

# 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](#)

# 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 *Content Filter...* to narrow down specific topics. Throughout this documentation you'll find a Downloaded PDF to allow you to take the material offline.

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.

## 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.

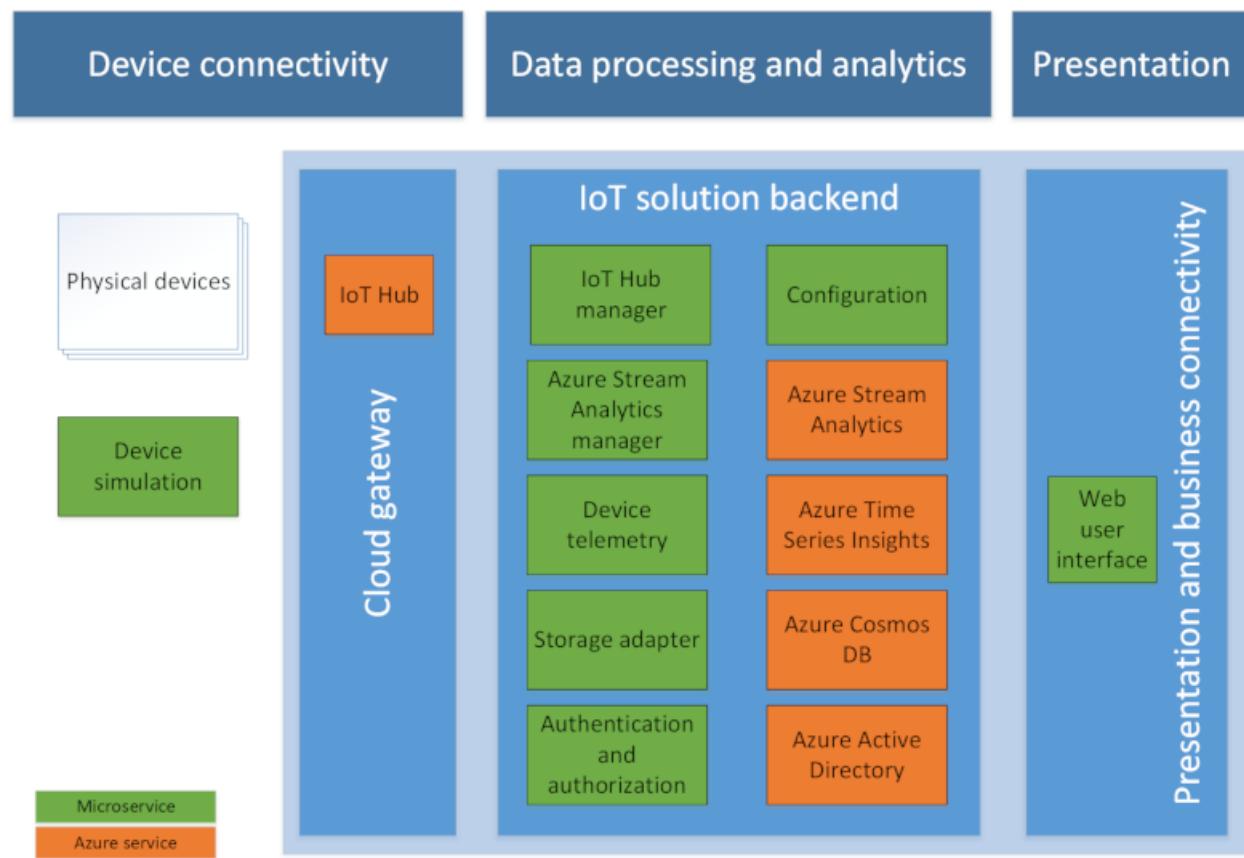
# 3M IoT Platform Overview

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.

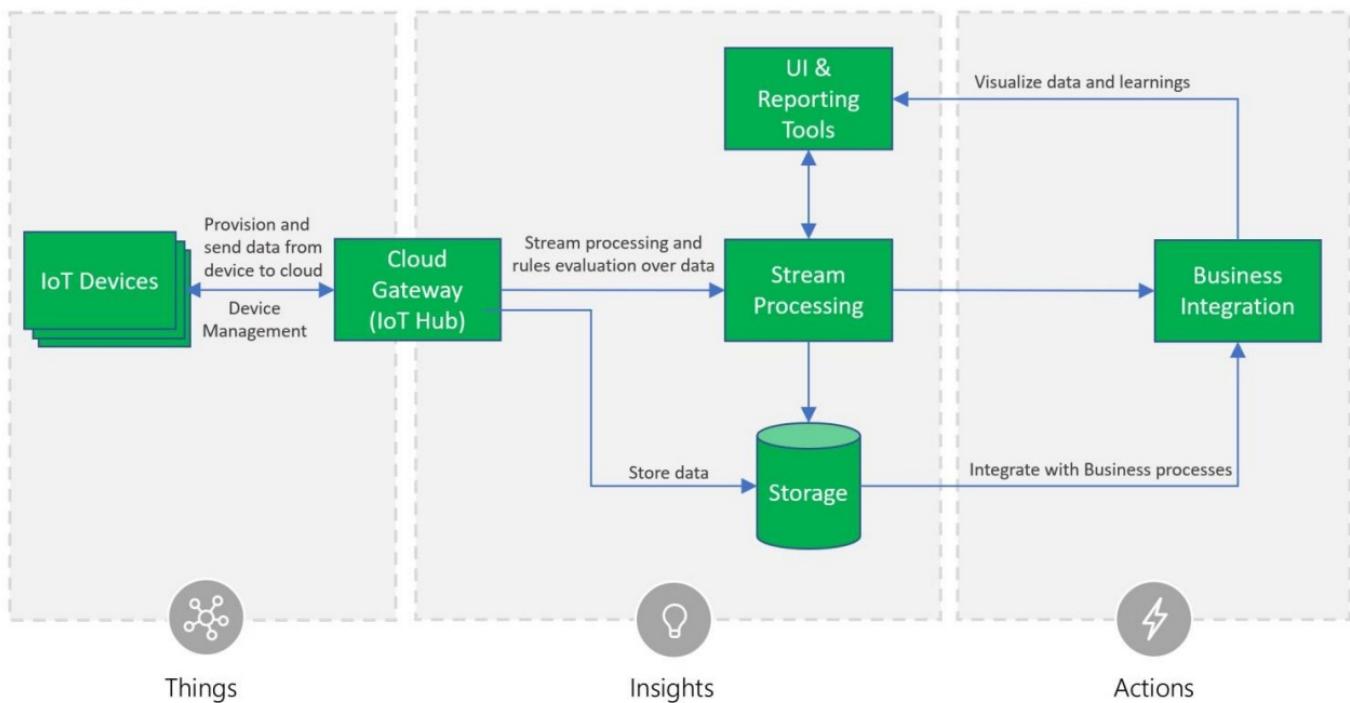
### Authentication and authorization microservice

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 platforms common services includes 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 dashboard offers:

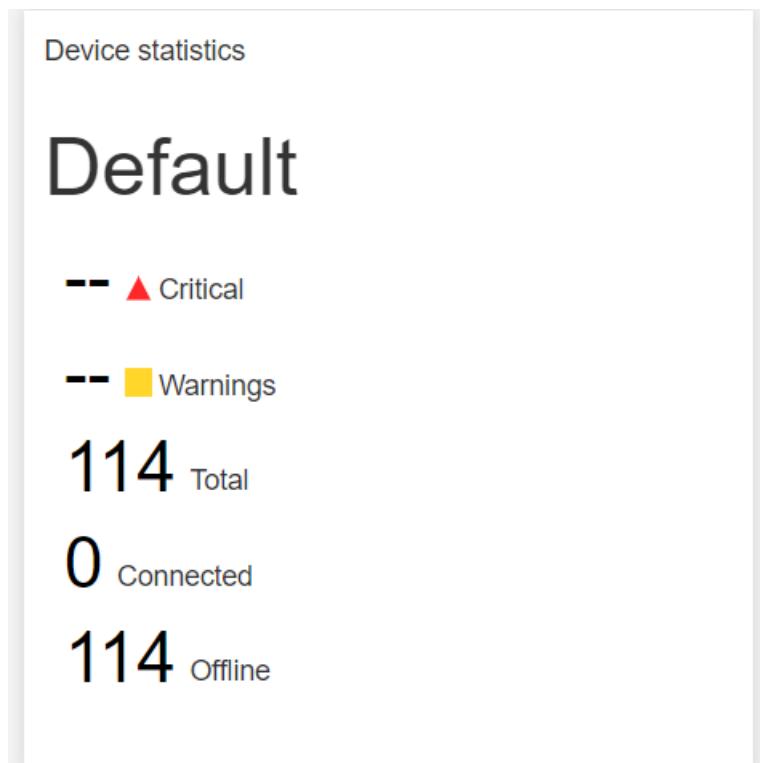
1. [Device Statistics](#)
2. [Device locations](#)
3. [Alerts](#)
4. [Telemetry](#)
5. [Analytics](#)

## Device Statistics

Device Statistics provides collective data on IoT Devices' state and health.

Device Statistics provides the following information

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

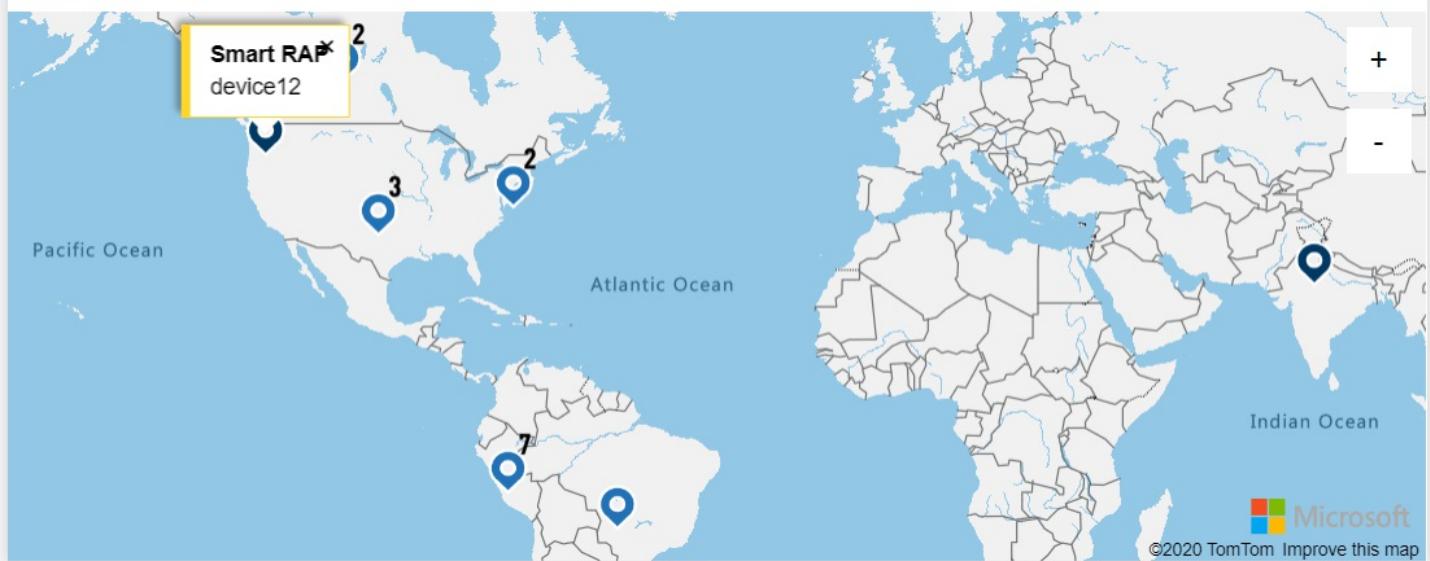


## Device locations [refresh]

Device locations provide the location of IoT Device in the world map, it only plots those devices configured with coordinates.

Upon clicking pin [info] on the map, it reveals the device type and device name.

## Device locations



## Alerts □□

Alerts list down the anomalies in telemetry based on rules.

Below are the details as follows:

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

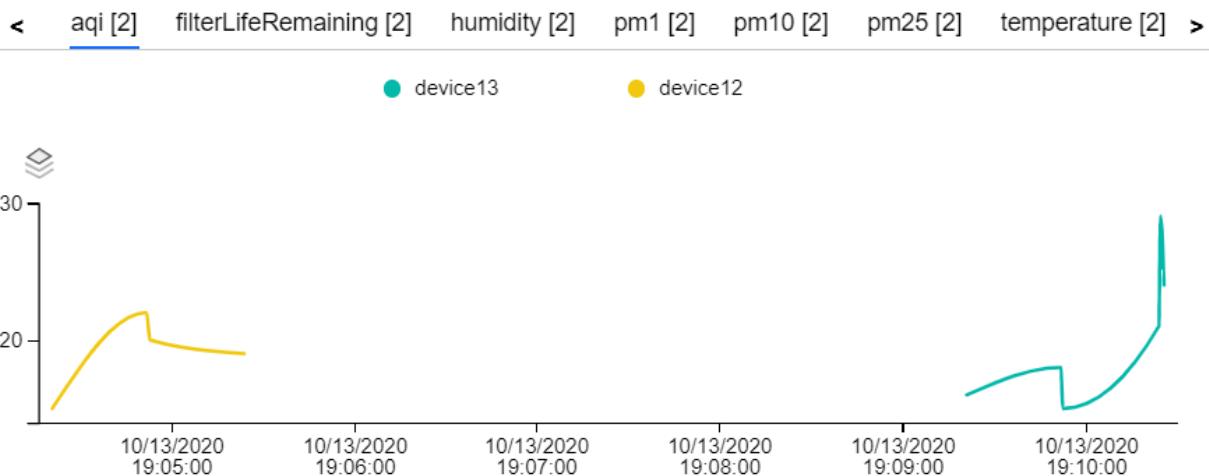
Alerts				
Rule name	S.	C.	E.	> Expand Columns
ShiftHumidityGreaterThan70	▲	C..	2	...
ShiftTemperatureGreaterT...	▲	C..	4	...
temp30	■	W..	10	...

