

# Table of Contents

[Overview](#)

[Get Started](#)

[Platform](#)

[Architecture](#)

[Services](#)

[Devices](#)

[Get Access](#)

[Overview](#)

[Deploy New](#)

[Request to Join](#)

[User Manual](#)

[Overview](#)

[Dashboard](#)

[Toolbar](#)

[Device Explorer](#)

[User Management](#)

[Rules](#)

[Packages](#)

[Deployments](#)

[Maintenance](#)

[User Profile](#)

[System Settings](#)

[Contribute](#)

[Overview](#)

[Community](#)

[Overview](#)

[Request a Feature](#)

[Report a Bug](#)

[Influence Priorities](#)

[Track Work](#)

[Track Releases](#)

[Improve Documentation](#)

[Platform](#)

[Overview](#)

[Setup by Role](#)

[Tools](#)

[All Products](#)

[Visual Studio 2019](#)

[Visual Studio Code](#)

[Virtualization](#)

[.NET Core 3.1](#)

[PowerShell](#)

[Azure CLI](#)

[Git](#)

[Bash](#)

[NodeJS](#)

[NPM](#)

[Azure Data Studio](#)

[Storage Explorer](#)

[Azure IoT Explorer](#)

[Docker](#)

[Terraform](#)

[Kubernetes](#)

[Helm](#)

[Redux](#)

[Postman](#)

[Dev spaces](#)

[XUnit](#)

[Source Code](#)

[Overview](#)

[Commit Changes](#)

[One Time Setup](#)

[Guide 1](#)

[Guide 2](#)

# Getting Started

Welcome to the 3M IoT Platform on Azure documentation. This section provides an overview of the platform and how to get started using it and contributing to its improvement.

## Content Organization

The left navigation is organized sequentially to provide a foundation for working with the platform. Use the *Filter Content...* located above the navigation to filter for specific topics.

As you navigate across links, you may see a PDF button for downloading documentation viable offline. The button's label will indicate the content scope of the PDF. For example, the content attached to the **Getting Started** link below will export all "Getting Started" content whereas is possible to download just the **User Manual** by navigating to that section.

## Content Accuracy

We have taken care to ensure the accuracy of this content but mistakes happen and things change over time. See [How to Improve This Documentation](#) to help us continue to improve the content and keep it up to date.

## Sections

The top navigation organizes content into sections as follows:

SECTION	DESCRIPTION
Getting Started	Introductory resources for those new to the platform
Use Cases	Examples of how the platform is being used at 3M.
Guides	Topics related to common questions and activities like how to manage devices.
APIs	Information about integrating the common services into custom solutions.

The *Search* box allows you to search for content across all sections.

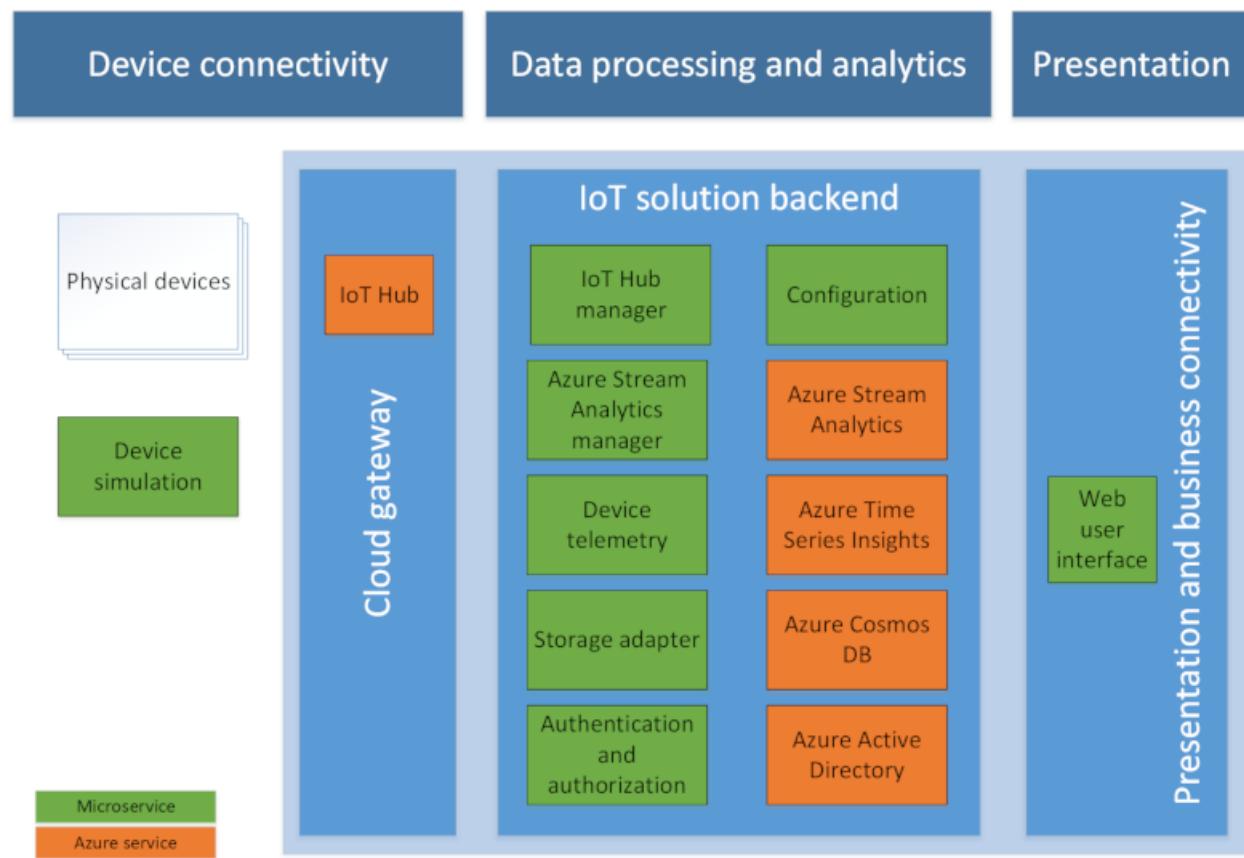
# 3M IoT Platform Overview

The 3M IoT Platform on Azure, called Serenity, provides essential services for managing devices at scale. It can be used directly via its web UI or programmatically via APIs. Product teams across 3M are using Serenity to accelerate innovation. The platform standardizes aspects of each innovation by providing core capabilities needed in any IoT scenarios, including device management services, security monitoring, and CI/CD. The platform allows adopters to focus on unique IoT objectives rather than having to build, validate, and independently maintain common services.

The code for the 3M IoT Platform is open source, available on [GitHub](#). Use it to request features, track improvements, and even make contribution to platform enhancements.

## Logical Components

The following diagram outlines the logical components of the platform and highlights the primary Azure Resources that it uses.



## IoT Hub

The [IoT Hub](#) is a Microsoft Azure Resource used to ingest telemetry sent from both real and simulated devices into the cloud. The hub makes the telemetry available to the services in the IoT solution for processing.

The IoT Hub also provides the following:

- Maintains an identity registry that stores the IDs and authentication keys of all the devices permitted to connect to the portal.
- Invokes methods on your devices on behalf of the solution accelerator.
- Maintains device twins for all registered devices. A device twin stores the property values reported by a device. A device twin also stores desired properties, set in the solution portal, for the device to retrieve when it next connects.
- Schedules jobs to set properties for multiple devices or invoke methods on multiple devices.

## Web UI

A web user interface is available for device management. This presentation offers a user friendly way to register and configure devices, deploy firmware, and manage alerts, and provides a dashboard for generally observing device telemetry.

The web user interface is a [React Javascript](#) application that:

- Is served to the browser via backend services
- Is styled with Css and SaSS
- Interacts with other backend services like the authentication and authorization service to protect user data and communicate with Azure resources

## Backend Services

The platform is made up of several services running in [Azure Kubernetes Service \(AKS\)](#). The platform containerizes services to offer enterprise-grade security and scalability. The services - often called *microservices*- are written in .NET (C#) and provide RESTful endpoints that can be used for direct integration in business specific solutions.

## Data processing and analytics

The solution includes the following components in the data processing and analytics part of the logical architecture:

### IoT Hub manager

The solution includes the IoT Hub manager microservice to handle interactions with your IoT hub such as:

Creating and managing IoT devices. Managing device twins. Invoking methods on devices. Managing IoT credentials. This service also runs IoT Hub queries to retrieve devices belonging to user-defined groups.

The microservice provides a RESTful endpoint to manage devices and device twins, invoke methods, and run IoT Hub queries.

### Device telemetry

The device telemetry microservice provides a RESTful endpoint for read access to device telemetry stored in Time Series Insights. The RESTful endpoint also enables CRUD operations on rules and read/write access for alarm definitions from storage.

### Storage adapter

The storage adapter microservice manages key-value pairs, abstracting the storage service semantics, and presenting a simple interface to store data of any format using Azure Cosmos DB.

Values are organized in collections. You can work on individual values or fetch entire collections. Complex data structures are serialized by the clients and managed as simple text payload.

The service provides a RESTful endpoint for CRUD operations on key-value pairs.

### Azure Cosmos DB

Deployments use Azure Cosmos DB to store rules, alerts, configuration settings, and all other cold storage.

### Azure Stream Analytics

The Azure Stream Analytics manager microservice manages Azure Stream Analytics (ASA) jobs, including setting their configuration, starting and stopping them, and monitoring their status.

The ASA job is supported by two reference data sets. One data set defines rules and one defines device groups. The rules reference data is generated from the information managed by the device telemetry microservice. The Azure Stream Analytics manager microservice transforms telemetry rules into stream processing logic.

The device groups reference data is used to identify which group of rules to apply to an incoming telemetry message. The device groups are managed by the configuration microservice and use Azure IoT Hub device twin queries.

The ASA jobs deliver the telemetry from the connected devices to Time Series Insights for storage and analysis.

Azure Stream Analytics is an event-processing engine that allows you to examine high volumes of data streaming from devices.

### Azure Time Series Insights

Azure Time Series Insights stores the telemetry from the devices connected to the solution accelerator. It also enables visualizing and querying device telemetry in the solution web UI.

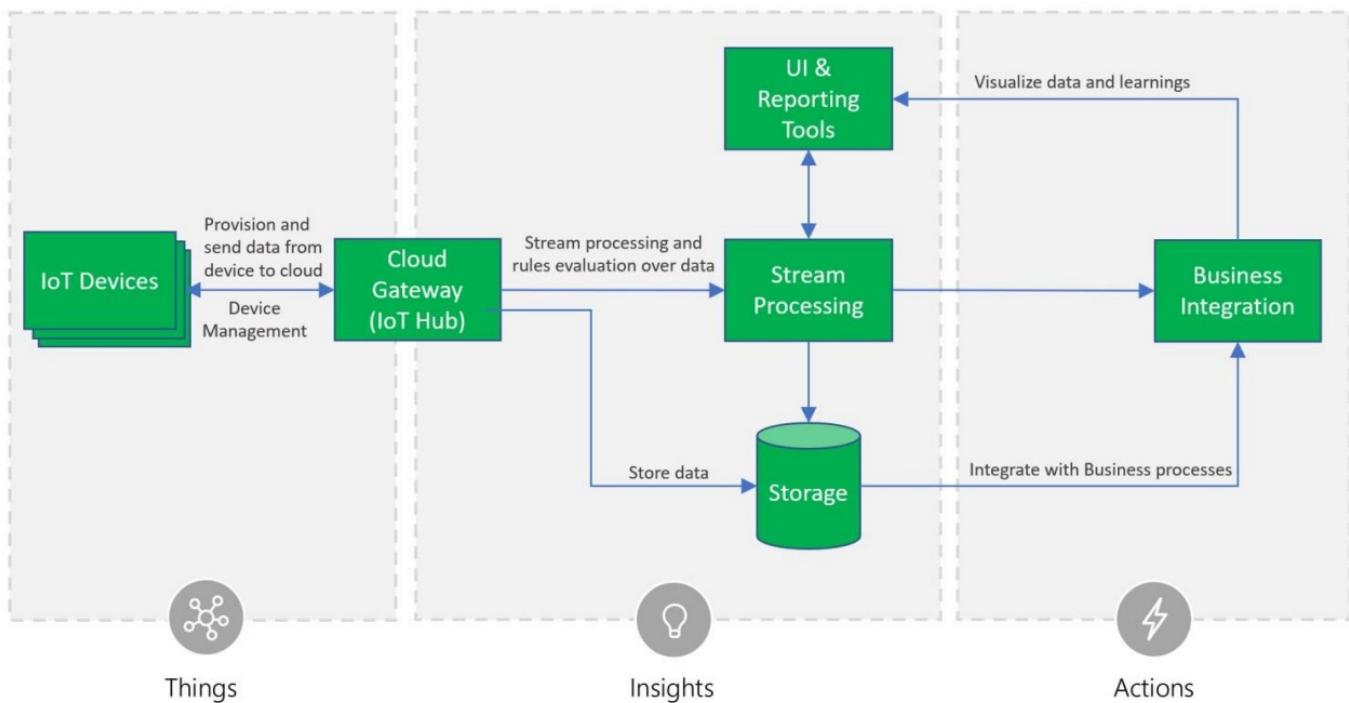
### Authentication and authorization microservice

The authentication and authorization microservice manages the users authorized to access the solution accelerator. User management supports a variety of identity service providers that support OpenId Connect, including Azure B2C.

# Architecture

The 3M IoT Platform on Azure consists of the following subsystems: 1) devices (and/or on premise edge gateways) that have the ability to securely register with the cloud, and connectivity options for sending and receiving data with the cloud, 2) a cloud gateway service, or hub, to securely accept that data and provide device management capabilities, 3) stream processors that consume that data, integrate with business processes, and place the data into storage, and 4) a user interface to visualize telemetry data and facilitate device management.

## Core Subsystems



Security is a critical consideration in each of the subsystems. The platform protects IoT devices, data, and communication by securely provisioning devices, secure connectivity between devices, edge devices, and the cloud, secure access to the backend solutions, and secure data protection in the cloud during processing and storage (encryption at rest).

The platform's use of Azure IoT Hub offers a fully-managed service that enables reliable and secure bi-directional communication between IoT devices and Azure services such as Azure Machine Learning and Azure Stream Analytics by using per-device security credentials and access control. Both Azure Cosmos DB for warm path storage and Azure Blob Storage for cold storage are available and support encryption at rest.

All data access and permissions are governed by Azure Active Directory. External access is available via OpenID Connectivity as needed.

# Overview of Platform Services

The 3M IoT Platform on Azure has several scalable services for addressing core Device Management capabilities.

The platform provides the following services:

NAME	DESCRIPTION
Identity Gateway	
Config	
Device Telemetry	
IoT Hub Manager	
Storage	
Tenant Manager	
Device Twin	
Device Properties Cache	



## IoT Devices

The 3M IoT Platform provides a Web UI and set of APIs that can be tailored to unique industry and product requirements related to device enrollment, discovery, connectivity, remote configuration, and software updates.

The platform's common services include the following device management capabilities:

1. Device provisioning and discovery
2. Device access management
3. Remote control
4. Remote administration and monitoring
5. Remote configuration
6. Remote firmware and software update

To start working with devices, see the [Device Guide](#) for more information



## First Step

Before you can begin, you need access to a instance of the IoT Platform. You have two options:

1. Deploy a new instance into Azure
2. Access an existing instance







# User Guide Overview

This guide provides a walk-through us using the application user interface for the 3M IoT Platform on Azure You can take the guide offline by downloading the PDF.

# Dashboard

The dashboard is the homepage where the user can visualize and monitor real-time Device status.

Below are the panels the dashboard offers:

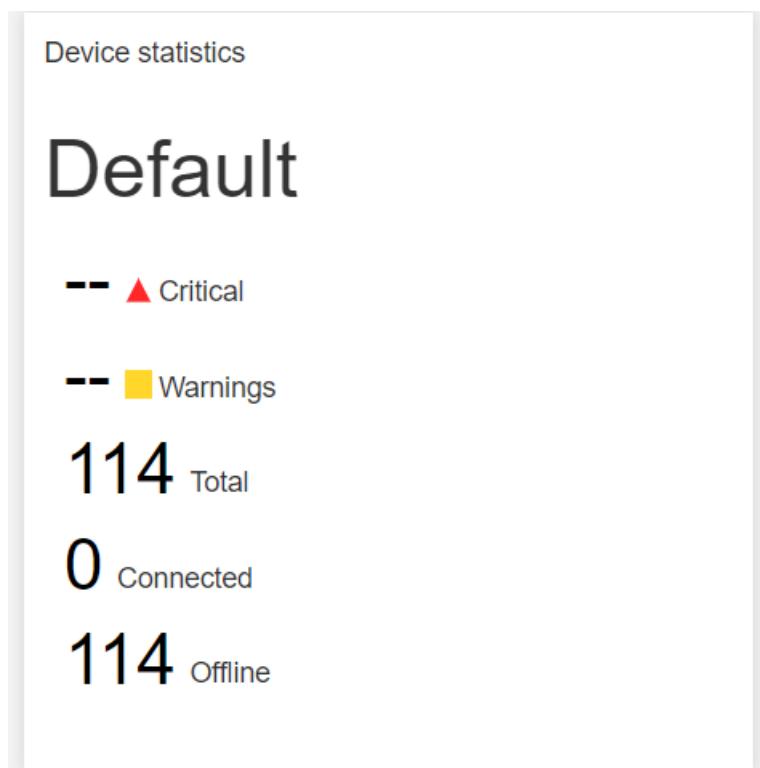
1. [Device statistics](#)
2. [Device locations](#)
3. [Alerts](#)
4. [Telemetry](#)
5. [Analytics](#)

## Device statistics

The device statistics panel provides collective data on IoT Devices' state and health.

Device statistics show the following information

- Device Group name.
- Number of critical alerts recorded in a device group.
- Number of warning alerts recorded in a device group.
- Total number of IoT Devices present in the device group.
- Number of IoT Devices that are online or connected to the internet.
- Number of IoT devices that are offline.

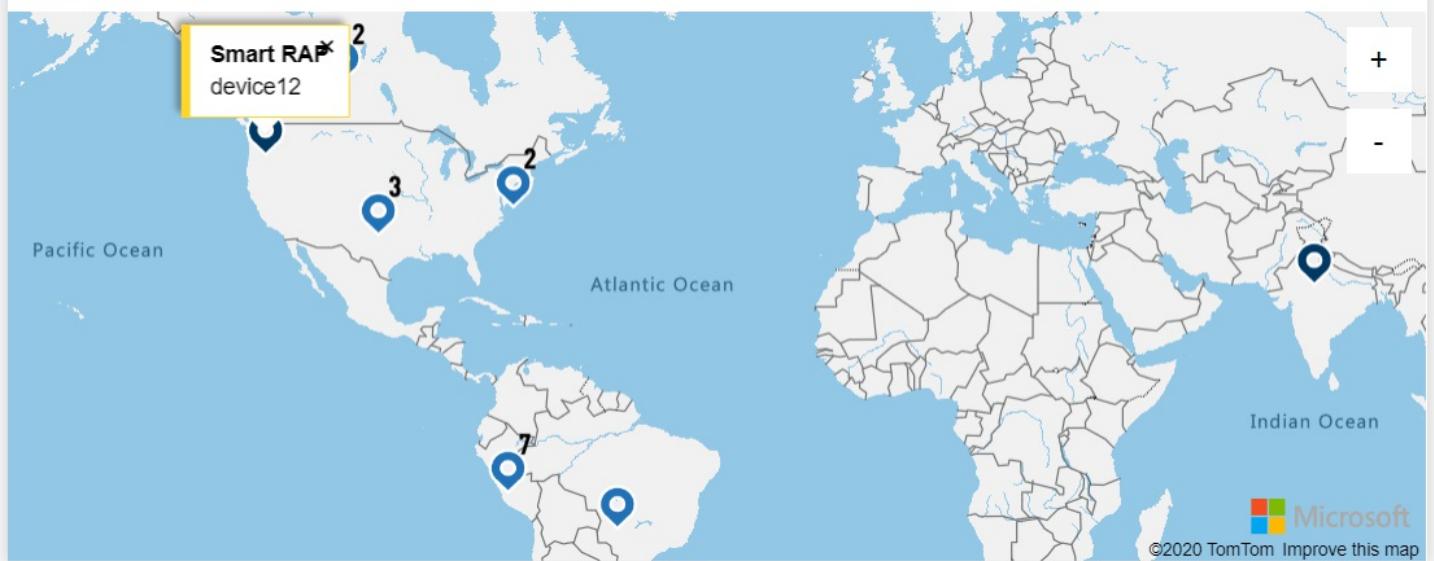


## Device locations [refresh]

Device locations provide the location of IoT Devices on the world map; it only plots those devices configured with coordinates.

Clicking a pin [info] on the map reveals the device type and device name.

## Device locations



## Alerts □□

Alerts list the anomalies in telemetry based on rules.

Below are the details:

1. Rule Name
2. Severity
3. Count - Number of alerts observed.
4. Explore - A link that routes to more details on the alert.

Alerts				
Rule name	S.	C.	E.	> Expand Columns
ShiftHumidityGreaterThan70	▲	C..	2	...
ShiftTemperatureGreaterT...	▲	C..	4	...
temp30	■	W..	10	...

1 to 3 of 3   First   Previous   Page 1 of 1   Next   Last

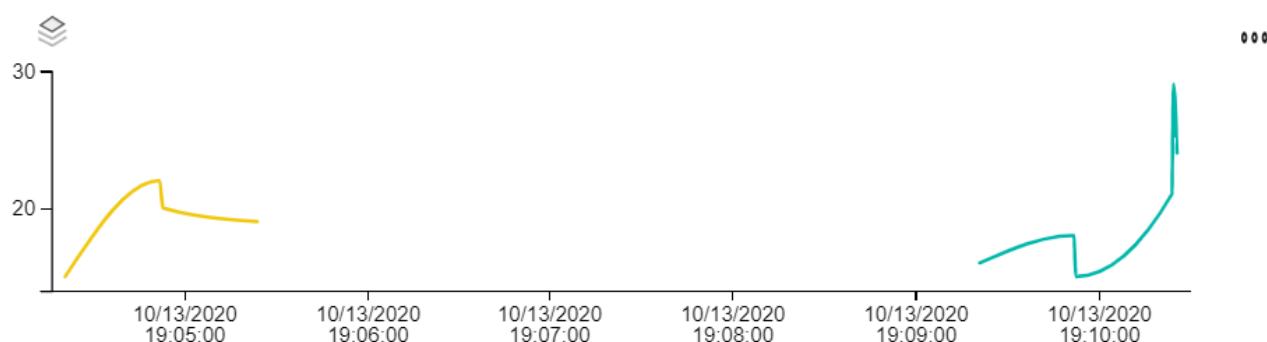
## Telemetry □

Telemetry shows a real-time chart based on data sent by the devices.

## Telemetry

< aqi [2] filterLifeRemaining [2] humidity [2] pm1 [2] pm10 [2] pm25 [2] temperature [2] >

device13 device12



Displaying in local timezone: Asia/Calcutta

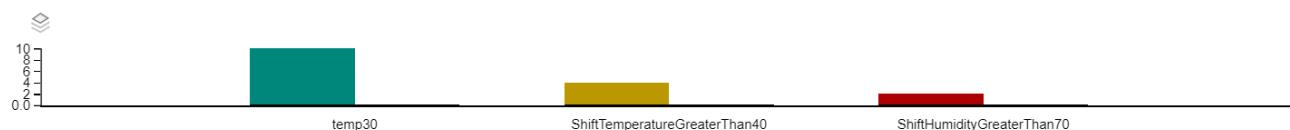
## Analytics

Analytics provide the visual representation of alert data in the form of charts □ as follows:

1. Top rules with active alerts
  - o It is visualized in the form of a Bar Graph □.
  - o The graph represents the number of alerts recorded per rule.
2. Alert by device group
  - o It is visualized in the form of a Pie Graph □□.
  - o The graph shows the number of alerts recorded per device type.
3. Critical alerts
  - o Percentage of Critical alerts in the open state.

### Analytics

#### Top rules with active alerts



#### Alert by device group



100.00%

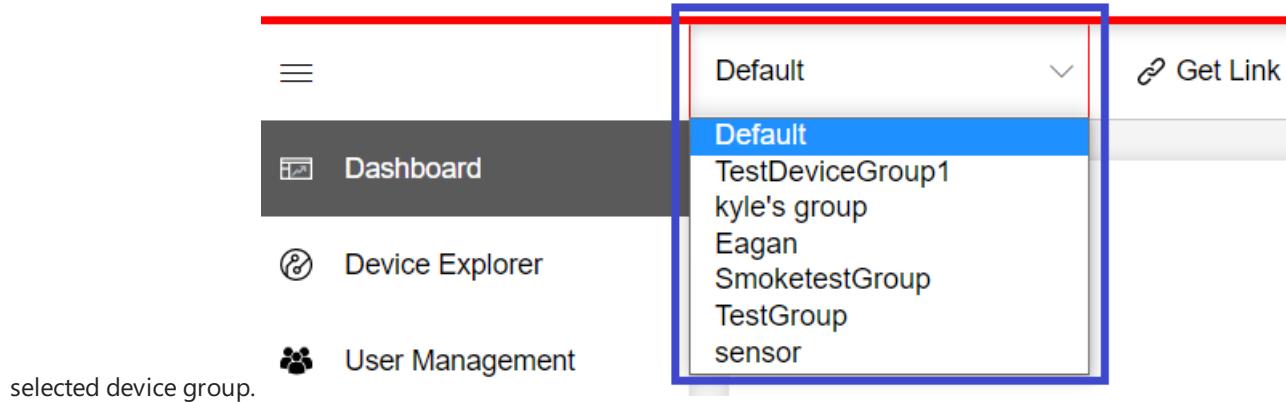
## Tool Bar

The tool bar has various options to access different functionalities. This section covers the tool bar functionalities that are common across the application. Below are some of them:

1. [Device Group](#)
2. [Get Link](#)
3. [Manage Device Groups](#)
4. [Time Period](#)
5. [Last Refreshed](#)

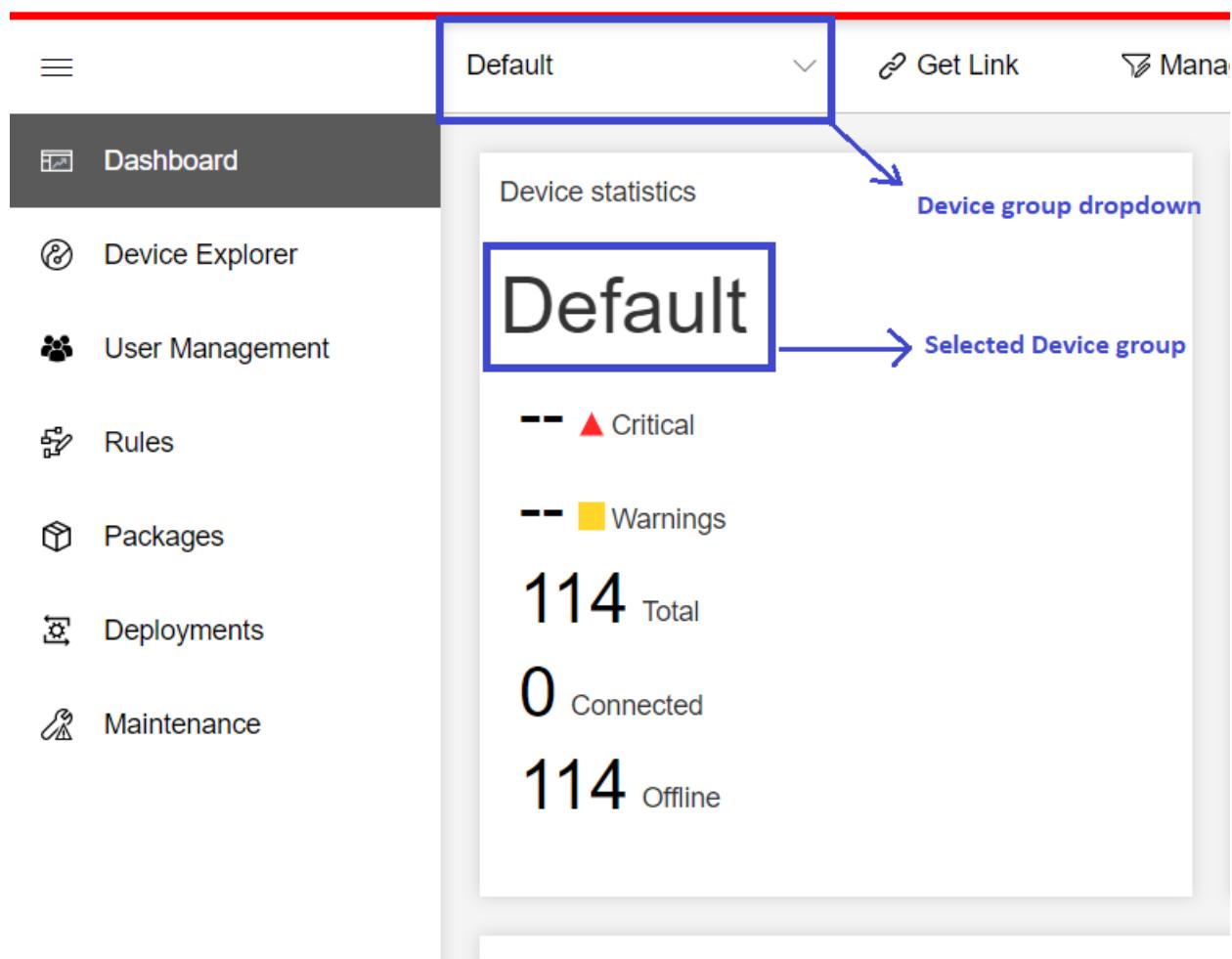
## Device Group

The **Device Group dropdown** in the tool bar is used to select a device group to filter the data in any given page. The selected device group remains same across all the pages for a user. And the data shown in most of the pages is filtered based on the



selected device group.

