

Azure IoT Central - 动手实验

场景

假设您经营一家管理冷藏卡车车队的公司。您在某个城市中有很多客户，基于客户的需求在货运中心来管理这些业务工作。您需要安排每辆卡车运输冷藏物品交付给客户。

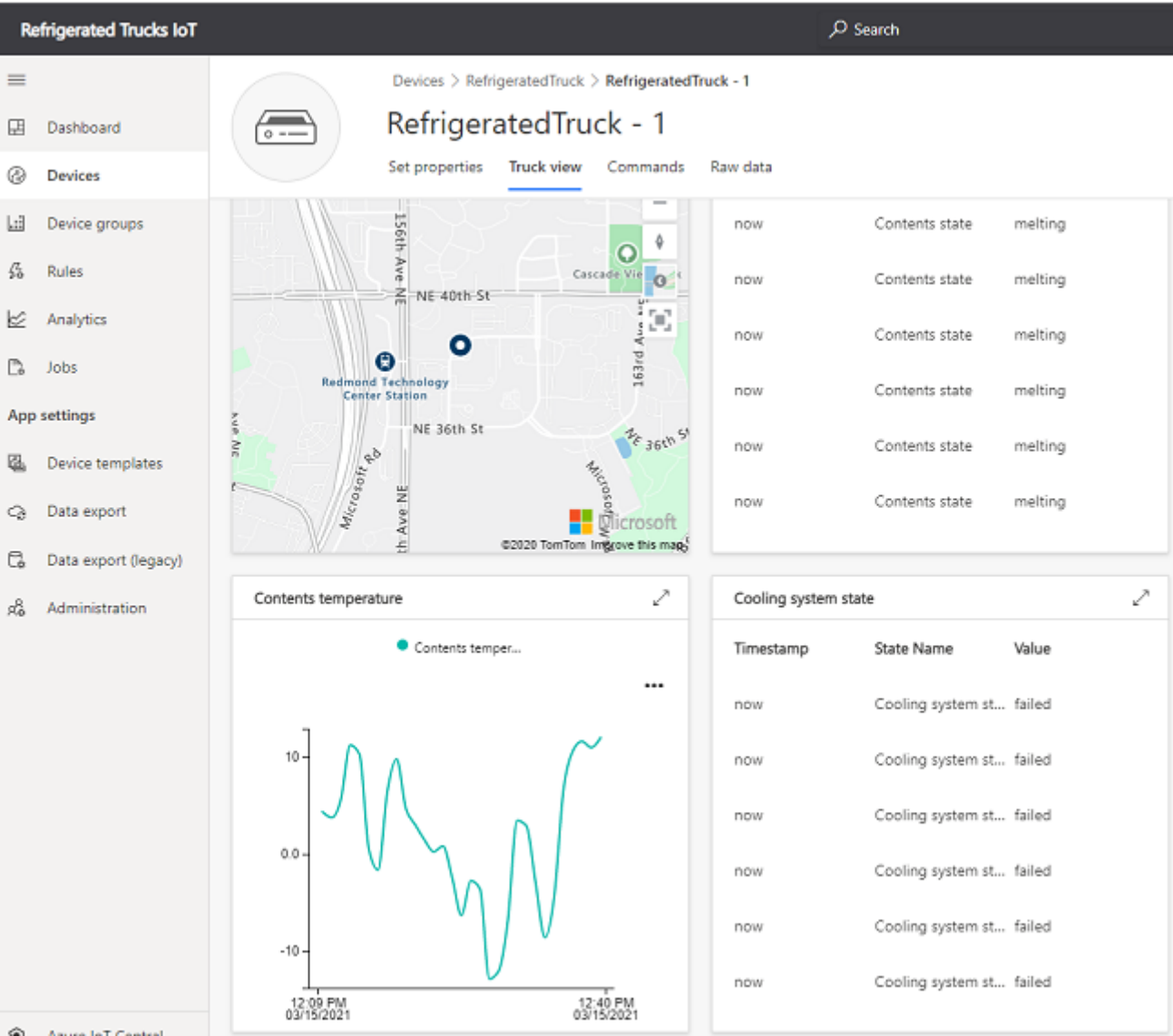
如果发现卡车上的冷却系统发生故障并且运输物品开始融化，您需要告诉卡车返回基地并卸下运输物品。或者，您也可以冷却系统出现故障时将运输物品交付给附近的客户。

要做出这些决定，您需要了解卡车发生问题的最新情况，包括每辆卡车在地图上的位置、冷却系统的状态以及运输物品的状态。

Azure IoT Central 提供了处理此场景所需的一切。例如，在下图中，彩色圆圈显示了一辆卡车在前往客户途中的位置。

交付的应用

在本动手实验结束时，您将会完成的应用程序应如下图所示



动手实验前所需要的准备

- Microsoft Azure 订阅
- 安装好 Visual Studio Code

动手实验步骤:

- 实验一: 创建自定义 Azure IoT Central 应用
 - 任务一: 创建应用
 - 任务二: 为您的应用添加遥测功能
 - 任务三: 为您的应用添加特性
 - 任务四: 为您的应用添加命令
- 实验二: 创建仪表板
 - 任务一: 让你的设备可视化
 - 任务二: 添加可写特性视图
 - 任务三: 创建设备
- 实验三: Azure 地图
- 实验四: 创建设备应用
 - 任务一: 创建你的开发环境
 - 任务二: 启动你的设备
 - 任务三: 设置特性
- 实验五: 创建规则
 - 实验一: 设置冷却系统状态规则
 - 实验二: 设置温度飙升规则
 - 实验三: 设置卡车离开货运中心规则
 - 实验四: 设置运输物品的温度规则
- 实验六: 清空资源

