingInterval": 20,
      "offset": 45,
      "type": "COIL",
      "datatype": "Int",
      "rawType": "FLOAT64",
      "description": "Humidity Value",
      "access": "Read"
    }
  }
}
```

## Adding Asset Types for Modbus TCP Connection Type

After you select the MODBUS-TCP/IP, complete the following additional fields by adding the details on it.

The screenshot shows the configuration interface for a Modbus TCP connection. The 'Connection Type' is set to 'Modbus - TCP/IP'. The 'Advanced Settings' tab is active, displaying various configuration options:

- Zero On Failed Poll:** True (radio button selected)
- Contiguous Batch Request Only:** True (radio button selected)
- Timeout (In milliseconds):** 500
- Max Read Bit Count:** 2000
- Max Write Register Count:** 120
- Use Batch Polling:** True (radio button selected)
- Use Multiple Write Commands:** As Appropriate
- Retries:** 2
- Max Read Register Count:** 125

The 'Attribute Definitions' table is also visible, showing a single row with the following values:

#	Name *	Label *	Data Type	Polling Interval (ms) *	Type	Offset *	RawType	Access Mode	Action
1			String		COIL		UINT16	Read Only	

Field	Description
<b>Configuration Details</b>	
<b>IP Address or Host Name</b>	Enter an IP address or hostname.
<b>Port</b>	Enter a port number.
<b>Slave ID</b>	This refers to the unique identifier assigned to a Modbus secondary device (such as a sensor or PLC) on a serial network. Modbus protocol requires each secondary device to have a unique ID (1–247) to distinguish it on a shared bus. This ID ensures that the Modbus primary (such as the Cisco EI agent) sends requests to the correct device and processes its responses.
<b>Advanced Settings</b>	
<b>Zero on Failed Poll</b>	The value if there is no response from the asset. This field is non-editable.
<b>Use Batch Polling</b>	To request batch responses from the asset. This field is non-editable.
<b>Contiguous Batch Request Only</b>	To request contiguous batch responses from the asset. This field is non-editable.
<b>Use Multiple Write Commands</b>	This field is non-editable.
<b>Timeout</b>	The time set to receive the data before it is reset. The default value is 500.
<b>Retries</b>	The number of times the server requests for retransmission of data. This field is non-editable.
<b>Max Read Bit Count</b>	The maximum number of bits that the server reads in one read request. This field is non-editable.
<b>Max Read Register Count</b>	The maximum number of registers that the server reads in one read request. This field is non-editable.
<b>Max Write Register Count</b>	The maximum number of registers that the server writes in one write request. This field is non-editable.

Field	Description
<b>Attribute Definitions</b>	<p>A data model explains how data is represented in the asset. You can create a data model using a JSON file. Enter the following details:</p> <ol style="list-style-type: none"> <li><b>Name:</b> Enter a name for the data model.</li> <li><b>Label:</b> Enter a label value.</li> <li><b>Data Type:</b> From the drop-down menu, select a measurement entity (string, integer, float, long, or boolean) for the MODBUS-TCP/IP data.</li> <li><b>Polling Interval:</b> Enter a value to define the fastest rate at which the server should test and debug.</li> <li><b>Type:</b> From the drop-down menu, choose a data type for industrial control of factory devices. The available choices are <b>COIL</b>, <b>HOLDING</b>, <b>DISCRETE</b>, and <b>INPUT</b>.</li> <li><b>Offset:</b> Enter a value to reference a specific register within the function.</li> <li><b>RawType:</b> From the drop-down menu, choose the value type for storing binary data or byte strings.</li> <li><b>Access Mode:</b> From the drop-down list, choose an access mode. The available choices are <b>Read Only</b>, <b>Read &amp; Write</b>, and <b>Write Only</b>.</li> </ol> <p>Click <b>Add</b> to add more data model attributes to the configuration</p>

The following is an example of a data model for Modbus-TCP/IP source type.

```
{
  "apiVersion": 1,
  "connectionType": "MODBUS_TCP",
  "fields": {
    "desired_temp": {
      "label": "Desired Temperature",
      "datatype": "Int",
      "description": "WO",
      "rawType": "UINT16",
      "type": "HOLDING",
      "pollingInterval": 5000,
      "offset": 5,
      "category": "TELEMETRY",
      "access": "Write"
    },
    "temp_to_display": {
      "label": "Temperature to be displayed",
      "datatype": "Int",
      "description": "RW",
      "rawType": "UINT16",
      "type": "HOLDING"
    }
  }
}
```

```
        "type": "HOLDING",
        "pollingInterval": 5000,
        "offset": 100,
        "category": "TELEMETRY",
        "access": "ReadWrite"
    },
    "temp": {
        "label": "Current Temperature",
        "datatype": "Int",
        "description": "RO",
        "rawType": "UINT16",
        "type": "HOLDING",
        "pollingInterval": 5000,
        "offset": 1,
        "category": "TELEMETRY",
        "access": "Read"
    }
}
```

## Configure OPC-UA asset type

After you select the asset type OPC-UA, enter the required details in the following fields:

Field	Description
<b>Configuration Details</b>	
<b>IP Address or Host Name</b>	Enter an IP address or hostname.