**Ex:** Dashboard- Device statistics, Device Explorer, Packages, etc.



#### Note:

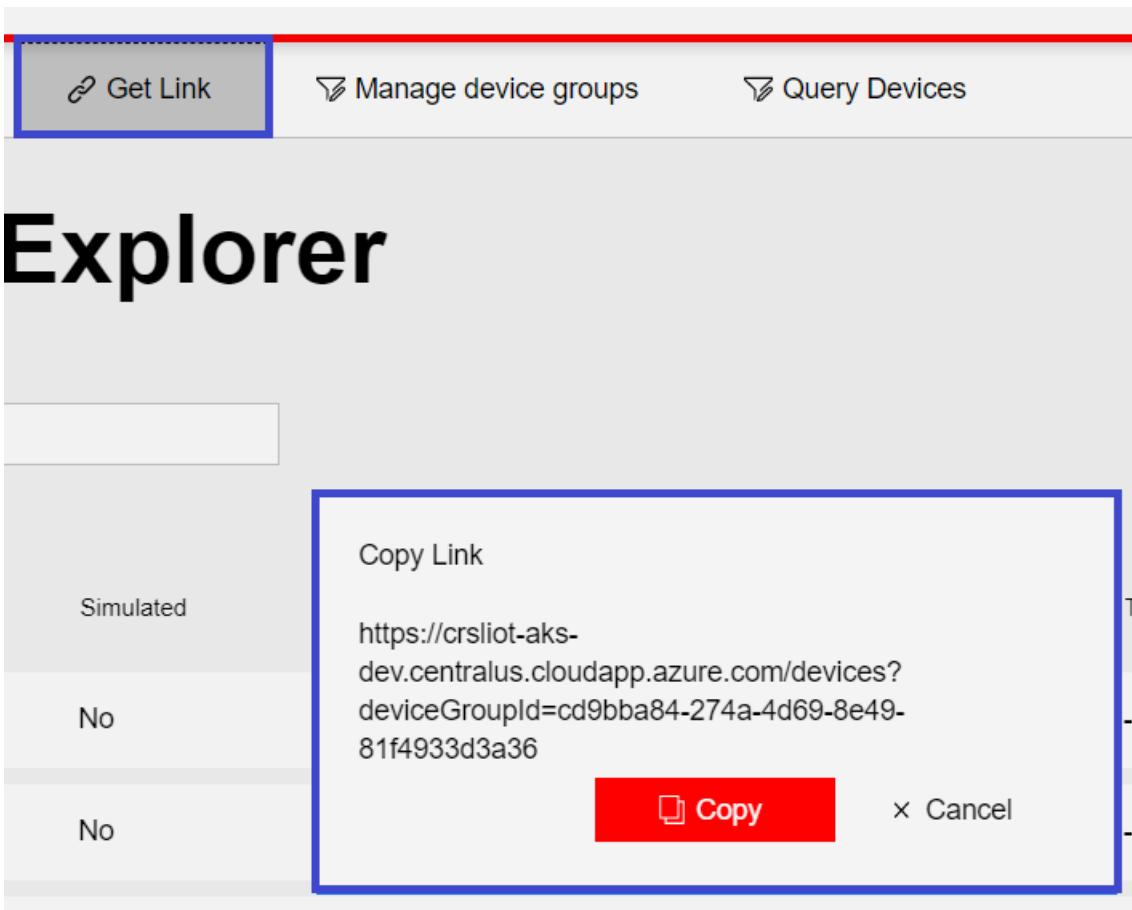
By default, the device group **Default** is selected for any user for the first login.

## Get Link

**Get Link** is a button which lets the user copy the link for that page appended with the Selected Device Group ID. This helps share links with other users that target a specific Device Group ID.

#### Steps to copy link

1. Navigate to the page for which the link (with Device Group ID) needs to be copied **Ex: Device Explorer**
2. Click the **Get Link** button
3. The Copy Link pop-up opens with the link that can be used to share
4. Click Copy : This copies the link to the clipboard



## Manage Device Groups

The **Manage Device Groups** button allows a user to:

1. [View Device group list](#)
2. [Create Device Group](#)
3. [Edit Device Group](#)
4. [Delete Device Group](#)
5. [Pin Device Groups](#)

**View Device group list**

To view list of device groups present:

1. Click the **Manage device groups** button from the toolbar
2. The **Manage device groups** flyout opens
3. The list of device groups is displayed under "Create new device group" button



## Manage device groups

+ Create new device group

Device group name

- Default
- TestDeviceGroup1
- kyle's group
- Eagan
- SmoketestGroup
- TestGroup
- sensor

Save

Reset

### Create Device Group

A user can create a new device group using this option. Steps to create a new device group:

1. Click **Manage device groups** button from the toolbar
2. Manage device groups flyout opens

The screenshot shows a user interface for managing device groups. At the top, there is a header 'Manage device groups'. Below it is a button labeled '+ Create new device group'. A list of device groups is displayed, each with a radio button next to it. The groups listed are: Default (selected), Blake\_group, Test2, joetest, this-is-long-to-resemble-chim-formatti..., and Andrew. At the bottom right are two buttons: 'Save' (in red) and 'Reset'.

3. Click **Create new device group** button

4. A **New device group** form opens

5. Below are the details to fill in:

- **Name(required)** : This is a mandatory field. Enter a unique Device group name

The screenshot shows a 'New device group' form. It has a header 'Manage device groups' and a sub-header 'New device group'. There is a 'Name \*' field with a placeholder 'Device group name'.

- **Conditions:** A user can either add/remove a condition on which a Device group needs to be created. These conditions are used as filters on Devices.

- For adding a condition:

- Click **Add Condition** or a new condition already appears while creating a new device group. Populate the fields to add the condition
- Field(required): Lets the user select the field on which the filter needs to be created on devices (Ex: tags, properties, etc.)
- Operator(required): Lets the user select a logical operator like Equals, Greater than, etc.
- Value(required): Lets the user enter the value to compare against the data for the selected field using the selected operator
- Type(required): Allows the user to select the type of value: Number/Text



## Manage device groups

### New device group

Name \*

+ Add condition

#### Condition 1

Field \*

Operator \*

Value \*

Type \*

Remove condition

- For removing a condition:
  - A **Remove condition** button is present below each condition

+ Add condition

### Condition 1 ^

Field \*

Tags.name

Operator \*

= Equals

Value \*

test-tag

Type \*

Text

Remove condition

- **Telemetry Format:** This is used to filter the telemetry data of devices under the device group based on telemetry keys.

- To configure telemetry format,
  - Click on **Add**
  - Enter Key: This key is used in sending the telemetry data

## Manage device groups

### Telemetry Format ^

KEY

DISPLAY NAME

test

Telemetry

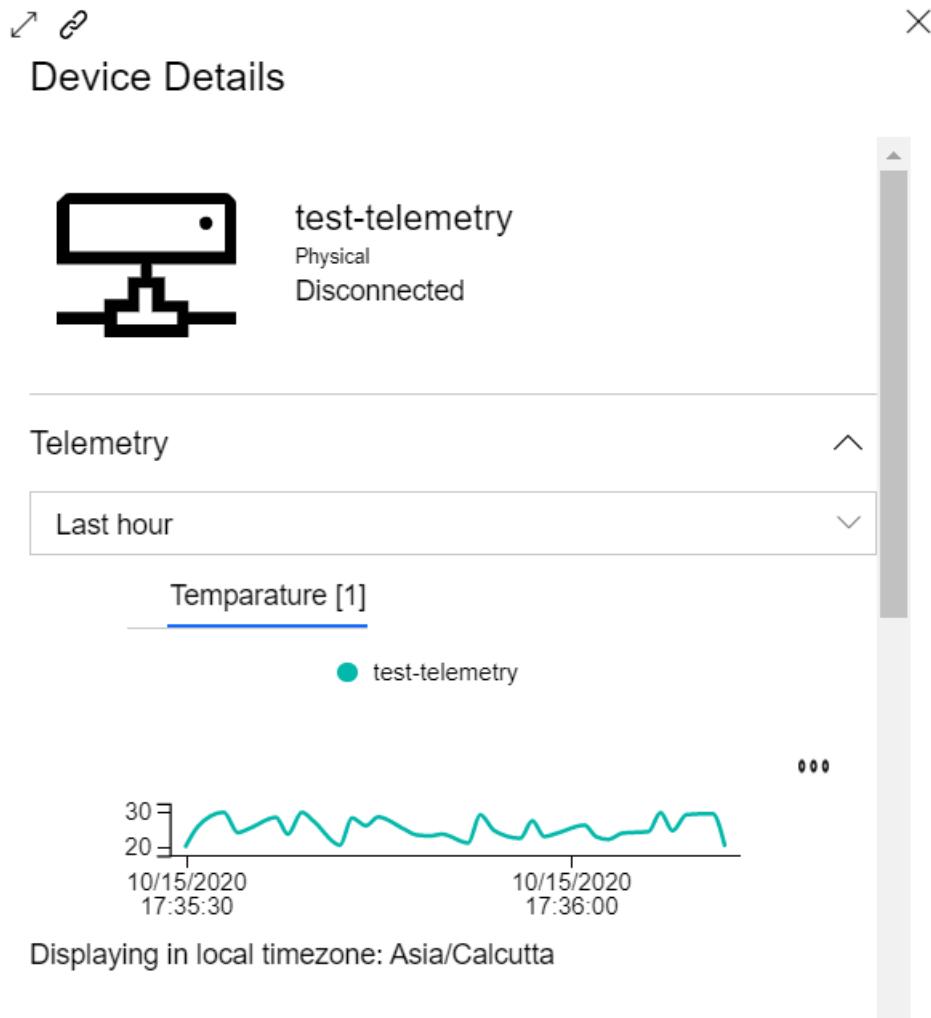


+ Add

- Enter Display Name

- To verify the telemetry format

- Add an entry under telemetry format (Ex: Key=temp and Display Name=Temperature)
- Create a device
- Simulate device telemetry by sending a message with different properties along with value for temp (Key from above) and any value **Ex:** { humidity: 60, temp: 20 } **Note:** Refer to this to simulate device telemetry using Azure online simulator
- Select the device and click "Show telemetry"
- You should be able to see only telemetry for temp(Temperature)



- **Supported Methods:** TODO

## Edit Device Group

A user can edit an existing device group using this option. Steps to edit a device group:

1. Click **Manage device groups** button from the toolbar
2. The **Manage device groups** flyout opens
3. The list of device groups is displayed under "Create new device group" button
4. Click on the name of the device group to edit

Manage device groups

+ Create new device group

Device group name

- Default
- TestDeviceGroup1
- kyle's group
- Eagan
- SmoketestGroup
- TestGroup
- sensor

Save Reset

5. The **Manage device groups** flyout display changes to show the "Edit device group" form where the existing device group details can be modified. Edit the details
6. Click **Save** to save the updates
7. Click **Cancel** in order to cancel and navigate back to Manage device groups

Manage device groups

Edit a device group

Name \*

Default

+ Add condition

Telemetry Format

Supported Methods

Save Cancel  Delete

Delete Device Group

A user can delete an existing device group using this option. Steps to delete a device group:

1. Click the **Manage device groups** button from the toolbar
2. The **Manage device groups** flyout opens
3. The list of device groups is displayed under the "Create new device group" button
4. Click on the name of the device group to delete
5. The display changes to "Edit device group". Scroll down the flyout to find the buttons Save, Cancel and Delete



+ Add condition

Telemetry Format



Supported Methods



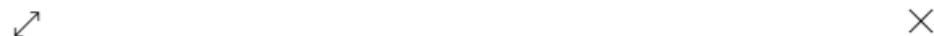
Save

x Cancel

 Delete

6. Click the **Delete** button. A confirmation section appears

7. Change the toggle to "Yes" and click **Delete**



Yes, please delete the device group.

Summary

 Delete

x Cancel

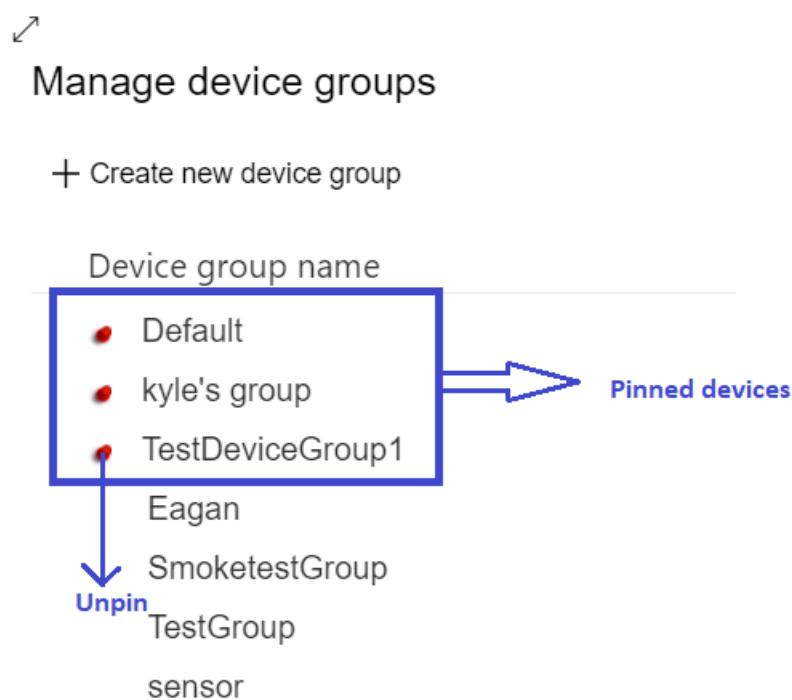
A user can pin one/multiple device groups using this option. Pinned device groups always show on top of the Device group list.

Steps to pin/unpin device groups:

1. Click the **Manage device groups** button from the toolbar
2. The **Manage device groups** flyout opens
3. The list of device groups is displayed under the "Create new device group" button
4. Hover on the name of the device group to pin or unpin
5. Pin Device groups: Click on the pin button that appears before the names of the Device groups



6. Unpin Device groups: Pinned devices can be unpinned. Pinned devices appear on the top of the list. Click on pin(unpin) button that appears before the name of the Device group



7. Click the **Save** button to update the changes
8. Click the **Reset** button to reset the changes



## Manage device groups

+ Create new device group

Device group name

- Default
- kyle's group
- TestDeviceGroup1

Eagan

SmoketestGroup

TestGroup

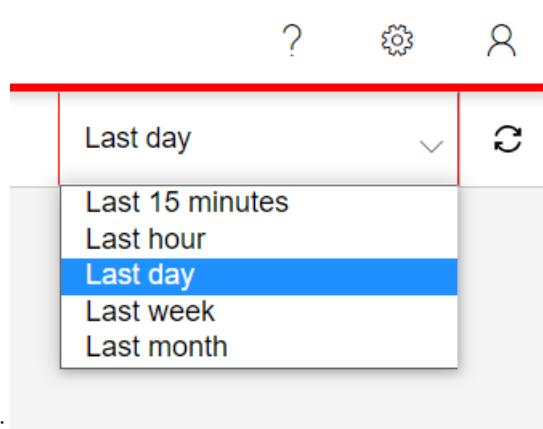
sensor

Save

Reset

## Time Period

The **Time period** dropdown is used in different pages to filter the data based on selected time interval.



Below are the available time period options:

### Note:

- The **Time period** dropdown can be found on pages like dashboard, maintenance and some other places in the application like Device Telemetry, etc.
- The default selected value for the Time period dropdown is **Last hour**

## Last Refreshed

**Last refreshed** is a label and button combination which can be used to:

1. Display the last-refreshed datetime of data on a given page when data is automatically refreshed
  - Hover on the refresh button to view the last-refreshed datetime

2. Manually refresh the data in a given page.

- o Click the refresh button to manually refresh the data in the page



# Device Explorer

The **Device Explorer** is used for maintaining the IoT devices. This section explores the creation and maintenance of devices.

Below are the features it offers:

1. [Device Explorer Grid](#)
2. [Add a Device](#)
3. [Device Details](#)
4. [Delete Device](#)
5. [Configure a Device](#)
6. [Organize devices](#)
7. [More](#)

## Device Explorer grid

The **Device Explorer** grid lists all the devices which are registered to the Azure IoT Hub.

- The following lists the columns and their usage in the **Device Explorer** grid

COLUMN NAME	DESCRIPTION
Device Name	Name of the Device
Simulated	Indicates whether device simulation is on or off
Device Type	Describes the type of device
Firmware	Latest device firmware version
Telemetry	Schema of the telemetry
Status	Indicates whether the device is connected to the Azure IoT hub
Last Connection	Last date when a device connected to Azure IoT hub

- We have a search bar associated with the grid where we can search the grid data.

The screenshot shows the 'Device Explorer' page. On the left, there's a sidebar with icons for Dashboard, Device Explorer (which is selected and highlighted in dark grey), User Management, Rules, Packages, Deployments, and Maintenance. The main area has a search bar labeled 'Search devices...'. Below it is a table with columns: Device name, Simulated, Device type, Firmware, Telemetry, Status, and Last connection. The table lists several devices, including 'groupTest2', 'JA-Device1', 'joetest' (which is checked), 'rule\_test', 'sah\_test', 'sah\_test\_device', and 'sah\_test,test1'. Most devices are marked as Offline.

Device name	Simulated	Device type	Firmware	Telemetry	Status	Last connection
groupTest2	No	---	---	---	Offline	---
JA-Device1	No	---	---	---	Offline	---
joetest	No	---	---	messageSchema	Offline	08:54:35 PM 08.06.20...
rule_test	No	---	---	---	Offline	---
sah_test	No	---	---	---	Offline	04:23:02 PM 09.11.20...
sah_test_device	No	---	---	---	Offline	05:02:16 PM 10.15.20...
sah_test,test1	No	---	---	---	Offline	---

## Add a Device

Steps for adding a device:

- Navigate to the **Device Explorer** page from the left menu. Click on **+New device**

This screenshot is identical to the one above, showing the 'Device Explorer' page. The 'Device Explorer' item in the sidebar is highlighted. The '+ New device' button in the top right corner is also highlighted with a blue box.

- Choose between the below device management services

- **IoT Edge device**
- **IoT device**

- Enter a device ID or choose **System generated device IDs** to generate a random device ID.

- Choose one of the below authentication types

- **Symmetric key**
- **X.509**

The screenshot shows the Device Explorer page with a sidebar containing links like Dashboard, Device Explorer (which is selected), User Management, Rules, Packages, Deployments, and Maintenance. A search bar at the top says "Search devices...". Below it is a table with columns: Device name, Simulated, Device type, Firmware, and Telemetry. The table lists several devices: grouptest2, JA-Device1, joetest, rule\_test, sah\_test, sah\_test\_device, and sah\_test,test1. To the right, a modal window titled "New device" is open. It has sections for "Device" (radio buttons for IoT Edge device or IoT device, with IoT device selected), "Number of devices" (set to 1), "Device ID" (radio buttons for Enter device ID or System generated device IDs, with Enter device ID selected), "Authentication type" (radio buttons for Symmetric key or X.509, with Symmetric key selected), "Authentication key" (radio buttons for Auto generate keys or Enter keys manually, with Auto generate keys selected), and "Primary Key" and "Secondary Key" fields which are empty.

- Choose one of the below authentication key options
  - Auto generate keys**
    - If selected, the system generates the primary and secondary keys for the device.
    - This option is only available when authentication type **Symmetric key** is selected.
  - Enter keys manually**
    - If selected, the user should provide Primary and Secondary Keys.

The screenshot is similar to the previous one, showing the Device Explorer page with the "Enter keys manually" option selected in the modal. The "Authentication key" section now shows "Enter keys manually" selected. The "Primary Key" and "Secondary Key" fields are filled with placeholder text: "SerenityKey" and "SerenityKey" respectively. The "Provision summary" section indicates "1 Devices to provision". At the bottom right, there are "Apply" and "Cancel" buttons.

- Now verify the details provided and click **Apply**.

The screenshot shows the Device Explorer interface with a modal window for creating a new device. The modal contains fields for authentication type (Symmetric key selected), authentication key (Auto generate keys selected), primary and secondary keys, provision summary (1 device to provision), and an Apply button.

- Once the device is successfully created, you will be presented with the created device information and the device will be reflected in Device Explorer Grid.

The screenshot shows the Device Explorer interface with a modal window for creating a new device. The modal displays a success message (1 device provisioned successfully) and lists the device ID, primary key, secondary key, connection string primary key, and connection string secondary key.

## Device Details

The **Device Details** panel displays information about a device such as telemetry, tag values, the methods it supports, and the properties reported by the device.

Click on the device name in the grid. It will open the **Device Details** panel.

Below are the details the **Device Details** panel provides.

### 1. Alert Grid

- The grid will be displayed when alerts are observed for a device.

**Device Explorer**

Device Details

test-telemetry  
Physical  
Disconnected

Device name: sah\_test\_device

Device name	Simulated	Device type	Firmware
sah_test_device	No	---	---
sahh_test	No	---	---
sahh_test,test1	No	---	---
test	No	---	---
test-telemetry	No	---	---
test-tel1	No	---	---
VisTest	No	---	---

Rule name: Rule11

Rule name	Severity	Status
Rule11	Critical	open
Rule3	Critical	open
Rule7	Critical	open
Rule8	Critical	open
Rule9	Critical	open

1 to 5 of 5 First Previous Page 1 of 1 Next Last

Telemetry

## 2. Telemetry

### 3. Tags

### 4. Methods

**Device Explorer**

Device Details

Telemetry

Last week

humidity [1] temp [1] temperature [1]

test-telemetry

Displaying in local timezone: Asia/Calcutta

Tags

To add, delete, or change a tag on one or more devices, close this pane, select the checkbox for the device(s), click **Jobs**, and then select **Tags**.

No tags found for this device.

Methods

To run a method on one or more devices, close this pane, select the checkbox for the device(s), click **Jobs**, and then select **Methods**.

No methods found for this device.

## 5. Properties

## 6. Diagnostics

## 7. Deployment messages

- Edge module messages on the device.

## 8. Device Uploads

- File Uploads from the device.

## 9. Device Deployments

- Deployment history of the device.

# Delete Device

Devices that are no longer needed or in use can be deleted.

Steps to delete a device:

- Select one or more devices by clicking on the checkbox to the left of the device name in the grid.

- After selecting devices the **Delete** button will be visible in the toolbar; click **Delete**.

**Device Explorer**

Device name	Simulated	Device type	Firmware	Telemetry	Status	Last c
sah_test_device	No	---	---	---	Offline	05
sah_test,test1	No	---	---	---	Offline	---
sahh_test	No	---	---	---	Offline	---
<b>Test</b>	No	---	---	---	Offline	---
test-tel1	No	---	---	---	Offline	05
test-telemetry	No	---	---	---	Offline	05
VisTest	No	---	---	messageSchema	Offline	---

- Click on the toggle button to delete the device(s) and the **Delete** button will be activated.

**Delete**

Delete device

This action will delete the devices permanently from the platform.

Yes, please delete the selected physical devices.

**Summary**

1 Affected devices

**Delete**

- After deleting the device(s) you will see a message that the device(s) have been deleted and the device details removed from the **Device Explorer** grid.

The screenshot shows the Device Explorer interface with a modal overlay titled "Delete". The modal contains a confirmation message: "Delete device" and "This action will delete the devices permanently from the platform." Below this is a radio button option: "(●) No, do not delete the selected physical devices." A summary box indicates "1 devices deleted successfully!" with a checkmark. The main grid lists several devices, including "sah\_test", "sah\_test\_device", "sahh\_test", "test-tel1", "test-telemetry", and "VisTest".

Device name	Simulated	Device type	Firmware	Telemetry
sah_test	No	---	---	---
sah_test_device	No	---	---	---
sahh_test	No	---	---	---
test-tel1	No	---	---	---
test-telemetry	No	---	---	---
VisTest	No	---	---	messageSche

## Configure a Device

The **Device Explorer** grid also allows updating device properties, below are the steps:

- Select a device by clicking on the check box to the left side of the device name in the grid.

The screenshot shows the Device Explorer interface with a toolbar at the top containing "Jobs" (which is highlighted with a blue box). The main grid lists devices, and one device, "Test", has its checkbox checked (also highlighted with a blue box). The grid includes columns for Device name, Simulated, Device type, Firmware, Telemetry, Status, and Last connection.