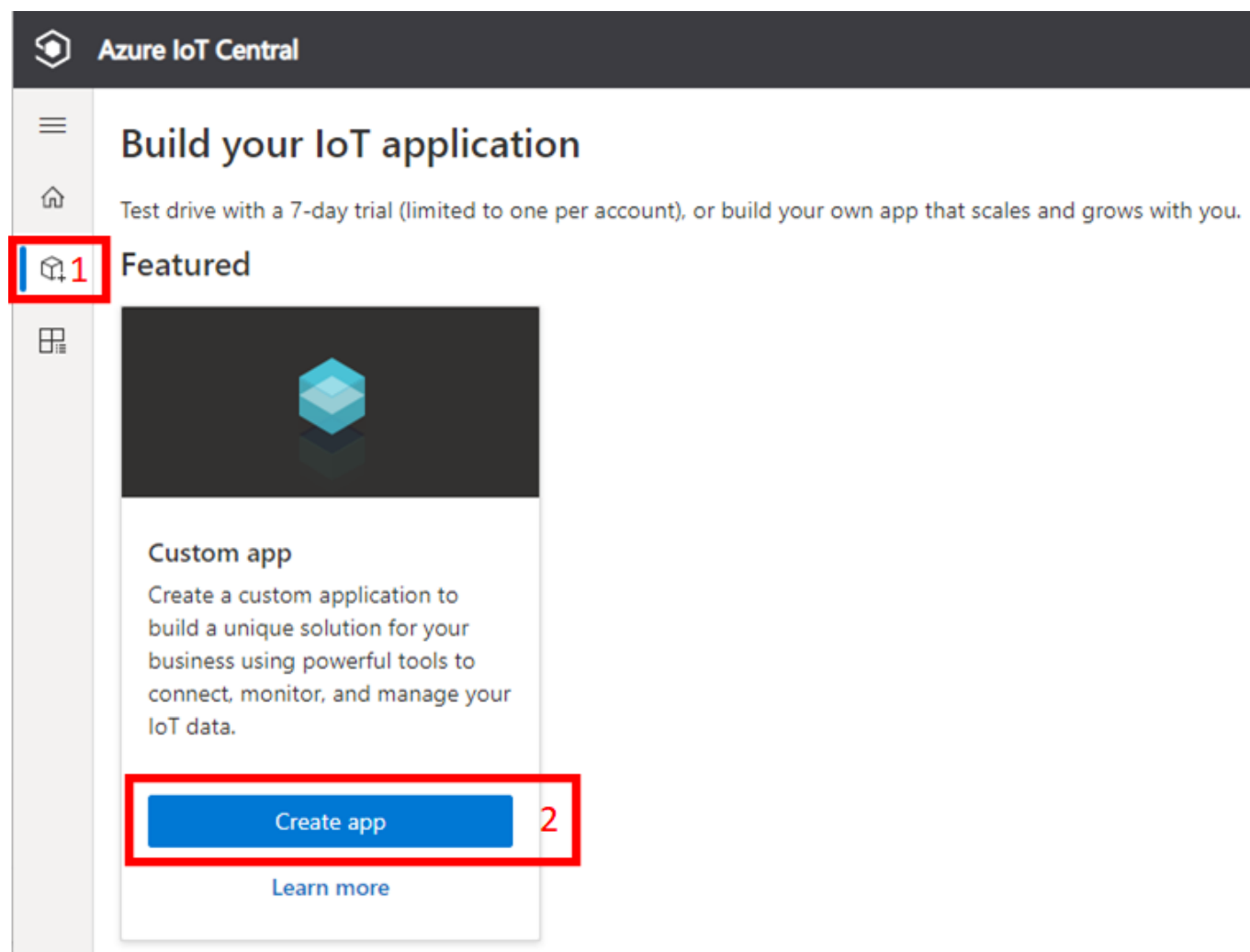
实验一: 创建自定义 Azure IoT Central 应用

任务一: 创建应用

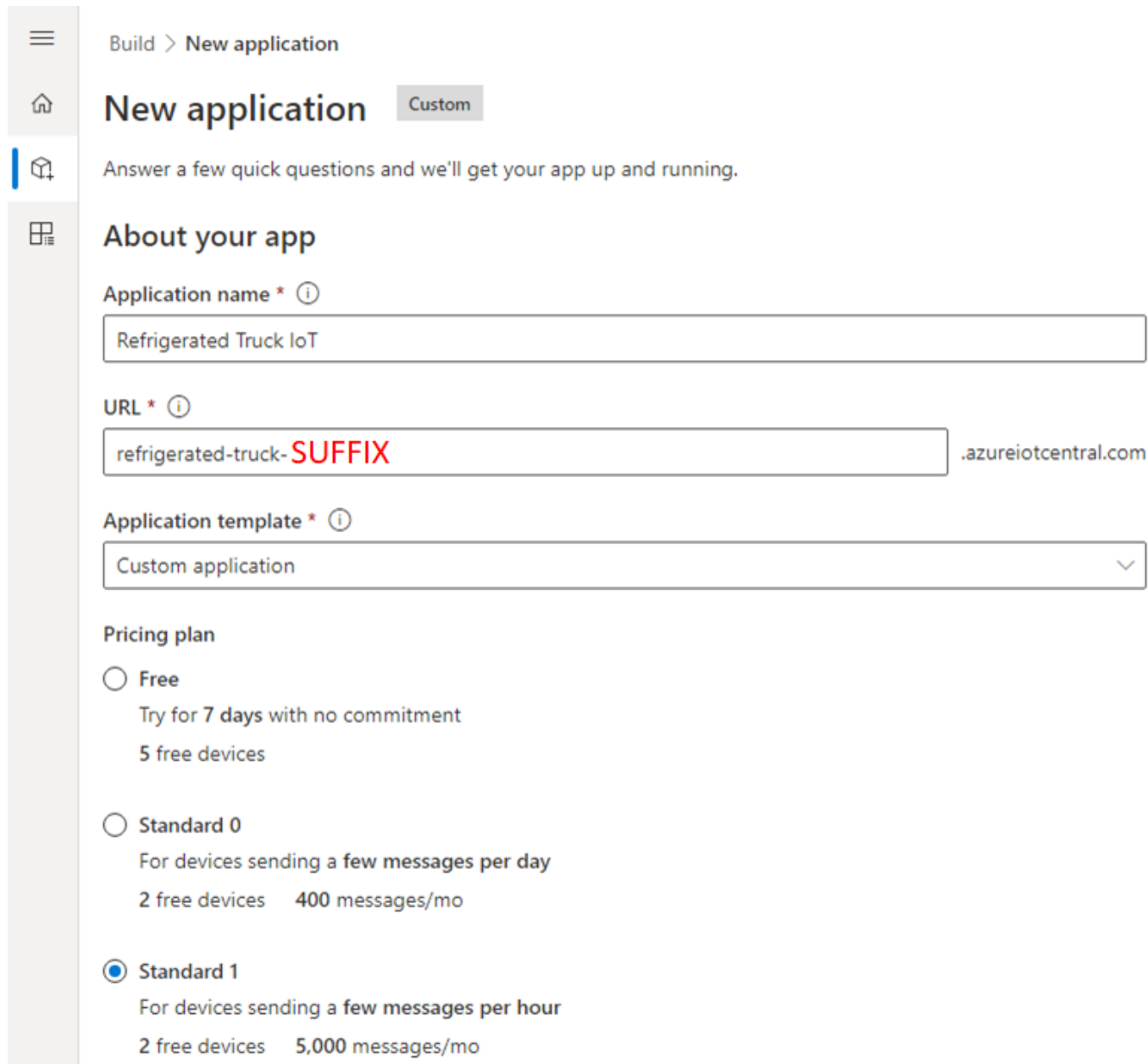
打开您的浏览器:

通过 Azure Pass 创建您的 Azure 订阅后, 登录 Azure 账户后进入 <https://portal.azure.com/>

打开 Azure IoT Central: <https://apps.azureiotcentral.com/>



选择左侧的创建图标，然后选择在 "自定义应用 / Custom app" 中创建应用 (Create app)。您需要填写申请表中的字段：



Build > New application

New application Custom

Answer a few quick questions and we'll get your app up and running.

About your app

Application name * ⓘ
Refrigerated Truck IoT

URL * ⓘ
refrigerated-truck-SUFFIX .azureiotcentral.com

Application template * ⓘ
Custom application

Pricing plan

☐ Free
Try for 7 days with no commitment
5 free devices

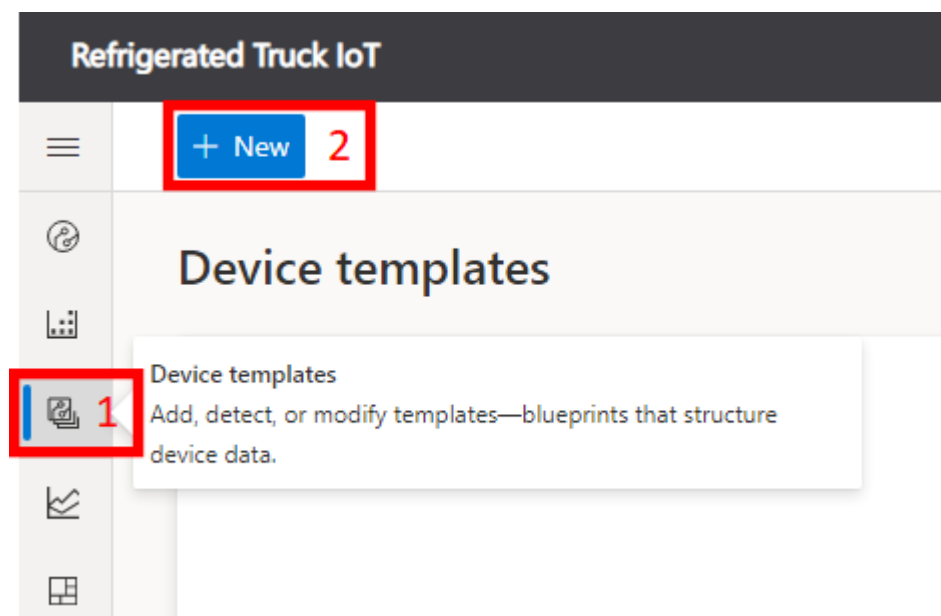
☐ Standard 0
For devices sending a few messages per day
2 free devices 400 messages/mo

☒ Standard 1
For devices sending a few messages per hour
2 free devices 5,000 messages/mo

- **应用名字 - Application Name:** 可以用与冷藏车项目有关的名字
- **URL:** refrigerated-trucks-后序 必须是唯一的 URL
- **应用模版 - Application Template:** 使用默认模版 - Custom Application
- **付费计划:** 请选择 " Standard 1 "
- **目录:** 您的订阅所在的目录，通常是与您的登录关联的“默认目录”。
- **Azure 订阅:** 选择您的 Azure 订阅
- **所在位置** 选择您所在 IoT Central 所在位置