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



Displaying in local timezone: Asia/Calcutta

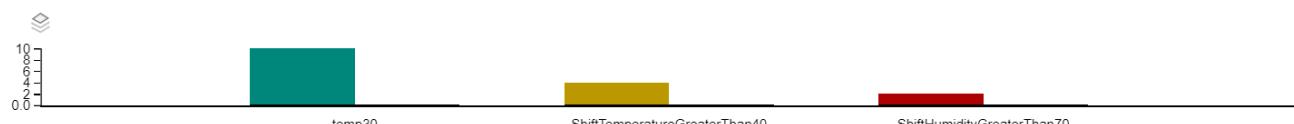
## Analytics

Analytics provide the visual representation of alerts data in the forms of charts □ as follows:

1. Top rules with active alerts
  - o It is visualized in the form of a Bar Graph □.
  - o Graph is represented for number of alerts recorded per rule.
2. Alert by device group
  - o It is visualized in the form of a Pie Graph □□.
  - o Graph is represented for 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%

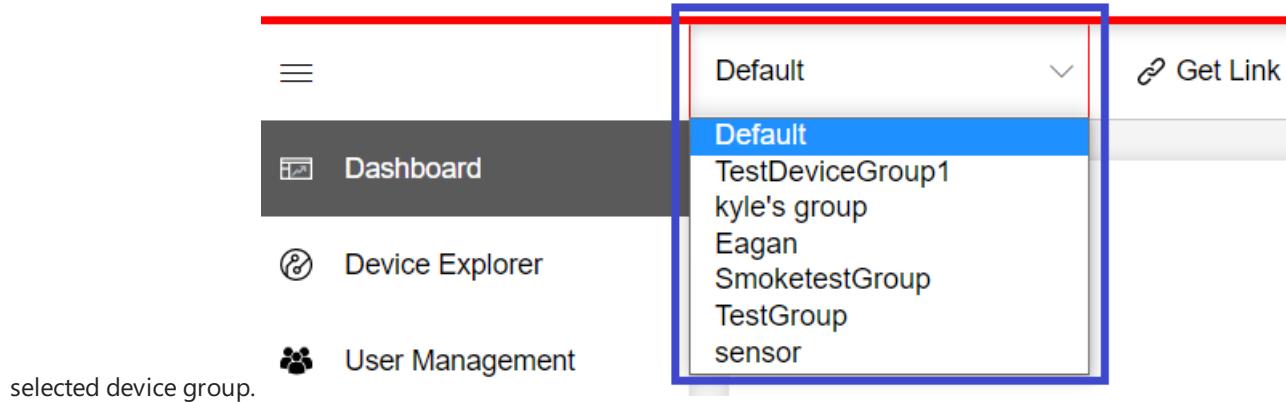
## Tool Bar

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.

The screenshot shows the Device Explorer dashboard. On the left is a sidebar with icons for Dashboard, Device Explorer, User Management, Rules, Packages, Deployments, and Maintenance. The 'Dashboard' icon is highlighted. The main area has a dropdown menu at the top labeled 'Default' with a blue border. To the right of the dropdown are 'Get Link' and 'Manage' buttons. Below the dropdown is a section titled 'Device statistics' with a large 'Default' heading. A blue arrow points from the 'Default' text in the dropdown to the 'Default' text in the stats section, labeled 'Selected Device group'. Another blue arrow points from the 'Default' text in the stats section to the 'Default' text in the dropdown, labeled 'Device group dropdown'. Below the stats are two status indicators: 'Critical' (red triangle) and 'Warnings' (yellow square). Underneath are three numerical counts: '114 Total', '0 Connected', and '114 Offline'.

#### Note:

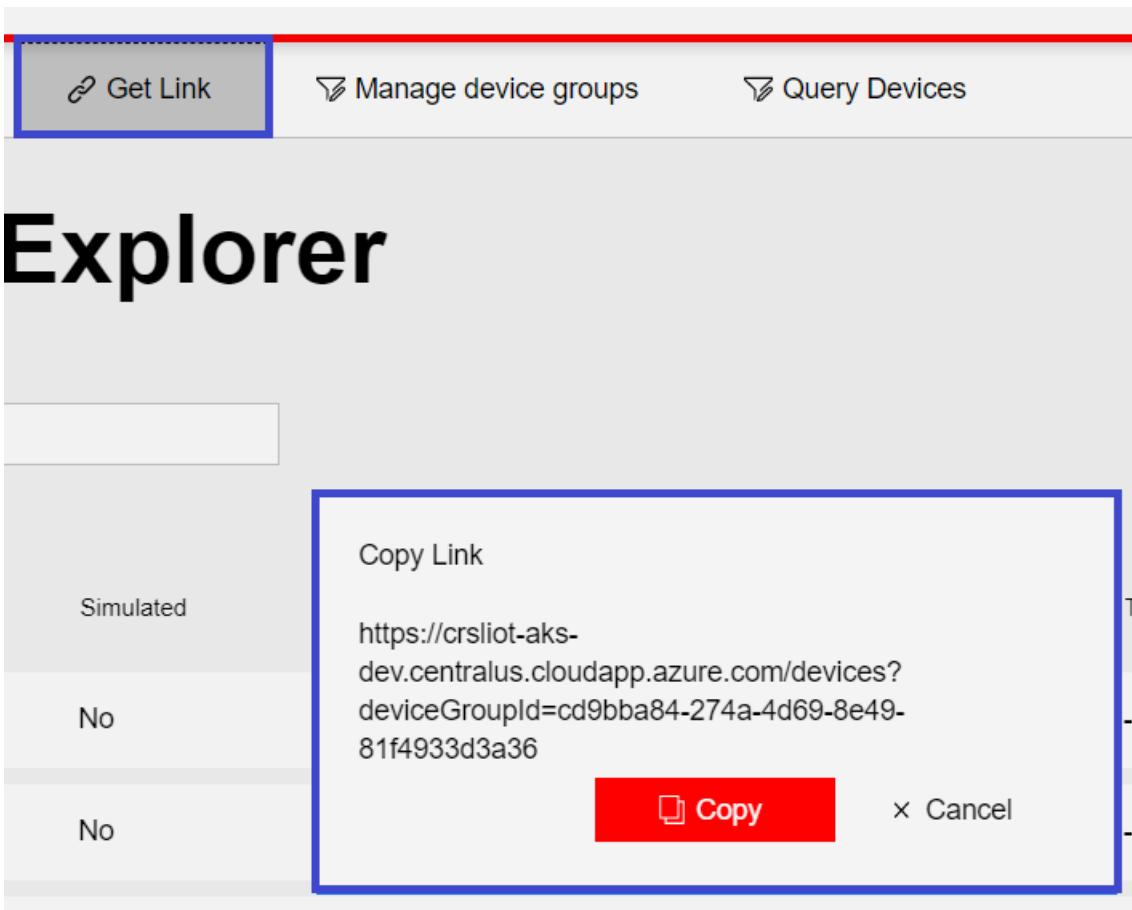
By default, the device group **Default** is selected for any user for the first login.

## 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

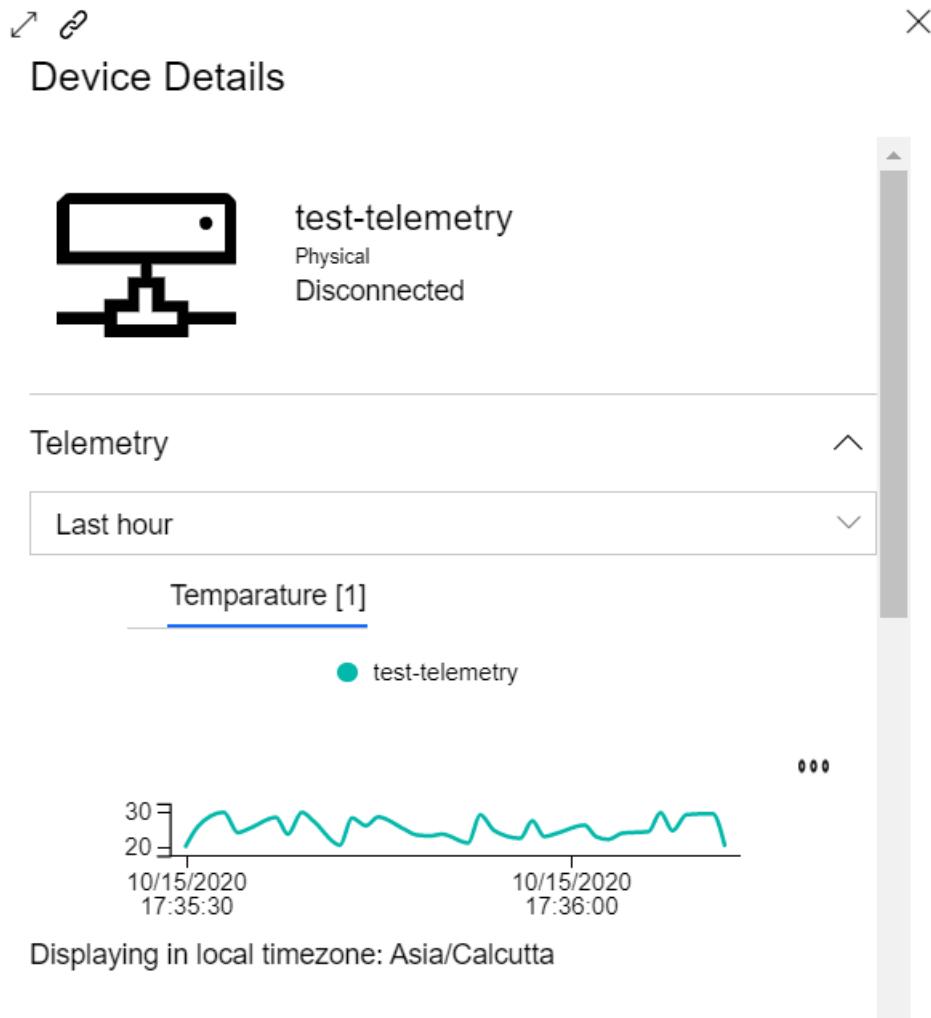


+ Add

- 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

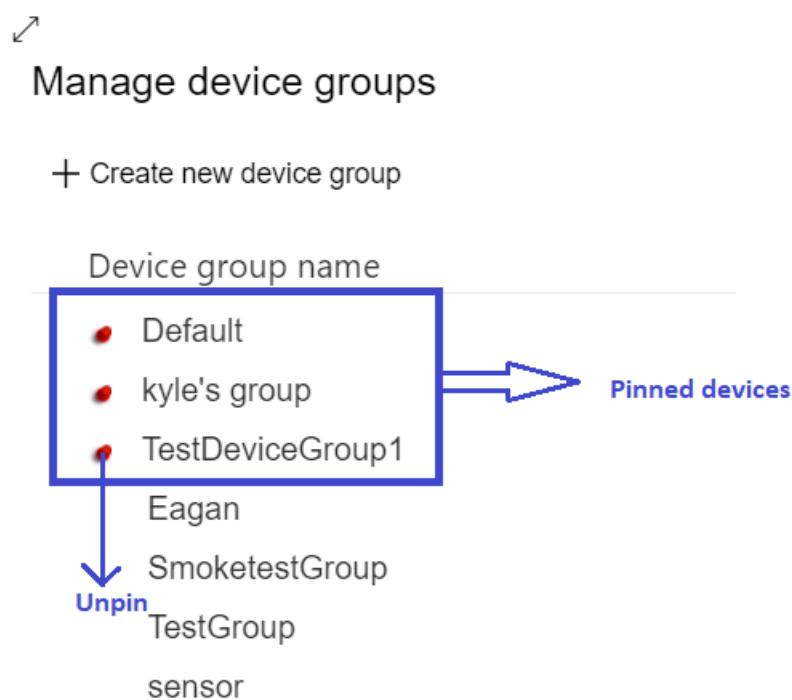
A user can pin one/multiple device groups using this option. Pinned device groups always show on top of the Device group list.

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

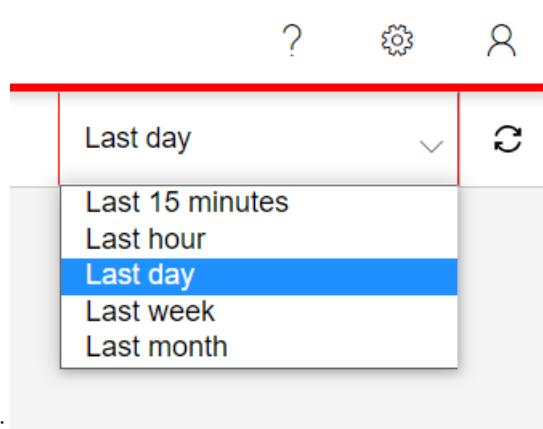
sensor

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

Device Explorer is used for maintaining the IoT devices. This section explores the creation, 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

Device Explorer Grid lists all the devices which are registered to Azure IoT Hub.

