InGateway Documentation

发布 *0.0.1*

zhangning

2023 年 05 月 06 日

用户手册

CHAPTER 1

InGateway 文档网站导航

InGateway902 系列边缘计算网关支持托管 docker 镜像，您可以将您的 docker 镜像发布到 IG902 上，快速部署和运行您自行开发的应用程序。

Docker 是一个开源的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的容器中, 然后发布到任何流行的 Linux 机器或 Windows 机器上, 也可以实现虚拟化, 容器是完全使用沙箱机制, 相互之间不会有任何接口。

1.1 InGateway902/974 Docker 用户手册

InGateway902/974 系列边缘计算网关（以下简称网关）支持托管 docker 镜像，您可以将您的 docker 镜像发布到网关上，快速部署和运行您自行开发的应用程序。为了说明如何使用网关的 Docker 环境，本文档将演示如何在网关上运行一个 Nginx 镜像，该镜像用于 HTTP，HTTPS，SMTP，POP3 和 IMAP 协议的开源反向代理服务器，以及负载平衡器，HTTP 缓存和 Web 服务器。Docker 是一个开源的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的容器中, 然后发布到任何流行的 Linux 机器或 Windows 机器上, 也可以实现虚拟化, 容器是完全使用沙箱机制, 相互之间不会有任何接口。

- *1.* 准备网关硬件设备及其网络环境
  - *1.1* 接通网关电源并使用网线连接 *PC*
  - *1.2* 访问网关
  - *1.3* 网关连接 *Internet*
  - *1.4* 更新网关固件版本

1.1.1 1. 准备网关硬件设备及其网络环境

1.1 接通网关电源并使用网线连接 PC

- 接通 IG902 的电源并按照拓扑使用以太网线连接 PC 和 IG902。

- 接通 IG974 的电源并按照拓扑使用以太网线连接 PC 和 IG974。

1.2 访问网关

- 访问 IG902，请参考访问 IG902。
- 访问 IG974，请参考访问 IG974。

1.3 网关连接 Internet

- 设置 IG902 联网，请参考IG902 连接 Internet。
- 设置 IG974 联网，请参考IG974 连接 Internet。

1.4 更新网关固件版本

- 如需获取 IG902 产品最新固件版本及其功能特性信息，请访问资源中心。如需更新 IG902 的固件版本，请参考更新 IG902 软件版本。（固件版本应为 2.0.0.r12057 及以上）
- 如需获取 IG974 产品最新固件版本及其功能特性信息，请访问资源中心。如需更新 IG974 的固件版本，请参考更新 IG974 软件版本。（固件版本应为 2.0.0.r14169 及以上）

1.1.2 2. 启用并配置 Docker 管理器

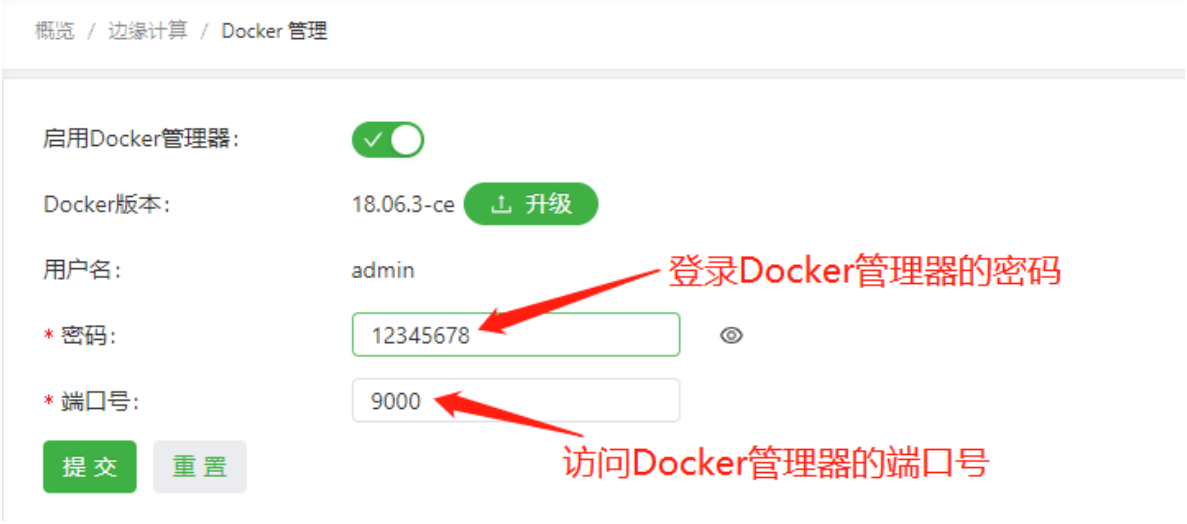
2.1 安装 Docker SDK 并启用 Docker 管理器

Docker SDK 集成了运行 docker 镜像所需的运行环境以及 docker 镜像管理器，在使用 Docker 前必须先安装 Docker SDK，你可以访问资源中心获取软件版本。

- 步骤 1：已有 Docker SDK 后，进入网关的 “边缘计算 >>Docker 管理” 页面，关闭 Docker 管理器并导入 Docker SDK。

- 步骤 2：导入后，网关将自动安装 Docker SDK，安装过程通常需要 1-2 分钟，请耐心等候。安装成功后，勾选启用 Docker 管理器并点击“提交”。