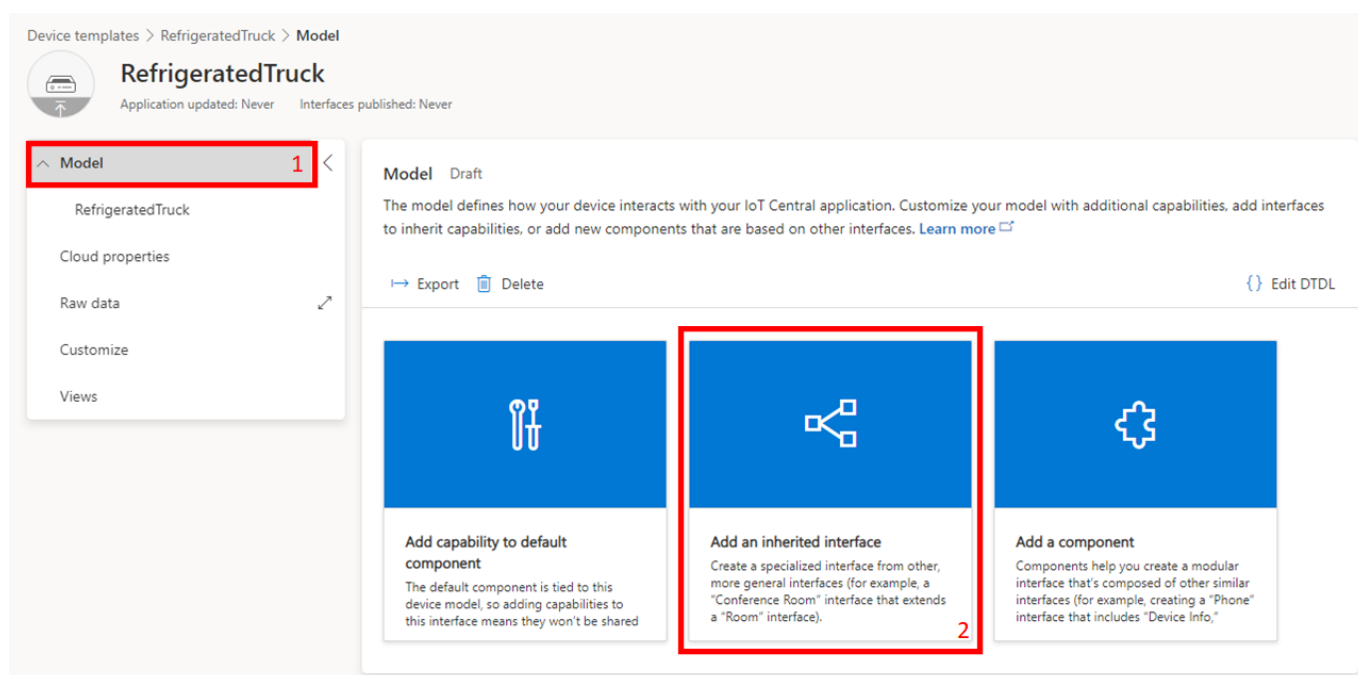
所有都确认填写好后，点击 **创建 - Create**

创建成功后，下一步需要创建设备模板。在左侧菜单中单击 **设备模版 - Device Templates** 按钮，然后再点击 **新建 - New**

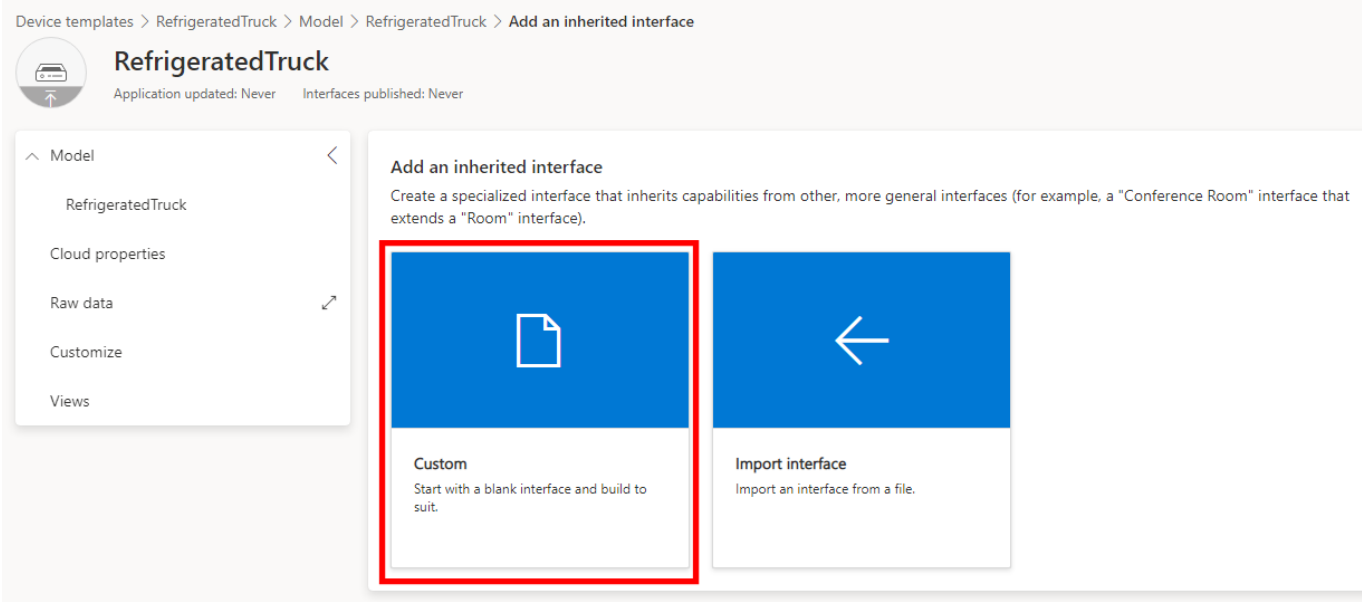


1. 选择 IoT 设备 - IoT Device 然后点击 下一步: 定制 - Next: Customize
2. 在自定义中分配一个 **Device Template name** - 设备模板名称 RefrigeratedTruck
3. 不要选择网关设备 - **Gateway device**框
4. 选择下一步: 查看 - **Next: Review**。然后选择创建 - **Create**
5. 在创建模型区域中, 选择自定义模型。您的视图现在应该类似于下图

6. 单击模型 - **Model**, 然后选择添加继承的接口 - **Add an inherited interface**



7. 选择自定义, 从空白界面开始构建



任务二: 为您的应用添加遥测功能

注意：接口名称必须完全按照本练习要求去输入。名称和条目必须与您稍后将在本模块中添加的代码对应。

1. 首先，请选择 “添加功能 - Add capability” 并单击向下图标以显示所有字段。

Display name

Contents temperature

Name *

ContentsTemperature

Capability type * ⓘ

Telemetry

Semantic type ⓘ

Temperature

×

▼

2. 在字段内依次输入以下值

输入对应字段	相关值
显示名称 - Display Name	Contents temperature
名称 - Name	ContentsTemperature
功能类型 - Capability type	Telemetry
语意类型 - Semantic Type	Temperature
结构类型 - Schema	Double
单位 - Unit	Degree celsius

确认您填写的功能如下图所示:

Display name

Contents temperature

Name *

ContentsTemperature

Capability type * ⓘ

Telemetry

Semantic type ⓘ

Temperature

Schema *

Define

Double

Unit

Degree celsius

Display unit

Comment

Description

3. 每个状态都是很重要。他们让操作员知道发生了什么。IoT Central 中的状态是与一系列值关联的名称。稍后您将选择与每个值关联的颜色。

使用 添加功能 - Add capability 控件为卡车的冷藏内容添加状态：**empty**、**full** 或 **melting**

输入对应字段	相关值
显示名称 - Display Name	Contents state
名称 - Name	ContentsState
功能类型 - Capability type	Telemetry
语意类型 - Semantic Type	State
结构类型 - Schema	String

选择 添加 - Add.

Contents state

ContentsState

Telemetry

State

State values

Value schema *

String

Please define the value and display name of each state

+ Add

对于显示名称和值，输入空。名称字段应自动填充为空。所以所有三个字段都是相同的，包含**empty**。添加另外两个状态值：**full** 和 **melting**。同理相同的文本应该出现在显示名称、名称和值的字段中。

State values

Value schema *

String

Display name

Name *

Value *

empty

empty

empty

×

full

full

full

×

melting

melting

melting

×

4. 如果冷却系统出现故障，正如您将在以下内容中看到的那样，货物融化的机会会大大增加。

输入对应字段	相关值
显示名称 - Display Name	Cooling system state
名称 - Name	CoolingSystemState
功能类型 - Capability type	Telemetry
语意类型 - Semantic Type	State
结构类型 - Schema	String

为冷却系统添加 **on**、**off** 和 **failed** 条目。 首先选择添加功能。 然后添加另外一个状态：

Cooling system state

CoolingSystemState

Telemetry

State

×

^

State values

Value schema *

String

Display name

Name *

Value *

on

on

on

×

off

off

off

×

failed

failed

failed

×

+ Add

Unit

Display unit

Comment

Description

None

5. 更复杂的状态是卡车本身的状态。如果一切顺利，卡车的正常路线可能已准备就绪，在途中、交付、返回、装载，然后再次准备就绪。还要添加倾倒状态以说明处理融化货物！要创建新状态，请使用与刚才最后两个步骤相同的过程。

输入对应字段	相关值
显示名称 - Display Name	Truck state
名称 - Name	TruckState
功能类型 - Capability type	Telemetry
语意类型 - Semantic Type	State
结构类型 - Schema	String

现在添加：**ready**、**enroute**、**delivering**、**returning**、**loading**、**dumping**，如下图：

Truck state

TruckState

Telemetry

State

×

^

State values

Value schema *

String

Display name	Name *	Value *	
ready	ready	ready	×
enroute	enroute	enroute	×
delivering	delivering	delivering	×
returning	returning	returning	×
loading	loading	loading	×
dumping	dumping	dumping	×

6. 添加事件功能。设备可能会触发一个事件 - 冲突指令。一个例子可能是当一辆从客户那里返回的空卡车收到将其内容交付给另一个客户的命令时。如果发生冲突，设备应触发事件以警告 IoT Central 应用程序的操作员。

另一个事件可能只是确认并记录卡车要交付的客户 ID。

要创建事件，请选择添加功能 - **Add capability**。然后填写以下信息。

输入对应字段	相关值
显示名称 - Display Name	Event
名称 - Name	Event
功能类型 - Capability type	Telemetry
语意类型 - Semantic Type	Event
结构类型 - Schema	String
严重性 - Severity	Information

设置如下图所示：

Event

Event

Telemetry

Event

×

^

Schema *

String

Severity ⓘ

Information

Unit

None

Display unit

Comment

Description

7. 使用以下信息添加位置功能:

输入对应字段	相关值
显示名称 - Display Name	Location
名称 - Name	Location
功能类型 - Capability type	Telemetry
语意类型 - Semantic Type	Location
结构类型 - Schema	Geopoint

任务三: 为您的应用添加特性

将卡车货物的最佳温度定义为特性。

1. 选择 "添加 - Add" 功能。然后添加卡车 ID 特性。

输入对应字段	相关值
显示名称 - Display Name	Truck ID
名称 - Name	TruckID
功能类型 - Capability type	Property
语意类型 - Semantic Type	None
结构类型 - Schema	String
可写 - Writable	Off
单位 - Unit	None

您的特性设置如下:

Truck ID

TruckID

Property

None

×

^

Schema *

Define

String

Writable

Off

Unit

None

Display unit

Comment

Description

2. 添加最佳温度的特性

输入对应字段	相关值
显示名称 - Display Name	Optimal Temperature
名称 - Name	OptimalTemperature
功能类型 - Capability type	Property
语意类型 - Semantic Type	Temperature
结构类型 - Schema	Double
可写 - Writable	On
单位 - Unit	Degree celsius

如下图所示:

Optimal Temperature

OptimalTemperature

Property

Temperature

×

^

Schema *

Define

Double

Writable

On

Unit

Degree celsius

Display unit

Comment

Description

任务四: 为您的应用添加命令

对于冷藏车，您应该添加两个命令：

将内容交付给客户的命令 将卡车召回基地的命令

1. 要添加命令，请选择添加功能 - Add capability 添加第一个命令

输入对应字段	相关值
显示名称 - Display Name	Go to customer

输入对应字段	相关值
名称 - Name	GoToCustomer
功能类型 - Capability type	Command

打开 **请求 - Request** 选项以输入更多命令相关详细信息。

输入对应字段	相关值
请求 - Request	On
显示名称 - Display Name	Customer ID
名称 - Name	Customer ID
结构类型 - Schema	Integer

使用下图验证您的输入是否一致：

Go to customer

GoToCustomer

Command

×

^

Queue if offline

Off

Comment

Description

Request

Display name

Customer ID

Name *

CustomerID

Schema *

Define

Integer

Comment

Description

Response

2. 创建一个命令来召回卡车。

输入对应字段	相关值
显示名称 - Display Name	Recall
名称 - Name	Recall
功能类型 - Capability type	Command

您的召回特性应如下图所示：

Recall

Recall

Command

×

^

Queue if offline

☒ Off

Comment

Description

Request ☒

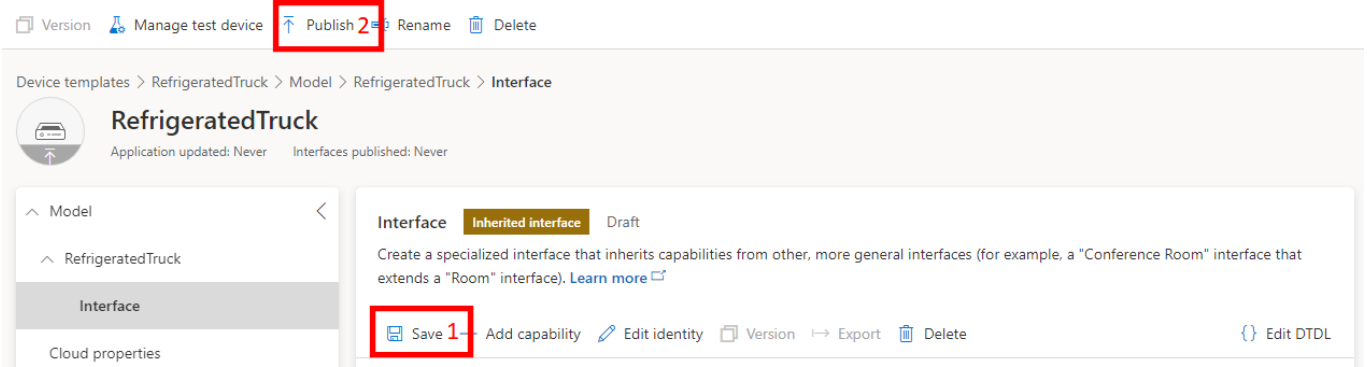
Response ☒

3. 在继续之前，请仔细检查您的界面。 发布界面后，编辑选项受到限制。所以你应该在发布之前把它弄好。

当您选择设备模板的名称时，以“视图”选项结尾的菜单总结了功能，合共6 个基于遥测，2 个特性和 2 个命令：

Display name	Name *	Capability type * ⓘ	Semantic type ⓘ		
Contents temperature	ContentsTemperature	Telemetry	Temperature	×	∨
Contents state	ContentsState	Telemetry	State	×	∨
Cooling system state	CoolingSystemState	Telemetry	State	×	∨
Truck state	TruckState	Telemetry	State	×	∨
Event	Event	Telemetry	Event	×	∨
Location	Location	Telemetry	Location	×	∨
Truck ID	TruckID	Property	None	×	∨
Optimal Temperature	OptimalTemperature	Property	Temperature	×	∨
Go to customer	GoToCustomer	Command		×	∨
Recall	Recall	Command		×	∨