Device name	Simulated	Device type	Firmware	Telemetry	Status	Last connection
sahh_test	No	---	---	---	Offline	04:23:02 PM 09.11.20...
sahh_test	No	---	---	---	Offline	05:02:16 PM 10.15.20...
sahh_test	No	---	---	---	Offline	---
Test	No	---	---	---	Offline	---
test-telemetry	No	---	---	---	Offline	05:08:39 PM 10.15.20...
test-telemetry	No	---	---	---	Offline	05:40:07 PM 10.15.20...

- Then click the **Jobs** button in the toolbar and then choose **Properties**.

**3M Serenity IoT DEV Platform**

The screenshot shows the Device Explorer page. On the left is a sidebar with navigation links: Dashboard, Device Explorer (which is selected and highlighted in dark grey), User Management, Rules, Packages, Deployments, and Maintenance. The main area is titled "Device Explorer" and contains a search bar "Search devices...". Below it is a table with columns: Device name, Simulated, Device type, Firmware, and Telemetry. The table lists several devices: sah\_test\_device, sah\_test,test1, sah\_test, Test, test-tel1, test-telemetry, and VisTest. The "Test" device has a checked checkbox next to its name. To the right of the table is a "Jobs" panel. At the top of the panel is a sub-panel titled "Select job" with three radio button options: Tags, Methods, and Properties (which is selected). Below this is a section titled "Properties on selected devices" with a sub-section "Job Name" containing a text input field with the value "Test". Further down are two rows for properties: "ATest1" with value "Test" and "ATest2" with value "Test". At the bottom of the panel is a "Summary" section.

- The **Jobs** panel shows the updateable property values for the selected device.
- To update the device properties, set the **Job Name** in the text box, update the properties value(s), and click **Apply**.

**3M Serenity IoT DEV Platform**

This screenshot is identical to the one above, but the "Job Name" field in the "Properties on selected devices" section now contains the value "Test1". At the bottom right of the "Jobs" panel, there is a red rectangular button with the text "Apply" in white, which is highlighted with a blue border. The rest of the interface, including the device list and other panels, remains the same.

- To track the status of the job, click **View job status**.

**3M Serenity IoT DEV Platform**

The screenshot shows the Device Explorer page. On the left sidebar, 'Device Explorer' is selected. The main area displays a table of devices with columns: Device name, Simulated, Device type, Firmware, and Telemetry. A search bar at the top says 'Search devices...'. To the right, a 'Properties' panel is open for a device named 'Test'. It shows two properties: 'ATest1' with value 'Test1' and 'TYPE' 'Text', and 'ATest2' with value 'Test1' and 'TYPE' 'Text'. A summary box indicates '1 Job submitted successfully! ✓'. A red box highlights the 'View job status' button.

- After the job completes, navigate to the **Device Explorer** and verify the updated properties details in the device details panel under the **Properties** section.

**Note:**

The job status does not show live status, to refresh the status click the **refresh** button.

**3M Serenity IoT DEV Platform**

The screenshot shows the Device Explorer page with a completed job. The top part shows a table with columns: Job Name, Status, Operation, No. of devices, Succeeded, Failed, Start time, and End time. One row is highlighted with a blue border. Below it, another table shows detailed information for the same job, including Job Name, Status, Device ID affected, Last return message, Start time, and End time.

## Organize devices

To make it easier as a user to organize and manage your devices, you can tag them with a team name.

To display all your devices, navigate to the **Device Explorer** page and choose the **Default** device group:

Default Get Link Manage device groups Query Devices + New device

## Device Explorer

Search devices... Expand Columns

<input type="checkbox"/> Device name	Simulated	Device type	Firmware	Telemetry	Status	Last connection
<input type="checkbox"/> BridgetteDemo	No	---	---	---	Offline	09:38:25 PM 08.18.20...
<input type="checkbox"/> DemoDevice	No	---	---	---	Offline	---
<input type="checkbox"/> grouptest2	No	---	---	---	Offline	---
<input type="checkbox"/> JA-Device1	No	---	---	---	Offline	---
<input type="checkbox"/> joetest	No	---	---	messageSchema	Offline	08:54:35 PM 08.06.20...
<input type="checkbox"/> rule_test	No	---	---	---	Offline	---
<input type="checkbox"/> sah_test	No	---	---	---	Offline	04:23:02 PM 09.11.20...

### Add tags

1. Select the devices in the grid for which you want to add a tag and click the **Jobs** button.

Default Get Link Manage device groups Query Devices **Jobs** Delete ... + New device

## Device Explorer

Search devices... Expand Columns

<input type="checkbox"/> Device name	Simulated	Device type	Firmware	Telemetry	Status	Last connection
<input type="checkbox"/> sah_test	No	---	---	---	Offline	04:23:02 PM 09.11.20...
<input type="checkbox"/> sah_test_device	No	---	---	---	Offline	05:02:16 PM 10.15.20...
<input type="checkbox"/> sah_test,test1	No	---	---	---	Offline	---
<input type="checkbox"/> sahh_test	No	---	---	---	Offline	---
<input checked="" type="checkbox"/> Test	No	---	---	---	Offline	---
<input type="checkbox"/> test-te1	No	---	---	---	Offline	05:08:39 PM 10.15.20...
<input type="checkbox"/> test-telemetry	No	---	---	---	Offline	05:40:07 PM 10.15.20...

2. In the **Jobs** panel, select **Tag**, provide a name to the job, and then add a text tag (for example **TestKey** as the key and **TestValue** as the value). Then click **Apply**.

**Serenity IoT DEV Platform**

Default | Get Link | Manage device groups | Query Devices | Jobs | Delete | Send Message

Jobs

Select job

- Tags
- Methods
- Properties

Tags on selected devices

Tags in common on selected devices

Job Name

Only letters and numbers are allowed

Test

KEY	VALUE	TYPE
TestKey	TestValue	Text

Summary

1 Affected devices

Apply | Cancel

Device Explorer

Search devices...

Device name	Simulated	Device type	Firmware	Telemetry
joetest	No	---	---	messageSche
rule_test	No	---	---	---
sah_test	No	---	---	---
sah_test_device	No	---	---	---
sah_test,test1	No	---	---	---
sahh_test	No	---	---	---
Test	No	---	---	---
test-tel1	No	---	---	---
test-telemetry	No	---	---	---
VisTest	No	---	---	messageSche

- To track the status of the job, click **View job status**.

**Serenity IoT DEV Platform**

Default | Get Link | Manage device groups | Query Devices | Jobs

Jobs

Tags on selected devices

Tags in common on selected devices

Job Name

Only letters and numbers are allowed

Test

KEY	VALUE	TYPE
TestKey	TestValue	Text

Summary

1 Job submitted successfully! ✓

View job status | Close

Device Explorer

Search devices...

Device name	Simulated	Device type	Firmware	Telemetry
sahh_test	No	---	---	---
sah_test_device	No	---	---	---
sah_test,test1	No	---	---	---
sahh_test	No	---	---	---
Test	No	---	---	---
test-tel1	No	---	---	---
test_telemetry	No	---	---	---

- After the job completes, verify the tags on the device by using the **Device Details** panel.

**3M Serenity IoT DEV Platform**

Last hour ▾

# Test-2b4d351e-2fa2-46e3-8020-50dc754996d9

Expand Columns

Job Name	Status	Operation	No. of devices	Succeeded	Failed	Start time	End time
Test-2b4d351e-2fa2-46e3-8020-50dc754996d9	Completed		1	1	0	01:01:23 PM 10.21...	01:01:28 PM 10.21...

Expand Columns

Job Name	Status	Device ID affected	Last return message	Start time	End time
Test-2b4d351e-2fa2-46e3-8020-50dc754996d9	Completed	Test	Completed	01:01:23 PM 10.21.2020	01:01:28 PM 10.21.2020

**3M Serenity IoT DEV Platform**

Default  Manage device groups

# Device Explorer

<input type="checkbox"/> Device name	Simulated	Device type	Firmware	Telemetry
<a href="#">sah_test_device</a>	No	---	---	---
<a href="#">sah_test,test1</a>	No	---	---	---
<a href="#">sahh_test</a>	No	---	---	---
<input checked="" type="checkbox"/> <a href="#">Test</a>	No	---	---	---
<a href="#">test-tel1</a>	No	---	---	---
<a href="#">test-telemetry</a>	No	---	---	---
<a href="#">VisTest</a>	No	---	---	messageSche

Displaying in local timezone: Asia/Calcutta

Tags

To add, delete, or change a tag on one or more devices, close this pane, select the checkbox for the device(s), click **Jobs**, and then select **Tags**.

KEY	VALUE
TestKey	TestValue

Methods

To run a method on one or more devices, close this pane, select the checkbox for the device(s), click **Jobs**, and then select **Methods**.

No methods found for this device.

## Create filters

You can use tag values to create filters. You can do this two ways:

1. **Query Devices** panel.
  2. **Manage device groups** panel.

**Query Devices** is useful for creating a quick filter over the device grid, below are the steps to follow:

- On the **Device Explorer** page, click **Query Devices**.

Default + New device

## Device Explorer

<input type="checkbox"/> Device name	Simulated	Device type	Firmware	Telemetry	Status	Last connection
<input type="checkbox"/> sah_test_device	No	---	---	---	Offline	05:02:16 PM 10.15.20...
<input type="checkbox"/> sah_test,test1	No	---	---	---	Offline	---
<input type="checkbox"/> sahh_test	No	---	---	---	Offline	---
<input type="checkbox"/> Test	No	---	---	---	Offline	---
<input type="checkbox"/> test-tel1	No	---	---	---	Offline	05:08:39 PM 10.15.20...
<input type="checkbox"/> test-telemetry	No	---	---	---	Offline	05:40:07 PM 10.15.20...
<input type="checkbox"/> VisTest	No	---	---	messageSchema	Offline	---

- Create a condition to filter the devices, for example, create a text filter that uses the tag name **TestKey** and value **TestValue** in the condition. Then click **Query**.

Default

## Device Explorer

<input type="checkbox"/> Device name	Simulated	Device type	Firmware	Telemetry
<input type="checkbox"/> sah_test_device	No	---	---	---
<input type="checkbox"/> sah_test,test1	No	---	---	---
<input type="checkbox"/> sahh_test	No	---	---	---
<input type="checkbox"/> Test	No	---	---	---
<input type="checkbox"/> test-tel1	No	---	---	---
<input type="checkbox"/> test-telemetry	No	---	---	---
<input type="checkbox"/> VisTest	No	---	---	messageSche

Condition 1

Field \*

Operator \*

Value \*

Type \*

- To remove the filter, click **Reset query**.

The screenshot shows the 3M Serenity IoT DEV Platform's Device Explorer page. On the left is a sidebar with links: Dashboard, Device Explorer (which is selected and highlighted in grey), User Management, Rules, Packages, Deployments, and Maintenance. The main area has a title 'Device Explorer' and a search bar 'Search devices...'. Below is a table with columns: Device name, Simulated, Device type, Firmware, Telemetry, Status, and Last connection. A single row is shown for a device named 'Test'.

- To save the filter, click **Device group**.

This screenshot shows the same Device Explorer interface as above, but with a 'Query Devices' panel open on the right. The panel contains a query builder with fields for Field (Tags.TestKey), Operator (= Equals), Value (TestValue), and Type (Text). It also includes a 'Remove condition' button and buttons for 'Query', '+ Device group', and 'Reset'.

The **Manage device groups** panel is useful for creating new device groups and managing them, for more info [click here](#)

## More

Device Explorer also provides a few more features, listed below:

1. Show telemetry
2. Send Message
3. SIM Management

### Show telemetry

The **Show telemetry** page is useful to show one or more devices' telemetry on a separate page.

Select one or more devices in the grid and click the **Show telemetry** button.

Default Get Link Manage device groups Query Devices Jobs Delete Send Message Show telemetry + New device ☰

## Device Explorer

Search devices...

Expand Columns

Device name	Simulated	Device type	Firmware	Telemetry	Status	Last connection
joetest	No	---	---	messageSchema	Offline	08:54:35 PM 08.06.2020
rule_test	No	---	---	---	Offline	---
sah_test	No	---	---	---	Offline	04:23:02 PM 09.11.2020
sah_test_device	No	---	---	---	Offline	05:02:16 PM 10.15.2020
sah_test.test1	No	---	---	---	Offline	---
sahh_test	No	---	---	---	Offline	---
Test	No	---	---	---	Offline	---
<input checked="" type="checkbox"/> test-te1	No	---	---	---	Offline	05:08:39 PM 10.15.2020
<input checked="" type="checkbox"/> test-telemetry	No	---	---	---	Offline	05:40:07 PM 10.15.2020
VisTest	No	---	---	messageSchema	Offline	---



## Send Message

### TODO

### SIM Management

### TODO

# User Management

The **User Management** page is used to manage all the available users of the tenant.

This document covers the the following aspects of the **User Management** page:

1. [User Grid](#)
2. [New User Creation](#)
3. [Delete User](#)
4. [Add System Admin](#)
5. [Delete System Admin](#)
6. [Add Service principal](#)

## User Grid

The **User Grid** lists all the active users of the tenant.

The following shows the columns in the grid and their description:

COLUMN NAME	DESCRIPTION
Name	Name of the user
Role	Describes the role assigned to that particular user
Type	Indicates the current status of the user such as Invited, Member etc .

Name	Role	Type
95d3c662-23ea-4e2d-8d3d-ea2448706934	Admin	Client Credentials
aditya.tangirala@ggktech.com	Admin	Invited
Anant Choudhari CW	Admin	Member
Andrew Schmidt	Admin	Member
Anup Warade CW	Contributor	Member
atangirala.cw@mmm.com	Admin	Invited
Blake Ma	Admin	Member

- We have a search bar associated with the grid where we can search the grid data.

+ New user   + Add System Admin      + Add Service Principal

> Expand Columns

Name	Role	Type
TestAccount - Admin	Admin	Member
TestAccount - SystemAdmin	Admin	Member
TestAccount - User	ReadOnly	Member
testuser.selide@mmm.com	ReadOnly	Invited

1 to 7 of 7   First   Previous   Page 1 of 1   Next   Last

## New User

The following steps are used to add a new user for the tenant.

- Navigate to the **User Management** page from the left menu.
- Click on **+New User**.
  - Enter the email address of the user to be created.
  - Select the **UserRole** you want to assign.

+ New user   + Add System Admin

Invite users

Email Address

\* Is required

User Role

Select user role

Send Invite   Cancel

- Click the **Send Invite** button to send an email to the user.

The screenshot shows the User Management page with a modal window titled "Invite users". In the modal, an email address "vkotha.cw@mmm.com" is entered, and the user role is set to "Admin". A summary message indicates that the user was successfully invited with a checkmark icon.

Name	Role
95d3c662-23ea-4e2d-8d3d-ea2448706934	Admin
aditya.tangirala@ggktech.com	Admin
Anant Choudhari CW	Admin
Andrew Schmidt	Admin
Anup Warade CW	Contributor
atangirala.cw@mmm.com	Admin
Blake Ma	Admin

The screenshot shows the User Management page with a user account "Viswanadh Kotha CW" highlighted in a blue box. This account has the role "Admin" and the type "Invited".

Name	Role	Type
TestAccount - User	ReadOnly	Member
testuser.selide@mmm.com	ReadOnly	Invited
timothy.gunter@analysts.com	Admin	Invited
Viswanadh Kotha CW	Admin	Invited

- After the user accepts the invite via email, they will be added to the tenant and will be able to access the application.

The screenshot shows the User Management page with the same user account "Viswanadh Kotha CW" now listed as a "Member" instead of "Invited".

Name	Role	Type
TestAccount - User	ReadOnly	Member
testuser.selide@mmm.com	ReadOnly	Invited
timothy.gunter@analysts.com	Admin	Invited
Viswanadh Kotha CW	Admin	Member

## Delete User

To delete a user:

- Navigate to the **User Management** page from the left menu.

- Select the user to be deleted.

- Click the **Delete** button.

The screenshot shows the User Management interface. On the left is a sidebar with options: Dashboard, Device Explorer, User Management (selected), Rules, Packages, Deployments, and Maintenance. The main area is titled "User Management". It contains a search bar and a table with columns "Name" and "Role". A user named "vkotha.cw@mmm.com" is selected, indicated by a checked checkbox. At the top right, there are buttons for "+ New user" and "+ Add System Admin". A modal window titled "Delete" is open on the right, containing a "Delete users" section with a note about removing the selected user from the tenant and a toggle switch set to "Yes". Below it is a "Summary" section showing "1 Affected users".

- Activate the toggle button to delete the user and the **Delete** button will be activated.

This screenshot is similar to the previous one, showing the User Management interface with the "User Management" option selected in the sidebar. The modal window "Delete" is still open, but the "Delete users" section now includes a message: "This action will remove the selected user from this tenant." with a checked toggle switch. The "Delete" button inside the modal is highlighted with a red border.

- After deleting the user we see a message that the user has been deleted.

This screenshot shows the User Management interface after a user has been deleted. The sidebar and main table remain the same. The modal window "Delete" is closed. In the main area, a message box displays "1 Users deleted successfully! ✓".

# Add System Admin

This feature is used to add an existing user in the system as a system admin.

The system admin is the user who has access across all the tenants.

- Navigate to the **User Management** page from the left menu
- Click on **+Add System Admin**
  - A flyout opens with a dropdown where we can see the list of existing users who are non-system admins.

The screenshot shows the 'User Management' page in the 3M Serenity IoT DEV Platform. On the left, there's a sidebar with links like Dashboard, Device Explorer, User Management (which is selected and highlighted in dark grey), Rules, Packages, Deployments, and Maintenance. At the top right, there are buttons for '+ New user' and '+ Add System Admin'. A modal window titled 'Add System Admin' is open on the right. It has a dropdown labeled 'Select user' containing several options: 'TestAccount - User', 'testuser.selide@mmm.com', 'testuser.selide@mmm.com', 'testuser.selide@mmm.com', 'testuser.selide@mmm.com', 'timothy.gunter@analysts.com', and 'Viswanadh Kotha CW'. Below the dropdown are 'Add' and 'Cancel' buttons. The main 'User Management' table below the modal shows a list of users with checkboxes and roles: TestAccount - User (ReadOnly), testuser.selide@mmm.com (ReadOnly), testuser.selide@mmm.com (ReadOnly), testuser.selide@mmm.com (ReadOnly), testuser.selide@mmm.com (ReadOnly), timothy.gunter@analysts.com (Admin), and Viswanadh Kotha CW (Admin).

- Select the user from the dropdown; the **Add** button will be enabled.

The screenshot is similar to the previous one, but the 'Select user' dropdown now has a single option selected: 'TestAccount - Admin'. The 'Add' button in the flyout is now highlighted with a red border, indicating it is enabled. The rest of the interface remains the same, with the sidebar, main table, and other UI elements visible.

- On clicking on add button, the user will be added as system admin.

The screenshot shows the User Management page with a sidebar containing links like Dashboard, Device Explorer, User Management (selected), Rules, Packages, Deployments, and Maintenance. The main area has a search bar and a table listing users with columns for Name, Role, and Action (checkbox). A summary message in a modal says "Added system admin successfully! ✓".

Name	Role
TestAccount - User	ReadOnly
testuser.selide@mmm.com	ReadOnly
testuser.selide@mmm.com	ReadOnly
testuser.selide@mmm.com	ReadOnly
timothy.gunter@analysts.com	Admin
Viswanad Kotha CW	Admin

## Delete System Admin

This feature is used to delete a system admin from the system.

- Navigate to **User Management** page from the left menu
- Click on **Delete System Admin**
  - A flyout opens with a dropdown where we can see the list of existing system admins.

The screenshot shows the User Management page with a sidebar containing links like Dashboard, Device Explorer, User Management (selected), Rules, Packages, Deployments, and Maintenance. The main area has a search bar and a table listing users with columns for Name, Role, and Type. A dropdown menu is open, showing the same list of users. An "Expand Columns" button is visible at the top right of the table.

Name	Role	Type
TestAccount - User	ReadOnly	Member
testuser.selide@mmm.com	ReadOnly	Invited
testuser.selide@mmm.com	ReadOnly	Invited
testuser.selide@mmm.com	ReadOnly	Invited
timothy.gunter@analysts.com	Admin	Invited
Viswanad Kotha CW	Admin	Member

The screenshot shows the User Management interface. On the left is a sidebar with navigation links: Dashboard, Device Explorer, User Management (selected), Rules, Packages, Deployments, and Maintenance. The main area has a search bar and a table listing users with checkboxes and roles. A modal window titled "Delete System Admin" is open, prompting the user to "Select user".

Name	Role
TestAccount - User	ReadOnly
testuser.selide@mmm.com	ReadOnly
timothy.gunter@analysts.com	Admin
Viswanadha Kotha CW	Admin

- Select the user from the dropdown; the **Delete** button will be enabled.

This screenshot is similar to the previous one, but the "Delete" button in the modal is now highlighted in red, indicating it is active.

- Click the **Delete** button; the user will be deleted as a system admin.

The screenshot shows the User Management interface after a deletion. The modal now displays a "Summary" message: "Users deleted successfully! ✓".

Name	Role
95d3c662-23ea-4e2d-8d3d-ea2448706934	Admin
aditya.tangirala@ggktech.com	Admin
Anant Choudhari CW	Admin
Andrew Schmidt	Admin
Anup Warade CW	Contributor
atangirala.cw@mmm.com	Admin
Blake Ma	Admin

# Add Service Principal

This feature is used to add an existing user in the system to the role of service principal.

- Navigate to **User Management** page from the left menu.
- Click on **+Add Service Principal**.
  - A flyout opens to enter the **Service principal app id** and select the user role.

The screenshot shows the 'User Management' page in the 3M Serenity IoT DEV Platform. On the left is a sidebar with navigation links: Dashboard, Device Explorer, User Management (which is selected and highlighted in dark grey), Rules, Packages, Deployments, and Maintenance. The main content area has a title 'User Management' and a search bar labeled 'Search users...'. Below the search bar is a table with columns: Name, Role, and Type. The table lists eight users:

Name	Role	Type
95d3c662-23ea-4e2d-8d3d-ea2448706934	Admin	Client Credentials
aditya.tangirala@ggktech.com	Admin	Invited
Anant Choudhari CW	Admin	Member
Andrew Schmidt	Admin	Member
Anup Warade CW	Contributor	Member
atangirala.cw@mmm.com	Admin	Invited
Blake Ma	Admin	Member

The screenshot shows the 'User Management' page with an open flyout titled 'Invite users'. The flyout contains two input fields: 'Service principal app id' (with a placeholder value) and 'User Role' (with a dropdown menu). At the bottom right of the flyout are 'Add' and 'Cancel' buttons. The rest of the page is identical to the first screenshot, showing the list of users in the main content area.

- Enter the **Service principal app id** and Select the user role from the dropdown; the **Add** button will be enabled.

The screenshot shows the User Management page with a modal window titled "Invite users". The modal contains fields for "Service principal app id" (95d3c662-23ea-4e2d-8d3d-ea2448706934) and "User Role" (Admin). A red "Add" button is at the bottom right of the modal.

Name	Role
95d3c662-23ea-4e2d-8d3d-ea2448706934	Admin
aditya.tangirala@ggktech.com	Admin
Anant Choudhari CW	Admin
Andrew Schmidt	Admin
Anup Warade CW	Contributor
atangirala.cw@mmm.com	Admin
Blake Ma	Admin

- Click the **Add** button, and the user will be added to the chosen role.

The screenshot shows the User Management page with a summary message in a modal: "Summary: Added service principal ✓". A "Close" button is at the bottom right of the modal.

Name	Role
95d3c662-23ea-4e2d-8d3d-ea2448706934	Admin
95d3c662-23ea-4e2d-8d3d-ea2448706935	Admin
aditya.tangirala@ggktech.com	Admin
Anant Choudhari CW	Admin
Andrew Schmidt	Admin
Anup Warade CW	Contributor
atangirala.cw@mmm.com	Admin

- After adding the service principal, the user will appear in the grid with the type **Client Credentials**.

The screenshot shows the User Management page with a table. One row is highlighted with a blue border, showing a user with the "Type" column value "Client Credentials".

Name	Role	Type
95d3c662-23ea-4e2d-8d3d-ea2448706934	Admin	Client Credentials
aditya.tangirala@ggktech.com	Admin	Invited
Anant Choudhari CW	Admin	Member
Andrew Schmidt	Admin	Member
Anup Warade CW	Contributor	Member
atangirala.cw@mmm.com	Admin	Invited
Blake Ma	Admin	Member

# Rules

The **Rules** page is used to manage all the active rules. A rule ensures that warnings will be issued whenever a device satisfies the rule's condition.

This document covers the the following aspects of the **Rules** page:

1. [Rules Grid](#)
2. [Rule Details Flyout](#)
3. [Editing Rules](#)
4. [Creating a New Rule](#)
5. [Deleting a Rule](#)
6. [Rule Status](#)

## Rules Grid

- Navigate to the **Rules** page from the left menu
- The **Rules Grid** lists all the active rules for a device group.

The following table lists the columns and their usage in the **Rules Grid**.

COLUMN NAME	DESCRIPTION
Rule Name	Name of the rule
Description	Describes the rule
Severity	Describes the severity level of the rule
Device Group	Indicates the device group that the rule belongs to
Trigger	Describes the rule's triggering condition
Notification Type	Describes the notification type when the rule triggers
Status	Shows whether the rule is enabled or disabled
Last Trigger	Displays the rule's last trigger date

**3M Serenity IoT DEV Platform**

Rule name	Description	Severity	Device group	Trigger	Notification type	Status	Last trigger
New rule	Sample rule to test	Info	joetest	temperature	Maintenance log	Enabled	---
Rule1	TestDesc	Critical	Default	temperature	Maintenance log	Enabled	05:42:00 PM 10.15.2020
Rule11	TestDesc	Critical	Default	temperature	Maintenance log	Enabled	05:42:00 PM 10.15.2020
Rule12	TestDesc	Critical	Default	temperature	Maintenance log	Enabled	05:42:00 PM 10.15.2020
Rule13	TestDesc	Critical	Default	temperature	Maintenance log	Enabled	05:42:00 PM 10.15.2020
Rule14	TestDesc	Critical	Default	temperature	Maintenance log	Enabled	05:42:00 PM 10.15.2020
Rule15	TestDesc	Critical	Default	temperature	Maintenance log	Enabled	05:42:00 PM 10.15.2020

1 to 17 of 17 First Previous Page 1 of 1 Next Last

- Note the search bar located above the grid.

Rule name	Description	Severity	Device group	Trigger	Notification type	Status	Last trigger
New rule	Sample rule to test	Info	joetest	temperature	Maintenance log	Enabled	---

1 to 1 of 1 First Previous Page 1 of 1 Next Last

## Rule Details

- Click on a rule name
- The **Rule Details** flyout displays the details of the selected rule.