- Following lists the columns and usage in 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 device either connected or disconnected to 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 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

Step for adding a device.

- Navigate to **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 below device managed service
  - **IoT Edge device**
  - **IoT device**
- Enter **Device ID** or choose **System generated device IDs** to generate random device ID.
- Choose 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 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, but the "Enter keys manually" option is selected in the "Authentication key" section of the modal. The "Primary Key" and "Secondary Key" fields now contain placeholder text: "SerenityKey" and "SerenityKey2" respectively. The "Apply" button is visible at the bottom right of the modal.

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

**3M Serenity IoT DEV Platform**

The screenshot shows the Device Explorer interface. On the left is a sidebar with icons for Dashboard, Device Explorer (selected), User Management, Rules, Packages, Deployments, and Maintenance. The main area is titled "Device Explorer" and contains a search bar and a table of devices. The table columns are: Device name, Simulated, Device type, Firmware, and Telemetry. Devices listed include "groupitest2", "JA-Device1", "joetest", "rule\_test", "sah\_test", "sah\_test\_device", and "sah\_test,test1". A modal window titled "New device" is open on the right, containing fields for Authentication type (Symmetric key selected), Authentication key (Auto generate keys selected), Primary Key, Secondary Key, and Provision summary (1 device to provision). An "Apply" button is highlighted with a blue box.

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

**3M Serenity IoT DEV Platform**

This screenshot is similar to the previous one, showing the Device Explorer interface and a "New device" modal. The modal now displays a success message: "1 Devices provisioned successfully ✓". It also shows the Device ID "Test", Primary Key, Secondary Key, Connection string primary Key, and Connection string secondary Key. The "Connection string primary Key" and "Connection string secondary Key" sections contain long, encoded strings. The "Close" button at the bottom right of the modal is highlighted with a blue box.

## Device Details

The Device Details panel displays information about the 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

- Grid will be available when alerts are observed for a device

The screenshot shows the Device Explorer interface. On the left, a sidebar lists navigation options: Dashboard, Device Explorer (selected), User Management, Rules, Packages, Deployments, and Maintenance. The main area is titled "Device Explorer" and contains a search bar ("Search devices..."). Below the search bar is a table with columns: Device name, Simulated, Device type, and Firmware. The table lists several devices: sah\_test\_device, sah\_test,test1, sahh\_test, ^Test, test-tel1, test-telemetry, and VisTest. To the right of the table is a "Device Details" pane for the device "test-telemetry". The pane includes a monitor icon, the device name, its physical status (Physical Disconnected), and a table of rules. The rule table has columns: Rule name, Severity, and Status. It lists six rules: Rule11, Rule3, Rule7, Rule8, Rule9, all marked as Critical and open. At the bottom of the pane are pagination controls: "1 to 5 of 5", "First", "Previous", "Page 1 of 1", "Next", and "Last".

## 2. Telemetry

### 3. Tags

### 4. Methods

This screenshot shows the same Device Explorer interface as the previous one, but the "Device Details" pane is expanded to show more information. The "Telemetry" section displays a line graph for the last week, showing data for humidity, temp, and temperature. The graph shows values fluctuating between 60 and 80. Below the graph, it says "Displaying in local timezone: Asia/Calcutta". The "Tags" section indicates "No tags found for this device". The "Methods" section also indicates "No methods found for this device".

## 5. Properties

- o Properties reported by the device.

## 6. Diagnostics

- o Diagnostic information from the device.

The screenshot shows the 'Device Details' pane open on the right side of the interface. It contains two main sections: 'Properties' and 'Diagnostics'.

**Properties:**

- To change a property on one or more devices, close this pane, select the checkbox for the device(s), click **Jobs**, and then select **Properties**.

PROPERTY	VALUE
weather.temperature	150 (Syncing [object Object])
weather.humidity	17 (Syncing [object Object])
...fields.temperature	Double (Syncing [object Object])
...a.fields.humidity	Double (Syncing [object Object])

**Diagnostics:**

- Diagnostic information from device

PROPERTY	VALUE
Status	Disconnected

At the bottom right of the pane, there is a 'Copy' button.

## 7. Deployment messages

- Edge module messages on the device.

## 8. Device Uploads

- File Uploads from Device.

## 9. Device Deployments

- Deployment History of device.

The screenshot shows the 'Device Details' pane open on the right side of the interface. It contains three main sections: 'Deployment messages', 'Device Uploads', and 'Device Deployments'.

**Deployment messages:**

- Edge module messages on the device

No messages found for this device

**Device Uploads:**

- File Uploads from Device

No Files are uploaded from Device.

**Device Deployments:**

- Deployment History of device

FIRMWARE VERSION	START DATE TIME	END DATE TIME
3.0.2	10:00:21 PM 09.16.2020	10:00:22 PM 09.16.2020
3.0.0	08:50:21 PM 09.16.2020	08:50:22 PM 09.16.2020

## 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 Check box left side of the device Name in the grid.

- After selecting devices the **Delete** button visible in the toolbar and click on the delete button.

The screenshot shows the 3M Serenity IoT DEV Platform interface. The left sidebar has tabs for Dashboard, Device Explorer (which is active), User Management, Rules, Packages, Deployments, and Maintenance. The main area is titled 'Device Explorer' and contains a search bar. Below it is a table with columns: Device name, Simulated, Device type, Firmware, Telemetry, Status, and Last c. Several devices are listed, including 'sah\_test\_device', 'sahh\_test', 'test-telemetry', and 'VisTest'. The device 'Test' is selected, indicated by a checked checkbox in its row. The 'Delete' button in the top right toolbar is also highlighted with a blue box.

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

This screenshot shows a confirmation dialog box titled 'Delete' over the Device Explorer grid. The dialog contains the text 'Delete device' and a message: 'This action will delete the devices permanently from the platform.' with a blue toggle switch. Below the message, there is a summary section with '1 Affected devices' and a red 'Delete' button, which is also highlighted with a blue box. The background grid shows the same list of devices as the previous screenshot, with 'Test' still selected.

- After deleting the device user sees a message **Device Deleted Successfully** and device details removed from the device explorer grid.

The screenshot shows the Device Explorer interface. 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 search bar labeled "Search devices...". Below it is a table with columns: Device name, Simulated, Device type, Firmware, Telemetry, and a checkbox header. Several devices are listed, including "sah\_test", "sah\_test\_device", "sah\_test,test1", "sahh\_test", "test-tel1", "test-telemetry", and "VisTest". A modal dialog titled "Delete" is open in the top right corner, asking if the user wants to "Delete device". It includes a note: "This action will delete the devices permanently from the platform." with two radio button options: "Yes, delete the selected physical devices." (selected) and "No, do not delete the selected physical devices.". A summary box at the bottom right says "1 devices deleted successfully! ✓".

## Configure a Device

In device explorer new device properties can also update, below are the steps:

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

The screenshot shows the Device Explorer interface with a similar sidebar and navigation as the previous screenshot. The main table now includes additional columns: Status and Last connection. A device named "Test" is selected, indicated by a blue border around its row and a checked checkbox in the first column. The "Jobs" button in the toolbar is highlighted with a blue box.

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

The screenshot shows the Device Explorer interface. 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 labeled "Search devices...". Below the search bar 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, where Properties 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". A table below shows two properties: ATest1 with value "Test" and ATest2 with value "Test1". 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 **Enter job name** text box, update the properties value, and click **Apply**.

This screenshot is identical to the one above, but it includes a blue rectangular highlight around the "Apply" button located at the bottom right of the "Jobs" panel. This highlights the step where the user should click the "Apply" button to save the changes made to the device properties.

- 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. Devices listed include 'sah\_test\_device', 'sah\_test,test1', 'sahh\_test', 'Test' (which is checked), 'test-tel1', 'test-telemetry', and 'VisTest'. A search bar at the top says 'Search devices...'. To the right, a modal window titled 'Jobs' shows properties for 'Test': ATest1 (Text) and ATest2 (Text). Below the table is a summary box stating '1 Job submitted successfully! ✓' with a 'View job status' button.

- After the job completes, navigate to device explorer and verify the updated properties details in the device details panel under the properties section.

**Note:**

Job status does not show live status, to refresh status click **refresh** button.

**3M Serenity IoT DEV Platform**

The screenshot shows the Device Explorer page with a completed job. The job details table has columns: Job Name, Status, Operation, No. of devices, Succeeded, Failed, Start time, and End time. One row shows 'Test-bd43e93a-cb12-4a12-baf1-e952d381df67' with a status of 'Completed'. Below it is another table with columns: Job Name, Status, Device ID affected, Last return message, Start time, and End time, also showing a completed entry for the same job.

## Organize devices

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

To display all your devices, navigate to the **Device Explorer** page and choose the All **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"/>	groupetest2	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 **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"/>	sahh_test,test1	No	---	---	---	Offline	---
<input checked="" 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...

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

**3M Serenity IoT DEV Platform**

The screenshot shows the Serenity IoT DEV Platform interface. On the left, there's a sidebar with navigation links: Dashboard, Device Explorer (which is currently selected and highlighted in dark grey), User Management, Rules, Packages, Deployments, and Maintenance. The main content area displays a table of devices with columns: Device name, Simulated, Device type, Firmware, and Telemetry. Several devices are listed, including 'joetest', 'rule\_test', 'sah\_test', 'sah\_test\_device', 'sahh\_test', 'Test', 'test-tef1', 'test-telemetry', and 'VisTest'. A modal window titled 'Jobs' is open on the right. Inside the modal, there's a section for 'Select job' with radio buttons for 'Tags' (selected), 'Methods', and 'Properties'. Below that, it says 'Tags on selected devices' and 'Tags in common on selected devices'. There's a 'Job Name' input field containing 'Test' with a note 'Only letters and numbers are allowed'. Underneath, there's a table for adding tags with columns 'KEY', 'VALUE', and 'TYPE'. A row is being added with 'TestKey' in the KEY column and 'TestValue' in the VALUE column. At the bottom of the modal, there's a 'Summary' section indicating '1 Affected devices' and a red 'Apply' button.

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

**3M Serenity IoT DEV Platform**

The screenshot shows the Serenity IoT DEV Platform interface. The sidebar and device list are identical to the previous screenshot. The 'Jobs' modal is open again, showing a summary of '1 Affected devices'. Below the summary, it says 'Job submitted successfully!' with a checkmark icon. At the bottom right of the modal, there's a red 'View job status' button and a 'Close' button.

- After the job completes, verify the tags in devices by using 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...

Maintenance

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

Now you can use the tag values to create filters. You can do it in 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 \*

+ Device group

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

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

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

## More

Device Explorer also provides few more features, below as follows

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

Show telemetry

**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 **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

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

Document covers the the following aspects of the User Management

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

## Users Grid

Users Grid lists all the active users, which are available for particular tenant to access.

Following lists the columns and usage in Users Grid

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   Delete 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

This is used to add new user for the tenant.

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

+ New user   + Add System Admin

Invite users

Email Address

\* Is required

User Role

Select user role

Send Invite   Cancel

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

The screenshot shows the User Management section of a software application. On the left is a sidebar with icons for Dashboard, Device Explorer, User Management (selected), Rules, Packages, Deployments, and Maintenance. The main area has a search bar and a table of users:

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

A modal window titled "Invite users" is open on the right, containing fields for "Email Address" (vkotha.cw@mmm.com) and "User Role" (Admin). A summary message at the bottom says "user successfully invited! ✓".

The screenshot shows the User Management section of a software application. The sidebar and table structure are identical to the previous screenshot. A specific user, "Viswanadh Kotha CW", is highlighted with a blue border in the table.

- User then accepting the invite through the email received, will be added as member to the tenant and will be able to access the application.

The screenshot shows the User Management section of a software application. The sidebar and table structure are identical to the previous screenshots. A specific user, "Viswanadh Kotha CW", is highlighted with a blue border in the table.

## Delete User

This is used to delete the user for the tenant.

- Navigate to **User Management** page from Left menu

- Select the user needs to be deleted.

- Click on **Delete** button

**User Management**

Search users...

Name	Role
testuser.selide@mmm.com	ReadOnly
timothy.gunter@analysts.com	Admin
Viswanadh Kotha CW	Admin
<input checked="" type="checkbox"/> vkotha.cw@mmm.com	Admin

**Delete**

Delete users  
This action will remove the selected user from this tenant.  
 No, do not remove selected users from this tenant.

**Summary**  
1 Affected users

**Delete** **Cancel**

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

**User Management**

Search users...

Name	Role
testuser.selide@mmm.com	ReadOnly
timothy.gunter@analysts.com	Admin
<input checked="" type="checkbox"/> vkotha.cw@mmm.com	Admin

**Delete**

Delete users  
This action will remove the selected user from this tenant.  
 Yes, please remove selected users from this tenant.

**Summary**  
1 Affected users

**Delete** **Cancel**

- After deleting the user we see a message **User Deleted Successfully**

**User Management**

Search users...

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

**Delete**

Delete users  
This action will remove the selected user from this tenant.  
 No, do not remove selected users from this tenant.

**Summary**  
1 Users deleted successfully! ✓

**Close**

# Add System Admin

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

System admin is the user who has access across all the tenants.

When we are adding user as system admin, then the selected user will be added as user for all the tenants.