<b>Port</b>	Enter a port number.

Field	Description
<b>Publishing Interval</b>	<p>This is the requested publishing frequency from the OPC-UA Server. The interval must be greater than or equal to 1000ms.</p> <p><b>Note</b> The OPC-UA server publishing frequency is independent of the metric-specific sample interval. In case the sampling interval (in the following attribute table) of an individual metric is smaller than the publishing interval, the OPC-UA server queues up and send all the sampled values for a metric between the last publish and the current publish.</p>
<b>Advanced Settings</b>	
<b>Authentication</b>	<p>Choose an authentication type:</p> <ul style="list-style-type: none"> <li>• <b>Anonymous:</b> The OPC-UA client inside the EI Agent does not authenticate at the OPC-UA server. Use this authentication type if your OPC-UA server does not have authentication that is enabled for connecting clients.</li> <li>• <b>Username &amp; Password:</b> Enter the username and password that the EI Agent must use to authenticate at the OPC-UA server.</li> </ul>
<b>Security Mode</b>	This field is non-editable. The default value is <b>None</b> .

Field	Description
<b>Attribute Definitions</b> <p>A data model explains how data is represented in the asset.</p> <ol style="list-style-type: none"> <li><b>Name:</b> Enter a name for the data model.</li> <li><b>Label:</b> Enter a label for the data model.</li> <li><b>Data Type:</b> From the drop-down menu, select a measurement entity (string, integer, float, long, or boolean) for the OPC-UA data.</li> <li><b>OPC_UA Type:</b> From the drop-down menu, choose <b>URI</b> or <b>INDEX</b>.</li> <li><b>NameSpace URI:</b> Enter a value to identify the naming authority that defines the identifiers of Node IDs.</li> </ol> <p><b>Note</b> This field is editable if you select <b>URI</b> under <b>OPC_UA Type</b>.</p> <ol style="list-style-type: none"> <li><b>NameSpace Index:</b> Enter a value to identify the naming authority that defines the identifiers of Node IDs.</li> </ol> <p><b>Note</b> This field is editable if you select <b>INDEX</b> under <b>OPC_UA Type</b>.</p> <ol style="list-style-type: none"> <li><b>Identifier:</b> Enter a value that is unique across different naming authorities.</li> <li><b>Sampling Interval:</b> Enter a value to indicate the fastest rate at which the server should sample its underlying source for data changes.</li> <li><b>Type:</b> From the drop-down menu, choose the data type for the identifier. The available choices are <b>Numeric</b> and <b>String</b>.</li> </ol> <p>Click <b>Add</b> to add more data model attributes to the configuration</p>	

The following is an example of a data model for OPC-UA source type.

```
{
  "apiVersion": 1,
  "connectionType": "OPC_UA",
  "fields": {
    "temperature": {
      "label": "Temperature",
      "description": "",
      "datatype": "Float",
      "nodeId": {
        "uri": "ns=1;i=1000"
      }
    }
  }
}
```

## Configure Serial asset type

```

        "namespaceUri": "2",
        "identifier": "2",
        "type": "numeric"
    },
    "samplingInterval": 1000,
    "category": "TELEMETRY"
}
}
}
}

```

## Configure Serial asset type

After you select the asset type **Serial**, enter the required details in the following fields.

Connection Type \*

Serial No \*

**Configuration Details**

Serial port * <small>①</small>	Baud Rate *
<input type="text"/>	9600 <small>▼</small>
Parity	Stop Bits
None <small>▼</small>	<input checked="" type="radio"/> 1 <input type="radio"/> 2
Data Bits	
<input type="radio"/> 7 <input checked="" type="radio"/> 8	

**Attribute Definitions** + Add

#	Name *	Label *	Data Type *	Access	Start Code	End Code	Message Size	Tin
1	<input type="text"/>	<input type="text"/>	String <small>▼</small>	READ <small>▼</small>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Field	Description
<b>Configuration Details</b>	
<b>Serial Port</b>	Enter a port number. For example, <code>/dev/ttys0</code>
<b>Baud Rate</b>	From the drop-down menu, choose 9600, 19200 or you can add a new custom value on it.
<b>Parity</b>	From the drop-down menu, choose <b>None</b> , <b>Odd</b> , or <b>Even</b>
<b>Stop Bits</b>	Choose the radio button for <b>1</b> or <b>2</b> .
<b>Data Bits</b>	From the drop-down menu, choose <b>7</b> or <b>8</b> .