The screenshot shows the Serenity IoT DEV Platform interface. On the left is a navigation sidebar with links like Dashboard, Device Explorer, User Management, Rules (which is selected), Packages, Deployments, and Maintenance. The main area displays a table of rules:

Rule name	Description	Severity	Device group	Trigger	Notification type	Status
New rule	Sample rule to test	Info	joetest	temperature	Maintenance log	Enabled
Rule1	TestDesc	Critical	Default	temperature	Maintenance log	Enabled
Rule11	TestDesc	Critical	Default	temperature	Maintenance log	Enabled
Rule12	TestDesc	Critical	Default	temperature	Maintenance log	Enabled
Rule13	TestDesc	Critical	Default	temperature	Maintenance log	Enabled
Rule14	TestDesc	Critical	Default	temperature	Maintenance log	Enabled
Rule15	TestDesc	Critical	Default	temperature	Maintenance log	Enabled
Rule16	TestDesc	Critical	Default	temperature	Maintenance log	Enabled
Rule17	TestDesc	Critical	Default	temperature	Maintenance log	Enabled
Rule2	TestDesc	Critical	Default	temperature	Maintenance log	Enabled
Rule20	Rule Description	Critical	Default	temperature	Maintenance log	Enabled

To the right of the table, a modal window titled "Rule Details" is open for Rule20. It contains sections for Rule Description, Device group (Default), Calculation (Average), Time period (in minutes) (1), Conditions (with a table showing temperature = Equals 1), Severity level (Critical), and Rule status (Enabled). At the bottom of the modal, it says "13 devices affected by this rule".

## Editing a Rule

- You can use the **Rule Details** flyout to edit a rule. Click **Edit**.

This screenshot is identical to the one above, showing the Serenity IoT DEV Platform dashboard with the "Edit" button highlighted in the "Rule Details" flyout for Rule20.

- You will see the details of rule.

- Update the data accordingly and click **Apply**, saving the new rule data.

## New Rule

This option is used to create a new rule associated with a device group.

- Click on **+New Rule**
- Enter your **Rule name**.
- Enter your **Rule Description**.
- Select a **Device group** for the rule.
- Select the **Calculation Type**.
- If the **Calculation Type** is **Average** then select **Time Period**.
- Add a Condition for the rule.
  - Select the **Field** to be added.
  - Select the **Operator** to be compared.
  - Enter a **Value** to be compared.
- Select a **Severity Level** you want to assign to the rule.

- Select the **Rule Status**.

The screenshot shows the Serenity IoT DEV Platform interface. On the left is a navigation sidebar with options: Dashboard, Device Explorer, User Management, Rules (selected), Packages, Deployments, and Maintenance. The main area displays a 'Rules' grid with columns: Rule name, Description, Severity, Device group, Trigger, Notification type, and Status. A search bar at the top of the grid allows searching by rule name. To the right of the grid is a 'New rule' configuration dialog. The 'Rule name' field contains 'Rule20'. The 'Severity level' section has 'Critical' selected. The 'Action' section has 'Email disabled' checked. The 'Rule status' section has 'Enabled' checked. A note at the bottom right of the dialog states '13 devices affected by this rule'.

- Click the **Apply** button to save the new rule.

This screenshot is similar to the previous one, showing the 'New rule' configuration dialog. The 'Rule name' is still 'Rule20'. The 'Severity level' section has 'Critical' selected. The 'Action' section has 'Email disabled' checked. The 'Rule status' section has 'Enabled' checked. The note at the bottom right now includes a red 'Apply' button and a 'Cancel' button. The note also states '13 devices affected by this rule'.

- After saving the rule, the rule will appear in the **Rules Grid**.

The screenshot shows the 'Rules' grid after the rule has been saved. The 'Rule name' column now includes 'Rule20' in the list. The 'Severity' column for 'Rule20' shows 'Critical'. The 'Status' column for 'Rule20' shows 'Enabled'. The note at the bottom right of the grid states '13 devices affected by this rule'.

# Deleting a Rule

- Select the rule to be deleted by clicking in the checkbox next to the rule's name.
- Click the **Delete** button

The screenshot shows the 'Rules' section of the Serenity IoT DEV Platform. On the left is a sidebar with navigation links: Dashboard, Device Explorer, User Management, Rules (which is selected and highlighted in dark grey), Packages, Deployments, and Maintenance. The main area is a table listing rules. The columns are: Rule name, Description, Severity, Device group, Trigger, Notification type, Status, and Last trigger. A search bar at the top says 'Search rules...'. The 'Delete' button in the top right corner is highlighted with a blue border. The table lists 18 rules, with 'Rule20' being the one selected for deletion.

- Activate the toggle button to delete the rule and the **Delete** button will be activated.

The screenshot shows the same interface as the previous one, but with a modal dialog box overlaid. The dialog is titled 'Delete Rule' and contains the following information:

- Rule20
- Rule Description
- 13 devices affected by this rule

A red rectangular button labeled 'Delete' is prominently displayed at the bottom right of the dialog. The rest of the page remains visible in the background.

- After deleting the rule you will see a message confirming that the rule has been deleted.

The screenshot shows the Serenity IoT DEV Platform interface. On the left, there's a sidebar with navigation links: Dashboard, Device Explorer, User Management, Rules (selected), Packages, Deployments, and Maintenance. The main area displays a table of rules. One rule, 'Rule20', is selected, and a context menu is open over it. The menu includes options like 'Delete Rule', 'Rule Description', and a status indicator '13 devices affected by this rule'. A confirmation message 'Rule has been deleted' with a checkmark is shown, along with a note about deleting alerts. A red 'Close' button is at the bottom right of the confirmation box.

## Rule Status

This feature is used to enable/disable a rule.

- Select a rule by clicking the checkbox next to its name, and then click the status toggle button in the toolbar.
- A flyout opens to change the status of the **Rule Status**.

The screenshot shows the Serenity IoT DEV Platform interface. The 'Rules' tab is selected in the sidebar. The main area shows a table of rules. The 'Disable' button in the toolbar is highlighted with a blue border. A context menu is open over Rule1, with the 'Disable' option selected. The table includes columns for Rule name, Description, Severity, Device group, Trigger, Notification type, Status, and Last trigger. The status column for Rule1 shows 'Enabled'. The last trigger for Rule1 is listed as '05:42:00 PM 10.15.2020'.

- After confirming that the status toggle button in the flyout is the correct action, click **Apply**.

Change status

Disable

Rule1  
TestDesc

13 devices affected by this rule

Apply Cancel

Rule name	Description	Severity	Device group	Trigger	Notification type
New rule	Sample rule to test	Info	joetest	temperature	Maintainanc
Rule1	TestDesc	Critical	Default	temperature	Maintenanc
Rule11	TestDesc	Critical	Default	temperature	Maintenanc
Rule12	TestDesc	Critical	Default	temperature	Maintenanc
Rule13	TestDesc	Critical	Default	temperature	Maintenanc
Rule14	TestDesc	Critical	Default	temperature	Maintenanc
Rule15	TestDesc	Critical	Default	temperature	Maintenanc

- The status of the rule will be updated accordingly.

Change status

Enable

Rule1  
TestDesc

13 devices affected by this rule ✓

Apply Cancel

Rule name	Description	Severity	Device group	Trigger	Notification type
New rule	Sample rule to test	Info	joetest	temperature	Maintenanc
Rule1	TestDesc	Critical	Default	temperature	Maintenanc
Rule11	TestDesc	Critical	Default	temperature	Maintenanc
Rule12	TestDesc	Critical	Default	temperature	Maintenanc
Rule13	TestDesc	Critical	Default	temperature	Maintenanc
Rule14	TestDesc	Critical	Default	temperature	Maintenanc
Rule15	TestDesc	Critical	Default	temperature	Maintenanc

# Packages

**Packages** are the device configuration files which are used in the deployment process to configure the desired properties of devices.

This document covers the following aspects of packages:

1. [Packages Grid](#)
2. [Package Creation](#)
3. [Deactivate Package](#)
4. [Activate Package](#)
5. [Delete Package](#)

## Packages Grid

The **Packages** grid lists all of the packages, whether active or deactivated, that are available for Deployments.

The following are the column headings in the **Packages** grid and their usage:

COLUMN NAME	DESCRIPTION
Name	Name of the package
Package Type	Describes the device category that the package is targeting
Configuration Type	Indicates whether the package is targeting firmware updates or custom updates
Date Created	Date the package was created
Active	Indicates Whether the package is active
Version	Indicates The version of the package
Last Modified Date	Date the package was created or was last modified
Last Modified By	The user who created the package or the user who last modified package

## Package Creation

Following are the steps for creating a package to update device firmware:

- Navigate to the **Packages** page from the left menu.
- Click on **+New Package**.
  - Select the **Package Type** based on the targeted device types, **Device Configuration** for IoT devices, or **Edge Manifest** for edge devices.

The screenshot shows the 3M Serenity IoT DEV Platform interface. On the left, there's a sidebar with navigation links: Dashboard, Device Explorer, User Management, Rules, Packages (which is selected and highlighted in dark grey), Deployments, and Maintenance. The main content area is titled "Packages" and lists several packages in a table:

	Name	Package Type	Configuration T...	Date Created	Active
<input type="checkbox"/>	be20e009-9664-4...	Device Configuration	Firmware	01:01:47 PM 10.0...	✓
<input type="checkbox"/>	be20e009-9664-4...	Device Configuration	Firmware	01:02:09 PM 10.0...	✓
<input type="checkbox"/>	Package1234	Device Configuration	Firmware	12:31:25 PM 08.2...	✗
<input type="checkbox"/>	sample	Device Configuration	Firmware	06:26:55 PM 09.2...	✓
<input type="checkbox"/>	SW1Package.json	Device Configuration	Firmware	01:18:38 PM 08.0...	✓
<input type="checkbox"/>	test.json	Device Configuration	Firmware	07:22:57 PM 08.2...	✓

A modal window titled "New Package" is open on the right side of the screen. It has a header "New Package" with a close button "X". Below the header, there are sections for "Upload a package" (with a placeholder "Add a package to your solution") and "Package Type \*". A dropdown menu titled "Select package type" is open, showing two options: "Edge Manifest" and "Device Configuration". The "Edge Manifest" option is currently selected. There are also input fields for "devicegroup.\*" and "devicegroup.cd9bba84-274a-4d69-8e49-81f4933d3a36", each with a small trash can icon. A "Cancel" button is located at the bottom right of the modal.

## ■ Package Type - Edge Manifest

- For edge devices, you should upload the edge device package content file for package creation.
- Click **Browse**.



## New Package

Upload a package

Add a package to your solution

Package Type \*

Edge Manifest

Browse for a package file

Tags

devicegroup.\*

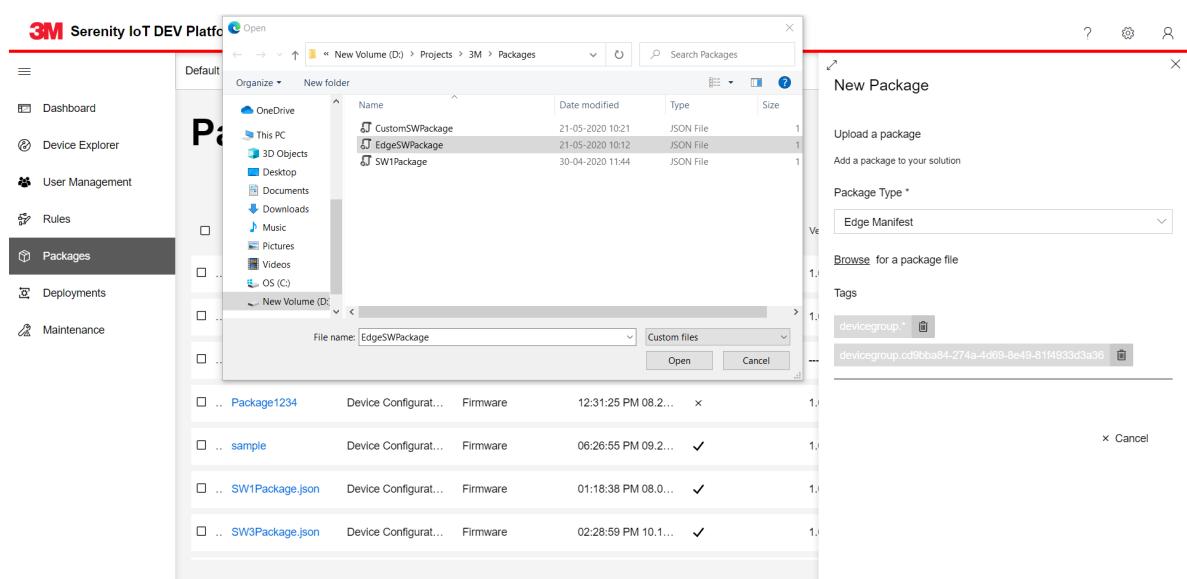


devicegroup.cd9bba84-274a-4d69-8e49-81f4933d3a36



Cancel

- Select the device configuration file.



- Once the file is selected, update the **Package Name**, which is prepopulated with FileName.



## New Package

Upload a package

Add a package to your solution

Package Type \*

Edge Manifest



[Browse](#) for a package file

Package Name \*

EdgeSWPackage.json

Tags

devicegroup.\*



devicegroup.cd9bba84-274a-4d69-8e49-81f4933d3a36



1

Package

EdgeSWPackage.json

- Click on Upload to create the package.



## New Package

Upload a package

Add a package to your solution

Package Type \*

Edge Manifest



[Browse](#) for a package file

Package Name \*

EdgeSWPackage

Tags

devicegroup.\*



devicegroup.cd9bba84-274a-4d69-8e49-81f4933d3a36



1 Package

EdgeSWPackage.json

Upload

Cancel

- Once the package is created, you will be presented with the created package information, which will also be reflected in the Packages Grid.



## New Package

Upload a package

Add a package to your solution

Package Type \*

EdgeManifest

Package Name \*

EdgeSWPackage

Tags

devicegroup.\*



devicegroup.cd9bba84-274a-4d69-8e49-81f4933d3a36



1 Package ✓

EdgeSWPackage.json

To deploy packages, go to the [Deployments page](#), and then click **+ New Deployment**.

Close

### ■ Package Type - Device Configuration

- Select the **Configuration type**, **Firmware** or **Custom**

The screenshot shows the 'Packages' section of the 3M Serenity IoT DEV Platform. On the left, there's a sidebar with navigation links: Dashboard, Device Explorer, User Management, Rules, Packages (which is selected and highlighted in grey), Deployments, and Maintenance. The main area displays a list of packages:

Name	Package Type	Configuration T...	Date Created	Active
be20e009-9664-4...	Device Configuration	Firmware	01:01:47 PM 10.0...	✓
be20e009-9664-4...	Device Configuration	Firmware	01:02:09 PM 10.0...	✓
Package1234	Device Configuration	Firmware	12:31:25 PM 08.2...	✗
sample	Device Configuration	Firmware	06:26:55 PM 09.2...	✓
SW1Package.json	Device Configuration	Firmware	01:18:38 PM 08.0...	✓
test.json	Device Configuration	Firmware	07:22:57 PM 08.2...	✓

A modal window titled 'New Package' is open on the right side of the screen. It contains fields for 'Upload a package' and 'Add a package to your solution'. Under 'Package Type \*', 'Device Configuration' is selected. Under 'Configuration type \*', 'Select configuration type' is shown with options 'Firmware' and 'Custom'. Below these are two input fields labeled 'devicegroup.\*' containing 'devicegroup.c09bba84-274a-4d69-8e49-81f4933d3a36'.

## ■ Configuration Type - Firmware

- After selecting the **Firmware** configuration type, you should upload the package content file with the desired device updates.

The screenshot shows the 'Packages' section of the 3M Serenity IoT DEV Platform. On the left, there's a sidebar with navigation links: Dashboard, Device Explorer, User Management, Rules, Packages (selected), Deployments, and Maintenance. The main area displays a list of packages. In the center, a file selection dialog is open, showing a list of files under 'OneDrive' and 'New Volume (D:)'. The file 'SW1Package' is selected. On the right, the 'New Package' modal is open with 'Firmware' selected as the configuration type.

- Once the file is selected, provide a name and version for the package.

A screenshot of a 'New Package' dialog box. At the top left is a back arrow icon. The title 'New Package' is centered above a dropdown menu set to 'Device Configuration'. Below it is another dropdown set to 'Firmware'. A 'Browse' button with the placeholder 'for a firmware file' is present. The 'Package Name \*' field contains 'SW1Package.json', which is highlighted with a blue border. The 'Version \*' field contains '1.0.0', also highlighted with a blue border. Under 'Tags', there are two entries: 'devicegroup.\*' and 'devicegroup.cd9bba84-274a-4d69-8e49-81f4933d3a36', each with a small trash can icon to its right. A horizontal line separates this from the 'Package' section. In the 'Package' section, the file 'SW1Package.json' is listed. At the bottom right are red 'Upload' and 'Cancel' buttons.

New Package

Device Configuration

Configuration type \*

Firmware

[Browse](#) for a firmware file

Package Name \*

SW1Package.json

Version \*

1.0.0

Tags

devicegroup.\* 

devicegroup.cd9bba84-274a-4d69-8e49-81f4933d3a36 

---

Package

SW1Package.json

- Then Click **Upload**; once uploading completes, you will be provided with the preview of the package file contents, which can be modified as necessary.



## New Package

### Firmware JSON

```
1      {
2        content: {
3          deviceContent: {
4            'properties.desired.softwareConfig': {
5              softwareName: 'Firmware',
6              version: '1.0.0',
7              softwareURL:
8                'https://crsliotstorageacctdev.blob.core.windows.net/233a1
9                ca2-6855-43c5-8b9c-c7f85a1dd520-software-
10               package/SW1Package.json?sv=2018-03-
11               28&sr=b&sig=ioYikP%2BAxGBTXonE45z3t33MscbEqPYcmI1ay%2Bsg
12               w%3D&st=2020-10-14T08%3A28%3A52Z&se=2020-10-
13               15T08%3A33%3A52Z&sp=rw',
14               fileName: 'SW1Package.json',
15               serialNumber: '',
16               checkSum:
17                 '16b557cb7b4828cefe71b9a9fdadb534bdce6089'
18             }
19           }
20         },
21         metrics: {
22           queries: {
23             current: 'SELECT deviceId FROM devices WHERE
24             configurations.[[${$deployment.id}]].status = \'Applied\''
25             AND properties.reported.softwareConfig.version =
26             properties.desired.softwareConfig.version AND
             properties.reported.softwareConfig.status=\'Success\'',
             applying: 'SELECT deviceId FROM devices WHERE
             configurations.[[${$deployment.id}]].status = \'Applied\''
             AND (
               properties.reported.softwareConfig.status=\\'Downloading\''
               OR properties.reported.softwareConfig.status=\\'Verifying\''
               OR
               properties.reported.softwareConfig.status=\\'Applying\''),
             rebooting: 'SELECT deviceId FROM devices WHERE
             configurations.[[${$deployment.id}]].status = \'Applied\''
```



- Once you verify the JSON, click **Upload** to create the package.



## New Package

```
19      }
20  },
21  metrics: {
22    queries: {
23      current: 'SELECT deviceId FROM devices WHERE
24 configurations.[[ ${deployment.id} ]].status = \'Applied\''
25 AND properties.reported.softwareConfig.version =
26 properties.desired.softwareConfig.version AND
27 properties.reported.softwareConfig.status=\'Success\'',
28      applying: 'SELECT deviceId FROM devices WHERE
29 configurations.[[ ${deployment.id} ]].status = \'Applied\''
30 AND (
31      properties.reported.softwareConfig.status=\'Downloading\''
32      OR properties.reported.softwareConfig.status=\'Verifying\''
33      OR
34      properties.reported.softwareConfig.status=\'Applying\'',
35      rebooting: 'SELECT deviceId FROM devices WHERE
36 configurations.[[ ${deployment.id} ]].status = \'Applied\''
37 
```

### Tags

devicegroup.\*



devicegroup.cd9bba84-274a-4d69-8e49-81f4933d3a36



1 Package

SW1Package.json

Upload

Cancel

- Once the package is created, you can find the package in the **Packages Grid**.



## New Package

Upload a package

Add a package to your solution

Package Type \*

DeviceConfiguration

Configuration type \*

Firmware

Package Name \*

TestPackage

Version \*

1.0.0

Tags

devicegroup.\*



devicegroup.cd9bba84-274a-4d69-8e49-81f4933d3a36



1 Package ✓

---

### ■ Configuration Type - Custom

- After selecting the **Custom** configuration type, enter a name for the configuration and browse for the **Configuration File**.



- Once the file is selected, enter a **Package Name**.



## New Package

Upload a package

Add a package to your solution

Package Type \*

Device Configuration

Configuration type \*

Custom

Custom configuration type \*

TestConfiguration

Browse for a package file

Package Name \*

TestCustomSWPackage

Tags

devicegroup.\*

devicegroup.cd9bba84-274a-4d69-8e49-81f4933d3a36

- Click on Upload to create the package.



## New Package

Configuration type \*

Custom



Custom configuration type \*

TestConfiguration

[Browse](#) for a package file

Package Name \*

TestCustomSWPackage

Tags

devicegroup.\*



devicegroup.cd9bba84-274a-4d69-8e49-81f4933d3a36



1 Package

CustomSWPackage.json

Upload

x Cancel

- Once Package is created, you will see the created package information, which will also be reflected in the **Packages Grid**.



## New Package

Package Type \*

DeviceConfiguration

Configuration type \*

Custom - TestConfiguration

Package Name \*

TestCustomSWPackage

Tags

devicegroup.\*

devicegroup.cd9bba84-274a-4d69-8e49-81f4933d3a36

1 Package ✓

CustomSWPackage.json

To deploy packages, go to the [Deployments page](#), and then click **New Deployment**.

Close

## Deactivate Package

Active packages can be deactivated, so that they are not available for deployment.

Steps to Deactivate a Package:

- Select an Active Package (which is marked with a ✓ in the **Active** column) by clicking on the checkbox adjacent to the package name in the grid.

Default Get Link Manage device groups + New Package

## Packages

> Expand Columns

<input type="checkbox"/>	Name	Package Type	Configuration T...	Date Created	Active	Version	Last Modified D...	Last Modified By
<input checked="" type="checkbox"/>	TestSWPackage	Device Configurat...	Firmware	11:25:05 AM 10.1...	✓	1.0.0	11:26:13 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	EdgeSWPackage	Edge Manifest	---	10:57:16 AM 10.1...	✓	---	10:57:16 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	TestCustomSWPa...	Device Configurat...	Custom - TestCon...	10:45:52 AM 10.1...	✓	---	10:45:52 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	SW3Package.json	Device Configurat...	Firmware	02:28:59 PM 10.1...	✓	1.0.0	02:28:59 PM 10.1...	Ragavender Bas...
<input type="checkbox"/>	TestPackage	Device Configurat...	Firmware	02:04:30 PM 10.1...	✓	1.0.0	11:23:51 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	EdgeSWPackage....	Edge Manifest	---	01:55:05 PM 10.1...	✗	---	11:22:53 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	sample	Device Configurat...	Firmware	06:26:55 PM 09.2...	✓	1.0.0	06:26:55 PM 09.2...	Jayasimha Nallag...

1 to 10 of 10 First Previous Page 1 of 1 Next Last

- You will be presented with two options in the toolbar:

- o Deactivate
- o Delete

Default Get Link Manage device groups x Deactivate >Delete + New Package

## Packages

> Expand Columns

<input type="checkbox"/>	Name	Package Type	Configuration T...	Date Created	Active	Version	Last Modified D...	Last Modified By
<input checked="" type="checkbox"/>	TestSWPackage	Device Configurat...	Firmware	11:25:05 AM 10.1...	✓	1.0.0	11:25:05 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	EdgeSWPackage	Edge Manifest	---	10:57:16 AM 10.1...	✓	---	10:57:16 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	TestCustomSWPa...	Device Configurat...	Custom - TestCon...	10:45:52 AM 10.1...	✓	---	10:45:52 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	SW3Package.json	Device Configurat...	Firmware	02:28:59 PM 10.1...	✓	1.0.0	02:28:59 PM 10.1...	Ragavender Bas...
<input type="checkbox"/>	TestPackage	Device Configurat...	Firmware	02:04:30 PM 10.1...	✓	1.0.0	11:23:51 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	EdgeSWPackage....	Edge Manifest	---	01:55:05 PM 10.1...	✗	---	11:22:53 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	sample	Device Configurat...	Firmware	06:26:55 PM 09.2...	✓	1.0.0	06:26:55 PM 09.2...	Jayasimha Nallag...

1 to 10 of 10 First Previous Page 1 of 1 Next Last

- Click on Deactivate
- The package is deactivated and is presented in the grid with a **X** in the **Active** column

Packages

Name	Package Type	Configuration T...	Date Created	Active	Version	Last Modified D...	Last Modified By
TestSWPackage	Device Configurat...	Firmware	11:25:05 AM 10.1...	<input checked="" type="checkbox"/>	1.0.0	11:25:55 AM 10.1...	Ragavender Bas...
EdgeSWPackage	Edge Manifest	---	10:57:16 AM 10.1...	<input checked="" type="checkbox"/>	---	10:57:16 AM 10.1...	Ragavender Bas...
TestCustomSWPa...	Device Configurat...	Custom - TestCon...	10:45:52 AM 10.1...	<input checked="" type="checkbox"/>	---	10:45:52 AM 10.1...	Ragavender Bas...
SW3Package.json	Device Configurat...	Firmware	02:28:59 PM 10.1...	<input checked="" type="checkbox"/>	1.0.0	02:28:59 PM 10.1...	Ragavender Bas...
TestPackage	Device Configurat...	Firmware	02:04:30 PM 10.1...	<input checked="" type="checkbox"/>	1.0.0	11:23:51 AM 10.1...	Ragavender Bas...
EdgeSWPackage....	Edge Manifest	---	01:55:05 PM 10.1...	<input checked="" type="checkbox"/>	---	11:22:53 AM 10.1...	Ragavender Bas...
sample	Device Configurat...	Firmware	06:26:55 PM 09.2...	<input checked="" type="checkbox"/>	1.0.0	06:26:55 PM 09.2...	Jayasimha Nallag...

1 to 10 of 10 First Previous Page 1 of 1 Next Last

## Activate Package

Deactivated packages can be activated, so that they are available again for deployment.

Steps to Activate a Package:

- Select a deactivated package (which is marked with a **X** mark in the **Active** Column) by clicking on the checkbox adjacent to the package name in the grid.

Packages

Name	Package Type	Configuration T...	Date Created	Active	Version	Last Modified D...	Last Modified By
TestSWPackage	Device Configurat...	Firmware	11:25:05 AM 10.1...	<input checked="" type="checkbox"/>	1.0.0	11:25:55 AM 10.1...	Ragavender Bas...
EdgeSWPackage	Edge Manifest	---	10:57:16 AM 10.1...	<input checked="" type="checkbox"/>	---	10:57:16 AM 10.1...	Ragavender Bas...
TestCustomSWPa...	Device Configurat...	Custom - TestCon...	10:45:52 AM 10.1...	<input checked="" type="checkbox"/>	---	10:45:52 AM 10.1...	Ragavender Bas...
SW3Package.json	Device Configurat...	Firmware	02:28:59 PM 10.1...	<input checked="" type="checkbox"/>	1.0.0	02:28:59 PM 10.1...	Ragavender Bas...
TestPackage	Device Configurat...	Firmware	02:04:30 PM 10.1...	<input checked="" type="checkbox"/>	1.0.0	11:23:51 AM 10.1...	Ragavender Bas...
EdgeSWPackage....	Edge Manifest	---	01:55:05 PM 10.1...	<input checked="" type="checkbox"/>	---	11:22:53 AM 10.1...	Ragavender Bas...
sample	Device Configurat...	Firmware	06:26:55 PM 09.2...	<input checked="" type="checkbox"/>	1.0.0	06:26:55 PM 09.2...	Jayasimha Nallag...