4. 选择保存 - Save，然后选择发布 - Publish



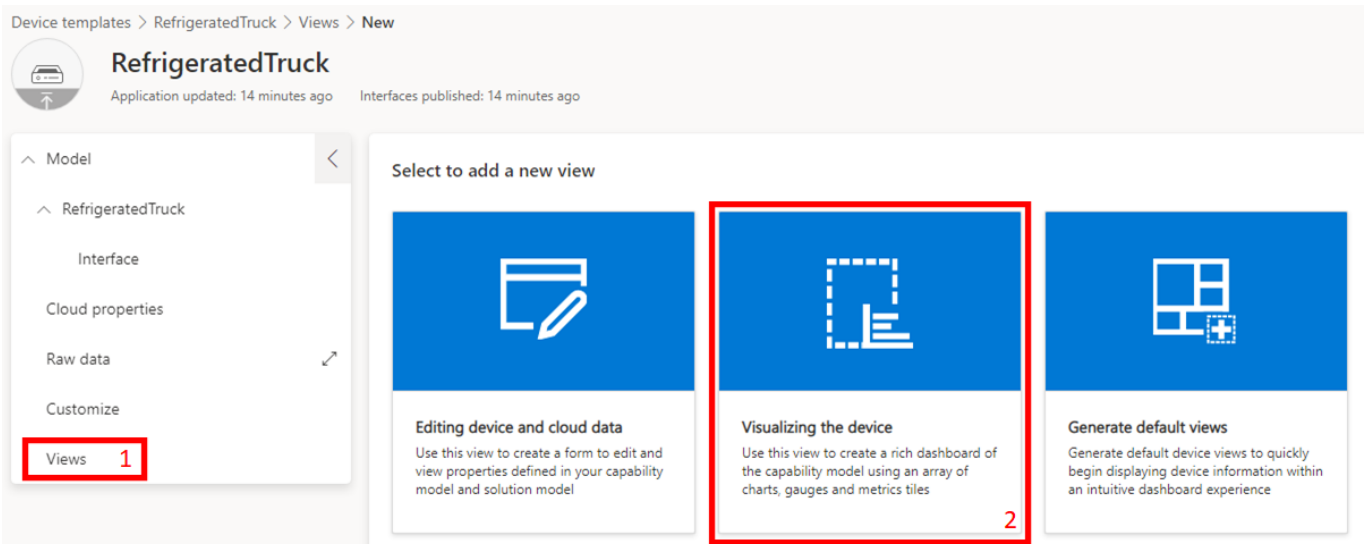
5. 在对话框中，再次选择 **发布 - Publish**。注释应从 **草稿 - Draft** 更改为 **已发布 - Published**。



实验二: 创建仪表盘

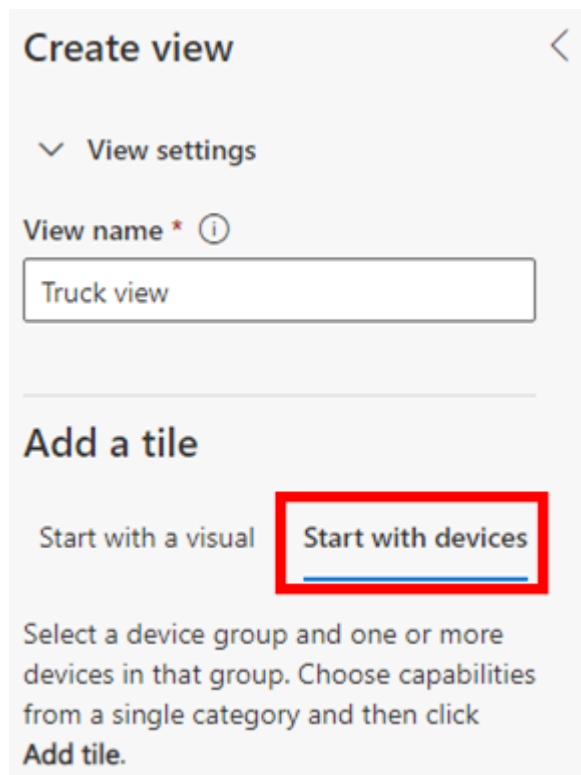
任务一: 让你的设备可视化

1. 选择视图 - Views。然后选择 **Visualizing the device** - 可视化设备



1. 将视图名称更改为更具体的名称，如 **Truck view**

2. 单击 - Add a tile 添加磁贴 - 下的从设备开始 - **Start with devices**。



Create view

View settings

View name * ⓘ

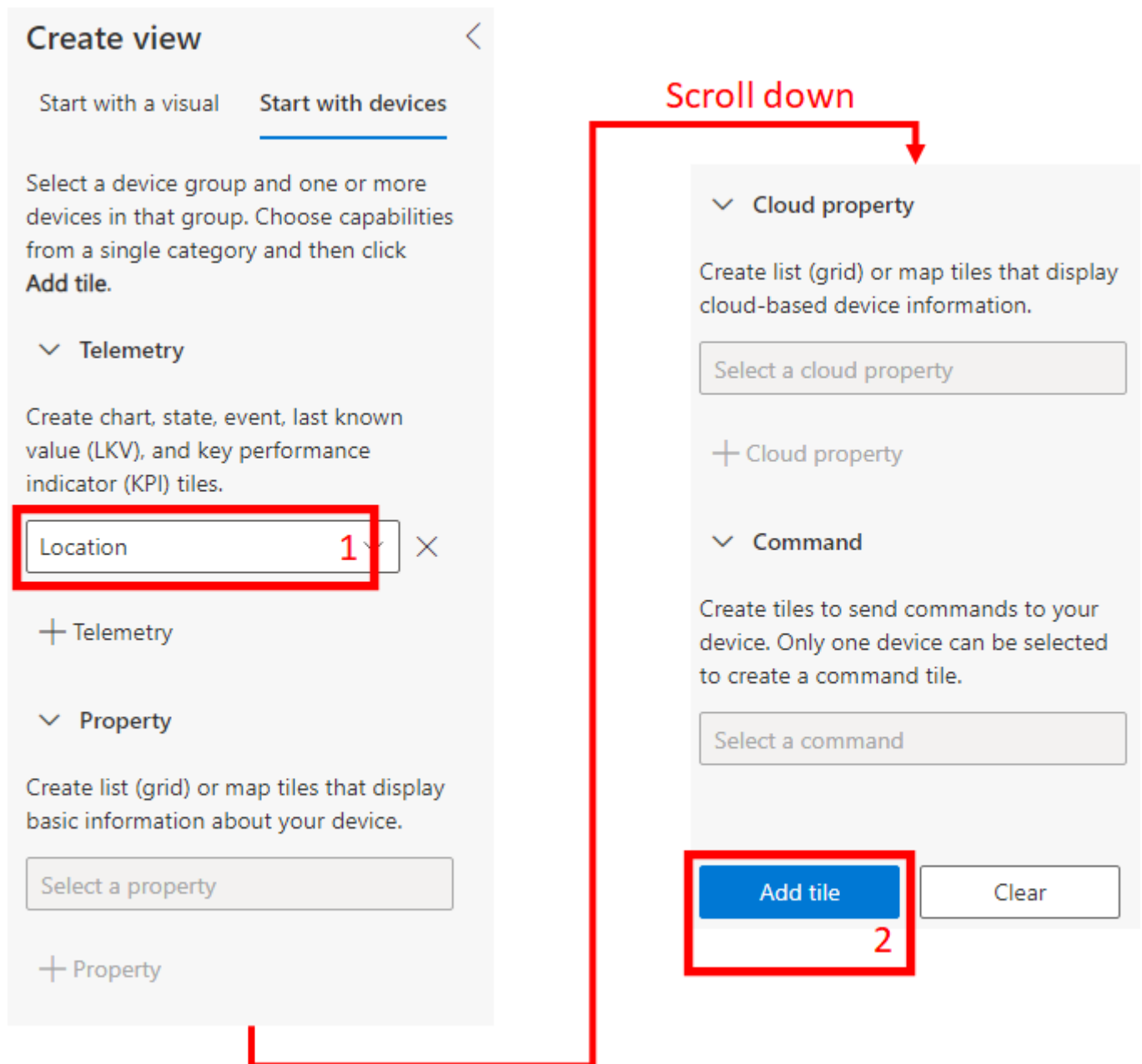
Truck view

Add a tile

Start with a visual **Start with devices**

Select a device group and one or more devices in that group. Choose capabilities from a single category and then click Add tile.