Field	Description
<b>Attribute Definitions</b>	<p>A data model explains how data is represented in an asset.</p> <ol style="list-style-type: none"> <li>1. <b>Name:</b> Enter a name for the data model.</li> <li>2. <b>Label:</b> Enter a label for the data model.</li> <li>3. <b>Data Type:</b> From the drop-down menu, choose <b>String</b> or <b>Binary</b>. If you select <b>Binary</b>, the data is delivered in binary form 1.             <ol style="list-style-type: none"> <li>a. For a data logic policy, the data is delivered as a binary buffer in the data logic script in the <i>on_update()</i> function.</li> <li>b. For a data rule policy, the data is sent to the northbound destination in the base64 format.</li> </ol> </li> <li>4. <b>Access:</b> From the drop-down menu, choose read, write, or read and write.</li> <li>5. <b>Start Code:</b> Enter the marker that indicates the start of a stream of bytes.</li> <li>6. <b>End Code:</b> Enter the marker that indicates the end of a stream of bytes.</li> <li>7. <b>Message Size:</b> This is the size, in bytes, between the start code and the end code.</li> <li>8. <b>Timeout:</b> Enter a time, in milliseconds, within which to receive data, before it is reset. This is enabled only for <b>READ</b> and <b>READWRITE</b> access modes. This field is mandatory only if the message size is configured.</li> </ol>

### Common errors and troubleshooting

Errors can easily occur when configuring a serial port. For example:

- The wiring must be accurate. For example, see the [Cisco IR829 installation guide](#).
- The serial relay service should be configured correctly for the Guest OS. For example, see the [Cisco Catalyst IR1101 documentation](#) and [Cisco Catalyst IR1800 documentation](#).
- The physical serial port must be correctly exposed to IOx through the Local Manager.

To troubleshoot a serial interface:

- Make sure that serial port is configured in propagate mode at the IOS level. A current workaround is to use just the 0x prefix as the StartCode to specify an empty StartCode.
- For testing interface options only:
  - Use a data model with a fixed message size of 1 byte and no start code.

- Verify that there is some data that is coming in to ensure that the connection is working.
- Once this is done, the actual data model can be defined.

### Serial Connector data model

A serial connector asset type has a reduced data type.

The following combinations are allowed:

- One read attribute
- One read plus one write attribute
- One read-write attribute

Read attribute allows the following configuration combinations:

- Message Size and Timeout
- StartCode and Message Size and an optional Timeout
- StartCode and EndCode and an optional Timeout

Start-/End-Code prefix handling:

- Prefix 0x allows to specify hex encoded binary data. For example, 0x1310 -> CR+LF)
- **Prefix** allows to specify **as-is**

### Serial data model example:

```
{
  "apiVersion": 1,
  "connectionType": "SERIAL",
  "fields": {
    "data_string1": {
      "label": "My Data String",
      "datatype": "String",
      "description": "serial read attribute",
      "access": "READ",
      "startCode": "$",
      "endCode": "0x0a",
      "messageSize": ""
    }
  }
}
```

## RSU asset type settings

The RSU asset type supports a set of static attributes in addition to the regular configurable attributes. Static attributes are always available and not required to be configured.

- The static attributes have a JSON string content and reflect incoming DSRC messages, except storeAndRepeatMessage and broadcastImmediately static attributes.
- The storeAndRepeatMessage must be set as an array of message objects. All previous messages will be overwritten by the new array.

- The current array of messages can be obtained by reading the attribute.
- The basic configuration for this asset type includes the host, port, and SNMP version.
- Advanced settings like community or authentication data must be set depending on the SNMP version.

In the **Source** tab, from the **Device Type** drop-down list, choose **RSU**. Then, fill the following fields.

The screenshot shows the configuration interface for an RSU asset type. It includes sections for Configuration Details (IP Address or Host Name, Port, SNMP Version), Advanced Settings (Security Name, Authentication password, Privacy password), and Default Attributes (Attribute Definitions table with columns for Name, Label, Data Type, Object ID(OID), OID Data Type, Polling Interval (ms), Access Mode, and Action). The IP Address or Host Name field is set to 'RSU'.

Field	Description
<b>Configuration Details</b>	
<b>IP Address or Host Name</b>	Enter the IP address or hostname.
<b>Port</b>	Enter the port number.
<b>SNMP Version</b>	Select a version from the drop-down list from 1, 2c and 3. 3 is the most secure version.
<b>Advanced Settings</b>	Complete these fields based on the selected SNMP Version.
<b>Default Attributes</b>	It shows the default attributes that are specific to RSU. Even if you will not add any additional attributes, it runs for the agents.

Field	Description
<b>Attribute Definitions</b>	<p>A data model explains how data is represented in an asset.</p> <ol style="list-style-type: none"> <li><b>Name:</b> Enter a name for the data model.</li> <li><b>Label:</b> Enter a label for the data model.</li> <li><b>Data Type:</b> From the drop-down list, choose <b>String, Integer, Float, Long, or Boolean</b>.</li> <li><b>Object ID (OID):</b> OID address is used to uniquely identify managed devices and their statuses.</li> <li><b>OID Data Type:</b> OID is the data type for the object. <b>String</b> or <b>Integer</b>.</li> <li><b>Polling Interval (ms):</b> This indicates the fastest rate at which the Server should test and debug. Select a unit from the up-down menu.</li> <li><b>Access Mode:</b> From the drop-down list, choose read, write, or read and write.</li> </ol> <p>Click <b>Add</b> to add more attributes to the asset configuration.</p>

## NTCIP1202, NTCIP1203, NTCIP1204 asset type settings

Cisco Edge Intelligence Local Manager supports three NTCIP devices. Use the Asset Type for the correct connection type.