- Navigate to **User Management** page from 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 with a sidebar containing 'Dashboard', 'Device Explorer', 'User Management' (selected), 'Rules', 'Packages', 'Deployments', and 'Maintenance'. At the top right are '+ New user' and '+ Add System Admin' buttons, with '+ Add System Admin' highlighted with a blue border. A modal window titled 'Add System Admin' is open on the right, showing a dropdown menu labeled 'Select user' with a list of users: '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). The 'Add' button at the bottom of the modal is greyed out.

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

The screenshot is identical to the previous one, but the 'Select user' dropdown now contains only one item: 'TestAccount - Admin'. The 'Add' button at the bottom of the modal is now highlighted with a red border, indicating it is enabled.

- On clicking on add button, 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 system admin from the system.

- Navigate to **User Management** page from 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 and a table listing users. A flyout window titled "Delete System Admin" is open, showing a list of users with checkboxes. The table includes an additional column for Type (Member or Invited).

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

- This is how flyout looks like.

The screenshot shows the User Management interface. On the left, a sidebar menu includes options like Dashboard, Device Explorer, User Management (which is selected), Rules, Packages, Deployments, and Maintenance. The main area displays a table of users with columns for Name and Role. 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. After selecting **Delete** button will be enabled.

This screenshot is similar to the previous one, but the "Delete" button in the "Delete System Admin" modal is now highlighted in red, indicating it is active or about to be clicked.

- On clicking on Delete button, User will be deleted as system admin.

The screenshot shows the User Management interface after a deletion. The "Delete System Admin" modal now displays a "Summary" section with the 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 user in the system using service principal Id.

- Navigate to **User Management** page from Left menu
- Click on **+Add Service principal**
  - A flyout opens to enter the service principal 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 links: Dashboard, Device Explorer, User Management (which is selected and highlighted in dark grey), Rules, Packages, Deployments, and Maintenance. The main area has a search bar ('Search users...') and a table with columns: Name, Role, and Type. The table contains eight rows of user data. A flyout window titled 'User Management' is overlaid on the right side of the screen. It has fields for 'Service principal app id' (containing placeholder text) and 'User Role' (a dropdown menu). At the bottom right of the flyout are 'Add' and 'Cancel' buttons, with the 'Add' button currently highlighted in blue.

This screenshot is similar to the one above, showing the 'User Management' page with a list of users. However, the 'User Management' flyout window is now fully open and centered. It includes a title 'Invite users', a 'Service principal app id' input field (containing placeholder text), a 'User Role' dropdown menu, and two buttons at the bottom right: 'Add' and 'Cancel'. The 'Add' button is highlighted with a blue border.

- Enter the Service Principal Id and Select the user role from the dropdown. After selecting **Add** button will be enabled.

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 displays a table of users with columns for Name and Role. A modal window titled "Invite users" is open, prompting for a "Service principal app id" (95d3c662-23ea-4e2d-8d3d-ea2448706934) and a "User Role" (Admin). A red "Add" button is visible 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

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

The screenshot shows the User Management page with the same layout and user list as the previous one. A summary message box is displayed, stating "Summary Added service principal ✓". A "Close" button is located at the bottom right of the message box.

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 user will be added with the type **Client Credentials**

The screenshot shows the User Management page with the sidebar and user list. A specific row in the table is highlighted with a blue border, showing the user "95d3c662-23ea-4e2d-8d3d-ea2448706934" with a role of "Admin" and a type of "Client Credentials". The table also includes columns for "Type" and "Status". Other users listed include aditya.tangirala@ggktech.com (Invited), Anant Choudhari CW, Andrew Schmidt, Anup Warade CW, atangirala.cw@mmm.com, and Blake Ma, all with Admin roles and Member status.

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

Rules are used to manage all the active rules. Rule is to ensure to receive a warning when it satisfies the condition.

Document covers the the following aspects of the User Management

1. [Rules Grid](#)
2. [Rule Details Flyout](#)
3. [Edit Rule Flyout](#)
4. [New Rule Flyout](#)
5. [Delete Rule Flyout](#)
6. [Rule Status Flyout](#)

## Rules Grid

- Navigate to **Rules** page from Left menu
- Rules Grid lists all the active rules, configured for device group.

Following lists the columns and usage in Rules 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
Device Group	Indicates the device group, the rule belongs to
Trigger	Describes for which kind of trigger this rule belongs to
Notification Type	Describes the Notification type when the rule triggers
Status	Determines the status of the rule whether its enabled or disabled
Last Trigger	Describes the rule's last trigger date

The screenshot shows the 'Rules' page of the 3M Serenity IoT DEV Platform. On the left is a navigation sidebar with links: Dashboard, Device Explorer, User Management, Rules (which is selected and highlighted in dark grey), Packages, Deployments, and Maintenance. The main content area has a header with 'Default' and dropdowns for 'Get Link' and 'Manage device groups'. A search bar at the top says 'Search rules...'. Below it is a table with columns: Rule name, Description, Severity, Device group, Trigger, Notification type, Status, and Last trigger. The table contains 15 rows, each representing a rule named 'New rule', 'Rule1', 'Rule11', 'Rule12', 'Rule13', 'Rule14', or 'Rule15'. The first row ('New rule') has a description of 'Sample rule to test', severity 'Info', device group 'joetest', trigger 'temperature', notification type 'Maintenance log', status 'Enabled', and last trigger '---'. The last row ('Rule15') has the same details. At the bottom right of the table are pagination controls: '1 to 17 of 17', 'First', 'Previous', 'Page 1 of 1', 'Next', and 'Last'.

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

This screenshot shows the same 'Rules' page as above, but with a search term 'New' entered into the search bar. The search bar is highlighted with a blue border. The rest of the interface is identical to the first screenshot, including the sidebar, table structure, and pagination controls.

## Rule Details

- Navigate to **Rules** page from Left menu
- Click on the rule name
- Rule details flyout is used to read the details of the selected rule.

The screenshot shows the Serenity IoT DEV Platform interface. On the left, there's a navigation menu with options like Dashboard, Device Explorer, User Management, Rules (which is selected), Packages, Deployments, and Maintenance. The main area displays a list of rules with columns for Rule name, Description, Severity, Device group, Trigger, Notification type, and Status. A search bar at the top allows searching for rules. To the right of the list is a detailed view for Rule20, which includes sections for Rule Details (Rule20, Rule Description, Device group set to Default, Calculation set to Average, Time period set to 1 minute), Conditions (a table with FIELD, OPERATOR, and VALUE columns showing 'temperature = Equals 1'), Severity level (set to Critical), and Rule status (Enabled). At the bottom right, it says '13 devices affected by this rule'.

## Edit Rule details

- Navigate to **Rules** page from Left menu
- Click on the rule name
- This flyout is used to edit the created rule. This option is available in Rule details flyout.

This screenshot is identical to the one above, showing the Serenity IoT DEV Platform interface. The 'Edit' icon in the Rule Details flyout for Rule20 is highlighted with a blue box. The rest of the interface, including the navigation menu, rule list, and detailed rule view, remains the same.

- On clicking on edit icon, user can able to edit the details of rule.

The screenshot shows the Serenity IoT DEV Platform interface. On the left, there's a navigation sidebar with options like Dashboard, Device Explorer, User Management, Rules (which is selected and highlighted in dark grey), Packages, Deployments, and Maintenance. The main content area displays a table of rules. One row, 'Rule20', is selected and expanded, showing its details: Rule name is 'Rule20', Description is 'Rule Description', Severity is 'Critical', Device group is 'Default', Trigger is 'temperature', and Notification type is 'Maintenance log'. To the right of the table, an 'Edit rule' modal is open for 'Rule20'. It contains fields for Rule name (set to 'Rule20'), Description (empty), and a large text area for Rule Description (empty). Below these are dropdowns for Device group ('Default'), Calculation ('Average'), and Time period (set to '1'). A 'Conditions' section is present with a 'Field' dropdown set to 'temperature', an 'Operator' dropdown set to '= Equals', and a 'Value' input field set to '1'. A '+ Add condition' button is available. At the bottom of the modal, there are 'Apply' and 'Cancel' buttons.

- Update the data accordingly and clicking on **Apply** button, saves the rule data.

This screenshot shows the same interface after changes have been made. The 'Edit rule' modal now has 'Rule name' set to 'Rule20', 'Description' set to 'Rule Description', and 'Severity' set to 'Critical'. The 'Value' field in the conditions section is now set to '1'. The 'Severity level' dropdown shows 'Critical' selected. The 'Action' section has 'Email disabled' checked. The 'Rule status' is set to 'Enabled'. At the bottom, the modal displays '13 devices affected by this rule' and contains 'Apply' and 'Cancel' buttons.

## New Rule

This is used to create new rule associating to the device group.

- Navigate to **Rules** page from Left menu
- Click on **+New Rule**
  - Enter your **Rule name**.
  - Enter your **Rule Description**.
  - Select **Device group** to be associated.
  - Select the **Calculation Type**.
  - If calculation Type is Average then select **Time Period**.
  - Add a Condition for the rule.
    - Select **Field** to be added.
    - Select the **Operator** to be compared.
    - Enter a **value** to be compared.

- Select **Severity Level** you wanted to assign.

- Select **Rule Status**.

The screenshot shows the Serenity IoT DEV Platform interface. On the left, there's a sidebar with 'Dashboard', 'Device Explorer', 'User Management', 'Rules' (which is selected), 'Packages', 'Deployments', and 'Maintenance'. The main area shows a list of rules with columns: Rule name, Description, Severity, Device group, Trigger, Notification type, and Status. There are 13 rows of rules, mostly named 'Rule1' through 'Rule17' with descriptions like 'TestDesc' and severity 'Critical'. To the right of this list is a 'New rule' dialog box. It contains fields for 'Rule name' (set to 'Rule20'), 'Description' (empty), 'Severity' (set to 'Critical'), 'Device group' (set to 'Default'), 'Calculation' (set to 'Average'), 'Time period (in minutes)' (set to '1'), 'Condition 1' (empty), 'Conditions' (empty), 'Severity level' (set to 'Critical'), 'Action' (set to 'Email disabled'), and 'Rule status' (set to 'Enabled').

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

This screenshot is similar to the previous one but shows the 'Apply' button at the bottom right of the dialog box. The dialog box contains the same fields as before, including the '13 devices affected by this rule' message. The 'Apply' and 'Cancel' buttons are visible at the bottom right of the dialog.

- After saving the rule, rule will be created sucessfully.

## Delete Rule

This is used to delete the rule.

- Navigate to **Rules** page from Left menu
- Select the rule needs to be deleted.
- Click on **Delete** button

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

The screenshot shows the Serenity IoT DEV Platform's Rules page. On the left, there's a navigation sidebar with links for Dashboard, Device Explorer, User Management, Rules (which is selected), Packages, Deployments, and Maintenance. The main area displays a table of rules with columns: Rule name, Description, Severity, Device group, Trigger, Notification type, and Status. One row, Rule20, is selected. A context menu is open over this row, containing 'Delete Rule' (highlighted in red), 'Rule Description', and a link indicating '13 devices affected by this rule'. There are also 'Delete' and 'Cancel' buttons at the bottom of the menu.

- After deleting the rule we see a message **Rule has been deleted**

This screenshot shows the same Rules page after a rule has been deleted. The context menu from the previous screenshot is still open, but now it includes a message box with the text 'Rule has been deleted.' and a checked checkbox. Below the message, there's a note: 'If you want to delete the alerts associated with this rule, go to the Maintenance Page and choose the alerts you want to remove.' There's also a 'Close' button at the bottom right of the message box.

## Rule Status

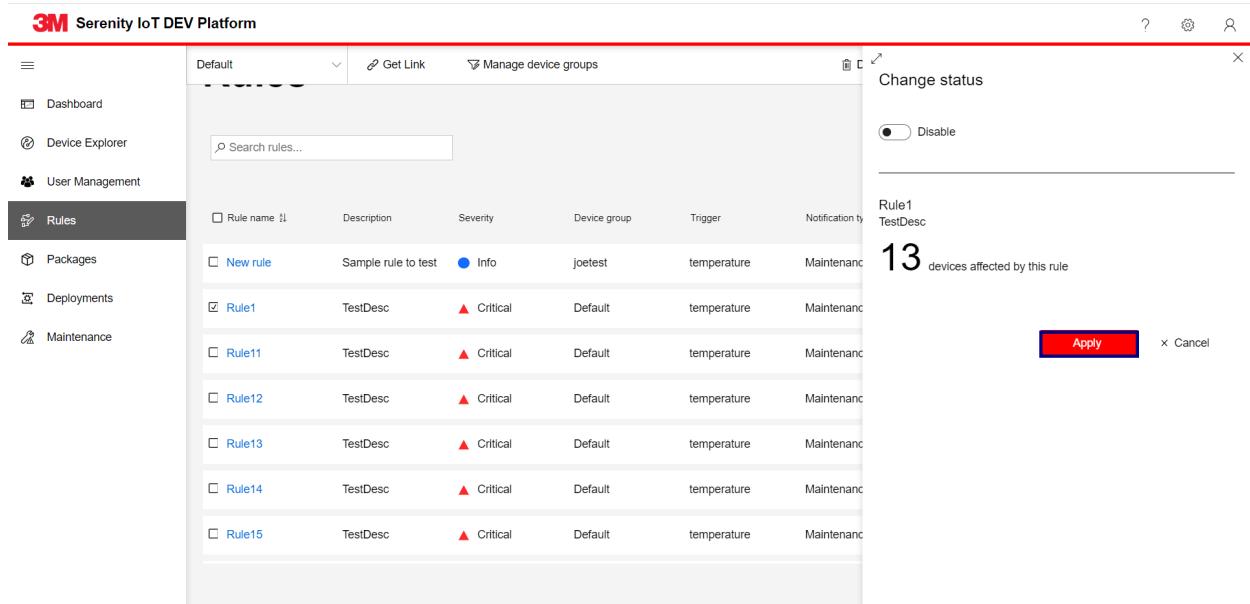
This feature is used to enable/disable the rule status.