3. 在 遥测 - Telemetry 下，通过下拉列表选择 位置 - Location，然后滚动到底部以点击 添加磁贴 - Add tile。仪表板由磁贴组成。我们首先选择地理位置磁贴。

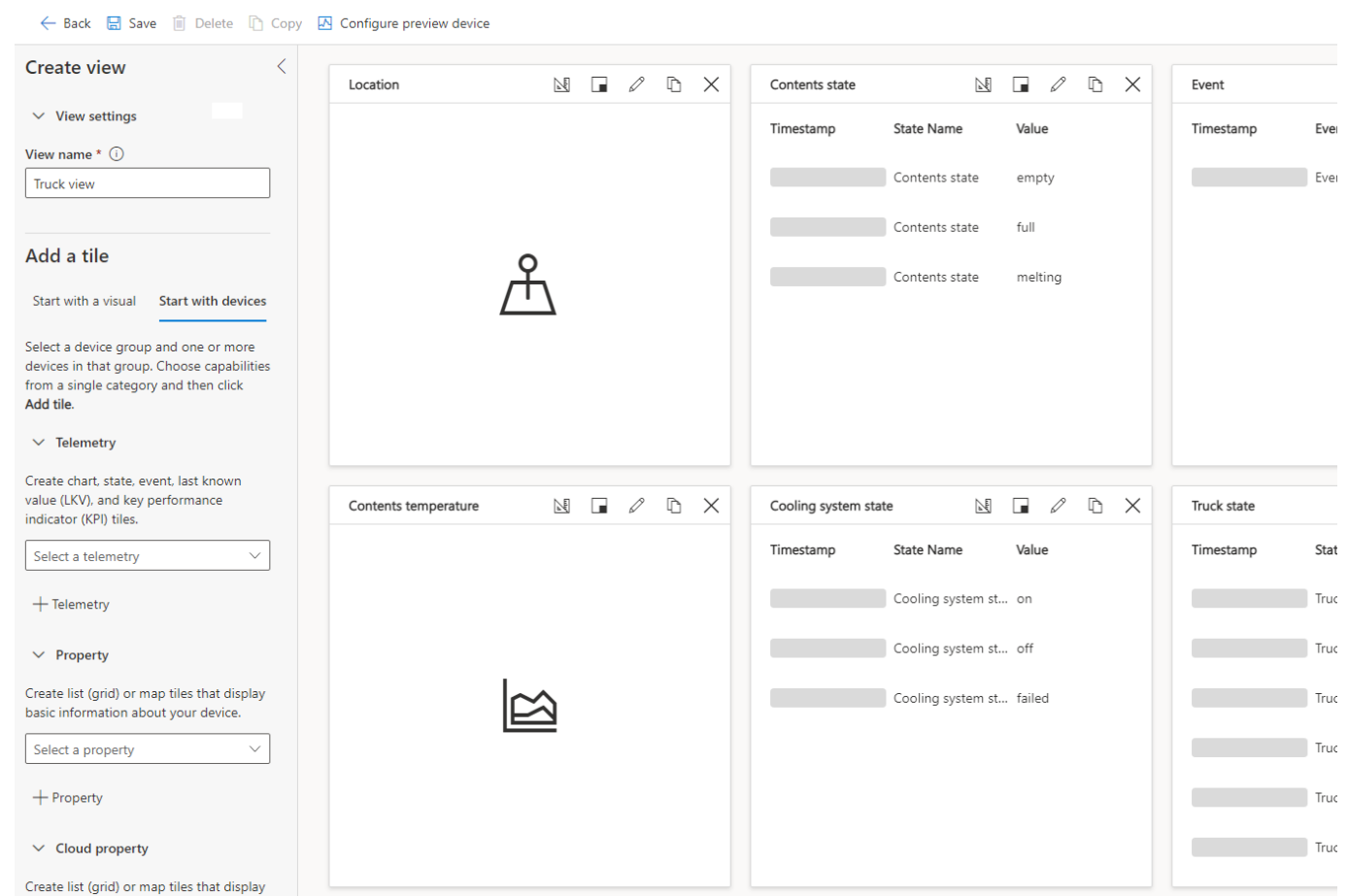


4. 从顶部开始，选择创建的遥测和特性功能。对于每个功能，都需要点击底部的 **添加磁贴 - Add tile**。

遥测: 位置, 货物状态, 货物温度, 制冷系统状态, 事件, 卡车状态

特性: 最佳温度, 卡车编号

您的新仪表板应如下图所示:

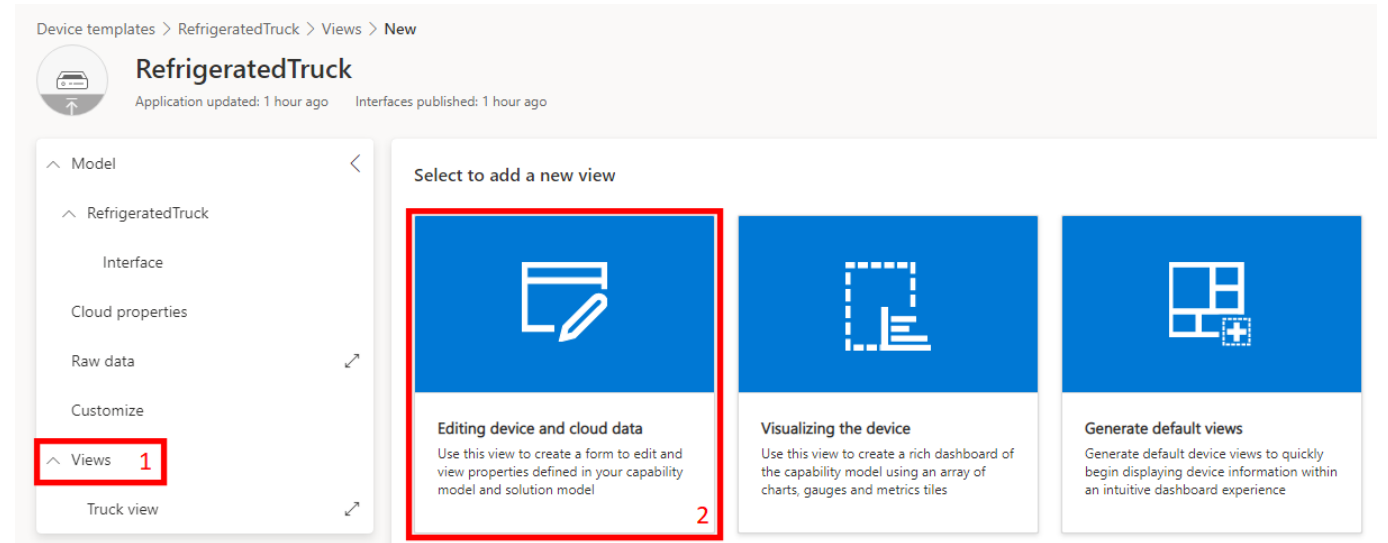


5. 点击 **保存 - Save** 保存该视图并点击 **返回 - Back** 按钮回到设备模板。

任务二: 添加可写特性视图

我们需要创建一个单独的视图。它的目的是设置可写特性。

1. 选择 **视图 - Views**，然后选择 **编辑设备和云数据 - Editing device and cloud data** 磁贴。



2. 将表单名称更改为 **Set properties** 之类的名称

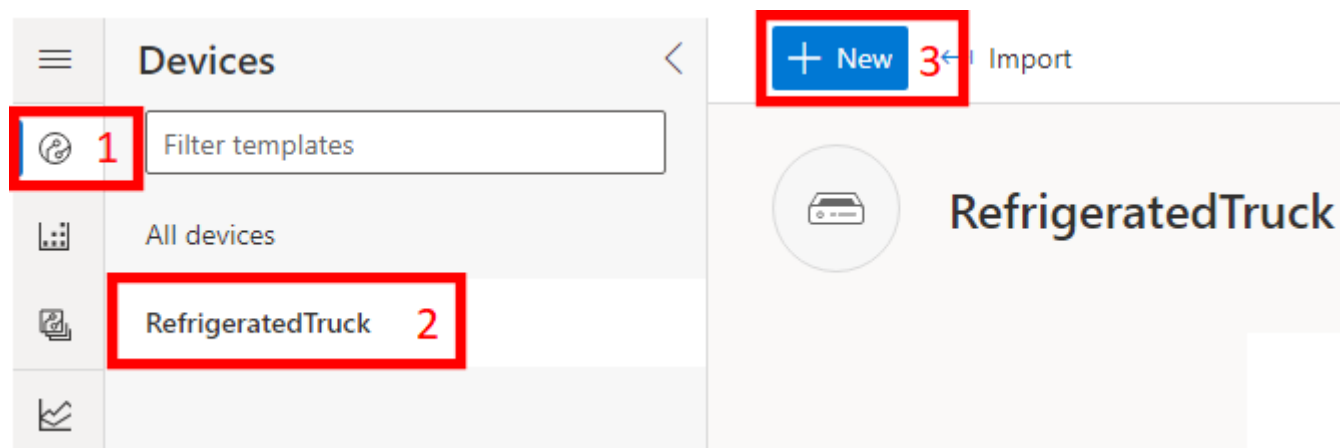
3. 选中 **最佳温度 - Optimal temperature** 属性复选框。然后单击 **添加部分 - Add section**