- 步骤 3：启用 Docker 管理器后，可以修改访问 Docker 管理器的端口号和登录密码。

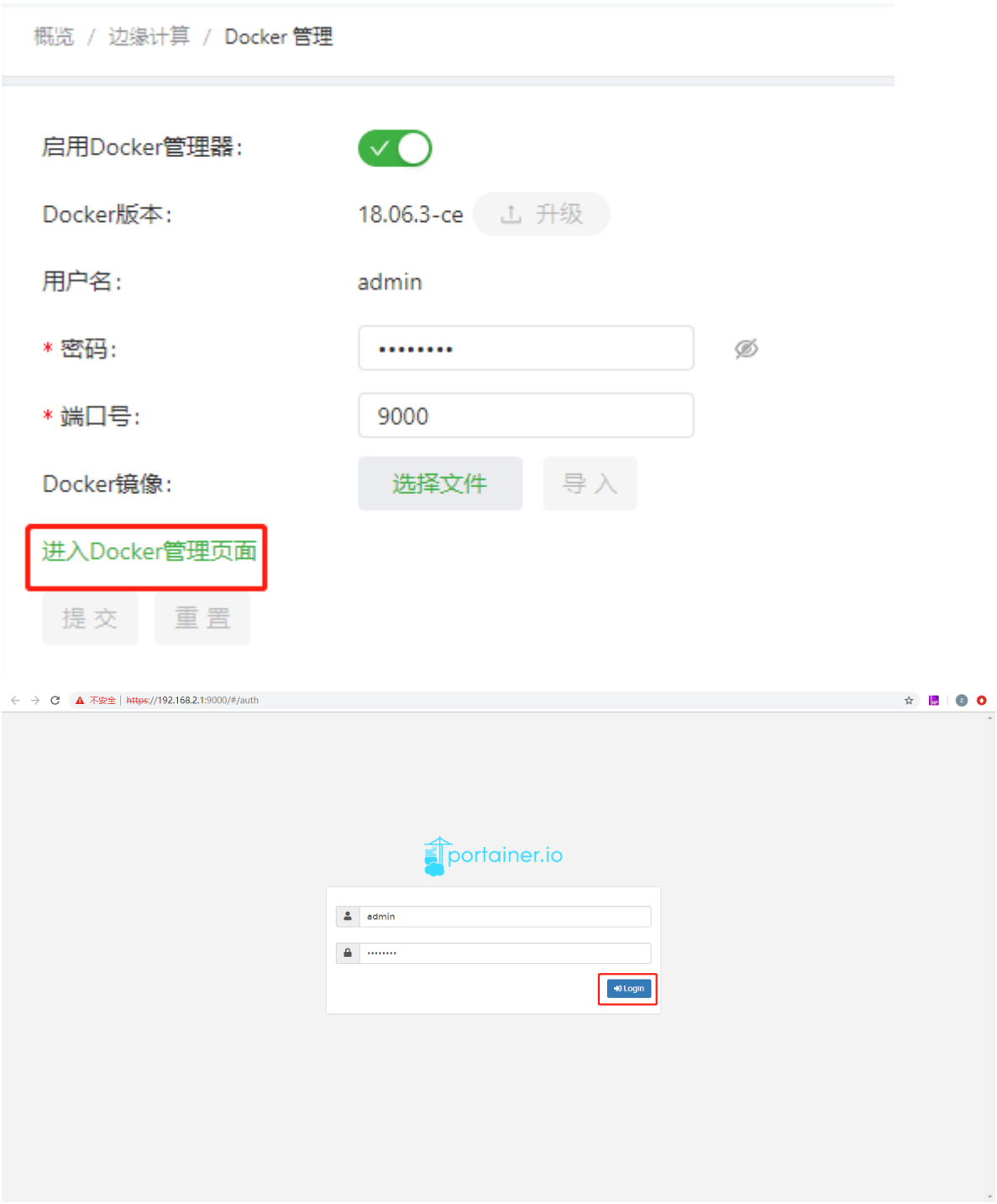


2.2 配置 Docker 管理器–Portainer

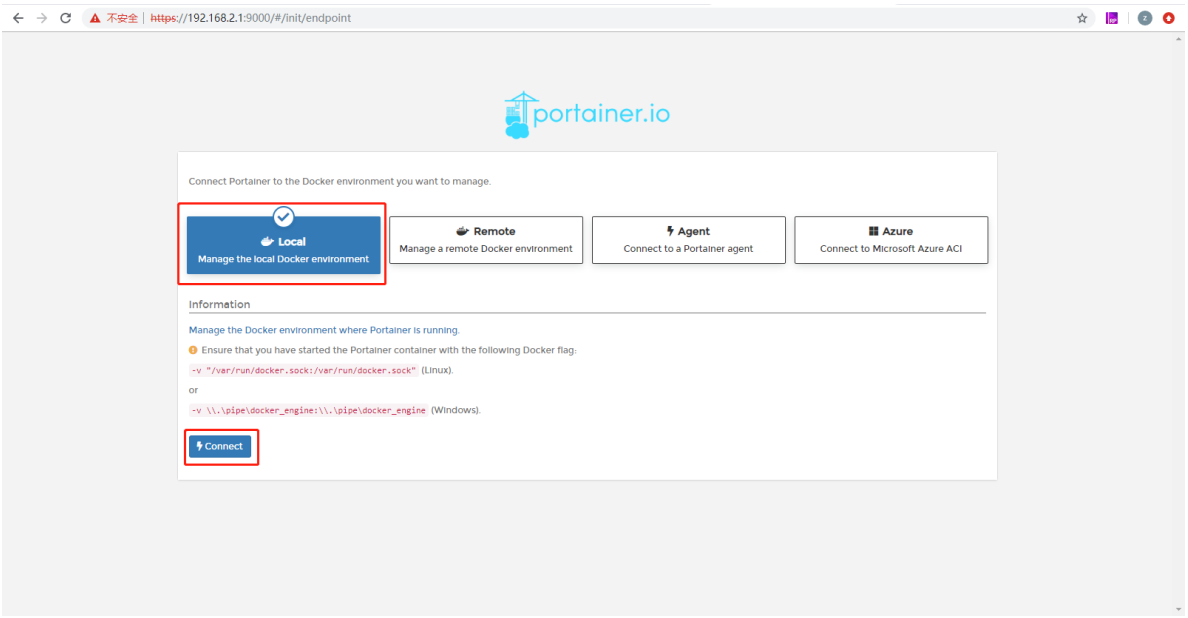
网关使用 Portainer 构建，管理和维护 Docker 镜像和容器。关于 Portainer 的详细介绍和使用说明请查看Portainer 官网。本文档将为您演示如何在网关上添加并部署运行一个 Nginx docker 镜像。

2.2.1 访问 Portainer

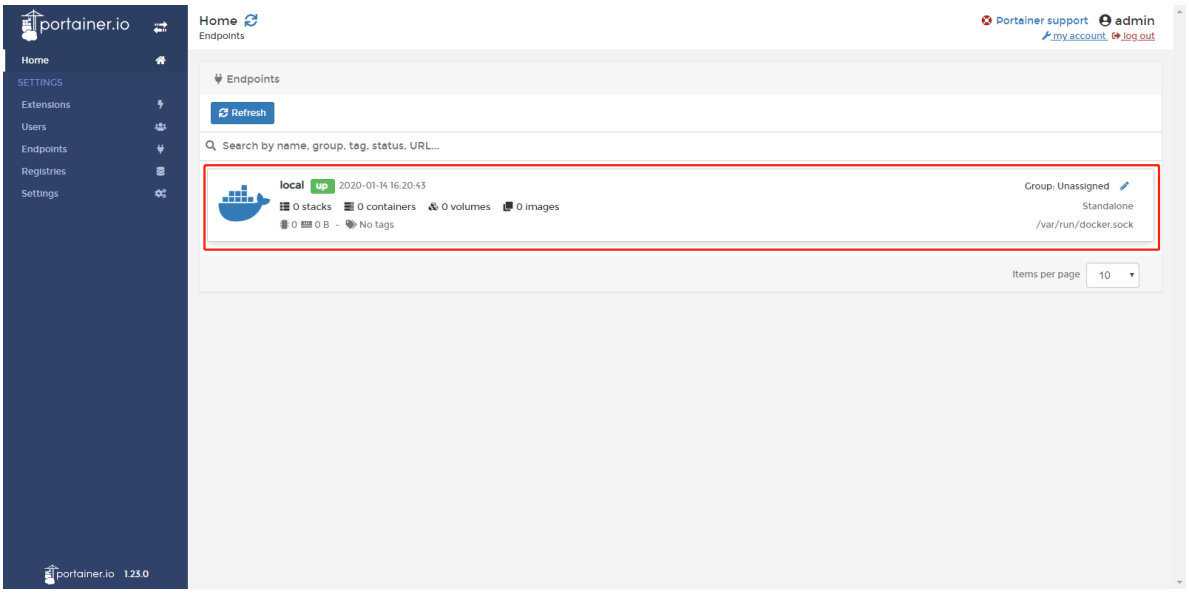
- 步骤 1：点击 Portainer 的访问按钮，随后 Portainer 会提示您需要输入用户名和密码。此时从网关的“边缘计算 >>Docker 管理”页面复制用户名和设置的密码后并点击“登录”即可。