- Navigate to **Rules** page from Left menu
- Click on **Status toggle button**
  - A flyout opens to change the status of the **Rule Status**.

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

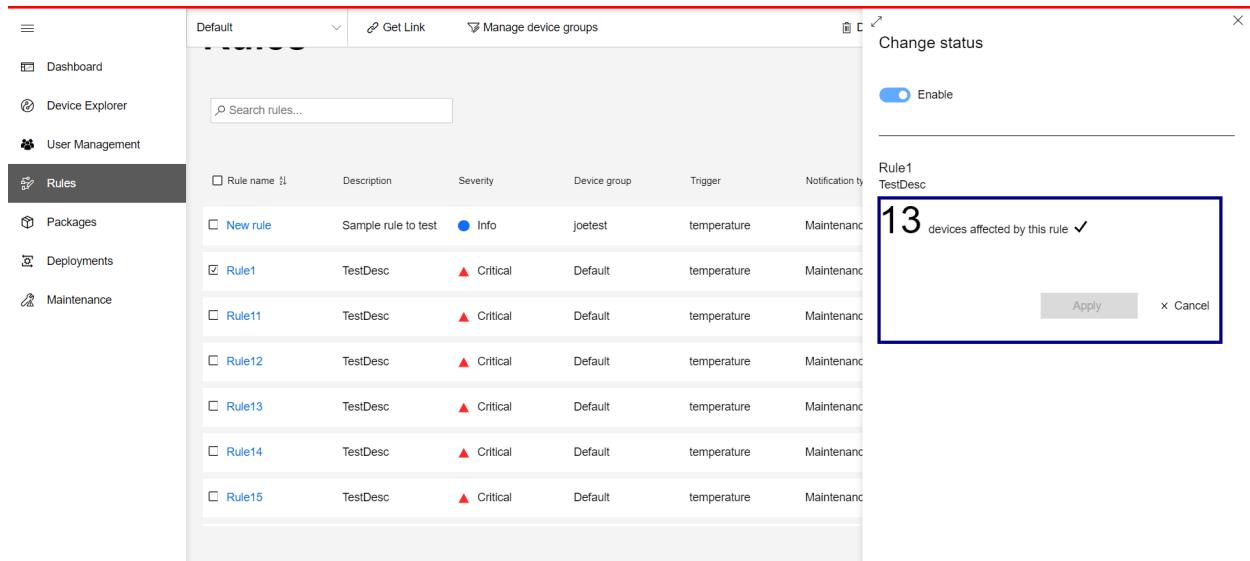
1 to 17 of 17 First Previous Page 1 of 1 Next Last

- Changing the status toggle button **Apply** button gets enabled



The screenshot shows the 'Change status' modal dialog. It contains a single toggle button labeled 'Disable'. The 'Disable' button is currently selected (indicated by a blue border), which has triggered the enablement of the 'Apply' button.

- On clicking on **Apply** button, Status of the rule gets updated accordingly.



The screenshot shows the 'Change status' modal dialog again. This time, the 'Enable' toggle button is selected (blue border). The message box in the dialog now displays '13 devices affected by this rule ✓', indicating that the rule has been successfully updated.

# Packages

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

Document covers the following aspects of the Packages

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

## Packages Grid

Packages Grid lists all the packages which are either active or deactivated, which are available for Deployments.

Following lists the columns and usage in Package Grid

COLUMN NAME	DESCRIPTION
Name	Name of the Package
Package Type	Describes the Device category the Package is targetting
Configuration Type	Indicates whether the package is targetting firmware updates or custom updates.
Date Created	Date the package is created
Active	Indicates whether the package is Active/Deactive
Version	Indicates version of the Package for differentiation purposes.
Last Modified Date	Date the package is created or date the package is modified lastly
Last Modified By	User who created the package or the user who modified package lastly.

## Package Creation

Package creation for Device Firmware update

- Navigate to **Packages** page from Left menu
- Click on **+New Package**
  - Select **Package Type** based on the targetted device types
    1. **Device Configuration** for IoT Devices
    2. **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), 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, prompting the user to "Upload a package" and "Add a package to your solution". It includes a dropdown for "Package Type" with "Edge Manifest" and "Device Configuration" options, and a file input field for "devicegroup.\*" with a browse button. The URL "devicegroup.cd9bba84-274a-4d69-8e49-81f4933d3a36" is shown in the input field.

■ **Package Type - Edge Manifest** For Edge Devices, User should upload the Edge Device Package content file for Package creation.

■ After selecting **Package Type**, User should upload the Edge Devices Configuration file by clicking **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.

The screenshot shows the Serenity IoT DEV Platform interface. On the left, there is a sidebar with navigation links: Dashboard, Device Explorer, User Management, Rules, Packages (which is currently selected), Deployments, and Maintenance. The main area has two windows open. One window is titled "New Package" and contains fields for "Upload a package", "Add a package to your solution", "Package Type" (set to "Edge Manifest"), and a "Browse" button for selecting a package file. It also lists "Tags" such as "devicegroup.\*" and "devicegroup.cd9bba84-274a-4d69-8e49-81f4933d3a36". The other window shows a list of packages in the "Packages" folder. The list includes "CustomSWPackage", "EdgeSWPackage", and "SW1Package". Below this list, there is a file selection dialog with the title "Open" and a file name "EdgeSWPackage". The file list in the dialog shows "Package1234", "sample", "SW1Package.json", and "SW3Package.json".

- 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 Package is created, user will be presented with created package information and will be reflected in

## 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

Packages Grid.

### ■ Package Type - Device Configuration

#### ■ Select the Configuration type

1. **Firmware**

2. **Custom**

**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...	✓

- **Configuration Type - Firmware** - Here user will be able to build package content based on the Firmware package file which can be uploaded.

After selection of Configuration Type, user should **upload the package content file** which consists of the desired updates to the devices.

**Packages**

Name	Type	Date modified
CustomSW2Package	JSON File	21-05-2020 10:21
EdgeSWpackage	JSON File	21-05-2020 10:12
EdgeSWpackage1	JSON File	08-09-2020 11:14
EdgeSWpackage2	JSON File	09-09-2020 12:34
PackagesTest_New	File	04-06-2020 15:16
PackagesTest_Old	File	04-06-2020 15:16
SW1Package	JSON File	30-04-2020 11:44
SW2Package	JSON File	30-04-2020 11:44
SW3Package - Copy - Copy	JSON File	30-04-2020 11:44
SW3Package	JSON File	30-04-2020 11:44
SW4Package - Copy	JSON File	30-04-2020 11:44

- Once the file is selected, Provide Name and Version for Package.

A screenshot of a 'New Package' dialog box. At the top, there's a back arrow icon and the title 'New Package'. Below the title, a dropdown menu shows 'Device Configuration' with a downward arrow. Under 'Configuration type \*', another dropdown menu shows 'Firmware' with a downward arrow. A 'Browse' button is followed by the placeholder text 'for a firmware file'. 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. The 'Tags' section lists 'devicegroup.\*' and 'devicegroup.cd9bba84-274a-4d69-8e49-81f4933d3a36', each with a small trash can icon to its right. At the bottom, the word 'Package' is followed by 'SW1Package.json'. On the far right, there are two buttons: a red 'Upload' button with a white arrow icon and a white 'Cancel' button with a black 'X' icon.

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 on Upload, once uploading of the file is done, User will be provided with the preview of the Package File contents, which can modified as per the need.



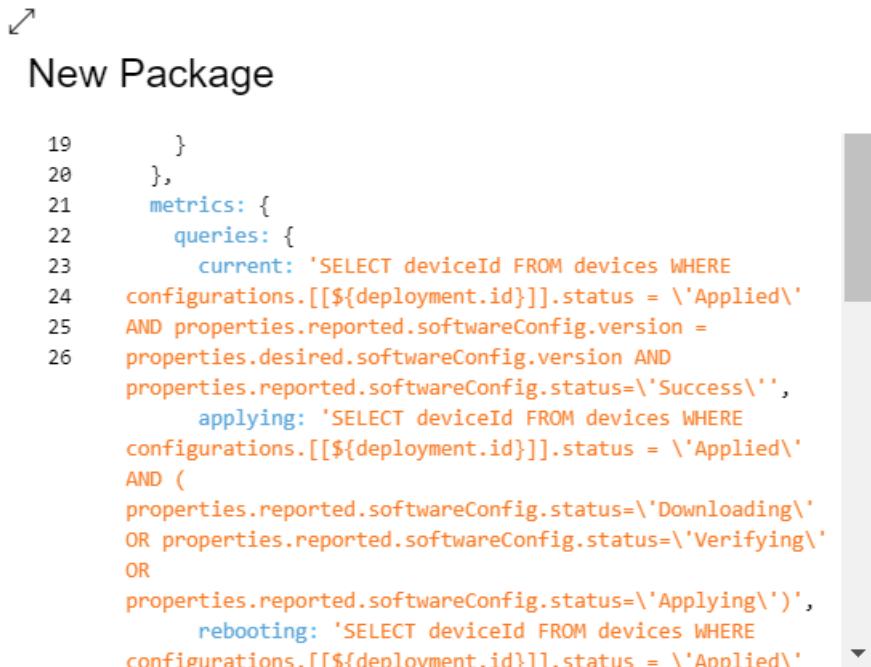
## New Package

### 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 the JSON is verified, Click on Upload for package creation.



## Tags

devicegroup.\* 

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

# 1 Package

## SW1Package.json



- Once Package is created, user can find the package in the package 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** Here user should provide the package content file for package creation.
    - After selection of Configuration Type, user should provide name for **Custom Configuration** and **Configuration File** which should contain details which are required for the deployment process.



- Once the file is selected, Provide **Package Name**, which will be prepopulated with **FileName** selected.



## 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 for package creation.

↗

## 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 [Delete]

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

---

1 Package

CustomSWPackage.json

↑ Upload

× Cancel

- Once Package is created, user will be presented with created package information and will be reflected in 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

Available Package which are Active, can be deactivated, so that they are not available for Deployment process.

Steps to Deactivate a Package.

- Select a Active Package(which is marked ✓ mark in the Active Column) by clicking on the check box adjacent to Package Name in 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

- User Will be presented with two options in the toolbar

- Deactivate
- 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
- Package is deactivated and is presented in the grid with X mark 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 process.

Steps to Activate a Package.

- Select a Deactivated Package(which is marked by X mark in Active Column) by clicking on the Check box adjacent to Package Name in 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

- User Will be presented with two options in the toolbar

- Activate
- Delete

The screenshot shows the 'Packages' page in the Serenity IoT DEV Platform. On the left, there's a sidebar with icons for Dashboard, Device Explorer, User Management, Rules, Deployments, Maintenance, and Packages (which is selected). At the top right, there are buttons for 'Activate' (highlighted in blue), 'Delete', '+ New Package', and a search icon. The main area has a table with columns: Name, Package Type, Configuration T..., Date Created, Active, Version, Last Modified D..., and Last Modified By. There are seven rows in the table, each with a checkbox in the first column. The first row, 'TestSWPackage', has its 'Active' column checked with a green checkmark. The last row, 'sample', also has its 'Active' column checked with a green checkmark. At the bottom of the table, there are links for '1 to 10 of 10', 'First', 'Previous', 'Page 1 of 1', 'Next', and 'Last'.

	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	11:26:55 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...	Jayashima Nallag...

- Click on Activate
- Package is Activated and is presented in the grid with ✓ mark in the Active Column

This screenshot is identical to the one above, showing the 'Packages' page. The difference is that the first row, 'TestSWPackage', is now highlighted with a blue selection bar around its entire row. All other rows remain unselected.

## Delete Package

Packages which are no longer needed or those whose Package content is expired can be deleted.

Steps to Delete a Package.

- Select a Package by clicking on the Check box adjacent to Package Name in 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

- User Will be presented with Delete in the toolbar

Default Get Link Manage device groups Deactivate  + 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 Delete

- User will be provided with Confirmation popup for Delete.

**3M Serenity IoT DEV Platform**

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			
TestPackage	Device Configuration	02:04:30 PM 10.1...	✓	1.0.0			
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 on Delete/Cancel to Delete or Cancel the operation.
- On Click of Delete Package is Deleted and is Grid will be refreshed with Active and Deactivated packages.

**3M Serenity IoT DEV Platform**

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.

Document covers the following aspects of the Deployments

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

## Deployments Grid

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

Following lists the columns and usage in 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 targetted by Deployment.
Priority	Priority specifies which deployment will impact the devices, Higher priority deployments take precedence over lower priority.
Configuration Type	Indicates whether the Configuration is updating Device Firmware or any other custom properties. Empty for Edge Deployments
Targeted	No of Devices configuration is targetted for.
Applied	No 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	No of Devices where configuration update is successful as reported by devices.
Failed	No of Devices where configuration update is failed as reported by devices.
Date Created	Date the package is created
Created On	Date the Deployment is created
Created By	User who created the Deployment.
Modified On	Date the Deployment is modified lastly

COLUMN NAME	DESCRIPTION
Modified By	User who modified Deployment lastly.