- NTCIP 1202—Actuate Signal Controller
- NTCIP 1203—Dynamic Message Sign
- NTCIP 1204—Road Weather Information System

The basic configuration for all these three asset types include the host, port, and SNMP version.

Advanced settings like community or authentication data must be set depending on the SNMP version.

Each NTCIP asset type supports a set of static attributes in addition to the regular configurable attributes.

The screenshot shows the configuration interface for NTCIP1202, NTCIP1203, and NTCIP1204 asset type settings. It includes sections for Configuration Details, Advanced Settings, and Attribute Definitions.

**Configuration Details:**

- Connection Type: NTCIP1202
- Serial No.: (empty)
- IP Address or Host Name: (empty)
- Port: 161
- Trap Port: 162
- SNMP Version: 3
- Enable Streaming: checked
- Standard or Asset Manufacturer: SAE J2735 Standard
- Intersection ID: 0
- Intersection Name: (empty)

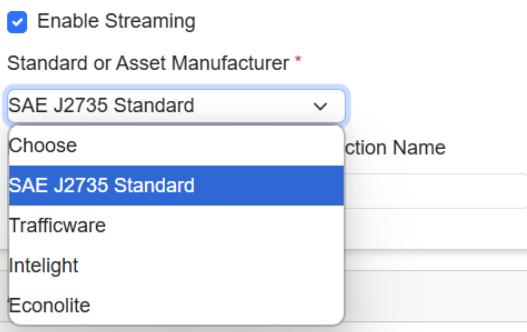
**Advanced Settings:**

- Security Name: (empty)
- Security Level: AuthPriv
- Authentication password: (empty)
- Authentication Protocol: SHA
- Privacy password: (empty)
- Privacy Protocol: AES
- OID Count per Request: 32

**Attribute Definitions:**

#	Name *	Label *	NTCIP Type *	Data Type	Object ID *	Trap OID *	OID Data Type *	Polling Interval (ms) *	Access Mode	Service	Action
1	(empty)	(empty)	Default	String	RawSpat	(empty)	Integer	(empty)	Read Only	TRAP_RECEIVE	+ Add
<input type="button" value="Add OID"/> <input type="button" value="+"/>											

Field	Description
<b>Configuration Details</b>	
IP Address or Host Name	Provide the IP address or Host Name.
SNMP Version	Select a version from the drop-down list from 1, 2c and 3. 3 is the most secure version.
Port	Provide a NTCIP connector port number. <b>Note</b> NTCIP1202 port number should never match Trap port number, and vice versa.
Trap Port (Applicable only for device type: NTCIP1202)	Provide Trap port number. <b>Note</b> Trap port number should never match NTCIP1202 port number, and vice versa.

Field	Description
<b>Enable Streaming</b> (Specific to device type: NTCIP1202)	Choose this checkbox to enable Streaming. When you choose to enable Streaming, an additional field <b>Standard or Asset Manufacturer</b> protocol is displayed. NTCIP 1202 allows the streaming of data using standard and Asset manufactured protocols. The different protocols are: Standard protocol: <ul style="list-style-type: none"> <li>SAE J2735 Standard</li> </ul> Asset manufactured protocols <ul style="list-style-type: none"> <li>Trafficware</li> <li>Intelight</li> <li>Econolite</li> </ul>
	 <p>Enable Streaming</p> <p>Standard or Asset Manufacturer *</p> <p>SAE J2735 Standard</p> <p>Choose</p> <p>SAE J2735 Standard</p> <p>Trafficware</p> <p>Intelight</p> <p>Econolite</p>
Intersection ID (Specific to device type: NTCIP1202)	Intersection ID is optional and depends on the user's specific use case. You can choose to configure this field if required for their particular scenario. For example, in traffic use cases, this information is mandatory.
Intersection Name (Specific to device type: NTCIP1202)	Intersection Name is optional and depends on the user's specific use case. You can choose to configure this field if required for their particular scenario.
<b>Advanced Settings</b>	The specific Advanced Setting details vary based on the SNMP Version. Provide the appropriate authentication information accordingly.

Field	Description
<b>Attribute Definitions</b>	<p>A data model explains how data is represented in an asset.</p> <ol style="list-style-type: none"> <li><b>Name:</b> Enter a name for the data model.</li> <li><b>Label:</b> Enter a label for the data model.</li> <li><b>NTCIP Type:</b> From the drop-down list, choose <b>Default</b> or <b>trap</b>.</li> <li><b>Data Type:</b> From the drop-down list, choose <b>String</b>, <b>Integer</b>, <b>Float</b>, <b>Long</b>, or <b>Boolean</b>.</li> <li><b>Object ID:</b> From the drop-down list, choose <b>RawSpat</b>, <b>NTCIP-1211 SRM</b>, and <b>SAE J2735 SRM</b>. Bottom to that add OID. OID address is used to uniquely identify managed devices and their statuses.</li> </ol> <p><b>Note</b> This field is applicable if you select <b>Default</b> under <b>NTCIP Type</b>.</p> <ol style="list-style-type: none"> <li><b>Trap OID:</b> Enter an ID for Trap NTCIP type.</li> </ol> <p><b>Note</b> This field is applicable if you select <b>Trap</b> under <b>NTCIP Type</b>.</p> <ol style="list-style-type: none"> <li><b>OID Data Type:</b> OID is the data type for the object. <b>String</b> or <b>Integer</b>.</li> <li><b>Polling Interval (ms):</b> It indicates the fastest rate at which the Server tests and debug. Select a unit from the up-down menu.</li> <li><b>Access Mode:</b> From the drop-down list, choose read, write, or read and write.</li> </ol> <p><b>Note</b> This field is applicable if you select <b>Default</b> under <b>NTCIP Type</b>.</p> <ol style="list-style-type: none"> <li><b>Service:</b> From the drop-down list, choose <b>TRAP_RECEIVE</b>.</li> </ol> <p><b>Note</b> This field is applicable if you select <b>Trap</b> under <b>NTCIP Type</b>.</p>

# Add data destinations

In the **Destination** tab, define where configured data policies must send data to.