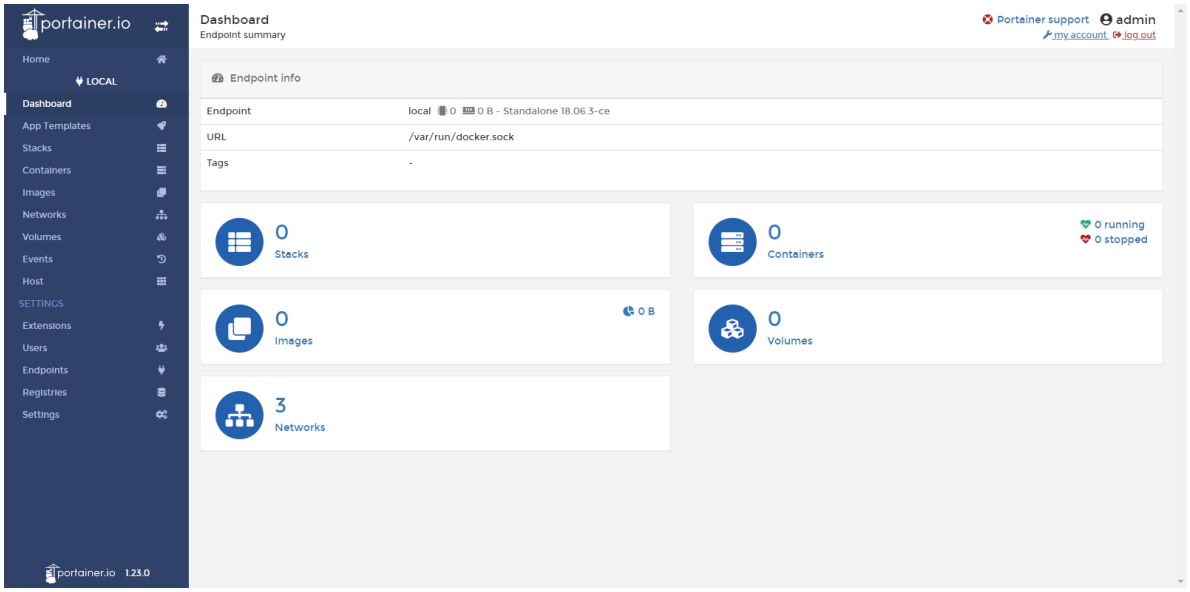
- 步骤 2：登录成功后如下图所示，选择“Local”以使用 Portainer 管理网关上的 docker 镜像，随后点击“Connect”。