## Deployment Creation

- Navigate to Deployments Page and Click on + New Deployment

The screenshot shows the 3M Device Management interface. The left sidebar has a navigation menu with items like Dashboard, Device Explorer, User Management, Rules, Packages, Deployments (which is highlighted), and Maintenance. The main content area is titled "Deployments" and displays a table with columns: Name, Parent, Description, Priority, Status, Configuration, Target, Application, Success, Failure, Create, Modify, and Delete. A date header "0/13/87" is shown above the table. At the bottom of the table, there are navigation 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 targetted 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 Priority for Deployment, Higher the Priority 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.

- Post Successful creation of Deployment, User 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

- Deployment Details screen provides data about the Metrics of Deployment such as No of Devices Targetted, Applied, Succeded, Pending and Failed.

**3M 3M** ? ⚙️ 🗑️

☰

Deployments

DEPLOYMENT NAME

## DeploymentSimulation

1 Applied	0 Failed
1 Targeted	0 Succeeded
	1 Pending

PRIORITY  
4

CONFIGURATION TYPE  
Firmware

DEPLOYMENT STATUS

DEVICE GROUP  
SimulationGroup

PACKAGE TYPE  
Device Configuration

START  
02:19:03 PM 10.16.2020

PACKAGE  
Package1 (1.0.0)

Devices Affected

▶ Expand Columns

Name	Deployment Status	Firmware	Start	End
SimulatedDevice	Pending	--	--	--

Following table depicts how the Metrics are defined.

Metric	Description
Targetted	No of Devices which will be impacted by Deployment
Applied	No of Devices which are impacted by Deployment
Failed	No of Devices which reported failure of Firmware update.
Succeded	No of Devices which reported Success of Firmware update to the Current Version.
Pending	No of Devices which are still to update the Firmware.

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

The screenshot shows the 3M Device Management interface. The left sidebar has a 'Deployments' tab selected. The main area displays a deployment named 'DeploymentSimulation' with the following status: 1 Applied, 0 Failed, 1 Targeted, and 1 Pending. A summary table provides details about the device group and package type. A box highlights the deployment configuration settings (Priority 4, Configuration Type Firmware). Below this, a table lists devices affected by the deployment.

DEPLOYMENT NAME

## DeploymentSimulation

Applied	Failed
1	0

Succeeded	Pending
0	1

DEVICE GROUP	PACKAGE TYPE
SimulationGroup	Device Configuration

START	PACKAGE
02:19:03 PM 10.16.2020	Package1 (1.0.0)

PRIORITY
4

CONFIGURATION TYPE
Firmware

### Devices Affected

Name	Deployment Status	Firmware	Start	End
SimulatedDevice	Pending	--	--	--

PROPERTY	DESCRIPTION
Device Group	Provides the information about the List of 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.
Configuration Type	Indicates whether the Configuration is updating Device Firmware or any other custom properties. 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 with the following details:

- DEPLOYMENT NAME:** DeploymentSimulation
- PRIORITY:** 4
- CONFIGURATION TYPE:** Firmware
- Devices Affected:**

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

- Following depicts the various stages how the Deployment is affecting the devices.

- Device Status as Pending when the Firmware update is pending from the Device.

The screenshot shows the 'Deployment Simulation' page with the following details:

- DEPLOYMENT NAME:** DeploymentSimulation
- PRIORITY:** 4
- CONFIGURATION TYPE:** Firmware
- Devices Affected:**

Name	Deployment Status	Firmware	Start	End
SimulatedDevice	Pending	---	---	---

- Device Status as Completed, when the Device successfully updated the 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 has a header 'Deployments' and a sub-header 'DEPLOYMENT NAME: DeploymentSimulation'. It displays two rows of deployment summary: one for 'Applied' (0 Failed) and one for 'Targeted' (1 Succeeded, 0 Pending). Below this, sections for 'PRIORITY' (4) and 'CONFIGURATION TYPE' (Firmware) are shown. A large section titled 'Devices Affected' contains a grid with columns: Name, Deployment Status, Firmware, Start, and End. One row in the grid is for 'SimulatedDevice' with status 'Succeeded', firmware '1.0.0', start '10:00:21 PM 10.16.2020', and end '10:00:23 PM 10.16.2020'. A 'Devices Affected' button is located at the bottom right of this section.

- Deployment Details screen also lists the Devices which are 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: Name, Deployment Status, Firmware, Start, and End. The single entry for 'SimulatedDevice' is visible with its details: Succeeded, 1.0.0, 10:00:21 PM 10.16.2020, and 10:00:23 PM 10.16.2020.

Grid displays the following columns

Column	Description
---	---
Name	Name of the Device.
Deployment Status	Status depicts the status of Firmware update by Device.
Firmware	Firmware Version.
Start	Date and time which depicts the start time when Device starts Configuration update.
End	Date and time which depicts the End time when Device completes Configuration update.

- Deployment Details screen also provides option to Download the Devices Affected Grid Details to an Excel.

☰

↳ 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 Status Toggle User will be presented with confirmation popup.

**3M 3M**

☰ Dashboard Device Explorer User Management Rules Packages Deployments Maintenance

DEPLOYMENT NAME DeploymentSimulation

1 Applied 0 Failed 1 current -- DEVICE GROUP SimulationGroup PACKAGE TYPE Device Configuration

1 Targeted 1 Succeeded 0 Pending TART 2:30:26 PM 10.16.2020 PACKAGE Package1 (1.0.0)

PRIORITY 4 CONFIGURATION TYPE Firmware

Devices Affected

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

- Deployment Details screen also provide option to Delete the Deployment.

**3M 3M**

☰ Dashboard Device Explorer User Management Rules Packages Deployments Maintenance

DEPLOYMENT NAME DeploymentSimulation

1 Applied 0 Failed -- DEVICE GROUP SimulationGroup PACKAGE TYPE Device Configuration

1 Targeted 1 Succeeded 0 Pending -- START 02:19:03 PM 10.16.2020 PACKAGE Package1 (1.0.0)

PRIORITY 4 CONFIGURATION TYPE Firmware

Devices Affected

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
-----------------	-----------	-------	------------------------	------------------------

- On Delete button click, user will be presented with popup for confirmation.

The screenshot shows the 3M IoT Hub interface. On the left, a sidebar menu includes options like Dashboard, Device Explorer, User Management, Rules, Packages, Deployments (which is selected), and Maintenance. The main content area is titled 'Deployment Simulation'. It displays two rows of deployment status: 'Applied' (1 item) and 'Targeted' (1 item). The 'Targeted' row has a 'Delete Deployment?' confirmation dialog open over it. The dialog contains a message about stopping the deployment if deleted, and two buttons: 'Delete' (highlighted in red) and 'Cancel'. Below the deployment grid, there's a section for 'Devices Affected' with a table showing one device: 'SimulatedDevice' with status 'Succeeded'. A note at the bottom right of the table says '> Expand Columns'.

## Deactivate Deployment

User can deactivate a Deployment to free up the Configurations in IoT Hub and still have a reference of the deployment for future.

Note:

IoT Hub currently have 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)

- User will be presented with Deployment Status Flyout, representing the current 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.

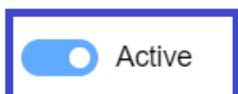
Apply Close

- Toggle the Button to change the Status of Deployment



Your environment is currently limited to 100 active deployments.

### DeploymentSimulation



### Related Deployments

No Related deployments exist for the selected deployment.

Apply Close

- Click on 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

- On Successful status change, User can verify the status of Deployment will be Inactive

The screenshot shows the 3M IoT Hub interface. On the left, there's a sidebar with navigation links: Dashboard, Device Explorer, User Management, Rules, Packages, **Deployments** (which is selected), and Maintenance. The main area is titled "Deployments" and shows a single deployment entry. The deployment details are as follows:

	Name	Package	Deployment	Priority	Platform	Configurations	Tariff	Applies	Status	Fails	Created	Crash	Mo.	Mo.
<input type="checkbox"/>	● Deploy...	Packa...	Simula...	4	Device...	Firmware	1	1	1	---	02:19...	Ragav...	04:44...	Ragav...

At the bottom right of the main area, there are buttons for "1 to 1 of 1", "First", "Previous", "Page 1 of 1", "Next", and "Last".

## Reactivate Deployment

User can Reactivate a Deployment based on the need, so that Devices can have the configuration.

Note:

IoT Hub currently have limit of 100 configurations.

Steps to Reactivate a Deployment.

- Click on the Circle adjacent to any Inactive Deployment

**Reference:**

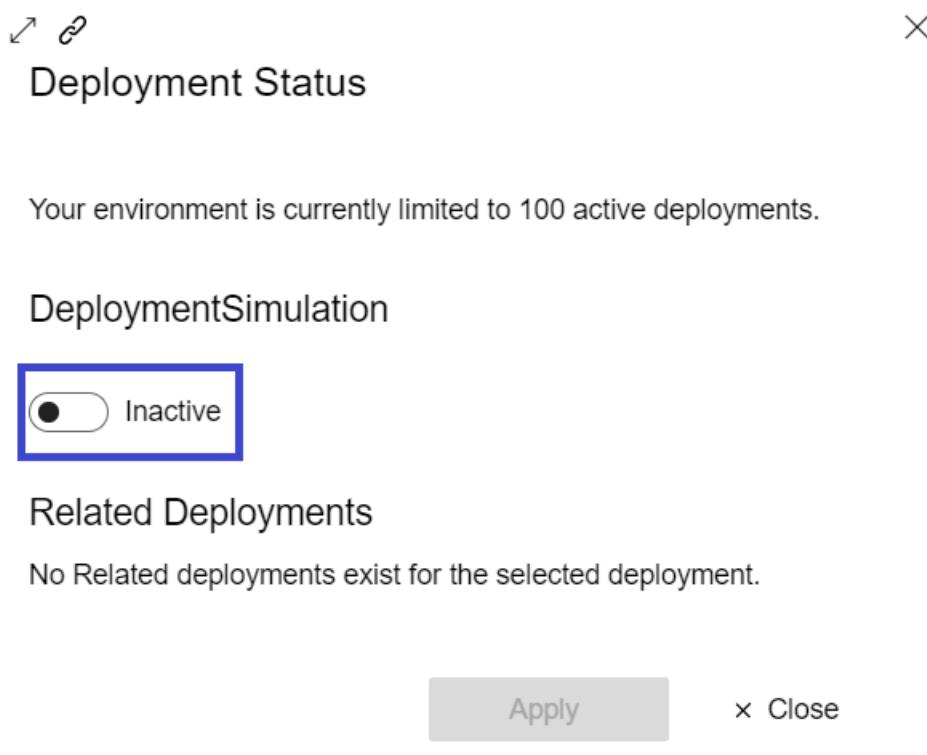
CIRCLE COLOR	STATUS OF DEPLOYMENT
Green	Active
Black	Inactive

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

- User will be presented with Deployment Status Flyout, representing the current status of the Deployment.



- Toggle the Button to change the Status of Deployment



- Click on Apply.



## Deployment Status

Your environment is currently limited to 100 active deployments.

### Deployment Simulation



### Related Deployments

No Related deployments exist for the selected deployment.



- On Successful status change, User can verify the status of Deployment will be Inactive

The screenshot shows a software interface titled "3M 3M". The left sidebar has a navigation menu with items like Dashboard, Device Explorer, User Management, Rules, Packages, **Deployments** (which is selected and highlighted in dark grey), and Maintenance. The main content area is titled "Deployments" and shows a table of deployment data. The table has columns for Selection (checkbox), Name (Deploy...), Package (Packa...), Simulation (Simula...), Priority (4), Device (Device...), Firmware (Firmware), Target (1), Success (1), Failure (---), Creation (02:19...), Created By (Ragav...), Modified (---), and Modified By (---). A single row is selected, indicated by a green dot next to the name. The top right of the main area shows the date "1/14/86". The bottom right of the main area shows pagination controls: "1 to 1 of 1", "First", "Previous", "Page 1 of 1", "Next", and "Last".

## Delete Deployment

User can delete any Deployment when they are considered not needed.

Steps to Delete a Deployment.

- Select any Deployment by Clicking on the Selection CheckBox.

**3M**

The screenshot shows a software interface for managing deployments. On the left, a sidebar menu includes options like Dashboard, Device Explorer, User Management, Rules, Packages, and Deployments (which is selected and highlighted). The main area is titled "Deployments" and shows a table with one row of data. The columns represent various deployment metrics such as Name, Package, Deployment ID, Priority, Status, Target, Application, Success, Failure, Create Time, and Modify Time. The single entry in the table is highlighted with a blue border.

	Name	Pa...	De...	Pri...	Sta...	Co...	Tar...	Ap...	Suc...	Fai...	Cr...	Cr...	Mo...	Mo...
<input type="checkbox"/>	Deploy...	Packa...	Simula...	4	Device...	Firmware	1	1	1	---	02:19:...	Ragav...	04:44:...	Ragav...

0/13/87

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

- Once User selects a Deployment, Delete option will be available in the ToolBar

**3M**

This screenshot is similar to the previous one but includes a toolbar at the top right with a "Delete" button, which is highlighted with a blue border. The deployment table and its single entry remain the same.

	Name	Pa...	De...	Pri...	Sta...	Co...	Target	App...	Success	Failure	Create	Modify	Mo...	Mo...
<input checked="" type="checkbox"/>	Deploy...	Packa...	Simula...	4	Device...	Firmware	1	1	1	---	02:19:...	Ragav...	04:44:...	Ragav...

0/13/87

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

- On Click of the Delete Button, Delete Confirmation popup will be presented for final confirmation.