The supported destinations are:

- MQTT servers
- AWS server
- Azure IoT Hub

From the **Type** drop-down list, choose the destination. Then, configure the connection settings for the chosen destination.

Before you configure a data destination in the pipeline, ensure that you have set up the servers or hubs. You must have the required identifying information ready to complete the destination configuration.

## Add a Microsoft Azure IoT Hub destination

### Procedure

In the **Destination** tab, from the **Type** drop-down list, choose **AZURE IoT**, and enter the required details in the following fields.

**Data Destination**

Type \*  
Azure IoT

**Azure IOT Connection Details**

ID Scope \*  
[Input field]

CA Certificate \*  
[Choose File button] or drag and drop your file here

CA Certificate Key \*  
[Choose File button] or drag and drop your file here

**Advanced Settings**

Device Provisioning Endpoint \*  
global.azure-devices-provisioning.net

Enable MQTT Over Websockets

Data Rule Classification \*  
Select an option

**Message Structure**

Asset Attribute

Include Asset Attributes

Group asset attributes under property

Telemetry Data

Group telemetry data under property

Include Timestamps

Example:  
{  
  "telemetryData": 22.4,  
  "telemetryData": 45.2  
}

Field	Description
<b>Azure IoT Connection Details</b>	
<b>ID Scope</b>	Enter the ID Scope that is displayed in the <b>Azure Device Provisioning Service Overview</b> page.
<b>CA Certificate</b>	<p>Upload the intermediate CA Certificate file that you have configured in your Azure Device Provisioning Service.</p> <p><b>Note</b> The status of this CA certificate in the Azure Provisioning Service must be marked as trusted, or device creation (and therefore metric sending) is not allowed by Azure.</p>
<b>CA Certificate Key</b>	<p>Upload the unencrypted private certificate key file that belongs to the intermediate CA certificate that you uploaded. The private key must be in PKCS8 format, and must not include a passphrase.</p> <p>To convert an existing key, use the command: <code>openssl pkcs8 -topk8 -inform PEM -outform PEM -nocrypt -in azure-iot-test-only.intermediate.key.pem -out azure-iot-test-only.intermediate.pkcs8.pem</code></p>
<b>Advanced Settings</b>	
<b>Device Provisioning Endpoint</b>	Enter the <b>Global Device Endpoint</b> value from the Azure IoT Hub Device Provisioning Service.
<b>Enable MQTT over WebSockets</b>	Check this check box to enable the browser to leverage all MQTT features.
<b>Data Rule Classification</b>	From the drop-down list, choose <b>DEVICE_PROPERTY</b> or <b>TELEMETRY</b> .
<b>Message Structure</b>	<p>Choose the required options in this area to customize the structure of the device-to-cloud message. You can choose <b>Include asset attributes</b>, <b>Group asset attributes under property</b> in the Asset Attributes section or choose <b>Group telemetry data under property</b>, <b>Include Timestamps</b> in the Telemetry Data section.</p> <p>The data can be sent in a flat structure or can be grouped with a key. Customization does not apply if the chosen data policy is of the type Device Properties.</p>

# Add an MQTT Server destination

## Procedure

In the **Destination** tab, from the **Type** drop-down list, choose **MQTT Server** and enter the required details in the following fields.

Source   **Destination**   Data Policy

MQTT Server

**Connection Details**

Broker \*  Port \*

ⓘ  Additional custom topics can be defined in your Data Logic(s). Refer the documentation for more details.

Topic \*

Enable TLS

Username  Password

**Advanced Settings**

QoS \*  Client ID \*

Retain Messages  ⓘ

**Message Structure**

Asset Attribute

Include Asset Attributes

Group asset attributes under property

Telemetry Data

Group telemetry data under property

Include Timestamps

ⓘ  attributeData

ⓘ  deviceData

Example:

```
{
  "assetAttribute1": "Herkules",
  "assetAttribute2": 120,
  "deviceData": {
    "telemetryData1": {
      "v": 22.4,
      "ts": 1545738873
    },
    "telemetryData2": {
      "v": 45.2,
      "ts": 1511913359
    }
  }
}
```

Field	Description
<b>Connection Details</b>	
Broker	Enter the URL or IP address of your MQTT broker.