- 步骤 3：在 Portainer 的“Home”页面，选择 local 以管理网关上的 docker 镜像。



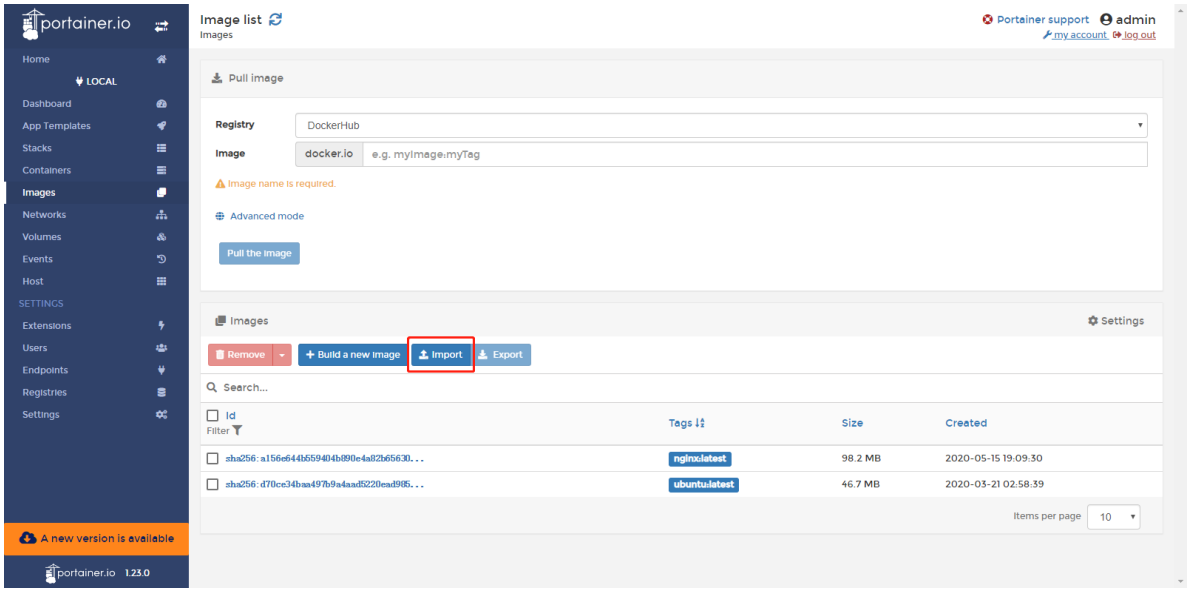
随后会跳转至本地仪表板，可在此页面概览网关的容器和镜像等信息。



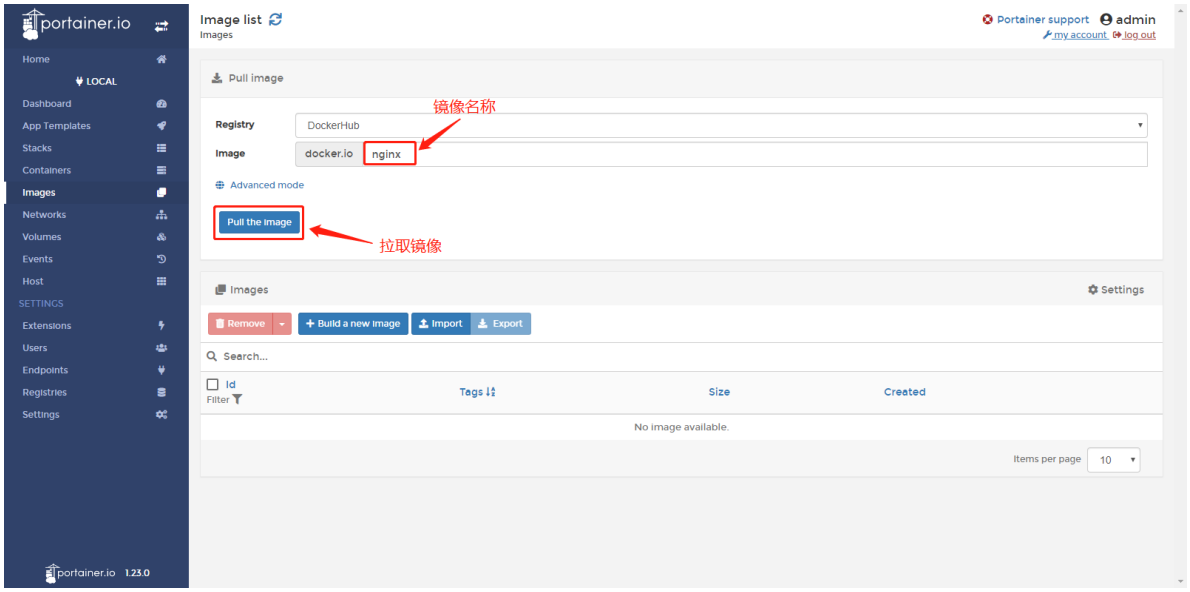
2.2.2 添加 docker 镜像

为 Portainer 添加 docker 镜像的方法有两种：