4. 确认您创建的视图是否类似于下图。然后单击 **保存 - Save** 保存该视图，单击 **返回 - Back** 返回设备模板。

5. 选择发布 - **Publish**。然后在对话框中，再次选择 发布 - **Publish**。

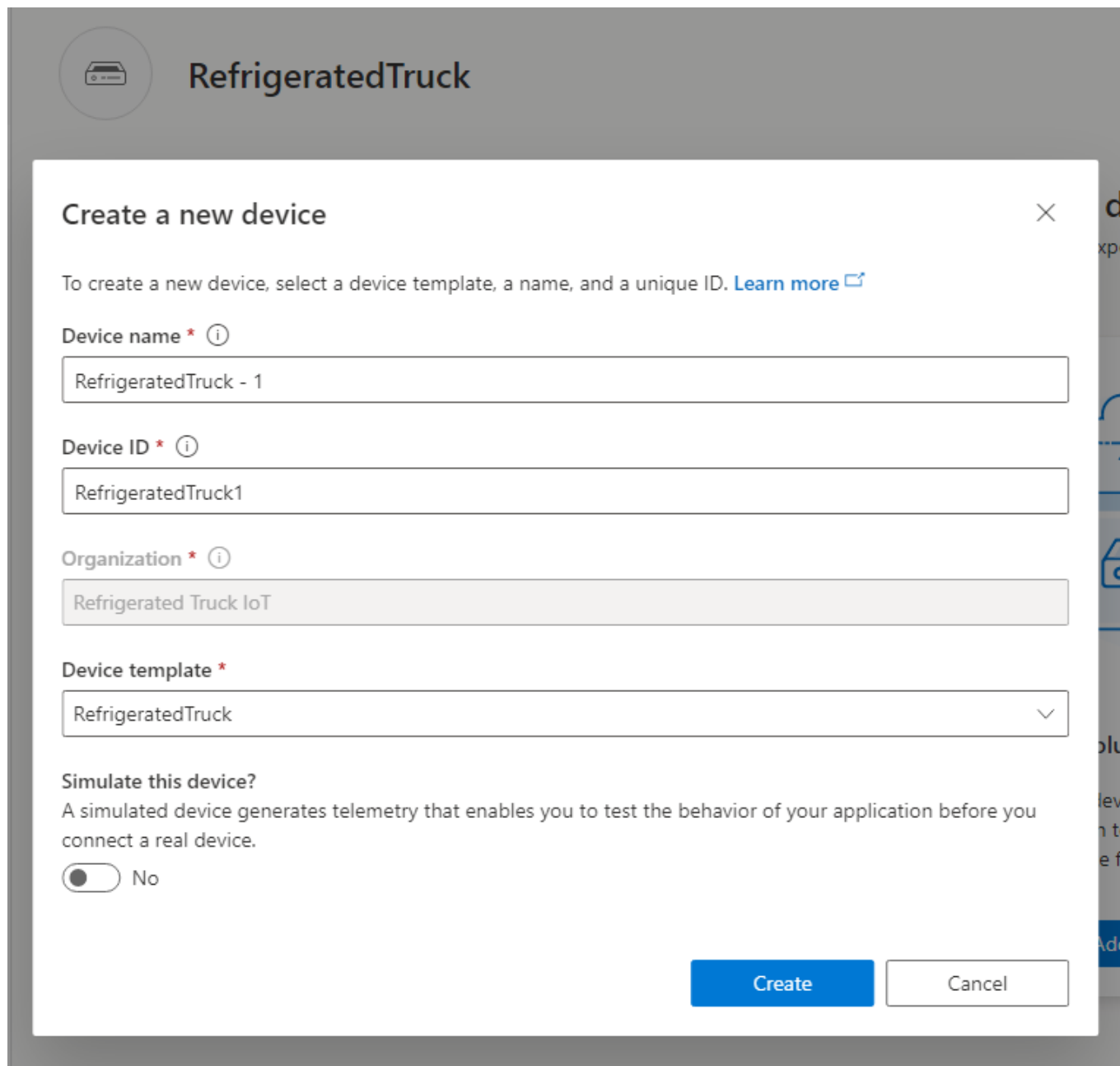
任务三: 创建设备

1. 在左侧菜单中，单击设备 - **Devices**，选择 **RefrigeratedTruck**，然后单击 **新建 - New**



2. 在 Create a new device 对话框中，验证设备模板是否为 **RefrigeratedTruck**

- 设备名称 - **Device name**: RefrigeratedTruck - 1
- 设备 ID - **Device ID**: RefrigeratedTruck1
- 设备模版 - **Device template**: RefrigeratedTruck (default)
- 是否模拟该设备 - **Simulate this device?**: No (default)



Create a new device

To create a new device, select a device template, a name, and a unique ID. [Learn more](#)

Device name * ⓘ
RefrigeratedTruck - 1

Device ID * ⓘ
RefrigeratedTruck1

Organization * ⓘ
Refrigerated Truck IoT

Device template *
RefrigeratedTruck

Simulate this device?
A simulated device generates telemetry that enables you to test the behavior of your application before you connect a real device.
☐ No

Create **Cancel**

3. 点击**创建 - Create**。请注意，设备状态为 **Registered**。只有在设备状态为 **已配置 - Provisioned** 后，IoT Central 应用才会接受与设备的连接。下面的编码单元显示了如何配置设备。
4. 单击 **RefrigeratedTruck - 1**，然后单击 **Truck view** 以查看实时仪表板，其中所有图块将显示 **找不到任何数据 - No data found** 因为我们还没有任何遥测数据。在包含 **Truck view** 的栏上，单击 **指令 - Commands**，您将看到您输入的两个命令已准备好运行。
5. 在右上角，单击**连接 - Connect**。

在打开的设备连接对话框中，仔细复制 **ID范围 - ID scope**、**设备ID - Device ID** 和 **Primary key - 主键**。ID 范围标识应用程序。设备 ID 标识真实设备。主键为您提供连接权限。

将此信息粘贴到文本文件中。

将身份验证类型设置保留为 **Shared access signature (SAS) - 共享访问签名 (SAS)**。

保存 ID 和密钥后，在对话框中选择关闭。

实验三: Azure 地图

1. 进入 Azure 门户: <https://portal.azure.com/>
2. 点击 **创建资源 - Create a Resource**, 然后在搜索框中键入 **Azure Maps**。打开 Azure Maps 服务页面, 然后单击 **创建 - Create**。

[Home](#) > [Create a resource](#) >

Azure Maps ...

Microsoft



Azure Maps [Add to Favorites](#)

Microsoft

★ 3.9 (41 Azure ratings)

Plan

Azure Maps

Create

Overview

Plans

Usage Information + Support

Reviews

填写创建表单:

- 订阅 - **Subscription**: 选择你的 Azure 订阅
- 资源组 - **Resource group**: 输入 IOTC
- 名称 - **Name**: 输入 mytrucksacademy后序
- 区域 - **Region**: 选择你在本次练习所需要的区域
- 定价 - **Pricing tier**: 使用 Gen1 (S1)
- 许可和隐私声明 - **License and Privacy Statement** 选中

点击页面底部的 **检查+ 创建 - Review + create**, 然后点击检查页面底部的**创建 - Create**。

[Home](#) > [Create a resource](#) > [Marketplace](#) > [Azure Maps](#) >

Create an Azure Maps Account resource ...

[Basics](#) [Identities](#) [Tags](#) [Review + create](#)

Azure Maps is a collection of geospatial services and SDKs that use fresh mapping data to provide geographic context to web and mobile applications. [Learn more](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	<div><div></div><div>Azure subscription 1</div><div>▼</div></div>
Resource group *	<div><div></div><div>IOTC</div><div>▼</div></div> <div>Create new</div>

Instance details

Name *	<div><div></div><div>mytruckacademySUFFIX</div><div>✓</div></div>
Region *	<div><div></div><div>West US 2</div><div>▼</div></div>
Pricing tier *	<div><div></div><div>Gen1 (S1)</div><div>▼</div></div>

[View full pricing details](#)

Terms

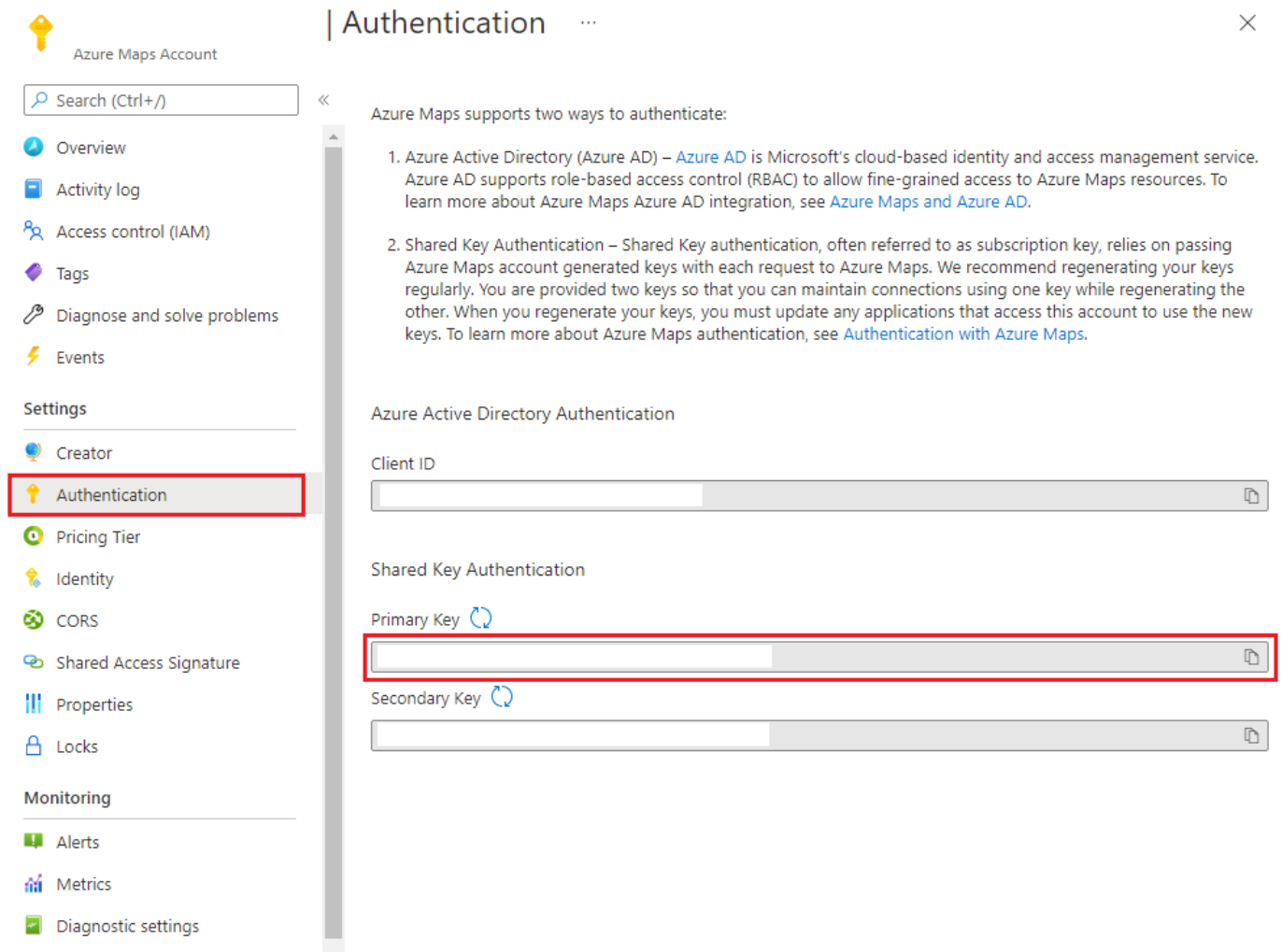
Azure Maps shares customer-provided address/location queries ("Queries") with third party TomTom for mapping functionality purposes. Queries are not linked to any customer or end-user when shared with TomTom and cannot be used to identify individuals. Microsoft is currently in the process of adding TomTom to the Online Services Subcontractor List.

[Learn more about Preview](#)

I confirm that I have read and agree to the License and Privacy Statement.

[Review + create](#)[< Previous](#)[Next : Identities >](#)

创建 Azure Maps 资源后，单击 **进入资源 - Go to resource** 然后在 **验证 - Authentication** 边栏选项卡上找到密钥。复制主键并将其粘贴到记事本中。



Azure Maps Account

Search (Ctrl+/)

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Events

Settings

- Creator
- Authentication**
- Pricing Tier
- Identity
- CORS
- Shared Access Signature
- Properties
- Locks

Monitoring

- Alerts
- Metrics
- Diagnostic settings

Authentication

Azure Maps supports two ways to authenticate:

1. Azure Active Directory (Azure AD) – [Azure AD](#) is Microsoft's cloud-based identity and access management service. Azure AD supports role-based access control (RBAC) to allow fine-grained access to Azure Maps resources. To learn more about Azure Maps Azure AD integration, see [Azure Maps and Azure AD](#).
2. Shared Key Authentication – Shared Key authentication, often referred to as subscription key, relies on passing Azure Maps account generated keys with each request to Azure Maps. We recommend regenerating your keys regularly. You are provided two keys so that you can maintain connections using one key while regenerating the other. When you regenerate your keys, you must update any applications that access this account to use the new keys. To learn more about Azure Maps authentication, see [Authentication with Azure Maps](#).