Port	Enter the port number used by the broker.

Field	Description
<b>Topic</b>	<p>Enter the topic to which device states and other data are published. For example, <code>cisco/edge-intelligence/telmetry/%deviceSerialNumber%deviceSerialNumber%</code>. The example topic matches the device or asset instance serial number that is configured previously in the source tab.</p> <p><b>Note</b> MQTT topic has a restricted number of characters that can be used in a topic name. For example, # or + cannot be part of a topic name.</p> <p>Topic names are URL-encoded to ensure that they do not violate MQTT specifications. URL-encoding also allows northbound applications to decode a topic easily to get to the original contents.</p>
<b>Enable TLS</b>	<p>Check the TLS check box to enable the protocol. When you choose TLS, the following fields are displayed:</p> <ul style="list-style-type: none"> <li>• <b>Verify Peer:</b> Check this checkbox to allow peer verification. When you select this option, the <b>Certificate</b> field is displayed where you can upload a CA certificate.</li> <li>• <b>Enable X.509:</b> Check this check box to use X.509 certificates. When you select this option, two fields are displayed where you can upload a CA certificate and a private key.</li> </ul> <p>Enabling the use of X.509 certificates allows you to turn on configure MQTT brokers (like <a href="#">Mosquitto</a>) to require certificates for authentication. You can also use X.509 client certificates instead of usernames and passwords to ensure that only trusted assets are allowed to send data to a cloud MQTT broker.</p> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>• The private key must be PKCS8-compatible.</li> <li>• The generated certificate for each asset contains the CN - Serial Number of the asset.</li> <li>• X.509 certificates can be used with the username and password authentication method, or as the only authentication method.</li> </ul>
<b>Username</b>	Enter the username to connect to the MQTT destination broker.
<b>Password</b>	Enter the password to connect to the MQTT destination broker.

## Add an MQTT Server destination

Field	Description
<b>Advanced Settings</b>	
<b>QoS</b>	From the drop-down list, choose <b>0</b> , <b>1</b> , or <b>2</b> .
<b>Client ID</b>	<p>Enter a Client ID.</p> <p>The Client ID field is not URL-encoded because there are no restrictions in MQTT specifications about allowed characters.</p>
<b>Retain Messages</b>	Check this check-box to retain messages on the broker for new subscribers.
<b>Message Structure</b>	<p>Choose the required options in this area to customize the structure of the device-to-cloud message. You can choose <b>Include asset attributes</b>, <b>Group asset attributes under property</b> in Asset Attributes section or choose <b>Group telemetry data under property</b>, <b>Include Timestamps</b> in Telemetry Data section.</p> <p>The data can be sent in a flat structure or grouped with a key. Customization does not apply if the chosen data policy is of the type Device Properties.</p>
<b>Cloud to Network Device</b>	
<b>Enable cloud to data logic commands</b>	<p>Use the cloud to data logic commands to send a command (with payload) from a cloud app to a data logic on an edge device. The data logic script parses the command.</p> <p>This feature provides bi-directional communication between the cloud and edge, allowing the application to send a command and receive a response. For example, a cold storage unit connected to an edge device can send commands to:</p> <ul style="list-style-type: none"> <li>• Set the temperature on the cold storage unit</li> <li>• Initiate a defrost action on the cold storage unit</li> </ul> <p>To enable cloud to data logic commands, enter the following topic values:</p> <ol style="list-style-type: none"> <li><b>Command Topic:</b> Enter the syntax and variables in the format, <i>cisco/edge-intelligence/commands/variable</i>.</li> <li><b>Response Topic:</b> The response topic must use the format <i>cisco/edge-intelligence/responses/variable</i>.</li> </ol> <p>Responses are optional. If a script doesn't send a response, nothing is published to the topic.</p>

# Add an AWS Server destination

## Procedure

From the **Destination** tab, from the **Type** drop-down list, choose **AWS** and enter the required details in the following fields.

Type \*

AWS

**Connection Details**

Broker \*

Port \*

ⓘ Additional custom topics can be defined in your Data Logic(s). Refer the documentation for more details.

Topic \*

Enable TLS

Username  Password

**Advanced Settings**

QoS \*

1  Client ID \*

Retain Messages  ⓘ

**Message Structure**

Asset Attribute

Include Asset Attributes

Group asset attributes under property

Telemetry Data

Group telemetry data under property

Include Timestamps

ⓘ  attributeData

ⓘ  deviceData

Example:

```
{
  "telemetryData1": 22.4,
  "telemetryData2": 45.2
}
```

Field	Description
<b>Connection Details</b>	
<b>Broker</b>	Enter the URL or IP address of your AWS broker. This info can be found from the AWS IoT <b>Settings</b> page
<b>Port</b>	Enter 8883.
<b>Topic</b>	Enter a topic to which device states and other data are published.