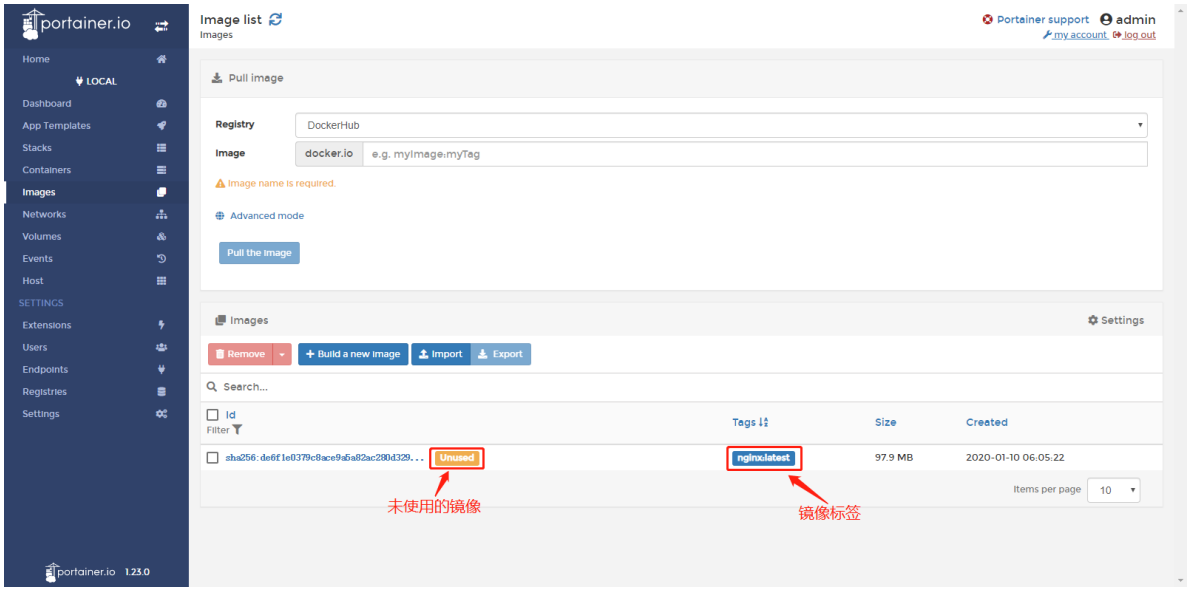
- 方法 1：进入 Portainer 的 “Local>>Images” 页面，点击 “Import” 导入镜像



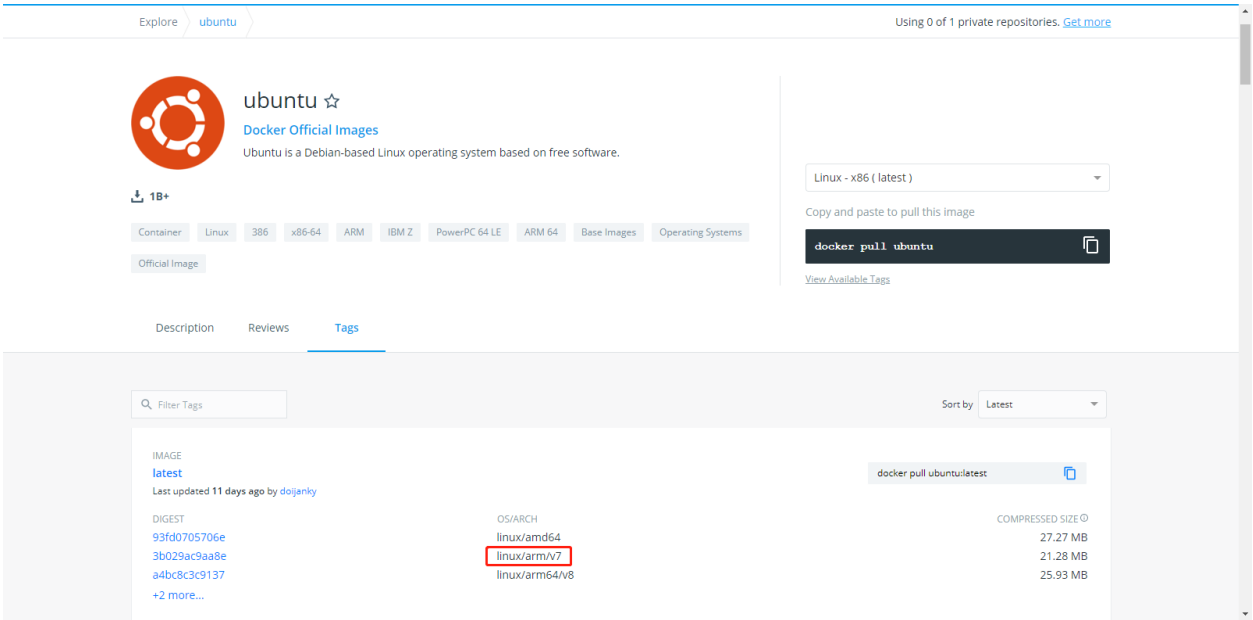
- 方法 2：进入 Portainer 的 “Local>>Images” 页面，从 DockerHub 中下载 “nginx” docker 镜像。（下载镜像所需时间根据镜像大小而不同；当 docker 镜像较大时，请耐心等待）