Azure Active Directory Authentication

Client ID

Shared Key Authentication

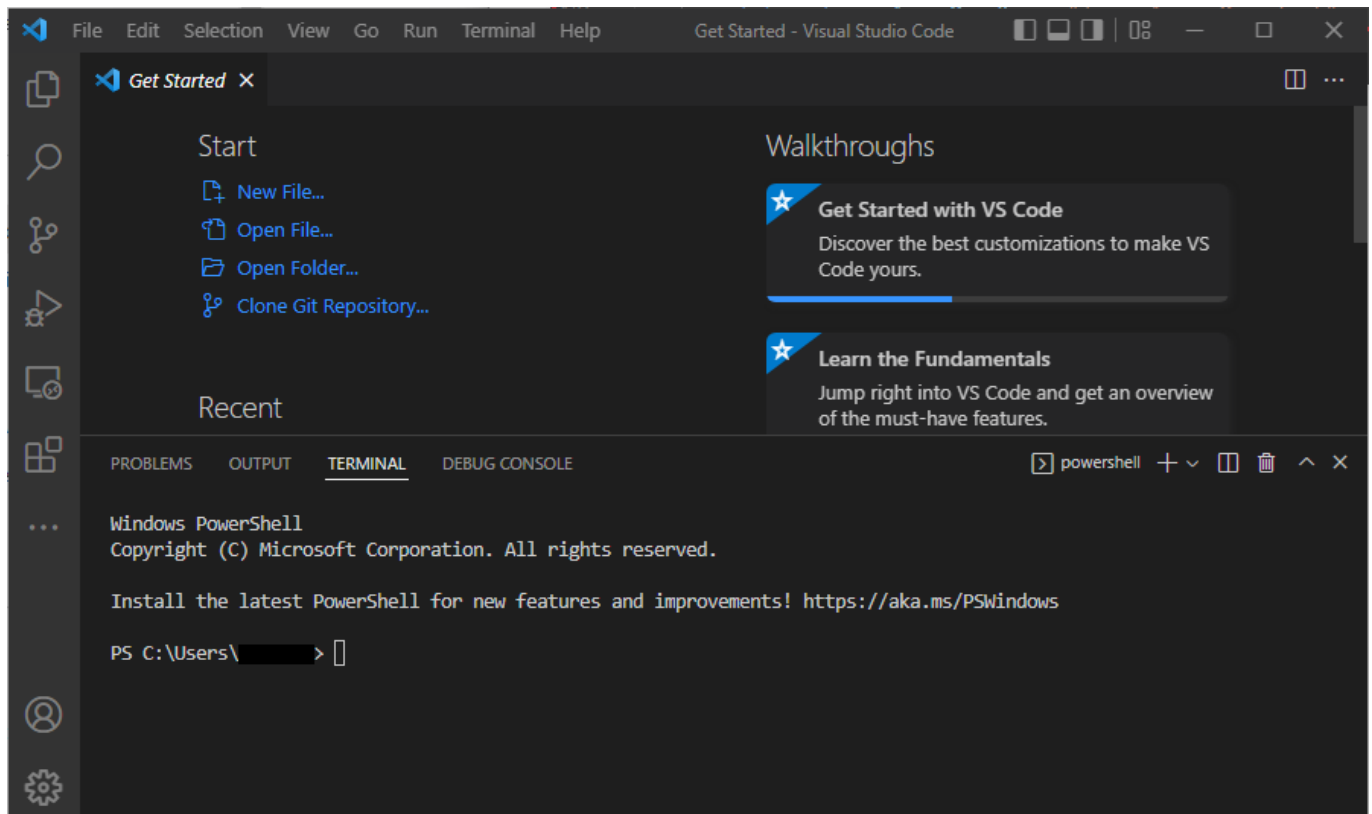
Primary Key

Secondary Key

实验四: 创建设备应用

任务一: 创建你的开发环境

1. 在本地打开 Visual Studio Code
2. 在顶部栏上选择 **终端 - Terminal**，然后在 Visual Studio Code 中选择 **新的终端 - New Terminal**。
3. 请确保您位于要创建项目的本地目录中。（通过 `cd ..` 在本地修改目录位置）



4. 运行以下命令创建目录，设置 dotnet 环境，并安装所需的库：

```
mkdir RefrigeratedTruck
cd RefrigeratedTruck
dotnet new console
dotnet restore
dotnet add package AzureMapsRestToolkit
dotnet add package Microsoft.Azure.Devices.Client
dotnet add package Microsoft.Azure.Devices.Provisioning.Client
dotnet add package Microsoft.Azure.Devices.Provisioning.Transport.Mqtt
dotnet add package System.Text.Json
```

5. 从“文件”菜单中，打开刚刚创建的 Program.cs 文件。在 github repo month 1/day 1 有一个名为 Code-Sample 的文件夹：从 https://github.com/AzureIoTGBB/iot-academy/tree/main/Month_1/Day_1/Code_sample 的 Program.cs 文件中复制此内容并粘贴到您的 Program.cs 文件中。。
6. 换文件内容后，我们需要添加密钥以连接我们的服务。查找行 **123** 到 **126**。根据您在之前的动手实验中添加到记事本的键进行相应的替换。


```
119     static string GlobalDeviceEndpoint = "global.azure-devices-provisioning.net";
120     static TwinCollection reportedProperties = new TwinCollection();
121
122     // User IDs.
123     static string ScopeID = "<your Scope ID>";
124     static string DeviceID = "<your Device ID>";
125     static string PrimaryKey = "<your device Primary Key>";
126     static string AzureMapsKey = "<your Azure Maps key>";
127
128     static double Degrees2Radians(double deg)
```

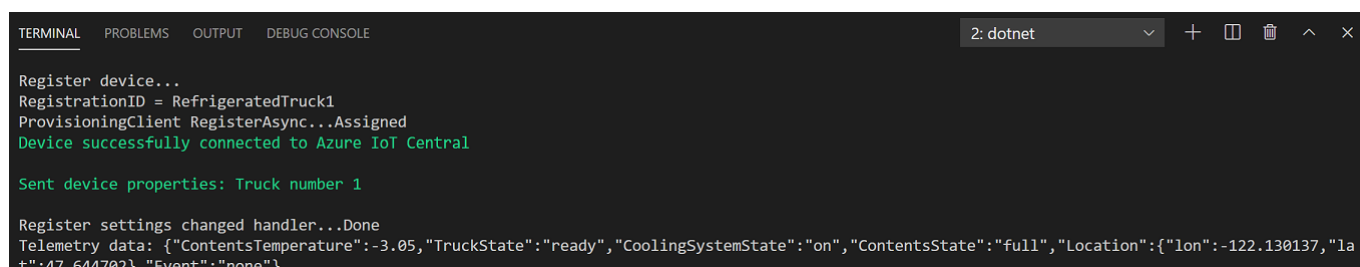
进行更改后，记得保存文件。点击文件 - 保存 - File - Save

任务二: 启动你的设备

1. 要开始测试，首先在浏览器中打开 Azure IoT Central 应用程序：<https://app.azureiotcentral.com/> 点击左侧的我的应用 - **My apps**。单击 **Refrigerated Truck IoT** 磁贴。
2. 在 VS Code 终端中，使用以下命令启动设备应用程序：

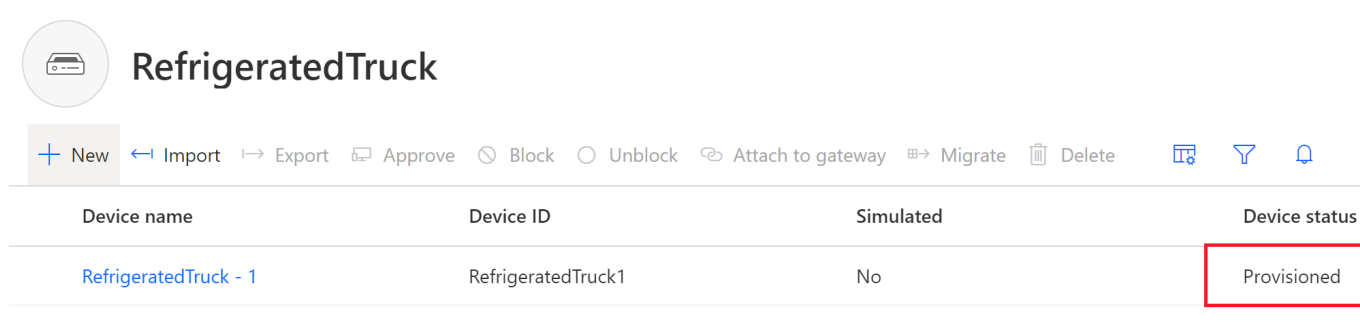
```
dotnet run
```

将打开一个控制台屏幕，其中显示消息“Starting Truck number 1”



```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE  2: dotnet
Register device...
RegistrationID = RefrigeratedTruck1
ProvisioningClient RegisterAsync...Assigned
Device successfully connected to Azure IoT Central
Sent device properties: Truck number 1
Register settings changed handler...Done
Telemetry data: {"ContentsTemperature":-3.05,"TruckState":"ready","CoolingSystemState":"on","ContentsState":"full","Location":{"lon":-122.130137,"lat":47.644702},"Event":"none"}
```

通过 VS Code 注册您的设备后，您应该会在 IoT Central 中看到状态更改为 **Provisioned**：



RefrigeratedTruck			
+ New ← Import → Export Approve Block Unblock Attach to gateway Migrate Delete			
Device name	Device ID	Simulated	Device status
RefrigeratedTruck - 1	RefrigeratedTruck1	No	Provisioned

此时，在 IoT Central 的 Truck View 仪表板中，您应该会看到通过它的数数据，地图上应该也会显示一个蓝点，并且接收遥测数据的图表并显示一些数据点。

3. 选择设备的 **指令 - Commands** 选项卡。此控件应位于卡车名称下，卡车视图控件的右侧。
4. 输入客户 ID，例如 **1**。（数字 0 到 9 是有效的客户 ID。）然后选择 **运行 - Run**。