1 to 10 of 10 First Previous Page 1 of 1 Next Last

- You will be presented with two options in the toolbar:

- Activate
- Delete

The screenshot shows the 'Packages' section of the Serenity IoT DEV Platform. A list of packages is displayed in a table. The first package, 'TestSWPackage', has its checkbox checked and its 'Active' column marked with a checkmark. Other packages listed include 'EdgeSWPackage', 'TestCustomSWPa...', 'SW3Package.json', 'TestPackage', 'EdgeSWPackage....', and 'sample'. The table includes columns for Name, Package Type, Configuration T..., Date Created, Active, Version, Last Modified D..., and Last Modified By.

	Name	Package Type	Configuration T...	Date Created	Active	Version	Last Modified D...	Last Modified By
<input checked="" type="checkbox"/>	TestSWPackage	Device Configurat...	Firmware	11:25:05 AM 10.1...	✗	1.0.0	11:26:55 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	EdgeSWPackage	Edge Manifest	---	10:57:16 AM 10.1...	✓	---	10:57:16 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	TestCustomSWPa...	Device Configurat...	Custom - TestCon...	10:45:52 AM 10.1...	✓	---	10:45:52 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	SW3Package.json	Device Configurat...	Firmware	02:28:59 PM 10.1...	✓	1.0.0	02:28:59 PM 10.1...	Ragavender Bas...
<input type="checkbox"/>	TestPackage	Device Configurat...	Firmware	02:04:30 PM 10.1...	✓	1.0.0	11:23:51 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	EdgeSWPackage....	Edge Manifest	---	01:55:05 PM 10.1...	✗	---	11:22:53 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	sample	Device Configurat...	Firmware	06:26:55 PM 09.2...	✓	1.0.0	06:26:55 PM 09.2...	Jayasimha Nallag...

1 to 10 of 10 First Previous Page 1 of 1 Next Last

- Click **Activate**
- The package is activated and has a **✓** mark in the **Active** column

This screenshot is identical to the one above, showing the 'Packages' section of the Serenity IoT DEV Platform. The 'TestSWPackage' row is selected, indicated by a blue highlight around its cells. The 'Delete' button is highlighted with a blue box.

## Delete Package

Packages which are no longer needed or that have expired package content can be deleted.

Steps to Delete a Package:

- Select a Package by clicking on the checkbox adjacent to the package name in the grid.

Default Get Link Manage device groups + New Package

## Packages

Expand Columns

	Name	Package Type	Configuration T...	Date Created	Active	Version	Last Modified D...	Last Modified By
<input type="checkbox"/>	TestSWPackage	Device Configurat...	Firmware	11:25:05 AM 10.1...	✓	1.0.0	11:26:13 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	EdgeSWPackage	Edge Manifest	---	10:57:16 AM 10.1...	✓	---	10:57:16 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	TestCustomSWPa...	Device Configurat...	Custom - TestCon...	10:45:52 AM 10.1...	✓	---	10:45:52 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	SW3Package.json	Device Configurat...	Firmware	02:28:59 PM 10.1...	✓	1.0.0	02:28:59 PM 10.1...	Ragavender Bas...
<input type="checkbox"/>	TestPackage	Device Configurat...	Firmware	02:04:30 PM 10.1...	✓	1.0.0	11:23:51 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	EdgeSWPackage....	Edge Manifest	---	01:55:05 PM 10.1...	✗	---	11:22:53 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	sample	Device Configurat...	Firmware	06:26:55 PM 09.2...	✓	1.0.0	06:26:55 PM 09.2...	Jayasimha Nallag...

1 to 10 of 10 First Previous Page 1 of 1 Next Last

- You will be presented with **Delete** in the toolbar.

Default Get Link Manage device groups Deactivate  + New Package

## Packages

Expand Columns

	Name	Package Type	Configuration T...	Date Created	Active	Version	Last Modified D...	Last Modified By
<input checked="" type="checkbox"/>	TestSWPackage	Device Configurat...	Firmware	11:25:05 AM 10.1...	✓	1.0.0	11:25:05 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	EdgeSWPackage	Edge Manifest	---	10:57:16 AM 10.1...	✓	---	10:57:16 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	TestCustomSWPa...	Device Configurat...	Custom - TestCon...	10:45:52 AM 10.1...	✓	---	10:45:52 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	SW3Package.json	Device Configurat...	Firmware	02:28:59 PM 10.1...	✓	1.0.0	02:28:59 PM 10.1...	Ragavender Bas...
<input type="checkbox"/>	TestPackage	Device Configurat...	Firmware	02:04:30 PM 10.1...	✓	1.0.0	11:23:51 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	EdgeSWPackage....	Edge Manifest	---	01:55:05 PM 10.1...	✗	---	11:22:53 AM 10.1...	Ragavender Bas...
<input type="checkbox"/>	sample	Device Configurat...	Firmware	06:26:55 PM 09.2...	✓	1.0.0	06:26:55 PM 09.2...	Jayasimha Nallag...

1 to 10 of 10 First Previous Page 1 of 1 Next Last

- Click **Delete**.

- You will see a confirmation popup to delete the package.

Packages

Name	Package Type	Version	Last Modified D...	Last Modified By			
TestSWPackage	Device Configuration	1.0.0	11:26:13 AM 10.1...	Ragavender Bas...			
EdgeSWPackage	Edge Manifest	---	10:57:16 AM 10.1...	Ragavender Bas...			
TestCustomSWPa...	Device Configuration	---	10:45:52 AM 10.1...	Ragavender Bas...			
SW3Package.json	Firmware	02:28:59 PM 10.1...	✓	1.0.0	02:28:59 PM 10.1...	Ragavender Bas...	
TestPackage	Device Configuration	02:04:30 PM 10.1...	✓	1.0.0	11:23:51 AM 10.1...	Ragavender Bas...	
EdgeSWPackage....	Edge Manifest	---	01:55:05 PM 10.1...	✗	---	11:22:53 AM 10.1...	Ragavender Bas...
sample	Device Configuration	06:26:55 PM 09.2...	✓	1.0.0	06:26:55 PM 09.2...	Jayasimha Nallag...	

- Click **Delete** or **Cancel** to delete the package or cancel the operation.
- Once you have deleted the package, the grid will be refreshed and populated with active and deactivated packages.

Packages

Name	Package Type	Configuration T...	Date Created	Active	Version	Last Modified D...	Last Modified By
EdgeSWPackage	Edge Manifest	---	10:57:16 AM 10.1...	✓	---	10:57:16 AM 10.1...	Ragavender Bas...
TestCustomSWPa...	Device Configuration	Custom - TestCon...	10:45:52 AM 10.1...	✓	---	10:45:52 AM 10.1...	Ragavender Bas...
SW3Package.json	Device Configuration	Firmware	02:28:59 PM 10.1...	✓	1.0.0	02:28:59 PM 10.1...	Ragavender Bas...
TestPackage	Device Configuration	Firmware	02:04:30 PM 10.1...	✓	1.0.0	11:23:51 AM 10.1...	Ragavender Bas...
EdgeSWPackage....	Edge Manifest	---	01:55:05 PM 10.1...	✗	---	11:22:53 AM 10.1...	Ragavender Bas...
sample	Device Configuration	Firmware	06:26:55 PM 09.2...	✓	1.0.0	06:26:55 PM 09.2...	Jayasimha Nallag...
test.json	Device Configuration	Firmware	07:22:57 PM 08.2...	✓	2.0.0	07:22:57 PM 08.2...	Jayasimha Nallag...

1 to 9 of 9 First Previous Page 1 of 1 Next Last

# Deployments

Deployments are used to manage the configuration of connected devices in bulk.

This document covers the following aspects of deployments view:

1. [Deployments Grid](#)
2. [Deployment Creation](#)
3. [Deployment Details](#)
4. [Deactivate Deployment](#)
5. [Reactivate Deployment](#)
6. [Delete Deployment](#)

## Deployments Grid

The Deployments Grid lists all the deployments which are either active or deactivated.

Following are the columns displayed and their usage in the Deployments Grid

COLUMN NAME	DESCRIPTION
Name	Name of the Deployment
Package	Name of the package that is used for configuration
Device Group	Device group that is targeted by the deployment
Priority	Priority specifies which deployment will impact the devices; higher priority deployments take precedence over lower priority ones
Configuration Type	Indicates whether the Configuration is updating Device Firmware or any other custom properties. Empty for Edge Deployments
Targeted	Number of devices a configuration is targeted for
Applied	Number of devices where the configuration is currently applied for the latest deployments and previously applied if the deployment is not the latest for the device group
Success	Number of Devices where the configuration update is successful as reported by devices
Failed	Number of Devices where the configuration update failed as reported by devices
Date Created	Date the package was created
Created On	Date the deployment was created
Created By	User who created the deployment
Modified On	Date the deployment was last modified

COLUMN NAME	DESCRIPTION
Modified By	User who last modified the deployment

## Deployment Creation

- Navigate to the Deployments page and click on **+ New Deployment**

The screenshot shows the 3M Device Management interface. At the top, there's a red banner with the text "3M 3M". Below it is a navigation bar with links for "SimulationGroup", "Get Link", "Manage device groups", "+ New deployment", and user icons. The main content area is titled "Deployments" and shows a date "0/13/87". On the left, a sidebar lists "Dashboard", "Device Explorer", "User Management", "Rules", "Packages", "Deployments" (which is selected and highlighted in blue), and "Maintenance". At the bottom right of the main area, there are buttons for "First", "Previous", "Page 0 of 0", "Next", and "Last".

- In the flyout that is presented, enter the name for the deployment



## New deployment

Name \*

Enter name

Package type \*

Select package type

## Package \*

Priority (Higher values indicate higher priority) ? \*

Enter priority

targeted devices

\* This deployment runs continuously. Every device (and any you add in the future) in the selected device group will receive this package.

- Select package type based on the targeted device types.



## New deployment

Name \*

Package type \*

Select package type

Edge Manifest  
Device Configuration

Package \*

Priority (Higher values indicate higher priority) ? \*

targeted devices

\* This deployment runs continuously. Every device (and any you add in the future) in the selected device group will receive this package.

- Select configuration type

Note:

**Configuration Type** is needed if **Package Type** is **Device Configuration**.



## New deployment

Name \*

Package type \*



Configuration type \*



Firmware

Custom - 1

Package \*



Priority (Higher values indicate higher priority) ? \*

---

1

- Select the package based on the configuration type



## New deployment

Name \*

DeploymentSimulation

Package type \*

Device Configuration



Configuration type \*

Firmware



Package \*

Select package



Package1  
SW8Package.json  
SW2Package.json  
SW2Package.json  
SW3Package.json  
Package2  
SW2Package.json  
test0309

1

- 
- Provide a priority for deployment, the higher the priority, the higher the precedence.
  - Click on Apply.



## New deployment

Configuration type \*

 ▼

Package \*

 ▼

Priority (Higher values indicate higher priority) ② \*

---

**1**

targeted devices

\* This deployment runs continuously. Every device (and any you add in the future) in the selected device group will receive this package.

- After successfully creating a deployment, you will be presented with the confirmation details.



## New deployment

Package type \*

DeviceConfiguration

Configuration type \*

Firmware

Package \*

Priority (Higher values indicate higher priority) ? \*

4

---

1

targeted devices ✓

View your deployment status detail for [DeploymentSimulation](#).

Close

## Deployment Details

- The Deployment Details screen provides data about the metrics of a deployment, such as the number of devices targeted, applied, succeeded, pending and failed.

☰

↳ Deployments

DEPLOYMENT NAME

## DeploymentSimulation

1 Applied	0 Failed
1 Targeted	0 Succeeded 1 Pending

DEVICE GROUP  
SimulationGroup  
---  
---  
---  
---  
START  
02:19:03 PM 10.16.2020  
PACKAGE TYPE  
Device Configuration  
---  
PACKAGE  
Package1 (1.0.0)

PRIORITY  
4

CONFIGURATION TYPE  
Firmware

Devices Affected

Name Deployment Status Firmware Start End

SimulatedDevice	Pending	---	---	---
-----------------	---------	-----	-----	-----

> Expand Columns

The following table depicts how the metrics are defined.

METRIC	DESCRIPTION
Targeted	Number of devices which will be impacted by the deployment
Applied	Number of devices which are impacted by Deployment
Failed	Number of devices which reported failure of firmware update
Succeeded	Number of devices which reported success of firmware update to the current version
Pending	Number of devices yet to be updated

- The Deployment Details screen also provides the following details of a Deployment.

☰

↳ Deployments

DEPLOYMENT NAME

## DeploymentSimulation

1 Applied	0 Failed
1 Targeted	0 Succeeded 1 Pending

DEVICE GROUP  
SimulationGroup  
---  
---  
---  
---  
START  
02:19:03 PM 10.16.2020  
PACKAGE TYPE  
Device Configuration  
---  
PACKAGE  
Package1 (1.0.0)

PRIORITY  
4

CONFIGURATION TYPE  
Firmware

Devices Affected

Name Deployment Status Firmware Start End

SimulatedDevice	Pending	---	---	---
-----------------	---------	-----	-----	-----

> Expand Columns

PROPERTY	DESCRIPTION
Device Group	Provides the information about the devices that are targeted
Package Type	Provides information about the type of package
Package	Name of the package that is used for configuration, with version
Priority	Priority specifies which deployment will impact the devices; higher priority deployments take precedence over lower priority ones
Configuration Type	Indicates whether the configuration is updating device firmware or any other custom properties; this is empty for edge deployments

- Deployment Details also provides information about the custom metrics configured for the configuration as depicted.

The screenshot shows the 'Deployment Simulation' page. On the left, there's a navigation sidebar with 'Deployments' selected. The main area displays deployment statistics: 1 Applied, 0 Failed, 1 Targeted, 1 Succeeded, and 0 Pending. It also shows priority (4), configuration type (Firmware), and deployment details (Device Group: SimulationGroup, Package Type: Device Configuration, Start: 02:30:26 PM 10.16.2020, Package: Package1 (1.0.0)). Below this is a table titled 'Devices Affected' with columns: Name, Deployment Status, Firmware, Start, and End. One row is shown: SimulatedDevice, Succeeded, 1.0.0, 10:00:21 PM 10.16.2020, 10:00:23 PM 10.16.2020.

- The following images depict the various stages of how a deployment affects the devices.

- The device status is Pending when the firmware update is pending on the device.

This screenshot is identical to the one above, but the 'Devices Affected' table shows one device in a 'Pending' state instead of 'Succeeded'. The table columns are: Name, Deployment Status, Firmware, Start, and End. The single row is: SimulatedDevice, Pending, --, --, --.

- The device status is Completed when the device has successfully updated its firmware as per the configuration.

The screenshot shows the 'Deployment Simulation' screen. On the left, a sidebar lists navigation options: Dashboard, Device Explorer, User Management, Rules, Packages, Deployments (which is selected and highlighted in grey), and Maintenance. The main content area is titled 'Deployment Simulation'. It displays summary statistics: 1 Applied (0 Failed), 1 Targeted (1 Succeeded, 0 Pending). Below this, configuration details are shown: Priority 4, Configuration Type Firmware, and Device Group SimulationGroup. A table titled 'Devices Affected' lists one device: SimulatedDevice, which succeeded with Firmware version 1.0.0, starting at 10:00:21 PM 10.16.2020 and ending at 10:00:23 PM 10.16.2020. A 'Download' button is visible in the top right corner.

- The Deployment Details screen also lists the Devices impacted by the deployment.

This screenshot is identical to the one above, showing the 'Deployment Simulation' screen. The 'Devices Affected' grid is highlighted with a blue border. The grid columns are labeled: Name, Deployment Status, Firmware, Start, and End. The single entry for 'SimulatedDevice' shows it succeeded with Firmware 1.0.0, starting at 10:00:21 PM 10.16.2020 and ending at 10:00:23 PM 10.16.2020.

The Devices Affected grid displays the following columns

Column	Description
---	---
Name	Name of the device
Deployment Status	Depicts the status of the firmware update on the device
Firmware	Firmware version.
Start	Date and time that a device began the configuration update.
End	Date and time that a device completed the configuration update.

- The Deployment Details screen also provides the option to download the Devices Affected grid details as an Excel spreadsheet.

☰

↳ Deployments

DEPLOYMENT NAME

## DeploymentSimulation

Applied	Failed	---	DEVICE GROUP	PACKAGE TYPE
1	0	--	SimulationGroup	Device Configuration
Targeted	Succeeded	---	START	PACKAGE
1	1	--	02:19:03 PM 10.16.2020	Package1 (1.0.0)
PRIORITY		---		
4				
CONFIGURATION TYPE	Firmware			

Devices Affected

> Expand Columns

Name	Deployment Status	Firmware	Start	End
SimulatedDevice	Succeeded	1.0.0	10:00:21 PM 10.16.2020	10:00:23 PM 10.16.2020

Active  Delete  Download

- Deployment Activation and Deactivation can also be performed from Deployment Details.

☰

IoT Platform

↳ Deployments

DEPLOYMENT NAME

## DeploymentSimulation

Applied	Failed	current	DEVICE GROUP	PACKAGE TYPE
1	0	--	SimulationGroup	Device Configuration
Targeted	Succeeded	---	START	PACKAGE
1	1	--	02:30:26 PM 10.16.2020	Package1 (1.0.0)
PRIORITY		---		
4				
CONFIGURATION TYPE	Firmware			

Devices Affected

> Expand Columns

Name	Deployment Status	Firmware	Start	End
SimulatedDevice	Succeeded	1.0.0	10:00:21 PM 10.16.2020	10:00:23 PM 10.16.2020

Inactive  Delete  Download

Type here to search

14:01  
ENG 19-10-2020

- On toggling the status indicator, you will be presented with a confirmation popup.

**3M 3M**

☰ Dashboard Device Explorer User Management Rules Packages Deployments Maintenance

DEPLOYMENT NAME DeploymentSimulation

Applied	Failed	current	DEVICE GROUP	PACKAGE TYPE
1	0	1	SimulationGroup	Device Configuration
Targeted	Succeeded	Pending	START	PACKAGE
1	1	0	2:30:26 PM 10.16.2020	Package1 (1.0.0)

PRIORITY 4

CONFIGURATION TYPE Firmware

Devices Affected > Expand Columns

Name	Deployment Status	Firmware	Start	End
SimulatedDevice	Succeeded	1.0.0	10:00:21 PM 10.16.2020	10:00:23 PM 10.16.2020

Update Deployment?  
Do you want to re-activate the Deployment?

✓ OK      ✕ Cancel

- The Deployment Details screen also provides the option to delete a deployment.

**3M 3M**

☰ Dashboard Device Explorer User Management Rules Packages Deployments Maintenance

DEPLOYMENT NAME DeploymentSimulation

Applied	Failed	---	DEVICE GROUP	PACKAGE TYPE
1	0	--	SimulationGroup	Device Configuration
Targeted	Succeeded	--	START	PACKAGE
1	1	--	02:19:03 PM 10.16.2020	Package1 (1.0.0)

PRIORITY 4

CONFIGURATION TYPE Firmware

Devices Affected > Expand Columns

Name	Deployment Status	Firmware	Start	End
SimulatedDevice	Succeeded	1.0.0	10:00:21 PM 10.16.2020	10:00:23 PM 10.16.2020

Active Delete Download

- On clicking the Delete button, you will be presented with a confirmation popup.

The screenshot shows the 3M IoT Platform interface. On the left, a sidebar lists various management options: Dashboard, Device Explorer, User Management, Rules, Packages, Deployments (which is selected), and Maintenance. The main content area is titled 'Deployment Simulation'. It displays summary statistics: 1 Applied, 0 Failed, 1 current, and 1 Targeted, 1 Succeeded, 0 Pending. Below this, details are shown for the targeted deployment: DEVICE GROUP (SimulationGroup), PACKAGE TYPE (Device Configuration), PACKAGE (Package1 (1.0.0)), and a timestamp (2:30:26 PM 10.16.2020). A 'Devices Affected' table is present, showing one device named 'SimulatedDevice' with status 'Succeeded', firmware '1.0.0', start time '10:00:21 PM 10.16.2020', and end time '10:00:23 PM 10.16.2020'. A modal dialog box is overlaid on the page, prompting the user to confirm the deletion of the deployment.

## Deactivate Deployment

You can deactivate a deployment to free up the configurations in IoT Hub and still have a reference of the deployment for future use.

Note:

IoT Hub currently has a limit of 100 configurations.

Steps to deactivate a deployment.

- Click on the circle adjacent to any active deployment

### Reference:

CIRCLE COLOR	STATUS OF DEPLOYMENT
Green	Active
Black	Inactive

! [Deployment Confirmation] (../../../../images/deployment/deploymentgrid-active-deployment.png)

- You will be presented with the Deployment Status flyout, representing the current status of the Deployment.

The screenshot shows a modal dialog box titled "Deployment Status". At the top right are close and apply buttons. The main area displays deployment statistics: "Your environment is currently limited to 100 active deployments." Below this is a section titled "DeploymentSimulation" with a toggle switch labeled "Active" which is currently set to "Active". A note below says "No Related deployments exist for the selected deployment." At the bottom are "Apply" and "Close" buttons.

- Toggle the button to change the status of the deployment

## Deployment Status

Your environment is currently limited to 100 active deployments.

### DeploymentSimulation

Active

### Related Deployments

No Related deployments exist for the selected deployment.

Apply

Close

- Click Apply.



## Deployment Status

Your environment is currently limited to 100 active deployments.

### Deployment Simulation



### Related Deployments

No Related deployments exist for the selected deployment.

Apply

x Close

- After a successful status change, you can verify the status of a deployment will be inactive

The screenshot shows the 3M IoT Hub interface. On the left, there's a sidebar with icons for Dashboard, Device Explorer, User Management, Rules, Packages, **Deployments** (which is selected), and Maintenance. The main area is titled "Deployments" and shows a table with one row. The table columns include: Deploy..., Packag..., Simula..., Pa..., Device..., Firmware, Tar..., Ap..., Su..., Fai..., Cr..., Cr..., Mo..., Mo..., and Mo... . The first column has a checkbox and a circle containing a dot. The second column contains the text "Deploy...". The third column contains "Packag...". The fourth column contains "Simula...". The fifth column contains "Pa...". The sixth column contains "Device...". The seventh column contains "Firmware". The eighth column contains "Tar...". The ninth column contains "Ap...". The tenth column contains "Su...". The eleventh column contains "Fai...". The twelfth column contains "Cr...". The thirteenth column contains "Cr...". The fourteenth column contains "Mo...". The fifteenth column contains "Mo...". The sixteenth column contains "Mo...". The date "0/13/87" is displayed at the top right of the main area. At the bottom right, there are links for "1 to 1 of 1", "First", "Previous", "Page 1 of 1", "Next", and "Last".

## Reactivate Deployment

You can reactivate a deployment if necessary.

Note:

IoT Hub currently has a limit of 100 configurations.

Steps to Reactivate a Deployment.

- Click on the circle adjacent to any inactive deployment

**Reference:**

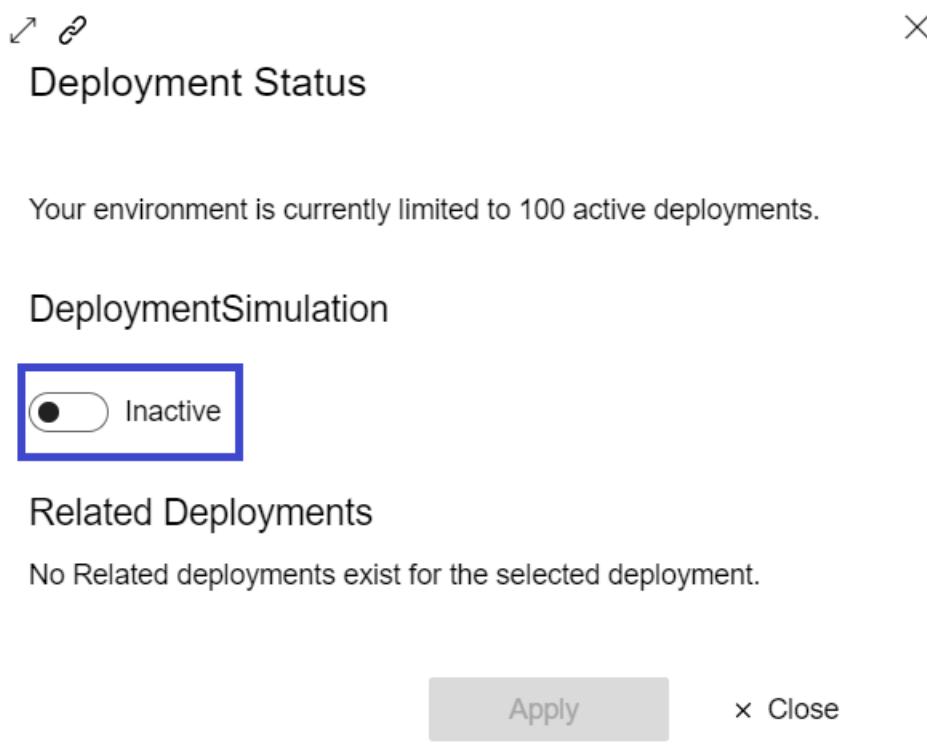
CIRCLE COLOR	STATUS OF DEPLOYMENT
Green	Active
Black	Inactive

! [Deployment Confirmation] (../../../../images/deployment/deploymentgrid-inactive-deployment.png)

- You will be presented with the Deployment Status flyout, representing the current status of the Deployment.



- Toggle the button to change the status of the deployment



- Click Apply.



## Deployment Status

Your environment is currently limited to 100 active deployments.

### Deployment Simulation



### Related Deployments

No Related deployments exist for the selected deployment.



- After successfully changing its status, you can verify the deployment is active

The screenshot shows the 3M Deployment Management interface. The left sidebar has a "Deployments" item selected. The main area displays a table titled "Deployments" with one row of data. The columns include: Selection CheckBox, Name (Deploy...), Package (Packa...), Simulation (Simula...), Priority (4), Device Type (Device...), Firmware (Firmware), Target (1), Success (1), Failure (---), Creation Time (02:19...), Creator (Ragav...), and two more columns with values (---). The top right corner shows the date "1/14/86".

	Name	Package	Simulation	Priority	Device Type	Firmware	Target	Success	Failure	Created	Creator	Mo...	Mo...
<input checked="" type="checkbox"/>	Deploy...	Packa...	Simula...	4	Device...	Firmware	1	1	---	02:19...	Ragav...	---	---

## Delete Deployment

You can delete any deployment when it is considered unnecessary.

Steps to Delete a Deployment.

- Select a deployment by clicking in the Selection CheckBox.