## Add an AWS Server destination

Field	Description
<b>Enable TLS</b>	For AWS destinations, you must enable the TLS protocol. When you choose TLS, the following fields are displayed: <ul style="list-style-type: none"> <li>• <b>Verify Peer:</b> Do not choose this option.</li> <li>• <b>Enable X.509:</b> Check this check box to use X.509 certificates. When you select this option, two fields are displayed where you can upload a CA certificate and a private key.</li> </ul>
<b>Username</b>	Do not enter any value in this field.
<b>Password</b>	Do not enter any value in this field.
<b>Advanced Settings</b>	
<b>QoS</b>	From the drop-down list, choose <b>1</b> .
<b>Client ID</b>	Enter a Client ID.
<b>Retain Messages</b>	For AWS destination type, do not choose the <b>Retain Messages</b> option.  This option retains messages on the broker for new subscribers.
<b>Message Structure</b>	Choose the required options in this area to customize the structure of the device-to-cloud message. You can choose to include asset attributes, include timestamps, group asset attributes, or group telemetry data in the device-to-cloud messages.  The data can be sent in a flat structure or can be grouped with a key. Customization does not apply if the chosen data policy is of the type Device Properties.
<b>Cloud to Network Device</b>	

Field	Description
<b>Enable cloud to data logic commands</b>	<p>Use the cloud to data logic commands to send a command (with payload) from a cloud app to a data logic on an edge device. The data logic script parses the command.</p> <p>This feature provides bi-directional communication between the cloud and edge, allowing the application to send a command and receive a response. For example, a cold storage unit connected to an edge device can send commands to:</p> <ul style="list-style-type: none"> <li>• Set the temperature on the cold storage unit</li> <li>• Initiate a defrost action on the cold storage unit</li> </ul> <p>To enable cloud to data logic commands, enter the following topic values:</p> <ol style="list-style-type: none"> <li><b>Command Topic:</b> Enter the syntax and variables in the format, <i>cisco/edge-intelligence/commands/variable</i>.</li> <li><b>Response Topic:</b> The response topic must use the format <i>cisco/edge-intelligence/responses/variable</i>.</li> </ol> <p>Responses are optional. If a script doesn't send a response, nothing is published to the topic.</p>

## About Data Policies

Data Policies define how data is sent from edge assets to a destination. There are two types of policies:

- **Data Logic:** Data is transformed before being sent to a destination. Data Logic scripts are developed using Microsoft VS Code and embedded UI editor.
- **Data Rule:** Data is sent from Assets to a destination without transformation.



### Restriction

Make sure to configure **Data Logic** when you have configured multiple Source type. Proceeding only with **Data Rule**, will pop-up an error while deployment.

## Create a Data Logic

Data logic is used to transform data from connected assets before it is delivered to a destination. Unlike Data Rules that send all the raw data for an Asset Type, Data Logic allows you to aggregate or average data, send only data that exceeds certain value, detect anomalies, and more.

Data Logic is developed and debugged using JavaScript in Microsoft Visual Studio (VS Code) and embedded UI editor. Scripts are synchronized to Cisco IoT where they can be deployed to EI Agents running on Edge Devices.

## Procedure

**Step 1** From the **Data Policy** tab, select **Data Logic** checkbox.

**Data Logic Configuration**

Data Logic Script File (.js) \*

[DL Editor](#)

Choose File

or drag and drop your file here

Data logic file is required

Invoke Periodically (in ms) [i](#) Enter interval in ms

Invoke on New Data

Cloud to Device Command [i](#)

Enable Raw Mode

### Note

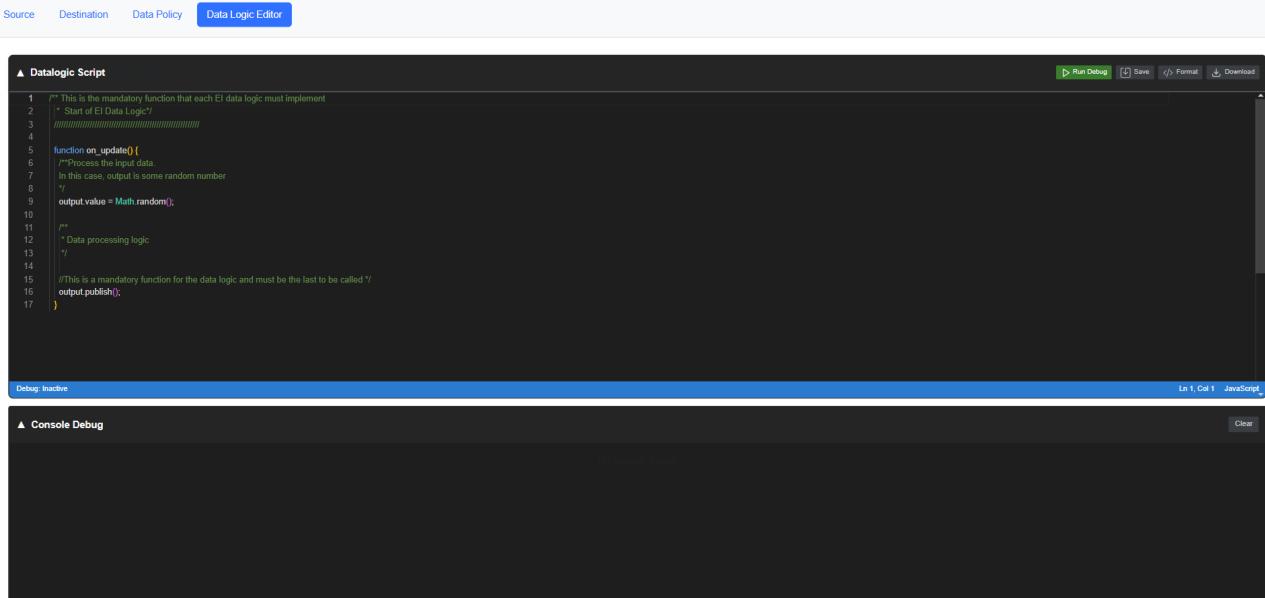
Clicking upon **Data Logic** radio button the **Data Logic Editor** tab appears next to Data Policy.