docker 镜像下载完成后如下图所示，在 Portainer 的 “Local>>Images” 中能够看到相应的 docker 镜像信息。

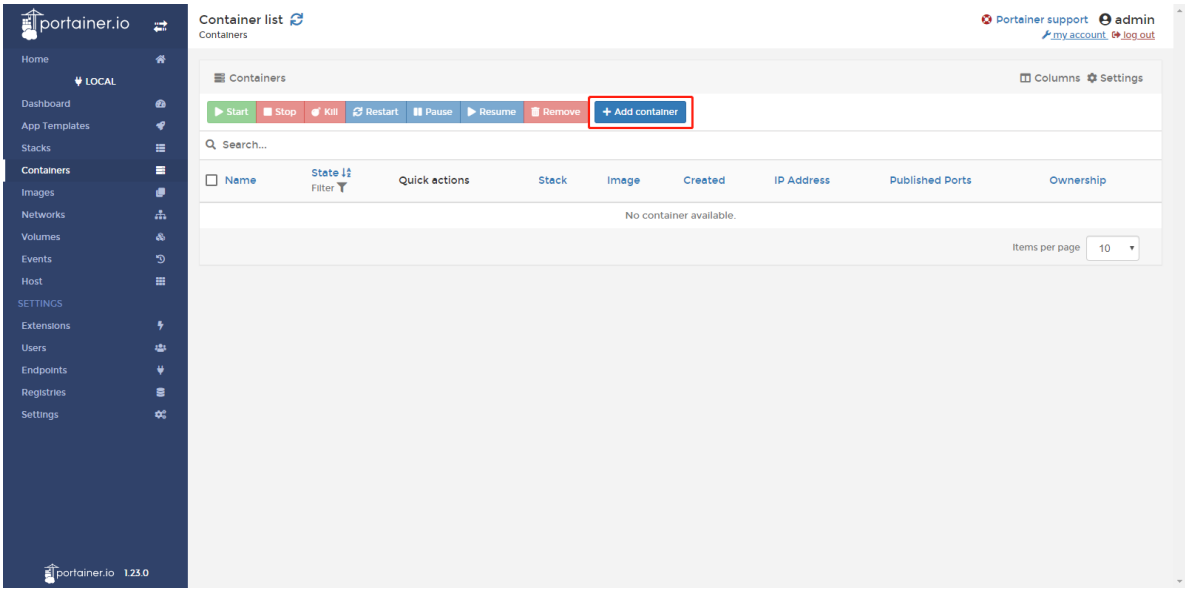


注意：因为网关的 CPU 架构为 linux/arm/v7，因此只有支持 linux/arm/v7 架构的镜像可以正常在网关中运行，其他如 window/amd64 等架构的镜像可能无法正常导入、拉取或在网关中运行。