**3M**

The screenshot shows the 3M Deployment Management interface. On the left, there is a navigation sidebar with icons for Dashboard, Device Explorer, User Management, Rules, Packages, Deployments (which is selected and highlighted in dark grey), and Maintenance. The main area has a header with 'SimulationGroup' dropdown, 'Get Link' button, 'Manage device groups' button, and '+ New deployment' button. The title 'Deployments' is displayed in large bold letters, with the date '0/13/87' to its right. Below the title is a table with columns: Na..., Pa..., De..., Pri..., Pa..., Co..., Tar..., Ap..., Su..., Fai..., Cr..., Cr..., Mo..., Mo... . A single row is visible, representing a deployment entry. The first column has a checkbox with a blue border around it. The second column has a black circle with a white dot. The third column contains the text 'Deploy...'. The table also includes summary statistics: Packa... 4, Device... Firmware, 1 1 1 ---, 02:19:..., Ragav..., 04:44:..., Ragav... . At the bottom of the table area, there is a horizontal scrollbar and a footer with pagination links: '1 to 1 of 1', 'First', 'Previous', 'Page 1 of 1', 'Next', and 'Last'.

- Once you select a deployment, the Delete option will be available in the toolbar

**3M**

This screenshot is similar to the previous one, but the 'Delete' button in the top right toolbar is highlighted with a blue border. The rest of the interface, including the sidebar, table structure, and footer, remains the same.

- After clicking the Delete button, the Delete Confirmation popup will be presented for final confirmation.

The screenshot shows the 3M Deployment Management interface. On the left, there's a sidebar with navigation links: Dashboard, Device Explorer, User Management, Rules, Packages, Deployments (which is selected and highlighted in dark grey), and Maintenance. The main area has a title 'Deployments' and a date '0/13/87'. At the top right are buttons for 'Delete' (with a trash icon) and '+ New deployment'. A modal dialog box is centered over the grid, titled 'Delete Deployment?'. It contains a message: 'Deleting selected deployment(s) will stop deployment from being applied to the devices. It may result in a lower priority deployment being applied if any'. Below the message are two buttons: a red 'Delete' button with a trash icon and a 'Cancel' button with a close icon. The grid below the dialog shows one deployment entry:

Su...	Fai...	Cr...	Cr...	Mo...	Mo...
1	---	02:19:...	Ragav...	04:44:...	Ragav...

At the bottom of the page, there are navigation links: '1 to 1 of 1', 'First', 'Previous', 'Page 1 of 1', 'Next', and 'Last'.

- Click on the Delete button in the popup to confirm deletion.

This screenshot is identical to the one above, showing the 'Deployments' screen with a 'Delete Deployment?' dialog. The dialog contains the same message and buttons. The grid below shows the same single deployment entry as the previous screenshot.

- After performing a successful deletion, you will not be able to find the deployment in the Deployment Grid.

SimulationGroup ▾ Get Link Manage device groups + New deployment

Dashboard Device Explorer User Management Rules Packages Deployments Maintenance

# Deployments

0/13/87

Expand Columns

□	Name	Path	Description	Priority	Package	Category	Target	App	Status	Failure	Created	Modified	File	More
0 to 0 of 0	First	Previous	Page 0 of 0	Next	Last									

# Maintenance

The **Maintenance** page allows users to view alerts from devices and different jobs that are running against devices. Below is the information a user can find on the **Maintenance** page:

1. [Summary](#)
  - [Alert counts](#)
  - [Job counts](#)
2. [Alerts Grid](#)
  - [Alert details](#)
3. [Jobs Grid](#)
  - [Job details](#)

## Summary

Click Maintenance from the left menu. A user can refer to this section to find the summary of counts for alerts and jobs based on their statuses.

### Alert counts

This section displays the number of alerts based on their statuses and severity:

- Open: Total open alerts
- Critical: Total Critical alerts
- Warning: Total Warning alerts

The screenshot shows the Maintenance page interface. At the top, there is a navigation bar with a menu icon, a 'Default' dropdown, a 'Get Link' button, and a 'Manage device groups' button. On the left, a sidebar lists various menu items: Dashboard, Device Explorer, User Management, Rules, Packages, Deployments, and Maintenance, with Maintenance being the active tab. The main content area has a title 'Maintenance'. Below it is a summary box containing the following data:

Category	Count	Severity
Open Alerts	16	6 Critical (red triangle), 10 Warning (yellow square)
Failed jobs	0	0 Total, 0 Succeeded

Below the summary box are two large buttons labeled 'Alerts' and 'Jobs'. Underneath these buttons is a table header with columns: Rule name, Description, Severity, and Total count.

### Job counts

This section displays the counts of jobs based on their statuses:

- Total: Total jobs
- Failed: Total failed jobs
- Succeeded: Total succeeded jobs

The screenshot shows the Maintenance dashboard. On the left, a sidebar lists navigation options: Dashboard, Device Explorer, User Management, Rules, Packages, Deployments, and Maintenance (which is selected). The main area has a title 'Maintenance' and two summary boxes. The first box shows '0 Open Alerts' with '0 Critical' (red triangle) and '0 Warning' (yellow square). The second box shows '0 Failed jobs' with '2 Total' (2 Succeeded). Below these is a section titled 'Alerts' and 'Jobs'. A grid table follows, with columns: Job Name, Status, Operation, and No. of devices.

Job Name	Status	Operation	No. of devices

## Alerts Grid

Clicking the **Alerts** tab reveals a grid that shows the list of alerts based on the rules defined on the [Rules page](#)

The screenshot shows the Maintenance dashboard with the 'Alerts' tab selected. It displays a summary of alerts: 16 Open Alerts (6 Critical, 10 Warning) and 0 Failed jobs (0 Total, 0 Succeeded). Below this is a grid table with columns: Rule name, Description, Severity, Total count, Open, Acknowledged, Closed, and Last occurrence. The table lists three specific rules.

Rule name	Description	Severity	Total count	Open	Acknowledged	Closed	Last occurrence
ShiftHumidityGreat...	Humidity greater th...	▲ Critical	2	2	0	0	07:05:24 PM 10.13....
ShiftTemperatureGr...	Temperature greate...	▲ Critical	4	4	0	0	07:05:24 PM 10.13....
temp30	temp30	■ Warning	10	9	1	0	07:10:26 PM 10.13....

Following are the columns displayed in the **Alerts** grid.

COLUMN NAME	DESCRIPTION
Rule Name	Name of the rule
Description	Describes the rule that is being created
Severity	Describes the severity of the rule
Total Count	Indicates the number of total alerts
Open Count	Indicates the number of open alerts
Acknowledged Count	Indicates the number of acknowledged alerts

COLUMN NAME	DESCRIPTION
Closed Count	Indicates the number of closed alerts
Last Occurrence	Describes the rule's last occurrence date

## Alert details

Clicking on the row of an alert will navigate to the **Alert details** view, which displays the following information:

- When the alert was triggered
- Statuses of the devices associated with the alert
- Telemetry from the devices associated with the alert

TOTAL	OPEN	ACKNOWLEDGED	CLOSED	LAST EVENT	SEVERITY
2	2	0	0	07:05:24 PM 10.13.2020	Critical

Manage alert occurrences associated to this rule in the section below, and use the associated information to troubleshoot each occurrence.

Rule name	Description	Severity	Device group	Trigger	Notification type	Status	Last trigger
ShiftHumidityGre...	Humidity greater ...	Critical	_SmartRap		Maintenance log	Enabled	07:05:24 PM 10.13.2020

Occurrence	Description	Severity	Trigger device	Time	Status
ShiftHumidityGreaterThan70		Critical	device12	07:05:24 PM 10.13.2020	open
ShiftHumidityGreaterThan70		Critical	device12	07:04:53 PM 10.13.2020	open

Details about the Rule are shown in the **Rule detail** grid. A user can perform the below operations from this grid, similar to the actions available on the [Rules page](#):

- View Rule details
- Edit Rule
- Disable Rule
- Delete Rule

The **Alert Occurrences** grid depicts the occurrences of alerts. A user can perform the below operations inside this grid:

- [Acknowledge Alerts](#)
- [Close Alerts](#)
- [Delete Alerts](#)

## Acknowledge Alerts

To acknowledge alerts:

1. Select the alerts to acknowledge from grid.
2. An **Acknowledge** button appears in the tool bar.
3. Click the button.

Default ▼ [Get Link](#) [Manage device groups](#) [Close](#) [Acknowledge](#) [Delete](#) Last month ▼ ↻

TOTAL	OPEN	ACKNOWLEDGED	CLOSED	LAST EVENT	SEVERITY
2	2	0	0	07:05:24 PM 10.13.2020	<span style="color: red;">▲ Critical</span>

Manage alert occurrences associated to this rule in the section below, and use the associated information to troubleshoot each occurrence.

**Rule detail**

> Expand Columns

<input type="checkbox"/> Rule name <span>&gt;All</span>	Description	Severity	Device group	Trigger	Notification type	Status	Last trigger
<input type="checkbox"/> ShiftHumidityGre...	Humidity greater ...	<span style="color: red;">▲ Critical</span>	_SmartRap		Maintenance log	Enabled	07:05:24 PM 10.13.2020

**Alert Occurrences**

> Expand Columns

<input type="checkbox"/> Occurrence	Description	Severity	Trigger device	Time	Status
<input checked="" type="checkbox"/> ShiftHumidityGreaterThan70		<span style="color: red;">▲ Critical</span>	device12	07:05:24 PM 10.13.2020	open
<input type="checkbox"/> ShiftHumidityGreaterThan70		<span style="color: red;">▲ Critical</span>	device12	07:04:53 PM 10.13.2020	open

1 to 2 of 2 First Previous Page 1 of 1 Next Last

## Close Alerts

To close alerts:

1. Select the alerts to close from grid.
2. A **Close** button appears in the tool bar.
3. Click the button.

Default ▼ [Get Link](#) [Manage device groups](#) [Close](#) [Acknowledge](#) [Delete](#) Last month ▼ ↻

TOTAL	OPEN	ACKNOWLEDGED	CLOSED	LAST EVENT	SEVERITY
2	2	0	0	07:05:24 PM 10.13.2020	<span style="color: red;">▲ Critical</span>

Manage alert occurrences associated to this rule in the section below, and use the associated information to troubleshoot each occurrence.

**Rule detail**

> Expand Columns

<input type="checkbox"/> Rule name <span>All</span>	Description	Severity	Device group	Trigger	Notification type	Status	Last trigger
<input type="checkbox"/> ShiftHumidityGre...	Humidity greater ...	<span style="color: red;">▲ Critical</span>	_SmartRap		Maintenance log	<span style="border: 1px solid blue; padding: 2px;">Maintenance log</span>	07:05:24 PM 10.13.2020

**Alert Occurrences**

> Expand Columns

<input type="checkbox"/> Occurrence	Description	Severity	Trigger device	Time	Status
<input checked="" type="checkbox"/> ShiftHumidityGreaterThan70		<span style="color: red;">▲ Critical</span>	device12	07:05:24 PM 10.13.2020	open
<input type="checkbox"/> ShiftHumidityGreaterThan70		<span style="color: red;">▲ Critical</span>	device12	07:04:53 PM 10.13.2020	open

1 to 2 of 2 First Previous Page 1 of 1 Next Last

## Delete Alerts

To delete alerts:

1. Select the alerts to delete from grid.
2. A **Delete** button appears in the tool bar.
3. Click the button.

Default ▼ 🔗 Get Link Manage device groups ✖ Close  ⓘ Acknowledge Delete Last month ▼ ⟳

TOTAL	OPEN	ACKNOWLEDGED	CLOSED	LAST EVENT	SEVERITY
2	2	0	0	07:05:24 PM 10.13.2020	<span>⚠ Critical</span>

Manage alert occurrences associated to this rule in the section below, and use the associated information to troubleshoot each occurrence.

Rule detail

> Expand Columns

Rule name	Description	Severity	Device group	Trigger	Notification type	Status	Last trigger
<a href="#">ShiftHumidityGre...</a>	Humidity greater ...	<span>⚠ Critical</span>	_SmartRap		Maintenance log	Enabled	07:05:24 PM 10.13.2020

Alert Occurrences

> Expand Columns

Occurrence	Description	Severity	Trigger device	Time	Status
<input checked="" type="checkbox"/> ShiftHumidityGreaterThan70		<span>⚠ Critical</span>	device12	07:05:24 PM 10.13.2020	open
<input type="checkbox"/> ShiftHumidityGreaterThan70		<span>⚠ Critical</span>	device12	07:04:53 PM 10.13.2020	open

1 to 2 of 2 First Previous Page 1 of 1 Next Last

## Jobs Grid

The **Jobs** tab displays a grid that lists the jobs for devices.

Default ▼ 🔗 Get Link Manage device groups Last hour ▼ ⟳

**Maintenance**

0 Open Alerts    0 ⚠ Critical    0 ⚠ Warning    0 Failed jobs    2 Total    2 Succeeded

Alerts Jobs > Expand Columns

Job Name	Status	Operation	No. of devices	Succeeded	Failed	Start time	End time
testjob1-920c560a-...	Running					11:02:40 AM 10.19....	---
Testjob-7cc37ae6-...	Completed		1	1	0	10:58:35 AM 10.19...	10:58:40 AM 10.19...
testtag1-30669087-...	Completed		1	1	0	11:01:29 AM 10.19....	11:01:34 AM 10.19....

The following columns are displayed in the **Jobs** grid.

COLUMN NAME	DESCRIPTION
Job Name	Name of the job
Status	Status of the job
Operation	Describes the severity of the rule
No. of devices	Total number of affected devices
Succeeded	Number of devices with a succeeded status

COLUMN NAME	DESCRIPTION
Failed	Number of devices with a failed status
Start time	Datetime when the job started running
End time	Datetime when the job completed running

## Job details

Clicking on the row of a job navigates to **Job details** view. This view shows:

- Jobs list
- Devices affected

The screenshot shows the 'Jobs' section of a management interface. On the left is a sidebar with navigation links: Dashboard, Device Explorer, User Management, Rules, Packages, Deployments, and Maintenance (which is currently selected). The main area has two tables.

The top table lists jobs with columns: Job Name, Status, Operation, No. of devices, Succeeded, Failed, Start time, and End time. One row is visible: 'Testjob-7cc37ae6-6e9f-405...' with Status 'Completed', 1 device, 1 succeeded, 0 failed, and times from 10:58:35 AM 10.19... to 10:58:40 AM 10.19... . A 'Last day' filter is applied.

The bottom table provides a detailed view of the selected job ('Testjob-7cc37ae6-6e9f-405...'). It includes columns: Job Name, Status, Device ID affected, Last return message, Start time, and End time. The data shows the job completed, affected device 'sah\_test\_device', and completion times.

At the bottom of the interface, there is a footer with navigation links: 1 to 1 of 1, First, Previous, Page 1 of 1, Next, and Last.

## Useful References

- [Alerts](#)

# Get Involved

There are many ways to help make this platform better. You can get involved in the community to share ideas or make direct contributions to the platform.

The following table describes the common ways to initialize improvements to the platform:

RESOURCE	DESCRIPTION
<b>Issues List</b>	Offers the following: <ul style="list-style-type: none"><li>• Report Bugs</li><li>• Request Features</li><li>• Influence Priorities</li><li>• Track Progress</li></ul>
<b>Documentation</b>	All documentation offers community contributions.
<b>Development</b>	You can make direct contributions to the code.
<b>Community Involvement</b>	Join the 3M IoT Platform Community.

## Issues List

Use the links below to get more information on each topic:

- [How to Report a Bug](#)
- [How to Request a Feature](#)
- [How to Influence Priorities](#)
- [How to Track Development](#)
- [How to Track Releases](#)

## Documentation

- [How to Improve Documentation](#)

## Development

- [How to Make Code Contributions](#)

## Community Involvement

- [How to Make Code Contributions](#)



# IoT Platform Community

Get involved in the 3M IoT Platform Community.

Details coming soon.

Emails will go out announcing the first event that is scheduled to start between October 15th and the 23rd.

# How to Request a Feature

To request a feature to improve the 3M IoT Platform on Azure, follow the steps below:

1. Open the [Issues List](#)
2. Navigate to the Issues List
3. Click **New issue**
4. Click **Get started** on the Feature request row
5. Populate the request (See section: *How to Complete Feature Request*)
6. Submit Issue
7. Track Progress (See [How to Track Progress](#))
8. Add Tag Name for the group requesting: e.g. CHIM

## How to Complete Feature Request

- Title
- Describe Need
- Describe Improvement
- Describe Alternatives
- Attachments



# How to Report a Bug

To report a bug related to the 3M IoT Platform on Azure, follow the steps below:

1. Open the [Issues List](#)
2. Navigate to the Issues List
3. Click **New issue**
4. Click **Get started** on the Bug report row
5. Populate the report (See section: How to Complete Bug Report)
6. Submit Issue
7. Track Progress (See [How to Track Progress](#))
8. Add Tag Name for the group requesting: e.g. CHIM

## How to Complete Bug Report

A Bug Report consists of the following sections:

- Title
- Description
- Steps to Reproduce
- Expected Behavior
- Screenshots
- Desktop
- Additional context
- Attachments

The remaining sections will describe how to complete each section.

### Title

TODO

### Description

### Steps to Reproduce

### Expected Behavior

### Screenshots

### Desktop

### Additional context

### Attachments



# How to Influence Priorities

This article describes how you can influence platform improvements by voting on issue prioritization.



# How to Track Progress

# How to Track Releases

Versions record what improvements are associated with a given release. When a new release is deployed to your environment, the version number is reflected in the UI. Versions organize platform improvements into identifiable releases. What's included in each version is recorded in the ChangeLog which is publicly available on [GitHub](#). A new version must be deployed to each environment. As such, environments typically run on different releases.

## How to check the current version

To see the list of improvements by version, open the [Change Log](#).

## How to check the improvements per version

From the releases page, click the link to the changelog for the version you're interested in or simply click the first link. Improvements for all other versions will also be listed. Here's what it looks like:

## How to check which version you're running

The easiest way to identify which version an environment has deployed is through the user interface. By clicking on the gear icon in the upper right, the current version is easily identified as depicted below:

The link **[View release notes](#)** offers a quick way to open the ChangeLog on the public repo.



# How to Improve Documentation



# Overview on Contributing Platform Enhancements



# Getting Started for Web Developers

This document provides setup instructions for contributing to the 3M IoT Platform as a *Web Developer*. In addition, it outlines relevant skills and helpful links to learn about or brush up on related topics.

## Common Contributions

Web Developers typically contribute to the platform in the following ways:

- Enhancing the web User Interface
- Communicating with the Backend Services

## Tools to Install

- [Git](#)
- [PowerShell](#)
- [NodeJS](#)
- [NPM](#)
- [VS Code](#)

## Useful Skills

Be familiar with the following:

- JavaScript
- CSS/Sass
- [ReactJS](#)
- [Redux \(a ReactJS event management tool\)](#)
- IoT
- [Azure IoT UX Fluent Controls](#)
- [Git](#)

## Helpful Resources

- [IoT School](#)
- [Microsoft IoT](#)



# Getting Started for API Developers

This document provides setup instructions for contributing to the 3M IoT Platform as an *API Developer*. In addition, it outlines relevant skills and helpful links to learn about or brush up on related topics.

## Useful Skills

- C#
- .Net core
- IoT
- Azure services- IoTHub, Azure Storage, Cosmos DB
- Containerization
- Dev spaces (optional)
- Azure functions
- Kubernetes (optional)
- Docker (optional)
- Git

## Common Contributions

API Developers typically contribute to the platform in the following ways:

- Updating/enhancing back-end services
- Building service to interact with IoTHub to configure devices and provide configuration updates to devices via IoTHub
- Building Analytics to check for anomalies
- Services to store data for Audit

## Tools to Install

- Visual Studio 2019 (or above) / VS Code
- Docker desktop (optional)
- Azure CLI (optional)
- Git

## Helpful Resources

- [IoT School](#)
- [Microsoft IoT](#)
- [Azure CLI](#)
- [Azure Storage](#)
- [Cosmos DB](#)
- [Azure IoT Hub](#)
- [Git](#)



# Getting Started for Azure Developers

This document provides setup instructions for contributing to the 3M IoT Platform as a *Azure Developer*. In addition, it outlines relevant skills and helpful links to learn about or brush up on related topics.

## Useful Skills

Be familiar with the following list of Azure services

- AKS
- Storage Account
- DPS (Device Provisioning Service)
- Function App
- App Configuration
- Cosmos DB
- Event Hub
- Key Vault
- Send Grid
- IOT Hub
- Stream Analytics Job
- Azure Maps

## Common Contributions

Azure Developer typically contribute to the platform in the following ways:

- Should have knowledge on Azure Services and how to deploy code to these services.
- Enhancing the Azure Services and integration between the services.

## Tools

- Az cli
- Azure Storage Explorer
- Azure Devspaces cli and extension
- Azure IOT Explorer
- VS Code
- Docker

## Helpful Resources

- [IoT School](#)
- [Microsoft IoT](#)
- [Azure Storage Explorer](#)
- [Az Cli](#)
- [Devspaces](#)

# Getting Started for DevOps Engineers

This document provides setup instructions for contributing to the 3M IoT Platform as a *DevOps Engineer*. In addition, it outlines relevant skills and helpful links to learn about or brush up on related topics.

## Useful Skills

- Azure Services
  - AKS
  - Storage Account
  - DPS (Device Provisioning Service)
  - Function App
  - App Configuration
  - Cosmos DB
  - Event Hub
  - Key Vault
  - Send Grid
  - IOT Hub
  - Stream Analytics Job
  - Azure Maps
  - Application Insights
  - Application Gateway
  - Log Analytics
  - Azure Automation Account
  - Virtual Network
  - Network Security Group
  - DNS Zone
  - Load Balancer
  - VM Scale set
- Azure Devops
- Powershell
- Bash
- Kubernetes
- Helm
- Docker
- Docker Compose

## Helpful Resources

- [Microsoft IoT](#)
- [Azure Cli](#)
- [Azure IOT Explorer](#)
- [Git](#)
- [Git Desktop](#)
- [Helm](#)
- [Kubernetes](#)
- [Storage Explorer](#)
- [Devspace](#)
- [Azure Devops](#)

DevOps Engineer typically contribute to the platform in the following ways:

- Enhancing existing CI/CD pipelines.
- Creating new pipelines if there are any
- Creating the infrastructure in Azure through Infrastructure as a Code.

## Tools

- Az cli
- Azure Storage Explorer
- Azure Devspaces cli and extension
- Azure IOT Explorer
- VS Code
- Docker
- Bash
- Kubernetes
- Helm
- Git
- GitHub Desktop
- Docker Desktop
- Docker Compose
- Powershell
- VS Code



# Getting Started for Document Contributors

This document provides setup instructions for contributing to the 3M IoT Platform as a *Document Contributor*. In addition, it outlines relevant skills and helpful links to learn about or brush up on related topics.

## Useful Skills

- Markdown
- IoT (optional)
- Git

## Common Contributions

Documentation Contributors typically contribute in the following ways:

- Add new content to the documentation
- Improve or organize the existing documentation

## Tools to Install

- VS Code
- [Docfx](#)
- Git

## Helpful Resources

- [Docfx User manual](#)
- [Docfx - API Documentation](#)
- [Docfx - Rest API](#)
- [IoT School](#)
- [Microsoft IoT](#)
- [Git](#)



# 3M IoT Development Environment Setup Instructions

This document provides detailed documentation on setting up a development environment to contribute to 3M's IoT Azure Platform, called Bluebird. Note: If you're using a 3M Virtual Machine, it's possible some of the products to install may exist already.

## Products to Install

Please install the following products in the order listed below. Click the project name to open the specific steps for each product below. Wherever relevant, OS specific steps will be provided for both Mac and PC. I would like to also point out that we are not allowed to be admins on our 3M machines, we use elevated access. This is accomplished by right clicking and selecting the elevated access option.

**Note:** Install Issues not specifically addressed in the specific product install guides below should be recorded here. Since we have moved into an open source environment for Odin it is not unreasonable to use our own machines for dev. Sometimes this proves less restrictive and has better up-time.

### Product Install Guides (by install order)

- [Virtualization](#) (only necessary if developing on Windows)
- [.NET Core 3.1](#)
- [PowerShell](#)
- [Azure CLI](#) (plus extensions)
- [Visual Studio Code](#)
- [Git / Bash](#)
- [NodeJS / npm](#)
- [Visual Studio 2019](#)
- [Azure Data Studio](#)
- [Storage Explorer](#)
- [Azure IoT Explorer](#)
- [Docker](#)
- [Terraform](#)
- [Kubernetes](#)
- [Helm](#)
- [Redux](#)
- [Postman](#)
- [Dev Spaces](#)
- [XUnit](#)

## Next Steps

Once you're done installing the tools, you need to get access to the solution's source code. To do so, please follow the steps here:  
[Source Code](#)



# Whats new in VS 2019

Check the details here

Use the community edition if you don't have a licensed version: <https://visualstudio.microsoft.com/downloads/>

## Download

Download Visual Studio 2019 from [here](#)



# Overview

VS Code is an editor developed by Microsoft for Windows, Linux and macOS.

## Prerequisites

Before installing VS Code, be sure the following are installed:

- [Git](#)
- [PowerShell](#)
- [NodeJS](#)
- [NPM](#)

## Download Media

VS Code can be downloaded from: <https://code.visualstudio.com/download>

Recommended install settings on Windows:

- Check Add to Windows Explorer Shortcut

## Extensions: Required

- [PowerShell](#)
- [Chrome Debugger](#)
- [React Native Tools](#)
- [YAML](#)

## Extensions: Optional

### General

- For additional React Sugar: <https://medium.com/productivity-freak/the-ultimate-vscode-setup-for-js-react-6a4f7bd51a2>

### For Kubernetes Development

Docker should already be installed. Then consider installing the following extensions:

- [Cloud Code](#) for
- [Docker](#)
- [C#](#)

### For Test Development

- [Nxunit Test Explorer](#)

### For Documentation

Have DocFx already installed

- [DocFX](#)
- [docs-preview]<https://marketplace.visualstudio.com/items?itemName=docsmsft.docs-preview>

# Overview

If your development environment runs on Windows you must setup virtualization in order to run docker containers locally since some of them run on Linux which Windows does not natively support. This is not necessary on a MacOS since it's based on Unix. Which virtualization product you use is a matter of preference and is typically influence by the OS of your host machine. Historically, Hyper-V was popular for servers and VirtualBox on clients.

## Virtualization Product

PRODUCT	VENDOR	GUIDES
Hyper-V	Microsoft	<a href="#">MS Install Guide</a>
Virtual Box	Oracle	- <a href="#">Docker for Mac - Using Docker with Virutal Box on Windows</a>

## Prepare Virtual Machine using Hyper-V

1. Enable Hyper-V. [Click for more details.](#)
2. Create a Virtual Machine. [Click for more details.](#) Note: Microsoft offers instances of Ubuntu and an evaluation copy of Windows. Currently the evaluation only last 5 days so it's not a great option unless you have an Win 10 Enterprise Key to use to upgrade. Otherwise you need an ISO and License Key for the OS you intend to install. Using the eval requires a 16GB download.
3. Prepare Installation Media
4. Create Virtual Machine

## Prepare Installation Media

If you're going to use your own copy of an operating system, you need to first create an ISO so Hyper-V or VirtualBox can use it to build the Virtual Machine. Microsoft provides a useful tool called **Create Windows 10 installation media** that can be downloaded here <https://www.microsoft.com/en-us/software-download/windows10>. The link includes instructions on how to use.

The above approach can also be used for Windows 8.1 and Windows 7. Alternatively there are several open source and 3rd party products that can be used to create ISO.