**Step 2** To add a data logic script, click **Choose File** to select precreated new script (example: **LogicExample.js**) from your local drive.

**Step 3** To review or to create a new data logic script, click **DL Editor**.

- Data logic functionalities where user can write his own transformation code to help to transform data from source to destination.
- It helps to verify and validate the existing script file by clicking **Run Debug**.

- You can write a new script and validate it.



The screenshot shows the Cisco Edge Intelligence Local Manager interface. At the top, there are tabs: Source, Destination, Data Policy, and Data Logic Editor. The Data Logic Editor tab is selected, showing a code editor with the following content:

```

1  /* This is the mandatory function that each EI data logic must implement
2  * Start of EI Data Logic*/
3  //////////////////////////////////////////////////
4
5  function on_update() {
6      /*Process the input data
7      In this case, output is some random number
8      */
9      output.value = Math.random();
10
11     /*
12     * Data processing logic
13     */
14
15     /*This is a mandatory function for the data logic and must be the last to be called */
16     output.publish();
17 }

```

Below the code editor, there are two panels: 'Debug: Inactive' and 'Console Debug'. The 'Console Debug' panel is empty.

**Step 4** After you upload a script file, the new download and delete icon appears. It helps to check/validate what uploaded from other agents. Another user can also check an existing pipeline and can download it.

**Step 5** From the following run-time options, select when the script will be run.

- **Invoke Periodically (in ms):** when you tick the checkbox, it enables the ms interval. Enter the interval time, in ms. For example, if you enter 500, the Data Logic script will be called every 500 ms. Enable this option if needed.
- **Invoke on New Data:** The script is called when data changes.
- **Cloud to Device Command:** This function is called when you receive a command from the cloud.
- **Enable Raw Mode:** A default output JSON data model is automatically created when a Data Logic script is created.
- (Optional) In the **Output Logic Data Model**, you can modify the default format of the Output Data Logic Model script (in JSON) and specify the output model with custom names.

Valid categories: TELEMETRY, PROPERTY, ATTRIBUTE

Valid types: string, int, binary, boolean, double

## Output Logic Data Model

```
[  
  {  
    "key": "value",  
    "type": "DOUBLE",  
    "category": "TELEMETRY"  
  }  
]
```

Valid categories: TELEMETRY, PROPERTY, ATTRIBUTE

Valid types: string, int, binary, boolean, double

---

## Create a Data Rule

Data rules define the flow of data, from connected assets to data destinations, without transformation.

### Procedure

---

**Step 1** From the **Data Policy** tab, select **Data Rule** checkbox.

**Step 2** To configure the Data Rule, select a data from **Data Sampling Interval (ms)** drop-down list.

Data Policy Type \*

Data Logic  Data Rule

#### Data Rule Configuration

Data Sampling Interval (ms)

Disabled

Disabled

100

250

500

1000

2000

5000

## Deploy or undeploy pipelines

When deployed, a pipeline runs on the Cisco network device where the EI agent is installed.

### Procedure

#### Step 1

To deploy a pipeline:

- a) Create a pipeline.
- b) Click **Deploy**.

The deployed pipeline is listed in the **Pipelines** area of the Cisco Edge Intelligence page.

#### Step 2

To undeploy a pipeline, in the **Pipeline** section, select a pipeline and click **Undeploy**.

When you undeploy a pipeline, it is entirely removed from Cisco Edge Intelligence. If you wish to retain a copy of the configuration, save the pipeline configuration as a template before you undeploy the pipeline.

# View health status

To track the health status of an EI agent, select a pipeline from the list of pipelines.

## Procedure

**Step 1** From the left pane, click any **Pipeline** for which you want to view the details.

**Step 2** Click **Health Status** tab. A combined list log of pipeline overview, source status, and destination status appears for the EI agent.

**Step 3** Click on any status tab to preview each log report.

Column Header	Description
Health Status	Displays the overall health of the data pipeline, indicating errors in source or destination connections.
Pipeline Status	Shows the current status of the data pipeline. For example, Error and so on.
Source Status	Indicates the connection status (online or offline) of the source asset.
Destination Status	Reflects the status (online or offline) of the data destination.