**3M 3M**

SimulationGroup

## Deployments

0/13/87

> Expand Columns

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

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

Deploy...

Delete Deployment?

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.

This screenshot shows the 3M Deployment Management interface. On the left, there's a sidebar with various navigation options like Dashboard, Device Explorer, User Management, Rules, Packages, Deployments (which is currently selected), and Maintenance. The main area is titled 'Deployments' and shows a single deployment entry. A modal dialog box is open over the grid, asking for confirmation to delete the deployment. The dialog contains a message about stopping deployment application and includes 'Delete' and 'Cancel' buttons. The deployment grid itself has columns for Success, Failure, Creation, Completion, and more, with one row of data visible.

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

**3M 3M**

SimulationGroup

## Deployments

0/13/87

> Expand Columns

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

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

Deploy...

Delete Deployment?

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.

This screenshot is identical to the one above it, showing the same deployment management interface and the 'Delete Deployment?' dialog box. The only difference is the absence of the list item 'On Successful Deletion, User will not be able to find the Deployment anylonger in the Deployment Grid.' from the list below.

- On Successful Deletion, User will not be able to find the Deployment anylonger 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	Applies To	Status	Failure Count	Critical	Modified By	Modified On
0 to 0 of 0	First	Previous	Page 0 of 0	Next	Last								

# Maintenance

This page allows users to view alerts from devices and different jobs that are running against devices. Below is the information a user can find in the maintenance page:

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

## Summary

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

**Navigation:** Click Maintenance from the left menu

### Alert counts

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

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

The screenshot shows the Azure IoT Platform Maintenance dashboard. On the left is a navigation sidebar with icons for Dashboard, Device Explorer, User Management, Rules, Packages, Deployments, and Maintenance (which is highlighted with a blue bar). The main area has a title 'Maintenance' and a summary card with a blue border. Inside the card, it says '16 Open Alerts' with sub-counts '6 ▲ Critical' and '10 ■ Warning'. To the right, it shows '0 Failed jobs' and '0 Total 0 Succeeded'. Below this is a section titled 'Alerts' and 'Jobs' with a red underline under 'Alerts'. At the bottom is a table header with columns: Rule name, Description, Severity, and Total count.

Rule name	Description	Severity	Total count
-----------	-------------	----------	-------------

### Job counts

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

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

The screenshot shows the Maintenance page with a sidebar on the left containing links: Dashboard, Device Explorer, User Management, Rules, Packages, Deployments, and Maintenance (which is highlighted). 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 are tabs for 'Alerts' and 'Jobs', with 'Alerts' being the active tab. A grid table follows, with columns: Job Name, Status, Operation, and No. of devices.

Job Name	Status	Operation	No. of devices

## Alerts Grid

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

The screenshot shows the Alerts grid with a title 'Maintenance'. It displays 16 Open Alerts, 6 Critical (red triangle), and 10 Warning (yellow square). The 'Alerts' tab is selected. The grid table has columns: Rule name, Description, Severity, Total count, Open, Acknowledged, Closed, and Last occurrence. Three rows of data are shown:

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....

A user can view the alerts grid by following below steps:

1. Click on Maintenance from the left menu to navigate to Maintenance page
2. Click on Alerts tab

Following lists the columns in 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 count of total alerts

Column Name	Description
Open Count	Indicates the count of open alerts
Acknowledged Count	Indicates the count of acknowledged alerts
Closed Count	Indicates the count of closed alerts
Last Occurrence	Describes the rule's last occurrence date

## Alert details

Alert details view shows information about:

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

The screenshot shows the 'Alerts' section of the Maintenance page. The sidebar on the left has a 'Maintenance' tab selected. The main area displays a summary for the rule 'ShiftHumidityGreaterThan70' with counts for Total, Open, Acknowledged, and Closed alerts, and the last event details. Below this is a table of rule details and a table of alert occurrences.

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

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

A user can view the alert details by following below steps:

1. Click on Maintenance from the left menu to navigate to Maintenance page
2. Click on Alerts tab
3. Alerts grid loads
4. Click on the row of the alert which navigates to details page

The following details are shown in the Alert details page:

1. **Rule name:** Name of the Rule 2.**Counts of alert occurrences:** Total, Open, Acknowledged, Closed
2. **Last Event date**
3. **Severity**
4. **Rule details:** Rules grid which is similar to the one on [Rules page](#). A user can perform below operations from this grid similar to rules page:
  - o View Rule details
  - o Edit Rule
  - o Disable Rule
  - o Delete Rule

5. **Alert Occurrences:** A grid that depicts the occurrences of alerts. A user can perform below operations from this grid:

- o Acknowledge Alerts
- o Close Alerts
- o Delete Alerts

## Acknowledge Alerts

To acknowledge alerts:

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

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 detail

> Expand Columns

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

Alert Occurrences

> Expand Columns

Occurrence	Description	Severity	Trigger device	Time	Status
<input checked="" type="checkbox"/> ShiftHumidityGreaterThan70		▲ Critical	device12	07:05:24 PM 10.13.2020	open
<input type="checkbox"/> ShiftHumidityGreaterThan70		▲ Critical	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 button to **Close** appears in the tool bar
3. Click the button

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

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>▲</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 black; 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 button to **Delete** appears in the tool bar
3. Click the button

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

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>▲</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

## Jobs Grid

Jobs tab has a grid that shows the list of jobs for devices

The screenshot shows the Maintenance page with the following details:

- Alerts:** 0 Open Alerts (0 Critical, 0 Warning)
- Jobs:** 0 Failed jobs (2 Total, 2 Succeeded)
- Table Headers:** Job Name, Status, Operation, No. of devices, Succeeded, Failed, Start time, End time
- Table Data:**

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....

A user can view the jobs grid by following below steps:

1. Click on Maintenance from the left menu to navigate to Maintenance page
2. Click on jobs tab

Following lists the columns in 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 Device count
Succeeded	Succeeded status Device count
Failed	Failed status Device count
Start time	Datetime when the job started running
End time	Datetime when the job completed running

## Job details

Job details view shows:

- Jobs list
- Devices affected

The screenshot shows a user interface for maintenance operations. On the left, a sidebar menu includes Dashboard, Device Explorer, User Management, Rules, Packages, Deployments, and Maintenance, with Maintenance selected. The main area displays a grid of job details. At the top right of the grid, there are filters for 'Last day' and a refresh icon. Below the grid, there is a link to 'Expand Columns'. The grid has columns for Job Name, Status, Operation, No. of devices, Succeeded, Failed, Start time, and End time. One row is visible, showing 'Testjob-7cc37ae6-6e9f-405...' with a status of 'Completed', 1 device affected, 1 succeeded, 0 failed, and times from 10:58:35 AM 10.19... to 10:58:40 AM 10.19.... At the bottom of the grid, there are navigation links for '1 to 1 of 1', 'First', 'Previous', 'Page 1 of 1', 'Next', and 'Last'.

Job Name	Status	Operation	No. of devices	Succeeded	Failed	Start time	End time
Testjob-7cc37ae6-6e9f-405...	Completed		1	1	0	10:58:35 AM 10.19...	10:58:40 AM 10.19...

> Expand Columns

Job Name	Status	Device ID affected	Last return message	Start time	End time
Testjob-7cc37ae6-6e9f-405...	Completed	sah_test_device	Completed	10:58:35 AM 10.19.2020	10:58:40 AM 10.19.2020

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

## Devices

A user can view the job details by following below steps:

1. Click on Maintenance from the left menu to navigate to Maintenance page
2. Click on Jobs tab
3. Jobs grid loads
4. Click on the row of the job which navigates to details page

## 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

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
  - o :::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.
  - o :::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.
  - o This way is done by the cli.
    - <https://help.github.com/en/github/collaborating-with-issues-and-pull-requests syncing-a-fork>
  - o This is the way to do it through the web gui. I think this is the ideal way to do it.
    - Microsoft GitHub Azure and 3M 2020

shots.

- <https://youtu.be/YhwBgYPfoVE>
  - As you can see my repo is 15 commits behind.
  - ::image type="content" source="../../images/pull-request1.PNG" alt-text="Pull request 1"::
  - After comparing it will tell me I am able to merge.
  - ::image type="content" source="../../images/pull-request2.PNG" alt-text="Pull request 2"::
  - You would then submit a PR to sync your fork.
  - ::image type="content" source="../../images/pull-request3.PNG" alt-text="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 enviornment 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)

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 NVS 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 orted please see: environment: Windows 64-bit with Unsupported runtim1.0e

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 node.js 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.

## 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.

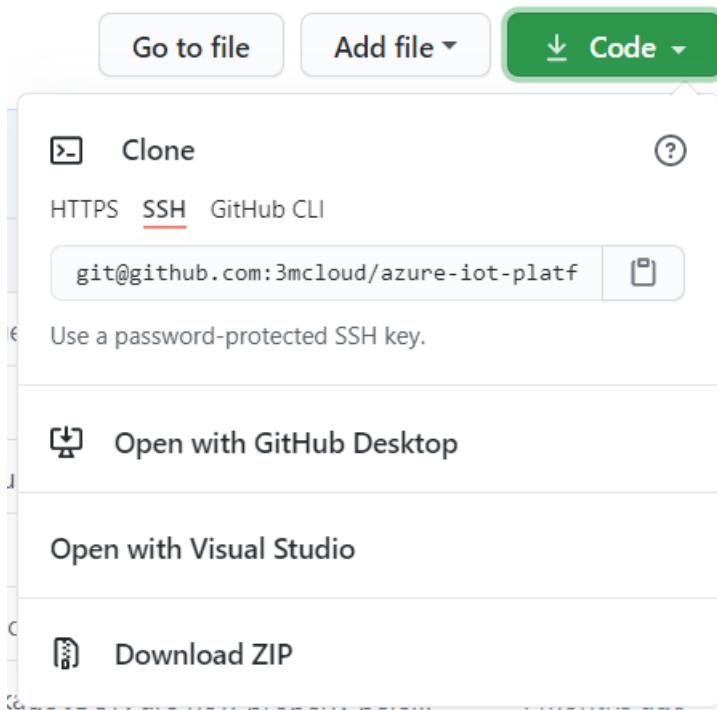
**Testing the SSH connection**

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



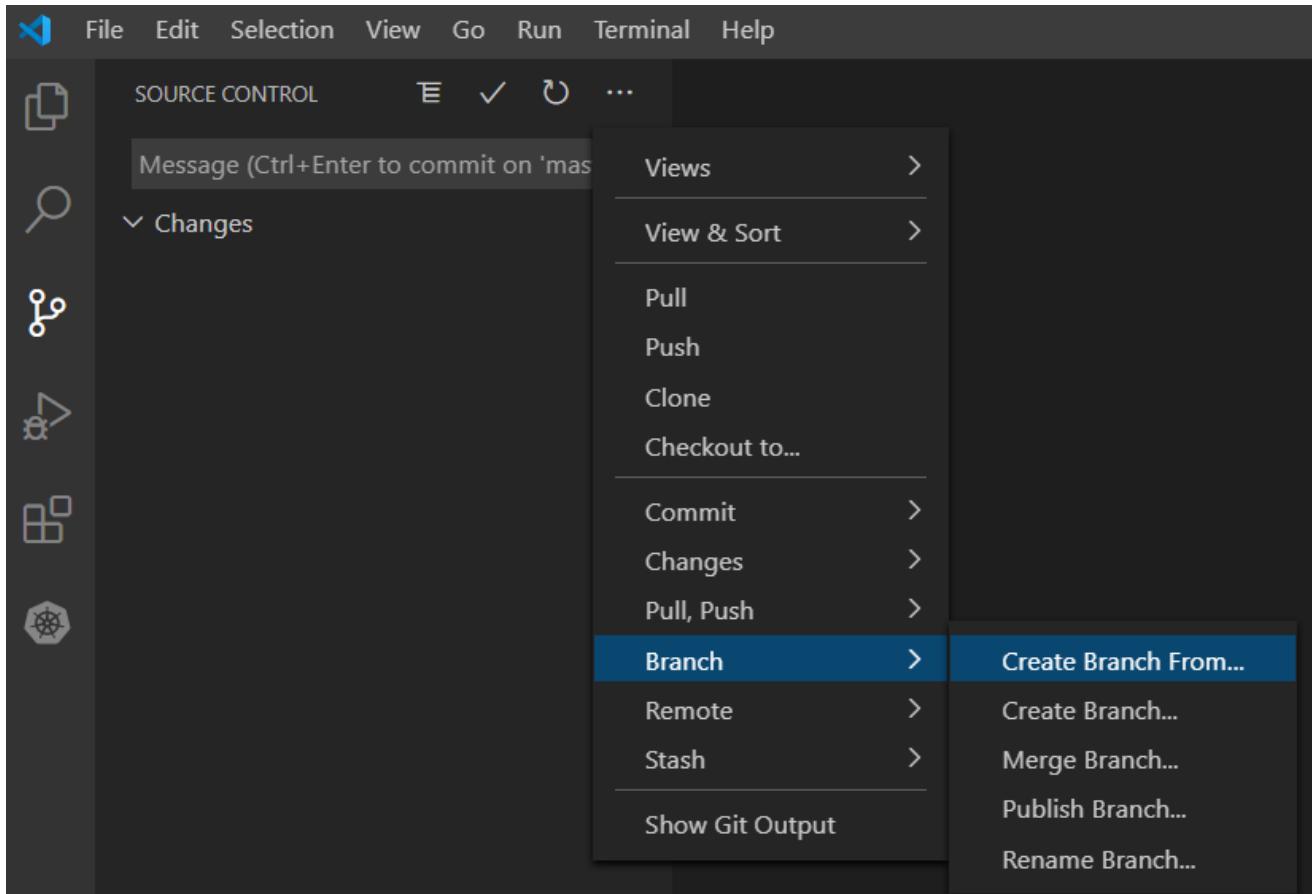
4. Clone the repository with this path from VS code/Visual Studio/Git bash or any other command liner