## Create Virtual Machine

Once you have your ISO, you can start the create virtual machine process. To continue, follow these steps:

<https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/quick-start/quick-create-virtual-machine>

## Alternative

As untried alternative to virtualization, if you're using Windows you could try the Linux subsystem for Windows.  
Note: We have not tried this. Key consideration is whether or not docker images can be instantiated.

Use Windows Subsystem for Linux for production: <https://docs.microsoft.com/en-us/windows/nodejs/setup-on-windows#use-windows-subsystem-for-linux-for-production>

The Windows Subsystem for Linux, introduced in the [Anniversary Update](#), became a stable feature in the [Fall Creators Update](#). You can now run Ubuntu and openSUSE on Windows, with Fedora and more Linux distributions coming soon.

This document might be a good guide: <https://docs.microsoft.com/en-us/virtualization/windowscontainers/quick-start/set-up-environment?tabs=Windows-10-Client>



## Overview

.NET Core is a cross-platform successor to the .NET Framework that runs on Windows, Linux, and macOS operating systems. You can download from [here](#). It was open sourced by Microsoft.

## TODO

Did you choose 3.1.0 or 3.1.3 (latest as of 4/8 )?

# Overview

PowerShell is a command-line shell and associated scripting language that can run on Windows, Linux and macOS as of version 7.

## Installation Guides

- [Windows](#)
- [macOS](#)
- [Linux](#)
- [Installing PowerShell in Azure Resources via ARM](#)

To access the download packages go here: <https://github.com/PowerShell/PowerShell>

## Additional

Windows PowerShell 7 is automatically part of Windows 10 IoT Enterprise. Additional details look here:

<https://docs.microsoft.com/en-us/powershell/scripting/install/installing-powershell-core-on-windows?view=powershell-7#deploying-on-windows-10-iot-enterprise>

# Overview

The Azure command-line interface (Azure CLI) is a set of commands used to create and manage Azure resources. The Azure CLI is available across Azure services and is designed to get you working quickly with Azure, with an emphasis on automation.

Click [here](#) to get more details Be sure to add [additional extensions, see below](#).

Note: Requires Elevated Permissions on 3M's Virtual Machine

## Install the Azure CLI

The Azure CLI is available to install in Windows, macOS and Linux environments. It can also be run in a Docker container and Azure Cloud Shell.

The current version of the Azure CLI is **2.3.1**. For information about the latest release, see the [release notes](#). To find your installed version and see if you need to update, run `az --version`.

- [Install on Windows](#)
- [Install on macOS](#)
- Install on Linux or Windows Subsystem for Linux (WSL)
  - [Install with apt on Debian or Ubuntu](#)
  - [Install with yum on RHEL, Fedora, or CentOS](#)
  - [Install with zypper on openSUSE or SLE](#)
  - [Install from script](#)
- [Run in Docker container](#)
- [Run in Azure Cloud Shell](#)

## Use extensions with Azure CLI

The Azure CLI offers the capability to load extensions. Extensions are Python wheels that aren't shipped as part of the CLI but run as CLI commands. With extensions, you gain access to experimental and pre-release commands along with the ability to write your own CLI interfaces.

Click [here](#) to get details on how to use extensions

## Additional Extensions

Add the extensions listed below. To do so, use the following command:

```
az extension add --name {extension_name}
```

To see the list of available extensions and their details such as if they're enabled, run the following:

```
az extension list-available
```

Note: if any of your extensions need to be updated, you can swap the **add** command with **update**.

### Dev Spaces

- **Description:** Dev Spaces provides a rapid, iterative Kubernetes development experience for teams.
- **Command:** `az extension add --name dev-spaces`

- [Documentation](#)
  - **Note:** there is a preview version as well. dev-spaces-preview

## IoT

- **Description:** Comprehensive data-plane functionality to manage Internet of Things (IoT) assets.
- **Command:** `az extension add --name azure-iot`
- [Documentation](#)

## DevOps

- **Description:** Manage Azure DevOps organization level operations including pipelines, boards, repos, artifacts, etc.
- **Command:** `az extension add --name azure-devops`
- [Documentation](#)

## Azure Kubernetes Service (AKS)

- **Description:** Manage Azure Kubernetes Services.
- **Command:** `az extension add --name aks-preview`
- [Documentation](#)

# Introduction

Git is a distributed version-control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows. It is designed to handle everything from small to very large projects with speed and efficiency.

## Pre-requisites

Be sure to installed VS Code first. Then you'll have the option of hooking Git into VS Code as you're editor.

If you're going to use recommended [NVM](#) (Node Version Manager) on Windows, you'll need to **Enable symbolic links** during the installation. See settings section, install step **Configuring extra options** below for more information.

## Git

For official installation instructions, click - [Git & GitHub](#).

Click [here](#) to download git for different OS.

## Settings

Use the default values except on the screenshots below. For those steps, use the settings indicated in the screenshot (assuming Win Install). :::image type="content" source="../../images/git-settings1.PNG" alt-text="Git settings 1":::

:::image type="content" source="../../images/git-settings2.PNG" alt-text="Git settings 2":::

:::image type="content" source="../../images/git-settings3.PNG" alt-text="Git settings 3":::

:::image type="content" source="../../images/git-settings4.PNG" alt-text="Git settings 4":::

## GitHub Desktop (optional)

Click [here](#) to download GitHub Desktop for macOS, Win, and Linux.

## Connecting Your GitHub Repository to 3M

1. The repository you are look for is <https://github.com/3Mcloud/azure-iot-platform-dotnet/>
2. You will then select Fork in the top right hand corner as displayed below
  - :::image type="content" source="../../images/git-repo-fork1.PNG" alt-text="Git repo Fork":::
3. This will then allow you to tie your personal account to the 3m cloud
4. You are able to verify that you are associated by clicking the number next to fork. It will display the different accounts attached as shown below.
  - :::image type="content" source="../../images/git-repo-fork2.PNG" alt-text="Git repo Fork":::
5. You can see mine is highlighted, using this method you are able to raise PR's and also submit Reviews. Keep in mind we need two reviews for QA purposes.
6. You can also see what your teammates have been working on by clicking on their account.
7. You are able to sync data from your repo to the 3m master. There are 2 main ways to do this.
  - This way is done by the cli.
    - <https://help.github.com/en/github/collaborating-with-issues-and-pull-requests/platforms/azure> - © 3M 2020

- This is the way to do it through the web gui. I think this is the ideal way to do this. I will attach a video and screen shots.
  - <https://youtu.be/YhwBgYPfoVE>
    - As you can see my repo is 15 commits behind.
    - ::image type="content" source="../../images/pull-request1.PNG" alt="Pull request 1"::
    - After comparing it will tell me I am able to merge.
    - ::image type="content" source="../../images/pull-request2.PNG" alt="Pull request 2"::
    - You would then submit a PR to sync your fork.
    - ::image type="content" source="../../images/pull-request3.PNG" alt="Pull request 3"::

## CLI Approach to Refresh Local Master with Upstream Changes

Note: This approach will completely replace your local master (forked 3M repo) with changes upstream (3M Cloud):

```
git remote add upstream /url/to/original/repo
git fetch upstream
git checkout master
git reset --hard upstream/master
git push origin master --force
```

## CLI Approach to Creating New Branch

By on a new copy of "master" (using above approach). Then create a new branch

```
git checkout -b 1245-fix(webui)-deployment-flyout
```

# Introduction

Bash is the shell, or command language interpreter, for the GNU operating system. The name is an acronym for the 'Bourne-Again SHell', a pun on Stephen Bourne, the author of the direct ancestor of the current Unix shell sh, which appeared in the Seventh Edition Bell Labs Research version of Unix. Bash is largely compatible with sh and incorporates useful features from the Korn shell ksh and the C shell csh. It is intended to be a conformant implementation of the IEEE POSIX Shell and Tools portion of the IEEE POSIX specification (IEEE Standard 1003.1). It offers functional improvements over sh for both interactive and programming use. While the GNU operating system provides other shells, including a version of csh, Bash is the default shell. Like other GNU software, Bash is quite portable. It currently runs on nearly every version of Unix and a few other operating systems - independently-supported ports exist for MS-DOS, OS/2, and Windows platforms.

The improvements offered by Bash include:

- Command line editing
- Unlimited size command history
- Job Control
- Shell Functions and Aliases
- Indexed arrays of unlimited size
- Integer arithmetic in any base from two to sixty-four

## Download

There are many ways to install bash. A couple options are as following:

### GNU

Bash can be found on the main GNU ftp server: <http://ftp.gnu.org/gnu/bash/> (via HTTP) and <ftp://ftp.gnu.org/gnu/bash/> (via FTP). It can also be found on the [GNU mirrors](#); please [use a mirror](#) if possible.

Click [here](#) for more details

### Install with Git

You can get bash on windows by [installing GIT](#). This might be the easiest approach for developers using Windows. During the install, choose Windows Explorer integration: :::image type="content" source="..../images/git-install.PNG" alt="Git install":::

### VS Code

Check out this: <https://stackoverflow.com/questions/42606837/how-do-i-use-bash-on-windows-from-the-visual-studio-code-integrated-terminal>

# Overview

NodeJS is a extension from [Joyant](#) that builds on Google's V8 Engine that offers a very fast run-time environment for solutions built using JavaScript that can run both client-side (web and desktop - via Electron) and server side. The speed comes from Google's non-blocking IO model.

□□□□□ BE SURE TO TARGET VERSIONS: **NPM** 6.4.1 **NODE** 10.14.1 (local) 11.1 (prod)

For the official installation documentation, check out: [NPM & Node](#)

We recommended using Node Version Manager (NVM) for Windows to be able to support switching between NodeJS versions (which also influences the active NPM version). We have found install issues on some Windows 10 Machines running McAfee. Follow these instructions to avoid install issues: <https://medium.com/@tysonpaul89/maintain-multiple-versions-of-node-js-in-windows-operating-system-using-nvm-3c6bf5b63f29>

Alternatively, to install NodeJS directly go to <https://nodejs.org/>, but you want be able to switch versions so be sure to install the correct version (see above).

Additional details about configuring and switching versions using NVM are below

Another helpful guide is: [Set up your Node.js development environment directly on Windows](#)

## NVM Alternatives

If you have install issues with NVM or are running on a non-windows machine, try the following:

While windows-nvm is currently the most popular version manager for node, there are alternatives to consider:

[nvs](#) (Node Version Switcher) is a cross-platform nvm alternative with the ability to [integrate with VS Code](#).

[Volta](#) is a new version manager from the LinkedIn team that claims improved speed and cross-platform support.

To install Volta as your version manager (rather than windows-nvm), go to the **Windows Installation** section of their [Getting Started guide](#), then download and run their Windows installer, following the setup instructions.

**Important:** You must ensure that [Developer Mode](#) is enabled on your Windows machine before installing Volta.

To learn more about using Volta to install multiple versions of Node.js on Windows, see the [Volta Docs](#).

## Node Version Manager (NVM)

Use Node Version Manager (NVM) to support multiple versions:

- Windows: <https://github.com/coreybutler/nvm-windows>
- Linux and MAC: <https://github.com/nvm-sh/nvm>

### Note

This solution uses older versions of NodeJS. Uses older versions will cause warning to be triggered like:

```
npm WARN npm npm does not support Node.js vXX.XX.XX These can be ignored.
```

It's best to do uninstall any direct node installs before installing NVM. Even though the install on windows seems to merge existing installs, you'll likely still run into issues with the Node Package Manager (NPM). See Uninstall Existing Node Documentation below:

Once installed, run the following to install the latest version of NodeJS:

## Useful NVM Commands

### List Versions currently installed

```
"engines": { "node": ">=0.12" }
```

```
list
```

### Install version

```
nvm install "latest"
```

```
or
```

```
nvm install 10.1
```

### Set current version

```
nvm use 10.1
```

### Set NPM Versions

You can change your NPM Versions up or down using the following

#### Getting Lastest NPM

Node comes with npm installed so you should have a version of npm. However, npm gets updated more frequently than Node does, so you'll want to make sure it's the latest version.

```
npm install npm@latest -g
```

#### Targeting Specific NPM Version

Just replace @latest with the version number you want to downgrade to. I wanted to downgrade to version 3.10.10, so I used this command:

```
npm install -g npm@3.10.10
```

**Note:** If you do a lot of version switching, you might run into the following issue:

Error: Node Sass does not yet support your current operating system: Windows 64-bit with Unsupported runtime 1.0e

There's an easy fix:

```
npm rebuild node-sass
```

Additional information can be found [here](#). This approach requires python to be installed.

## Uninstall Existing Node and NPM Versions

### Uninstall existing Node Version

Please note, you need to uninstall any existing versions of nodejs before installing NVM for Windows. Also delete any existing nodejs installation directories (e.g., "C:\Program Files\nodejs") that might remain. NVM's generated symlink will not overwrite an existing (even empty) installation directory.

### Uninstall existing NPM Version

You should also delete the existing npm install location (e.g. "C:\Users<user>\AppData\Roaming\npm"), so that the nvm install

location will be correctly used instead. Backup the global `npmrc` config (e.g.

`C:\Users\<user>\AppData\Roaming\npm\etc\npmrc`), if you have some important settings there, or copy the settings to the user config `C:\Users\<user>\.npmrc`.

## Helpful Tools

TOOL	DESCRIPTION	NOTES
NPM-Check	Display package versions and if updates are available.	Warning: Careful on making updates. Those are sweeping changes that shouldn't be taken lightly (unless of course you're not installing into the solution or changing dependencies).

## ESLint Issues

Sometimes ESLint causes a lot of issues. This often happens on a fresh clone that came from a repo built by a different OS. This issue can be easily resolved by running:

```
npm run lint -- --fix
```

# Overview

npm (Node Package Manager) is a package manager for the JavaScript programming language. It is the default package manager for the JavaScript runtime environment Node.js. It consists of a command line client, also called npm, and an online database of public and paid-for private packages, called the npm registry. The registry is accessed via the client, and the available packages can be browsed and searched via the npm website. The package manager and the registry are managed by npm, Inc.

□□□□□ BE SURE TO TARGET VERSIONS: **NPM** 6.4.1 **NODE** 10.14.1 (local) 11.1 (prod)

For the official installation documentation, check out: [NPM & Node](#)

## Command Line Client

npm includes a **CLI** (Command Line Client) that can be used to download and install software:

Windows Example

```
C:\>npm install <package>
```

Mac OS Example

```
>npm install <package>
```

## Installing npm

npm is installed with Node.js This means that you have to install Node.js to get npm installed on your computer. Download Node.js from the official Node.js web site: <https://nodejs.org>



## Overview

A light-weight editor that can run on-demand SQL queries, view and save results as text, JSON, or Excel. Edit data, organize your favorite database connections, and browse database objects in a familiar object browsing experience.

## Download

Azure Data Studio can be downloaded from [here](#).



## Overview

Upload, download, and manage Azure blobs, files, queues, and tables, as well as Azure Cosmos DB and Azure Data Lake Storage entities. Easily access virtual machine disks, and work with either Azure Resource Manager or classic storage accounts. Manage and configure cross-origin resource sharing rules.

## Download

Storage Explorer can be downloaded from [here](#)

# Introduction

The Azure IoT explorer is a graphical tool for interacting with and testing your IoT Plug and Play Preview devices. After installing the tool on your local machine, you can use it to connect to a device. You can use the tool to view the telemetry the device is sending, work with device properties, and call commands.

## Download

Azure IoT Explorer can be downloaded from [here](#)

## Install and use Azure IoT explorer

Click [here](#) to know how to install and use Azure IoT explorer

## Quick Demo

Check out this video for a quick e2e [demo](#).



# Installation guide for Docker

Docker is a set of platform as a service products that uses OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries and configuration files; they can communicate with each other through well-defined channels. To install it, following the guides below specific to your operating system.

## Docker on Mac

Installation instructions for installing Docker on a Mac can be found [here](#).

## Docker on Windows

### What to know before installation

- Windows 10 64-bit: Pro, Enterprise, or Education (Build 15063 or later).
- Hyper-V and Containers Windows features must be enabled.
- The following hardware prerequisites are required to successfully run Client Hyper-V on Windows 10:
  - 64-bit processor with [Second Level Address Translation \(SLAT\)](#)
  - 4GB system RAM
  - BIOS-level hardware virtualization support must be enabled in the BIOS settings. For more information, see [Virtualization](#).

For additional information about requirements see: [What to know before you install](#).

### Steps to setup Docker on Windows

- Download Docker Desktop <https://hub.docker.com/editions/community/docker-ce-desktop-windows/>
- Double-click Docker Desktop Installer.exe to run the installer.
- If you haven't already downloaded the installer (Docker Desktop Installer.exe), you can get it from [Docker Hub](#). It typically downloads to your Downloads folder, or you can run it from the recent downloads bar at the bottom of your web browser.
- Follow instructions on the installation wizard to accept the license, authorize the installer, and proceed with the install.
- When prompted, authorize the Docker Desktop Installer with your system password during the install process. Privileged access is needed to install networking components, links to the Docker apps, and manage the Hyper-V VMs.
- Click Finish on the setup complete dialog and launch the Docker Desktop application.

For additional information, see [Install Docker Desktop on Windows](#). To test your installation, follow [this guide](#).

### Alternative Setup

It might now be possible to use use Docker on Windows without having to virtualize thanks to the release of the Linux subsystem for Windows. This approach has not been tested to determine if it will work for this solution. It allows you to run Unbuntu and openSUSE on Windows, with Fedora and more Linux distributions coming soon.

As untried alternative to virtualization, if you're using Windows you could try the Linux subsystem for Windows.

Note: We have not tried this. Key consideration is whether or not docker images can be instantiated.

This documented might be a good guide: <https://docs.microsoft.com/en-us/virtualization/windowscontainers/quick-start/set-up-environment?tabs=Windows-10-Client>

For additional information on the Linux Subsystem: The Windows Subsystem for Linux, introduced in the [Anniversary Update](#), became a stable feature in the [Fall Creators Update](#).

# Installation guide for Terraform

Steps to set up terraform on Windows:

- To install Terraform, find the [appropriate package](#) for your system and download it. Terraform is packaged as a zip archive.
- After downloading Terraform, unzip the package. Terraform runs as a single binary named `terraform`. Any other files in the package can be safely removed and Terraform will still function.
- The final step is to make sure that the `terraform` binary is available on the PATH.
- Verify the installation by executing `terraform` on new terminal session

# Installation guide for Kubernetes

## Steps to set up a kubectl on Windows

- Download the kubectl.exe using a [link](#) and save the file in any folder on windows file system.
- Add the kubectl.exe folder location in path variable - "Advanced System Settings -> Advanced -> Environment Variables -> Path". For example, if you have saved file to C:/kube then add this folder path to the path variable.
- Open a command prompt and type kubectl and you should see all commands supported by kubectl.

## Install minikube \*(Not Mandatory)

- VT-x or AMD-v virtualization must be enabled in your computer's BIOS.
- Install the virtualization platform such as Virtualbox or KVM. You are not really required to configure the image.
- Download the minikube-windows-amd64 file from [here](#).
- Add this folder path location in path variable: "Advanced System Settings -> Advanced -> Environment Variables -> Path." For example, if you have saved the file to C:/kube then add this folder path to the path variable.
- Open the command prompt and fire a command minikube and you should see all the commands supported by minikube.

On windows, you can get similar kind of linux kind of user experience with Cygwin. Install Cygwin by following the steps listed on [its website](#).

### Start minikube :

- Open the Cygwin terminal and run command \$ minikube start.
- Run command kubectl version to confirm the working of minikube.

# Installation guide for Helm

## Prerequisites

- You will need the command line program kubectl installed on your Windows 10 computer and configured to work with a Kubernetes Cluster.
- 7-Zip compression / decompression program is needed to extract the Helm program for Windows from the compressed file-folder from the Helm site. You can download it here: <https://www.7-zip.org/download.html>

## Steps to Install

- Download the latest version of the compressed executable from the Helm GitHub site, <https://github.com/kubernetes/helm/releases>.
- Navigate to the folder you downloaded the helm-vX.X.X-windows-amd64.tar.gz compressed file from and move the file to its own directory.
- Navigate to the new directory and right click on the tar.gz file and with 7Zip, open the tar.gz archive.
- Double click the single tar file in that directory, helm-v2.7.2-windows-amd64.tar.
- You should now see a windows file folder in the 7Zip window, windows-amd64. Right click on the folder, select Copy To, and select the directory you want to copy the folder to.
- add the helm program to the System File path to make it easily accessible from the command line.
- Open the Control Panel's System panel.
- Select the Advanced system settings link on the left.
- Select Environment Variables.
- Under System variables, select Path, and then select Edit.
- Select the New button and then add the folder path where you copied the helm folder to and then press OK.
- Open a new command line window and type helm on the command line to make sure you have access to helm from the command line.
- Assuming you have the kubectl program configured for your Kubernetes cluster you can now initialize helm.
- Now you are ready to deploy Kubernetes applications to your kube cluster.

**Note:** Ubuntu and an evaluation copy of Windows. Otherwise you need an ISO and License Key for the OS you intend to install. Using the eval requires a 16GB download.

# Overview

Redux is an open-source JavaScript library for managing application state. It is most commonly used with libraries such as React or Angular for building user interfaces. It was created by Dan Abramov and Andrew Clark.

Redux is a predictable state container for JavaScript apps.

It helps you write applications that behave consistently, run in different environments (client, server, and native), and are easy to test. On top of that, it provides a great developer experience, such as [live code editing combined with a time traveling debugger](#).

You can use Redux together with [React](#), or with any other view library. It is tiny (2kB, including dependencies), but has a large ecosystem of addons available.

## Installation

To install the stable version:

`npm install redux` This assumes you are using [npm](#) as your package manager.

If you're not, you can [access these files on unpkg](#), download them, or point your package manager to them.

Click [here](#) to get more details

# Introduction

Postman is a collaboration platform for API development. Postman's features simplify each step of building an API and streamline collaboration so you can create better APIs—faster.

Get more detail [here](#)

## Installation and updates

Postman is available as a native app for Mac, Windows (32-bit / 64-bit), and Linux (32-bit / 64-bit) operating systems.

To get the latest version of the Postman app, visit the [download page](#) and click Download for your platform.

Installing Postman

- [Mac](#)
- [Windows](#)
- [Linux](#)



# Guide

A PowerShell script for configuring DevSpaces is available [here](#). Works on Windows and macOS. Not sure about Linux. But we also have a bash version (under Files\Tools\DevSpaces). Additional documentation is [here](#).

Here are instructions on how to use:

To use this file, download DevSpaces.ps1 to your machine

Then edit DevSpaces.ps1 line 10 and change the \$script:MmmSourceDirectory variable to point to the containing folder of the Serenity repository

Then edit your PowerShell profile and add a line to "dot-source" DevSpaces.ps1 like so:

```
. 'C:\Users\aa30hzz\DevSpaces.ps1'
```

Then, restart your PowerShell Core terminal

Then, create a new DevSpace like so (creates the default/kyle DevSpace):

```
New-DevSpace -Name kyle -Parent default
```

Then, deploy Serenity to the DevSpace like so:

```
Start-SerenityDevSpace -Name kyle
```

```
Use Stop-SerenityDevSpace and Remove-DevSpace to stop the DevSpace and remove the DevSpace.
```

## Scripts

NAME	WINDOWS	MACOS	LINUX
Configure Dev Spaces	<a href="#">DevSpaces.ps1</a>	<a href="#">DevSpaces.ps1</a>	<a href="#">new_dev_space.sh</a>



# Install Guide

You can download the tool from here: <https://xunit.net/>

XUnit is a unit testing tool for the .NET Framework



# Source Code Overview

The 3M IoT Platform Source Code is open-source and located on GitHub.com here: <https://github.com/3mcloud/azure-iot-platform-dotnet> You can work with the source code locally to better understand how the solution work or make improvements to the platform.

To get started, you

# Naming Branches and Pull requests

Branch and Pull Request naming conventions are used to keep track of different builds and now, as part of our move to open source, for change log documentation. This guide documents how to format the names (or "Titles") so we all standardize on the same convention.

## Title Format

The title of a Branch and PR format should conform to the conventions defined by [Conventional Commits v.1.0.0](#). The convention describes key flags that are picked up by the build process and recommends how to frame the body of the title, a human friendly description of the changes.

In summary, conventional syntax looks like this:

```
<type>[optional scope] <description>
```

In addition to following the conventions, all branches should be prefixed with the Work Item ID. Do not include the story type (Bug, User Story, etc.) or any separators between the id and the body, such as hyphens or colons. The next section summarizes these differences.

### Stage: On Create Branch

When creating a branch, prefix the title with the work item ID and then follow the conventional format.

For example:

```
15983 fix(webui) correct minor typos in code
```

### Stage: On Merge

When merging a branch, remove the prefixed work item id. Then Squash and Merge to commit the changes into master.

For example:

```
fix(webui) correct minor typos in code  
--
```



# Prerequisites

Once you've installed the appropriate tools according to your role, you need to configure your environment to begin working with the IoT Platform source code. To do so, complete the following setup instructions.

## One-Time Setup

Ensure the `AppConfigurationConnectionString` is set before building so that the `Mmm.Iot.Config.ClassGeneration` NuGet package can execute during build to automatically generate configuration classes in `./common/Services/Config` based on key-value pairs in Azure App Configuration and Azure Key Vault. This can be done in one of two ways:

1. Set an environment variable
2. Use `dotnet user-secrets` (recommended)

Either way, you will need to choose an Azure App Configuration instance and make note of its `<name>` and `<resource-group>` for use in the steps below.

### Set an environment variable

#### Windows

In a PowerShell shell:

```
[System.Environment]::SetEnvironmentVariable('AppConfigurationConnectionString', (az appconfig credential list --name <name> --resource-group <resource-group> --query "[?name=='Primary'].connectionString | [0]" --output tsv), 'User')
```

#### Non-Windows

Set the `AppConfigurationConnectionString` environment variable in the Bash configuration file of your choice.

### Use dotnet user-secrets

#### Windows

In a PowerShell shell:

```
dotnet user-secrets set --project ./src/services/common/Services/Services.csproj  
AppConfigurationConnectionString (az appconfig credential list --name <name> --resource-group <resource-group> --query "[?name=='Primary'].connectionString | [0]" --output tsv)
```

Then check the value of the secret:

```
dotnet user-secrets list --project ./src/services/common/Services/Services.csproj --json | Select-Object -Skip 1 | Select-Object -SkipLast 1 | ConvertFrom-Json | Select-Object -ExpandProperty AppConfigurationConnectionString
```

#### Non-Windows

In a Bash shell:

```
dotnet user-secrets set --project ./src/services/common/Services/Services.csproj  
AppConfigurationConnectionString `az appconfig credential list --name <name> --resource-group <resource-group> --query "[?name=='Primary'].connectionString | [0]" --output tsv`
```

Then check the value of the secret:

```
dotnet user-secrets list --project ./src/services/common/Services/Services.csproj --json | sed '1d;$d' | jq --raw-output '.AppConfigurationConnectionString'
```

# Building

## Build all services

```
dotnet build Mmm.Iot.sln
```

## Build an individual service

```
dotnet build ./<service-name>/<service-name>.sln
```

E.g., to build the Storage Adapter service:

```
dotnet build ./storage-adapter/storage-adapter.sln
```

## Build a Docker image for an individual service

You must provide a value for the `AppConfigurationConnectionString` environment variable to the Docker build. This value is a secret and must not be set directly in the Dockerfile via the `ENV` instruction. Therefore, you must provide the value in the `--build-args` option of the `docker build` command.

```
docker build --file ./<service-name>/WebService/Dockerfile --build-arg  
AppConfigurationConnectionString=$AppConfigurationConnectionString .
```

where `$AppConfigurationConnectionString` is either an environment variable or dotnet user secret.