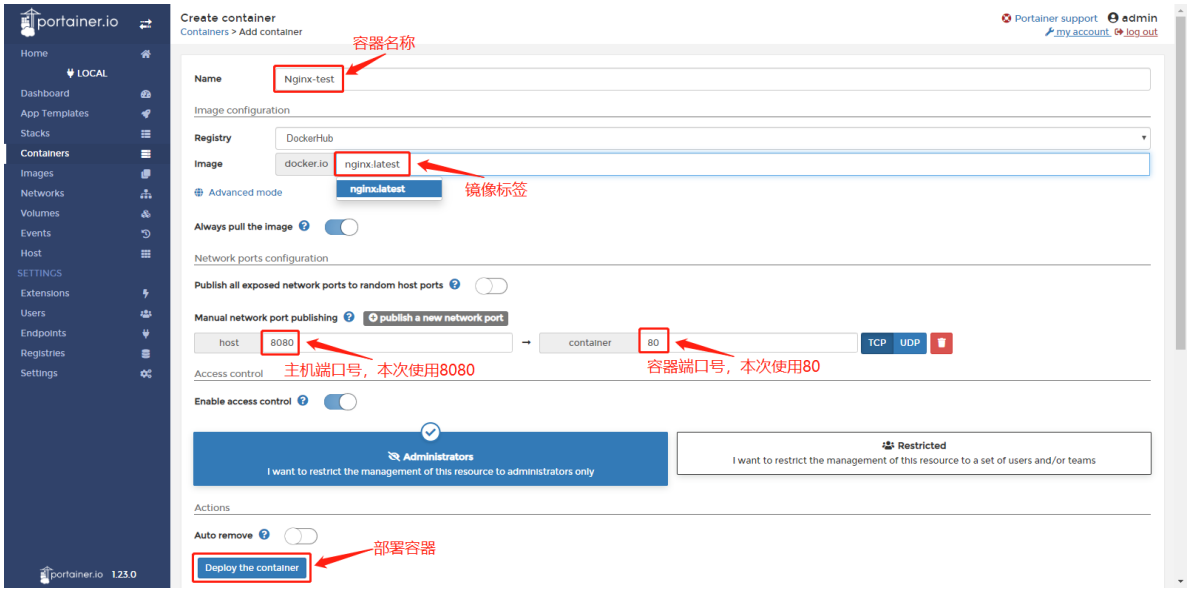


2.2.3 配置并部署容器

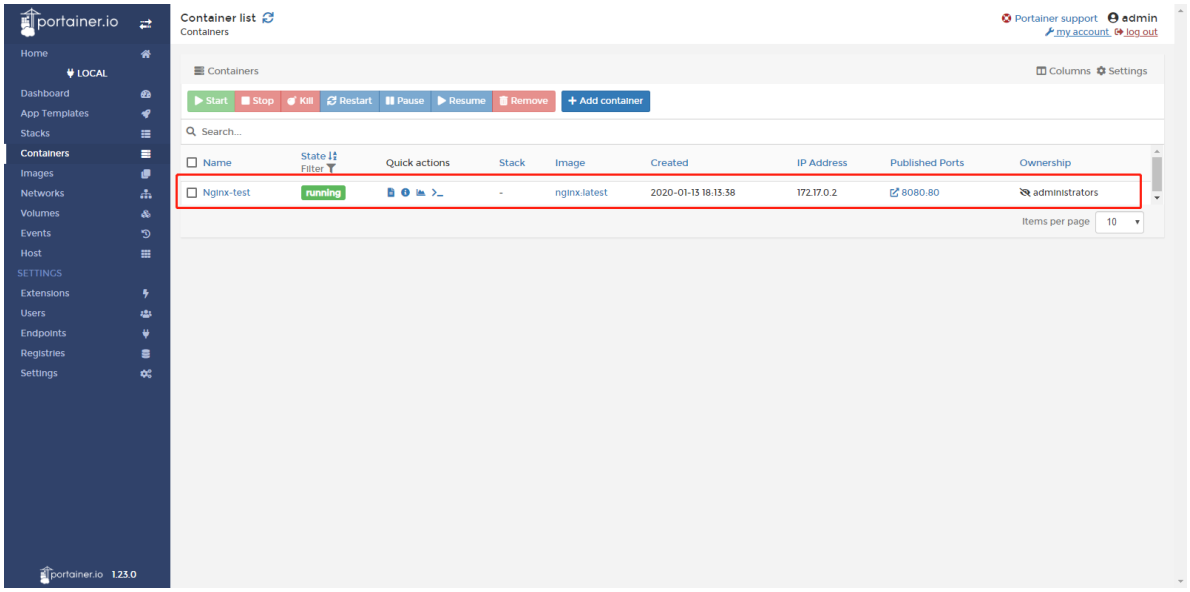
- 步骤 1：进入 Portainer 的“Local>>Containers”页面，点击“Add container”以添加一个新容器。