在设备应用程序的控制台中，您会看到新的客户事件和找到路由消息

```
Telemetry sent 3:02 PM

Telemetry data: {"ContentsTemperature":12,"TruckState":"ready","Co
,"lat":47.644702},"Event":"none"}
Telemetry sent 3:02 PM

Route found. Number of points = 673

Telemetry data: {"ContentsTemperature":12,"TruckState":"enroute",
0048344247,"lat":47.64604},"Event":"New customer: 1"}
Telemetry sent 3:02 PM

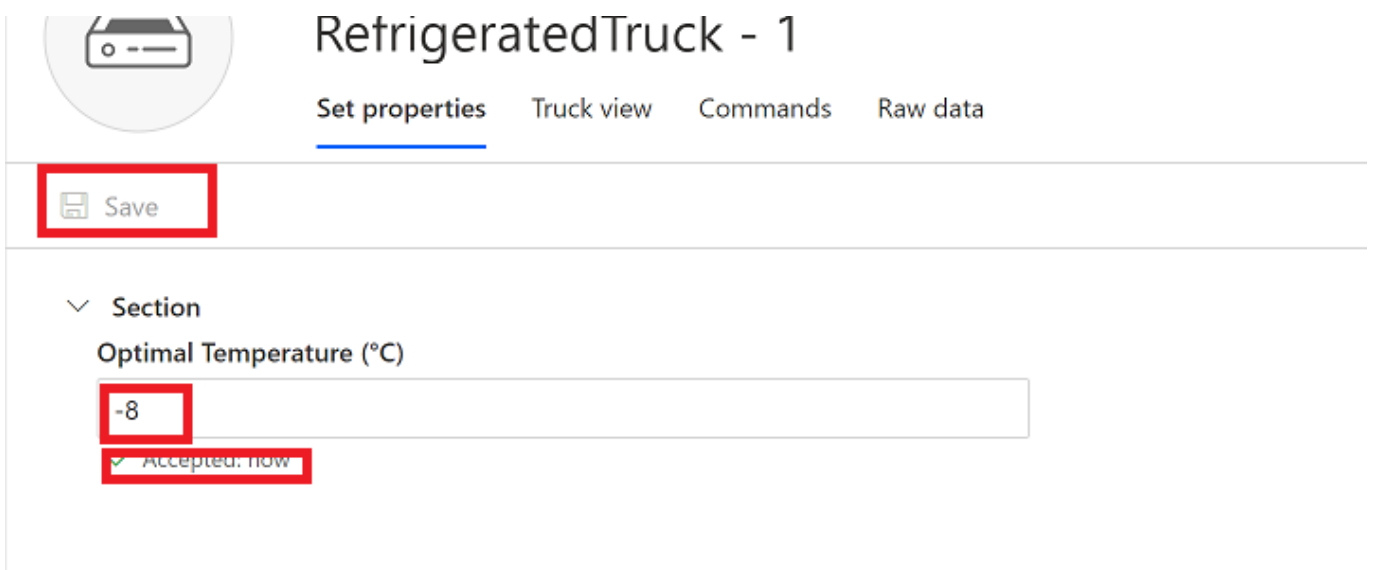
Telemetry data: {"ContentsTemperature":12,"TruckState":"enroute",
0997706499,"lat":47.641170152900116},"Event":"none"}
Telemetry sent 3:02 PM
```

5. 在仪表板的位置磁铁上，检查您的卡车是否在路上。可能需要等待一小段时间才能同步应用。
6. 通过验证事件磁贴上的事件文本，您应该会看到一个新的客户事件。
7. 当卡车返回基地并重新装载内容时，其状态准备就绪。尝试发出另一个交付命令。选择另一个客户 ID。
8. 在卡车到达客户之前，发出召回命令，检查卡车是否响应。

任务三: 设置特性

下一个测试是检查可写属性 **OptimalTemperature**。要更改此值，请选择 **选择特性 - Set properties** 视图。

将最佳温度设置为任何值，例如 **** -8 ****。选择 **Save**，然后注意 Pending 状态。

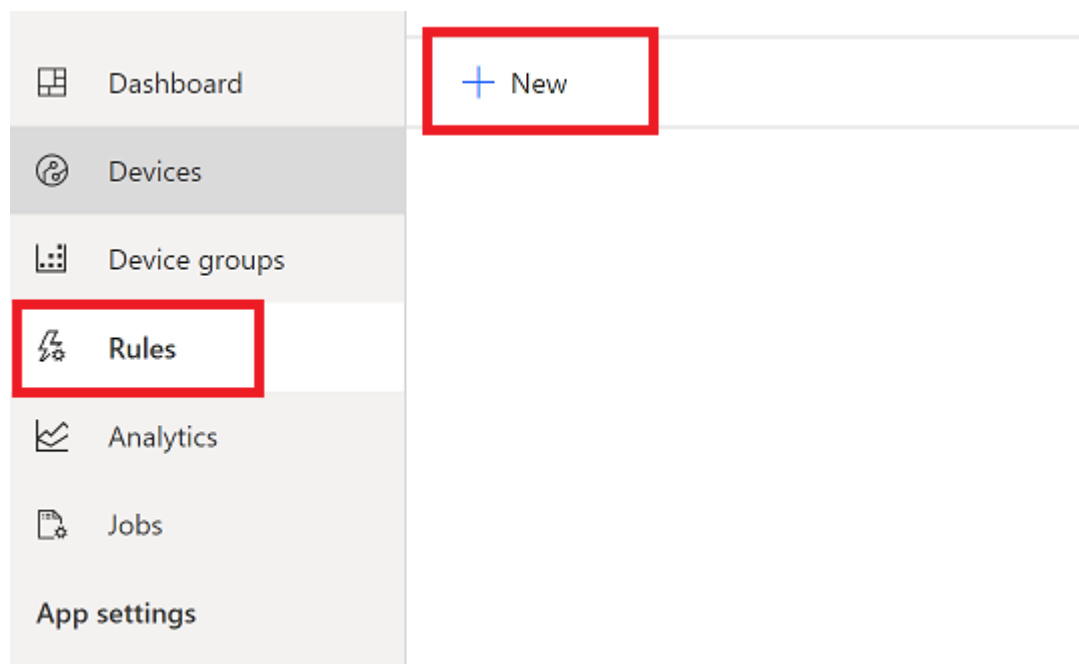


现在您应该看到新的最佳温度设置为 -8。在 **最佳温度** 磁贴中。

实验五: 创建规则

实验一: 设置冷却系统状态规则

1. 在 IoT 门户中, 选择左侧菜单中的 **规则 - Rules**, 然后选择 **+ 新建 - New**。为规则输入一个有意义的名称, 例如 **"Cooling system failed"**。按 Enter。



2. 为 ****设备模板 - device template**** 选择 **RefrigeratedTruck**。

3. 在条件 - **Conditions** 下, 请注意设备模板的所有遥测元素都可用。选择 **Cooling system state**。

对于运算符, 选择 **等于**。

对于值, 键入单词 **"failed"**, 然后单击 **Select:"failed"**。

将时间聚合设置为关闭。

对于 **动作 - Actions**, 点击 **+ 邮件 - + Email**。

在显示名称中, 输入电子邮件的标题, 可能是**"冷却系统失败! "**

对于收件人, 输入您用于 IoT Central 帐户的电子邮件。并为内容输入一些描述性文本, 这些文本将构成电子邮件的正文。

注意: 要接收电子邮件, 您选择的帐户必须至少登录 IoT Central 一次, 否则您将不会收到任何电子邮件。

您的新规则应如下图所示。

Save Cancel Rename

Rules > Cooling system failed Alert

Cooling system failed Alert

Enabled

Target devices

Select the device template your rule will use. If you need to narrow the rule's scope, add filters.

Device template *
RefrigeratedTruck

+ Filter

Conditions

Conditions define when your rule is triggered. Aggregation is optional—use it to cluster your data and trigger rules based on a time window.

Time aggregation
Off Select a time window

Telemetry *	Operator *	Value *
Cooling system state	Equals	failed

+ Condition

Actions

Choose what action your rule should take.

Email: Cooling system failed

实验二: 设置温度飙升规则

1. 创建一个新规则，名称如**"Contents temperature spiking"**。
2. 开启时间聚合，选择5分钟的时间间隔。
3. 为遥测选择内容温度。
4. 在范围聚合值中，选择最大值。
5. 对于操作员。选择大于或等于。然后输入**"0"**选择它作为值。
6. 对于操作，发送另一封电子邮件。给电子邮件一个适当的标题和注释。
7. 确保单击保存，以保存此规则。

Save Cancel Rename

Rules > Contents temperature spiking

Contents temperature spiking

Enabled

Target devices

Select the device template your rule will use. If you need to narrow the rule's scope, add filters.

Device template *
RefrigeratedTruck

+ Filter

Conditions

Conditions define when your rule is triggered. Aggregation is optional—use it to cluster your data and trigger rules based on a time window.

Trigger the rule if all of the conditions are true

Time aggregation
On 5 minutes

Telemetry * Operator * Aggregation *

Contents temperature is greater than or equal to Maximum

Enter a value Select a value

Value *
0

+ Condition

实验三: 设置卡车离开货运中心规则

1. 在左侧菜单中选择 **规则 - Rules**，然后选择 **+新建 - + New**。为规则输入一个有意义的名称，例如 **"Truck leave base"**。按 Enter。

现在，输入以下条件。

- 位置/纬度：不等于 => **47.644702**
- 位置/经度：不等于 => **-122.130137**
- 卡车状态：等于 => **途中**

2. 再次输入适当的 **email action**，然后单击 **保存 - Save**。

实验四: 设置运输物品的温度规则

1. 输入一个名称如 ****"Truck contents OK"***的规则。
2. 开启时间聚合，时间为**五分钟**。
3. 如果平均内容物温度低于 **-1** 摄氏度且高于 **-18** 摄氏度，则输入触发条件。
4. 再次输入适当的 **email action**，然后单击 **Save**。

此时您应该会看到所有规则，如下所示：

Rules	
+ New	
Name	Status
Cooling system failed Alert	Enabled
Contents temperature spiking	Enabled
Truck leaving base	Enabled
Truck contents OK	Enabled

是时候测试一下您定义的规则了 转到设备仪表板，发送命令以触发新的客户行程，记住使用 1 到 9 的数字。几分钟后，您应该开始接收电子邮件。

注意：要接收电子邮件，您选择的帐户必须至少登录 IoT Central 一次，否则您将不会收到任何电子邮件。

实验六: 清空资源

****完成所有实验后，转到 Azure 门户，查找 Azure IoT Central 应用程序并删除资源。**