E.g., to build the Storage Adapter container image:

```
docker build --file ./storage-adapter/WebService/Dockerfile --build-arg  
AppConfigurationConnectionString=$AppConfigurationConnectionString .
```

# Running

## Run all services

Azure DevSpaces

Non-Windows

First, setup your Azure DevSpaces for use

```
azds use \  
--name <Name of your managed kubernetes cluster> \  
--resource-group <Name of the resource group your cluster is within>
```

After setting up Azure DevSpaces, you can build and run each microservice in Azure DevSpaces. From the root of the project, run the script `up.sh`. To build and run each service in Azure DevSpaces.

```
./up.sh
```

Each service that is built and ran by `up.sh` has a unique url in the format of `http://my-dev-space.s.default.service-name.ABC1234DE.cus.azds.io`. You can use this url to begin making requests to each service's API endpoints.

## Run an individual service

The simplest is to use `dotnet run` to spin up a service on a random port on localhost:

```
dotnet run --project ./<service-name>/WebService/WebService.csproj
```

## Debugging

Use either Visual Studio or Visual Studio Code

# Working with 3M Cloud Repository

This documentation is about understanding the process of cloning the 3M cloud repository outside the 3M network to local machines and working with it.

1. Connect to Github with SSH
2. Clone solution
3. Creating new working branch
4. Commit and push the branch
5. Raise a pull request
6. PR Approval

## Connect to Github with SSH

Using the SSH protocol, you can connect and authenticate to remote servers and services. Use the steps below for connecting to github with SSH

**Check for existing keys/ Generate new key**

Check for existing keys: Before generating a new SSH key, you can check if you have any existing keys from [here](#). Generate a new key if you don't have an existing public and private key pair, or don't wish to use any that are available to connect to GitHub.

**Generate a new SSH key**

Generate a new SSH key: If there is any existing SSH key, this step can be skipped. Follow the [steps](#) to generate a new key.

**Add SSH key to ssh-agent**

Add SSH key to the ssh-agent: Once you have the key ready, follow the steps [here](#) to add your SSH key to the ssh-agent.

**Add SSH key to Github account**

Add SSH key to Github account: Once the key is added to the ssh-agent, login to Github and follow the steps [here](#) in order to add your SSH key to Gitub account.

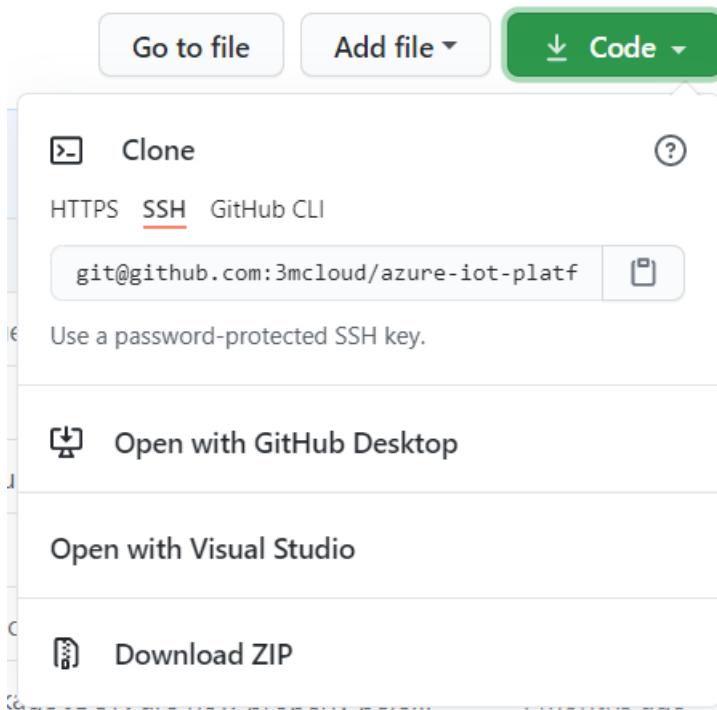
**Testing the SSH connection**

Testing the SSH connection: After you've set up your SSH key and added it to your GitHub account, you can test your connection by following the steps [here](#)

## Clone Solution

To clone the solution, do the following:

1. Login to Github
2. Navigate to 3M cloud [azure-iot-platform-dotnet](#) repository
3. Click on Code button as shown below, switch to SSH and copy the path to clone



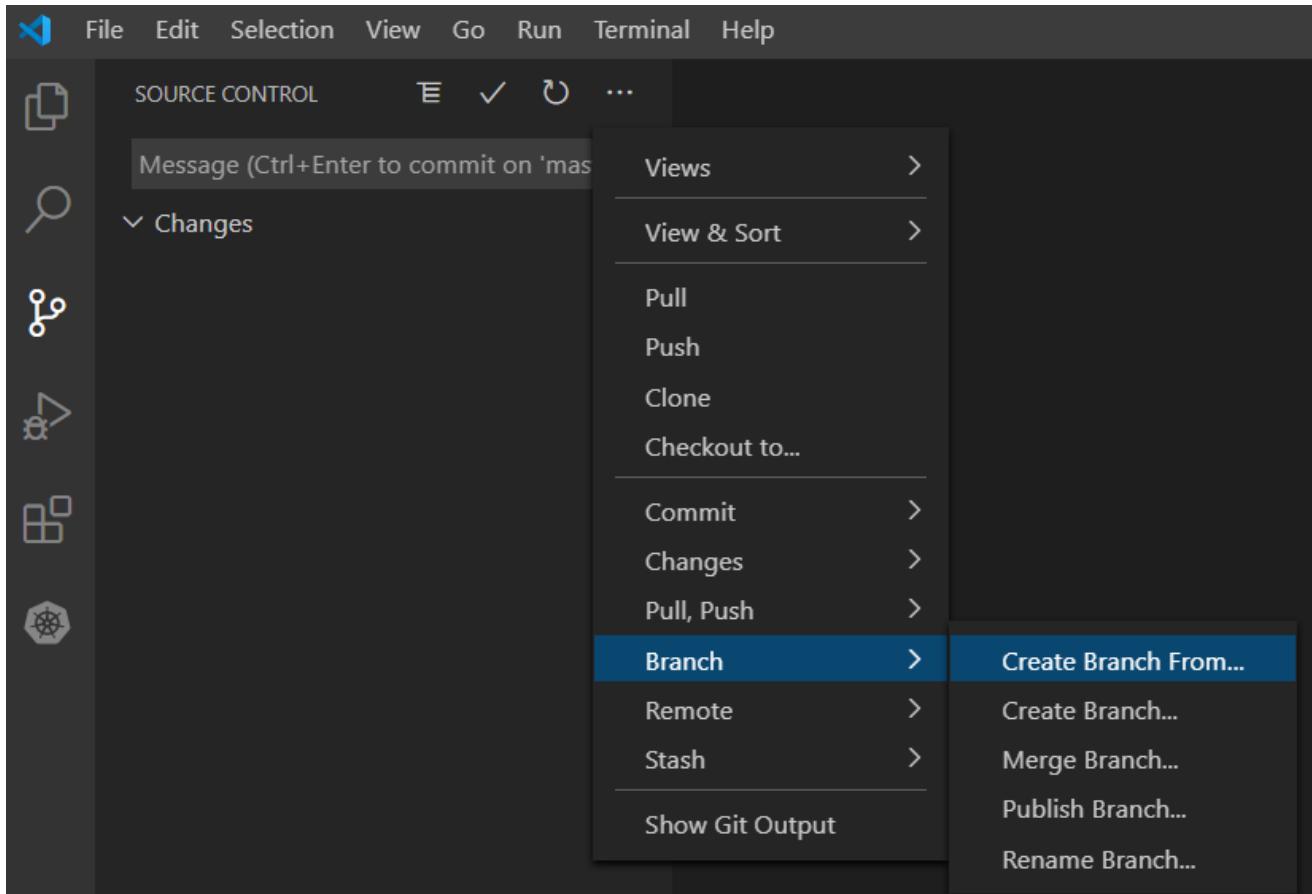
4. Clone the repository with this path from VS code/Visual Studio/Git bash or any other command liner

## Creating new working branch

1. Create a new branch from master branch either from terminal or from Source Control window/palette

Ex: Creating a new branch in VS Code:

1. Go to Source Control(Ctrl+Shift+G)
2. Click on Views and More Actions
3. Go to Branch => Create Branch From



2. The naming convention should be followed based on team guidelines for different work item types(user story/bug)

Ex: Branch name **for** a **user** story: userstory-<userstorynumber>-<title/shortdescription>

3. The new branch is setup and always make sure to keep your branch upto date before raising a pull request

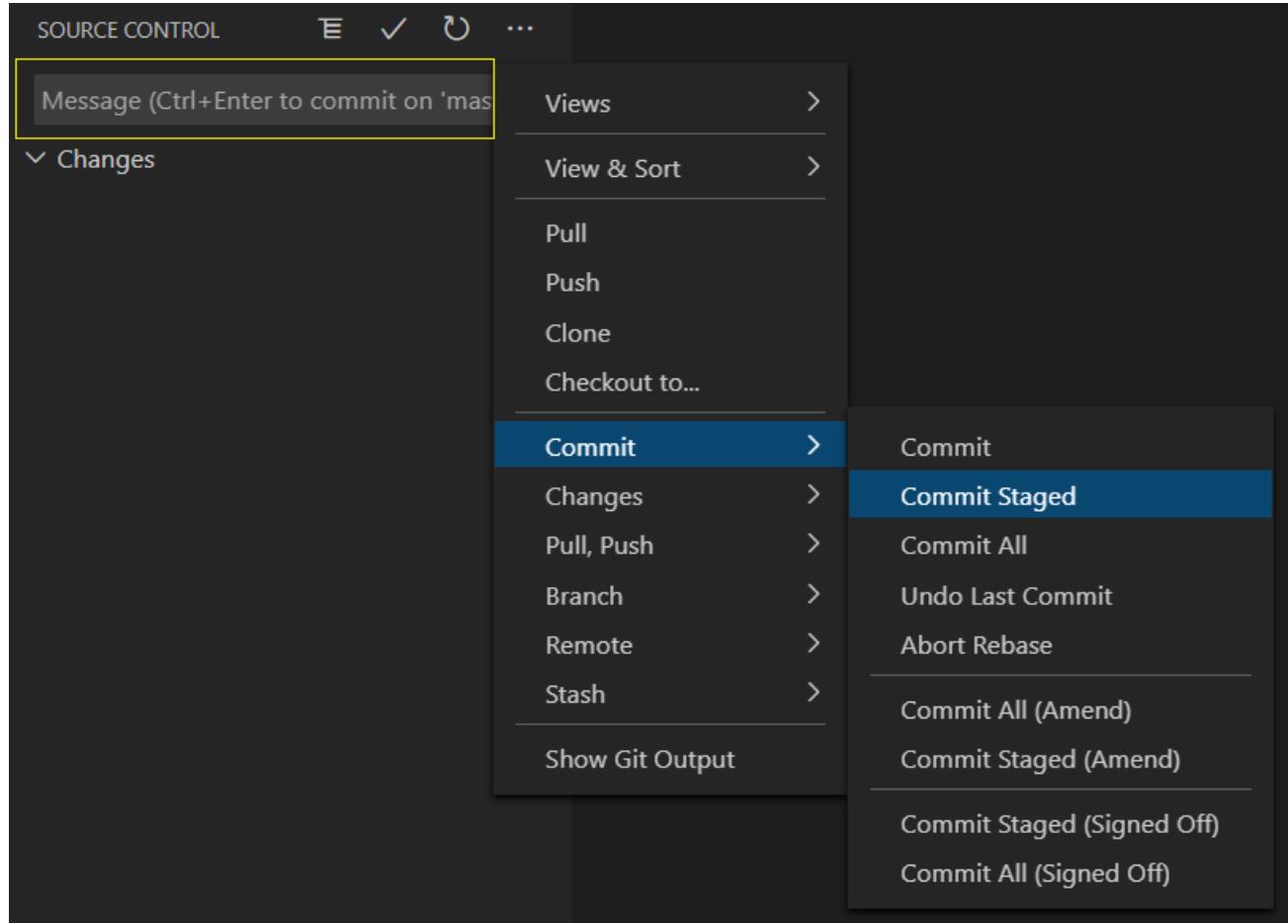
## Commit and push the branch

1. Once the changes are done, go to pending changes and stage the changes to check-in

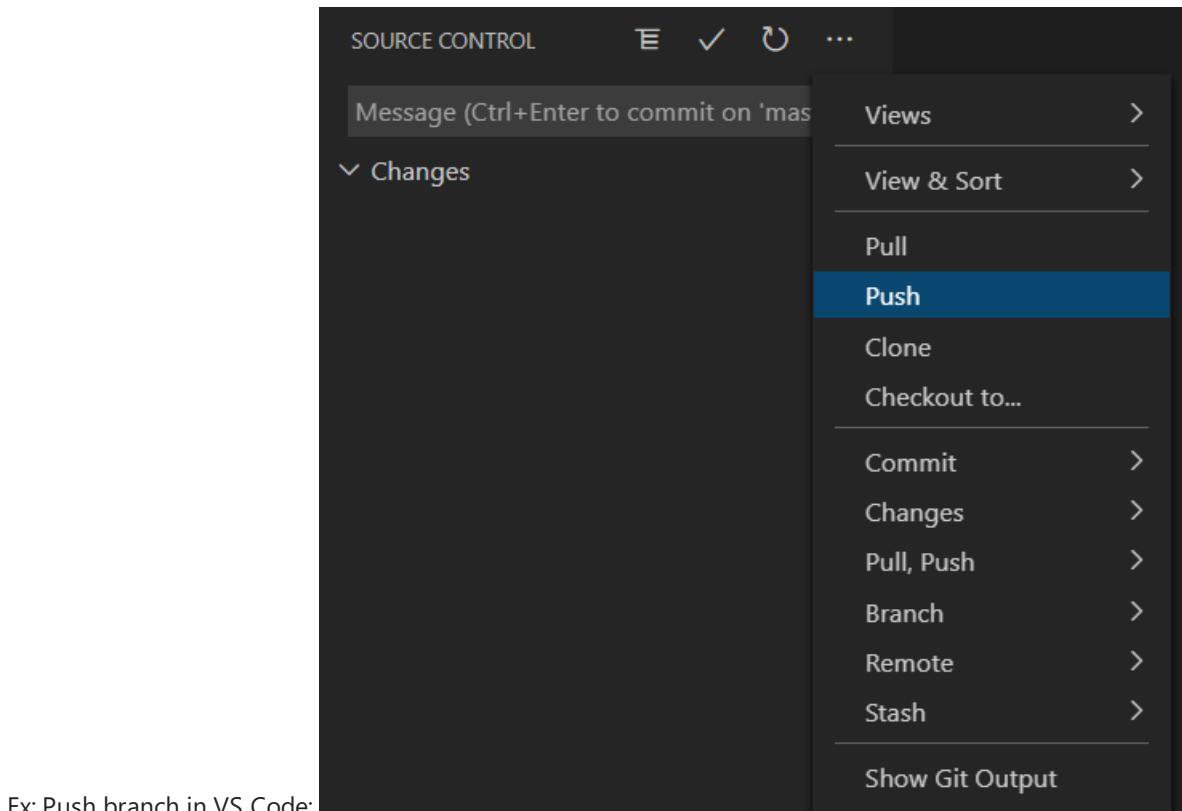
Note: Make sure to get latest/keep the branch in sync with master before committing in order to prevent any conflicts due to changes from the upstream

2. Enter the commit message with information about your changes and commit your changes

Ex: Committing staged changes in VS Code:



3. Once the changes are committed push the branch to origin



Ex: Push branch in VS Code:

## Raise a pull request

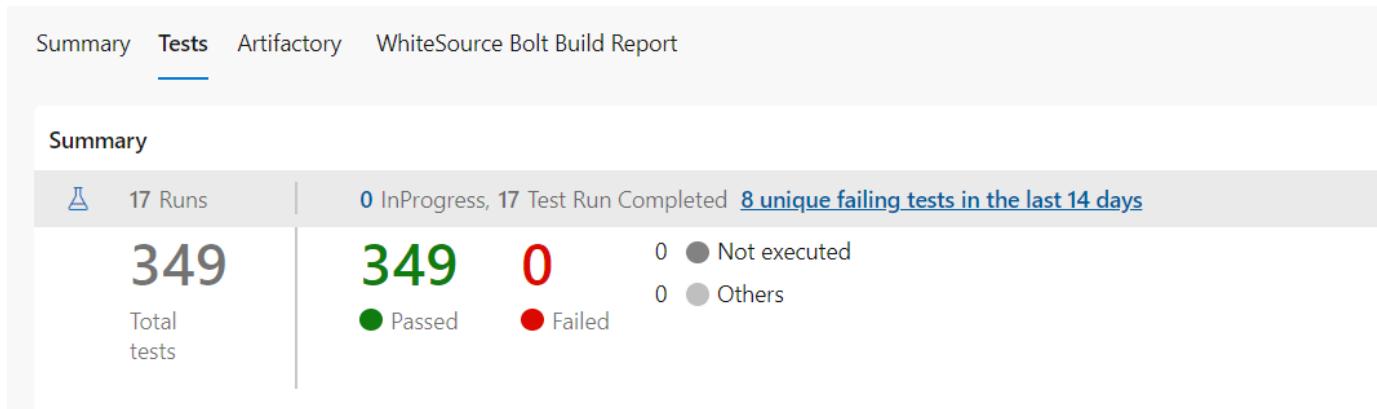
1. Login to VM
2. Open the browser and navigate to [pull requests tab](#)
3. Click "New pull request"
4. Select **base:master** and **compare:<your\_branch\_name>**
5. If required, review the commits and files changed
6. Give a brief description about the changes under Conversation tab
7. Add the revieweres if required and create the pull request

Note:

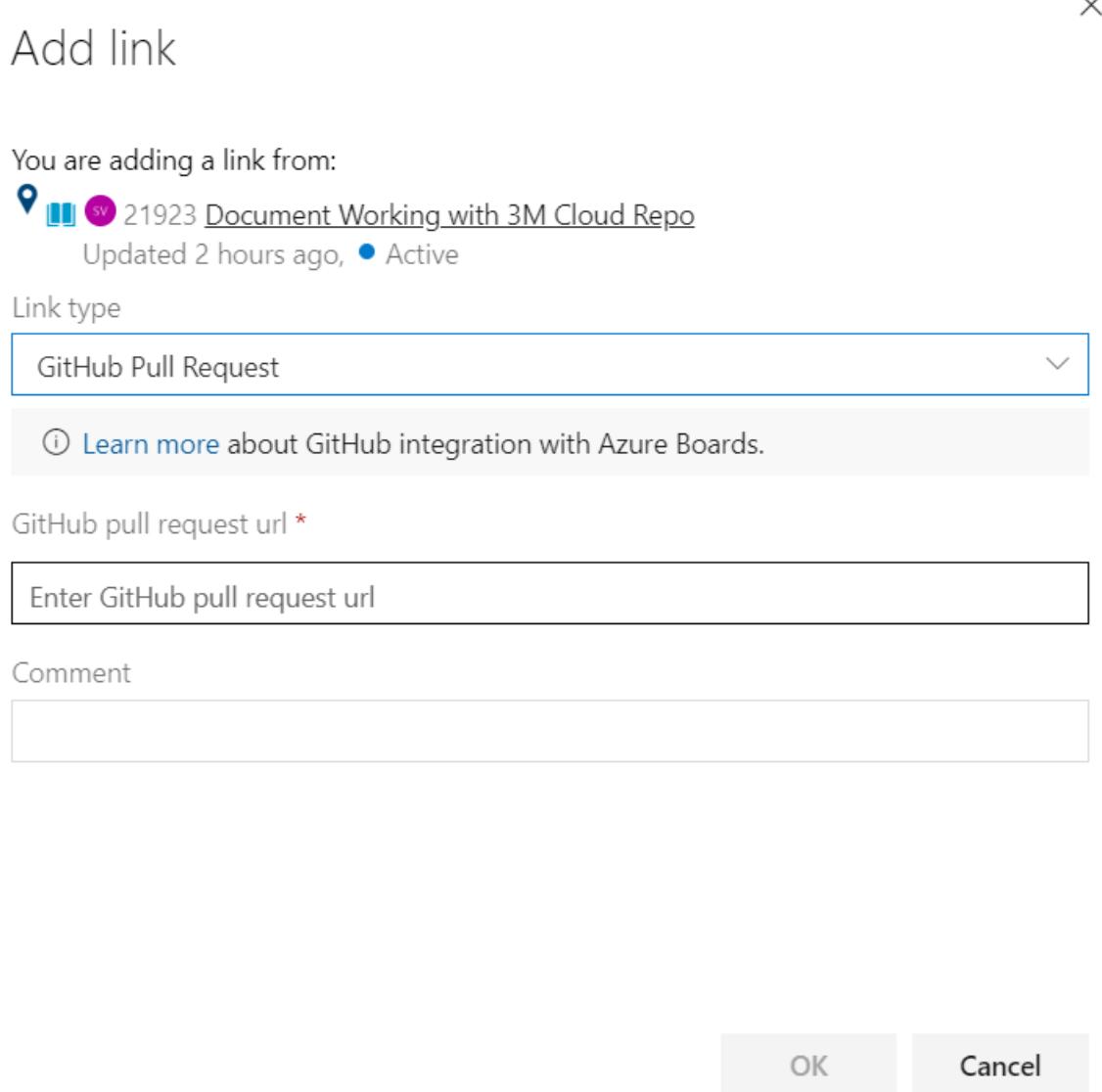
For more information on Naming branches and Pull requests, [check this](#)

## PR Approval

1. Notify others for review: Either post on Wiki or in a team's Communication channel regarding the Pull request
2. Address the changes requested or any merge conflicts
3. If a build runs with your PR, make sure that the checks are 100% passed. This can be viewed under "Checks" section.
4. Make sure that the tests are 100% passed(if there are any). Troubleshoot the errors if there are any. You can verify these under Tests tab in the build.



5. If the build fails, go through the log and address the issue/inform the team
6. If all the checks are passed, the PR is ready for approval.
7. Update the work item
  - o Set the status as Review
  - o Add discussion notes and attach any files if required for reference
  - o Link the Pull request. For this,
    1. Go to the respective work item
    2. Under Development section, select Add link
    3. Add link pop-up opens up
    4. Select the link type as "GitHub Pull request"
    5. Enter the Pull request url under "Github pull request url" field
    6. Add a comment if required and click "Ok"



- o Save the work item
8. Once the deployment to Development environment is completed:
  - o Test your changes
  - o Add appropriate notes
  - o Mark the work item as resolved

## Useful References

1. [Connecting to Github with SSH](#)

2. [Working with repositories in VS Code](#)
3. [Working with repositories in Git Extensions](#)



# Installing and Configuring 3M's Azure IoT Source Code

Assuming you've completed the major installs required to run the source code, you're now ready to clone the 3M repo and configure the solution on your local development environment.

To do so, requires the following steps:

1. Clone Repo
2. Configure Environment Settings
3. Update Source Code Dependencies
4. Build and Run

You might also want to watch the setup video: [https://www.youtube.com/watch?v=aslCCW\\_1uiI](https://www.youtube.com/watch?v=aslCCW_1uiI)

## Grant KeyVault Access to Project

```
dotnet user-secrets set --project C:{your path}\src\services\common\Services\Services.csproj AppConfigConnectionString  
"Endpoint=https://crsliot-appconfig-dev.azureconfig.io;id=AMGf-l4-  
s0:rAtTcp3u1hOQszVB49Tg;Secret=R4VYRXHcDx8FqQxc9xNC4F16Mxt0yG4FReBFVxQF7ls="
```

## TODO

Organize content into the sections below and provide coverage of how/when to use: Icons in React JS Fluent Framework:  
<https://developer.microsoft.com/en-us/fluentui#/styles/web/icons>

<crsliot-aks-dev.centralus.cloudapp.azure.com>

### Web UI:

```
dotnet user-secrets set --project C:{your path}\src\services\common\Services\Services.csproj AppConfigConnectionString  
"Endpoint=https://crsliot-appconfig-dev.azureconfig.io;id=AMGf-l4-  
s0:rAtTcp3u1hOQszVB49Tg;Secret=R4VYRXHcDx8FqQxc9xNC4F16Mxt0yG4FReBFVxQF7ls="
```

```
.env NODE_PATH src/
```

### App config:

```
const baseUrl = "https://crsliot-aks-dev.centralus.cloudapp.azure.com"; //process.env.REACT_APP_BASE_SERVICE_URL || "";
```

```
dotnet user-secrets set --project C:\DevOps\OpenSource\azure-iot-platform-dotnet\src\services\common\Services\Services.csproj  
AppConfigurationConnectionString "Endpoint=https://crsliot-appconfig-dev.azureconfig.io;id=AMGf-l4-  
s0:rAtTcp3u1hOQszVB49Tg;Secret=R4VYRXHcDx8FqQxc9xNC4F16Mxt0yG4FReBFVxQF7ls="
```

```
dotnet build mmm.ios.sln C:\DevOps\OpenSource\azure-iot-platform-dotnet\webui
```

<https://github.mmm.com/mmm/azure-iot-services-dotnet>

```
npm install
```

```
npm start
```

## Clone Repo

There are many ways to clone the 3M Repo. Doing so depends on having Git installed. 3M IoT Platform on Azure - © 3M 2020

- Internal Repo: <https://github.mmm.com/mmm/azure-iot-services-dotnet>
- Public Repo: <https://github.com/3M-Company/azure-iot-platform-dotnet/>

## Configure Environment Settings

Use the relevant sections below to configure environment Settings

Follow recommendations here: <https://github.com/3M-Company/azure-iot-platform-dotnet/blob/master/docs/DEVELOPMENT.md> More information on configuring environment variables here.

WebUI Environment Variables Use the section below that matches your environment.

- Windows
- MAC
- LINUX

Docker and Kubernetes Settings

## WebUI Development Settings

It's necessary to set a few environment settings to get things up and running. Use the section below that matches your environment:

Update Source Code Dependencies

## Build and Run

If you have SASS issues run:

```
npm run lint -- --fix
```

## Check-in Changes

TODO: <https://desigmodo.com/react-ci-cd/>

## Services Development Settings

TODO: 3MC02YM21KJG5J:webui a9q25zz\$ export REACT\_APP\_BASE\_SERVICE\_URL="<https://crsliot-aks-dev.centralus.cloudapp.azure.com/>"

### References

Azure Iot UX Fluent Controls	<a href="https://www.microsoft.com/design/fluent/">https://www.microsoft.com/design/fluent/</a>	<a href="https://github.com/Azure/iot-ux-fluent-controls">https://github.com/Azure/iot-ux-fluent-controls</a>

## Library version updates in 3M source code

Web UI section

update "node-sass" version to "4.13.1"

**To install, use:**

```
npm install node-sass@4.13.1
```