## 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

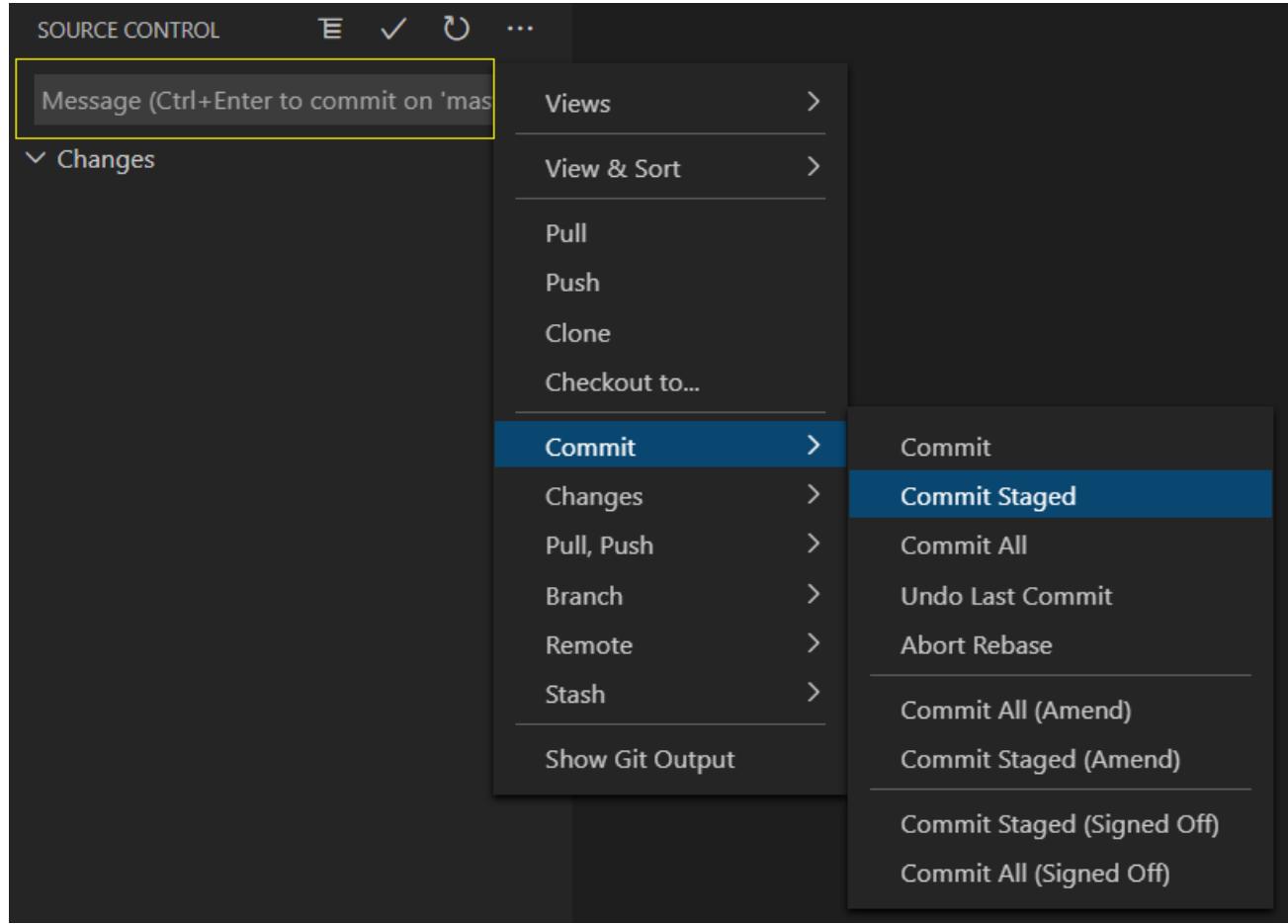
## Commit and push the branch

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

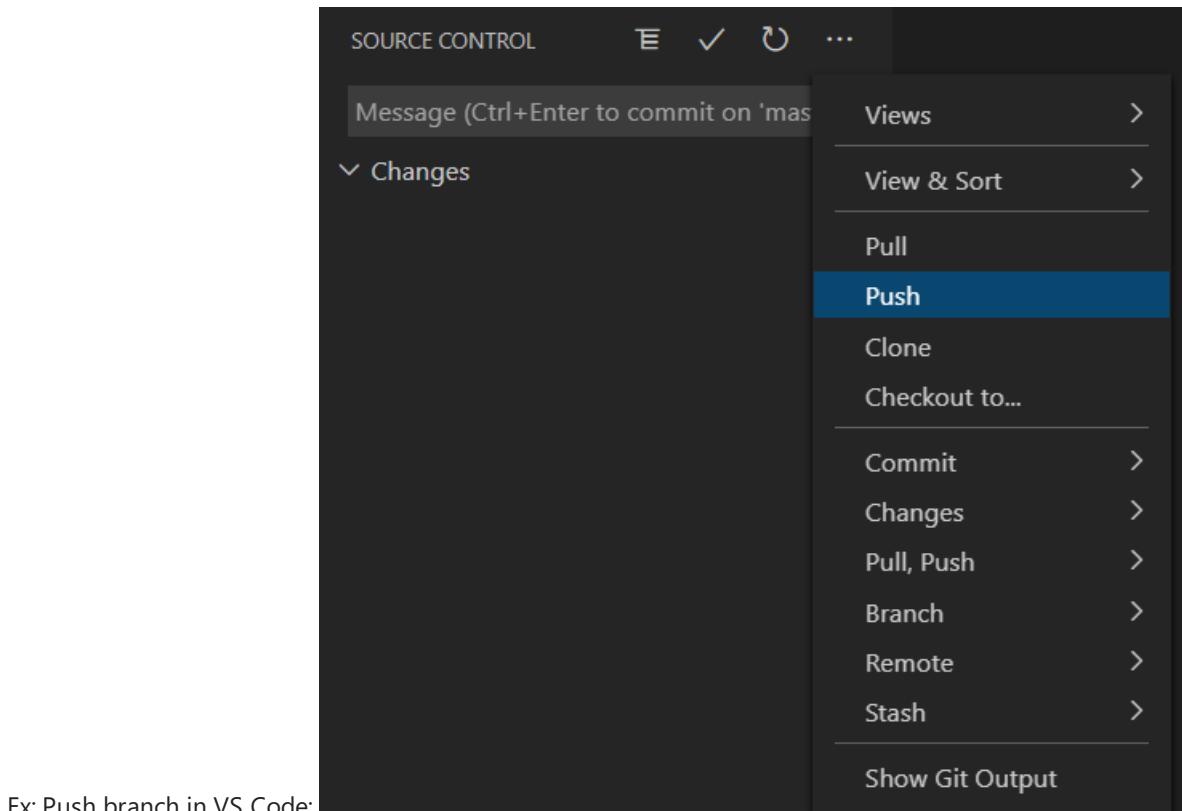
Note: Make sure to get latest/keep the branch in sync with master before committing in order to prevent any conflicts due to changes from the upstream

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

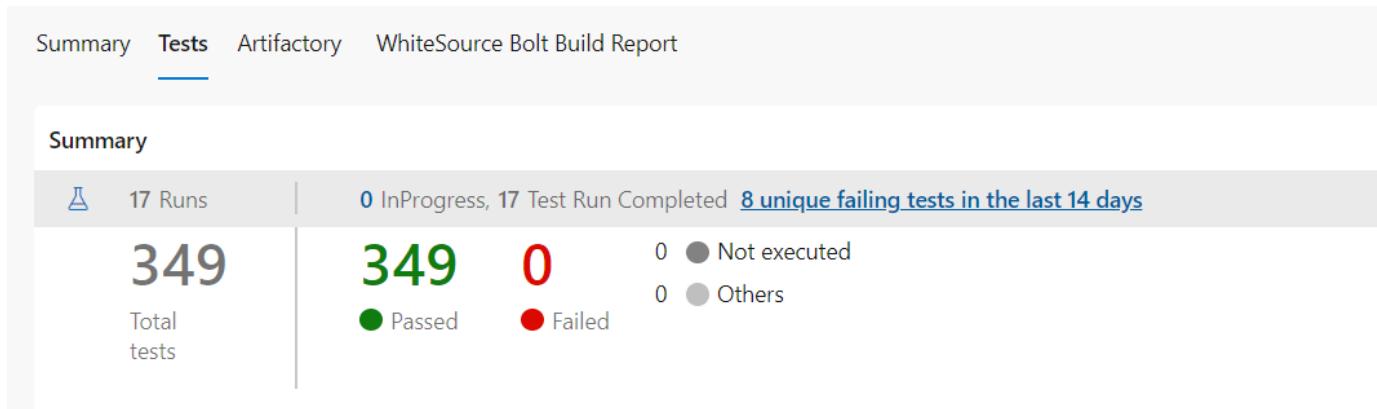
1. Login to VM
2. Open the browser and navigate to [pull requests tab](#)
3. Click "New pull request"
4. Select **base:master** and **compare:<your\_branch\_name>**
5. If required, review the commits and files changed
6. Give a brief description about the changes under Conversation tab
7. Add the revieweres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

Note:

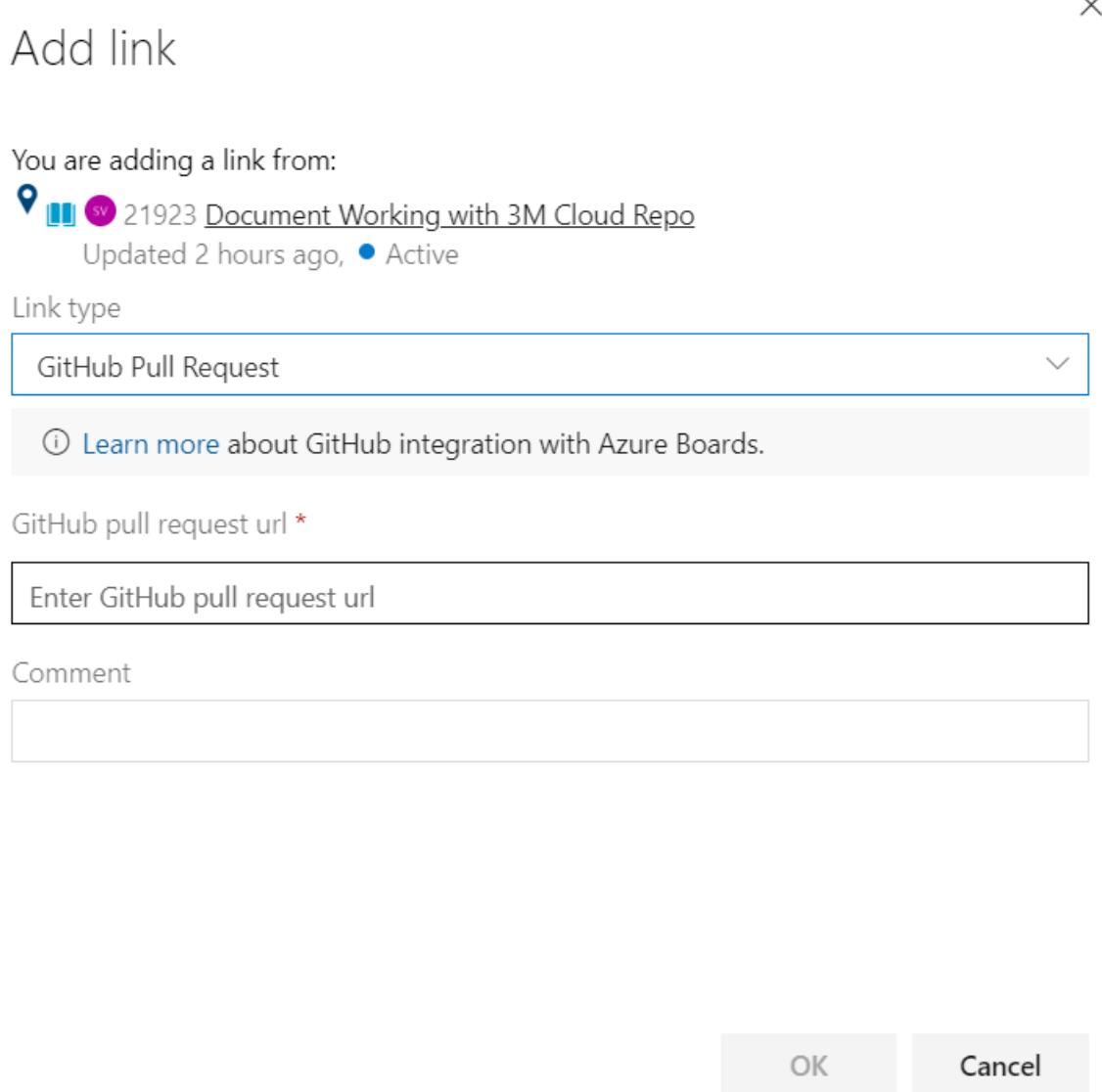
For more information on Naming branches and Pull requests, [check this](#)

## 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
```