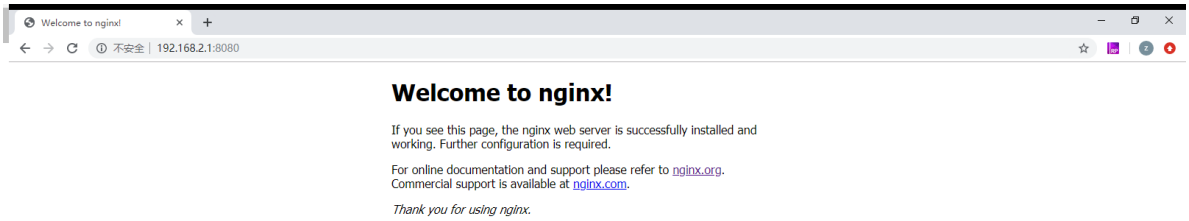
- 步骤 2：为容器配置运行参数并部署容器。



- 步骤 3：部署后容器会自动运行，在 Portainer 的“Local>>Containers”页面可以查看容器运行情况。



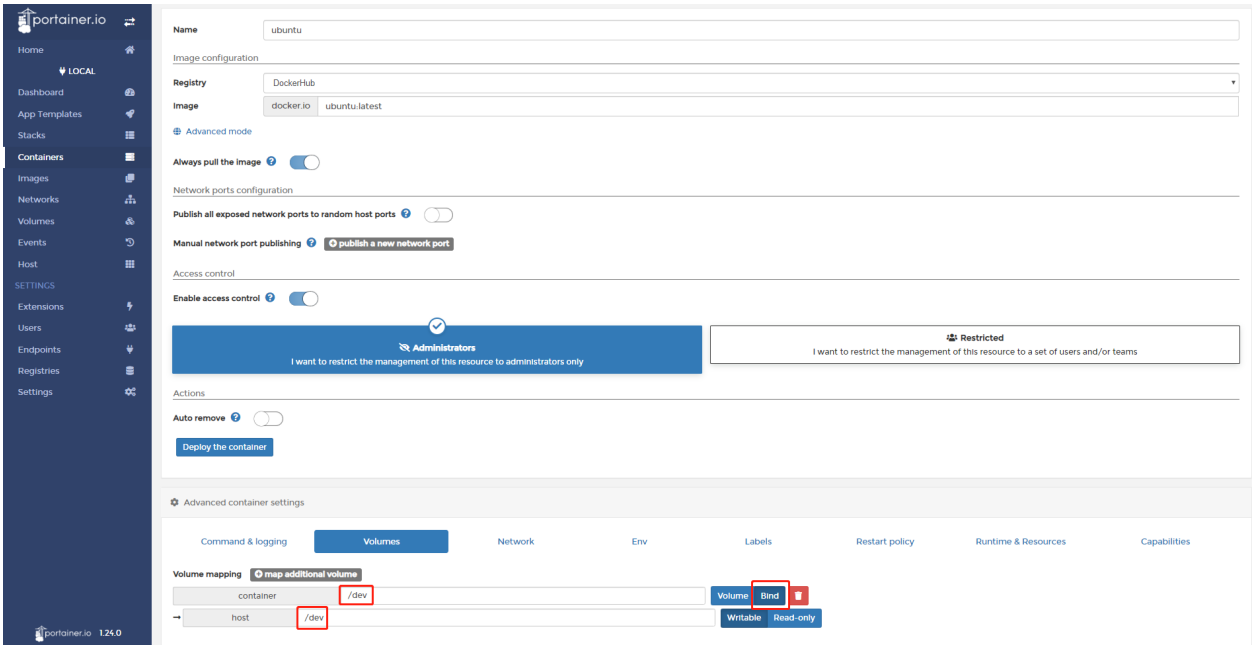
- 步骤 4：在浏览器中输入容器中配置的 Nginx 访问链接（网关的 IP 地址 + 端口号）后可以看到 Nginx 的欢迎页面。说明 Nginx docker 镜像已正常运行在网关上；至此，完成了在网关上添加并部署运行一个 Nginx docker 镜像。



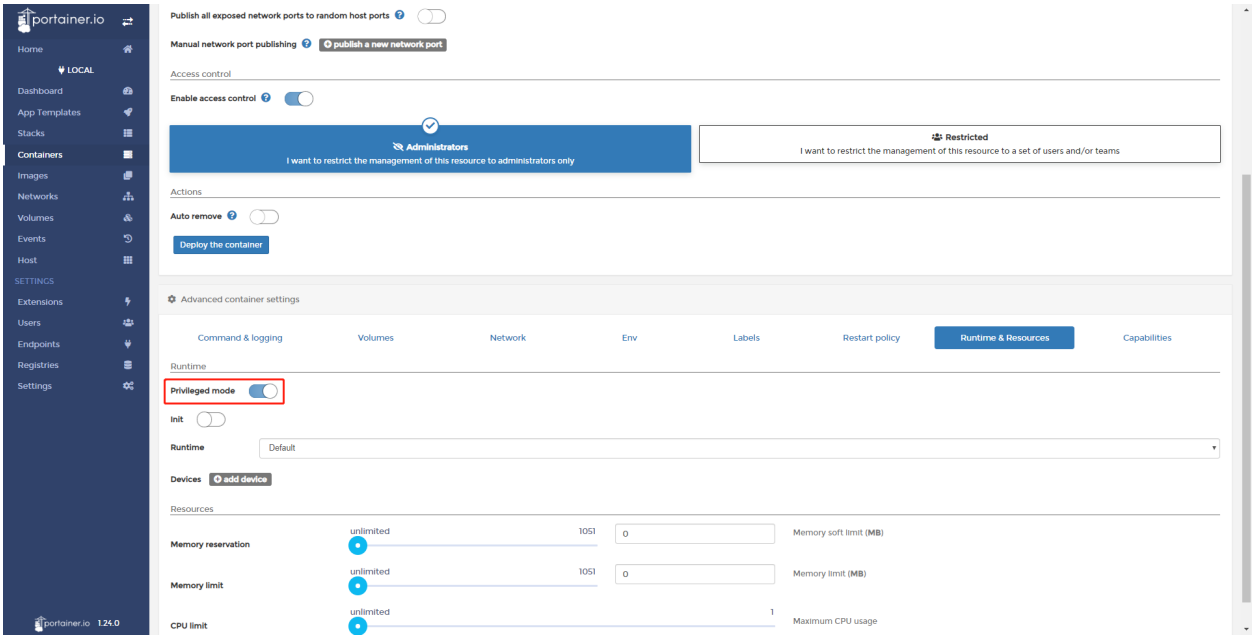
1.1.3 附录

在容器中调用串口进行通讯

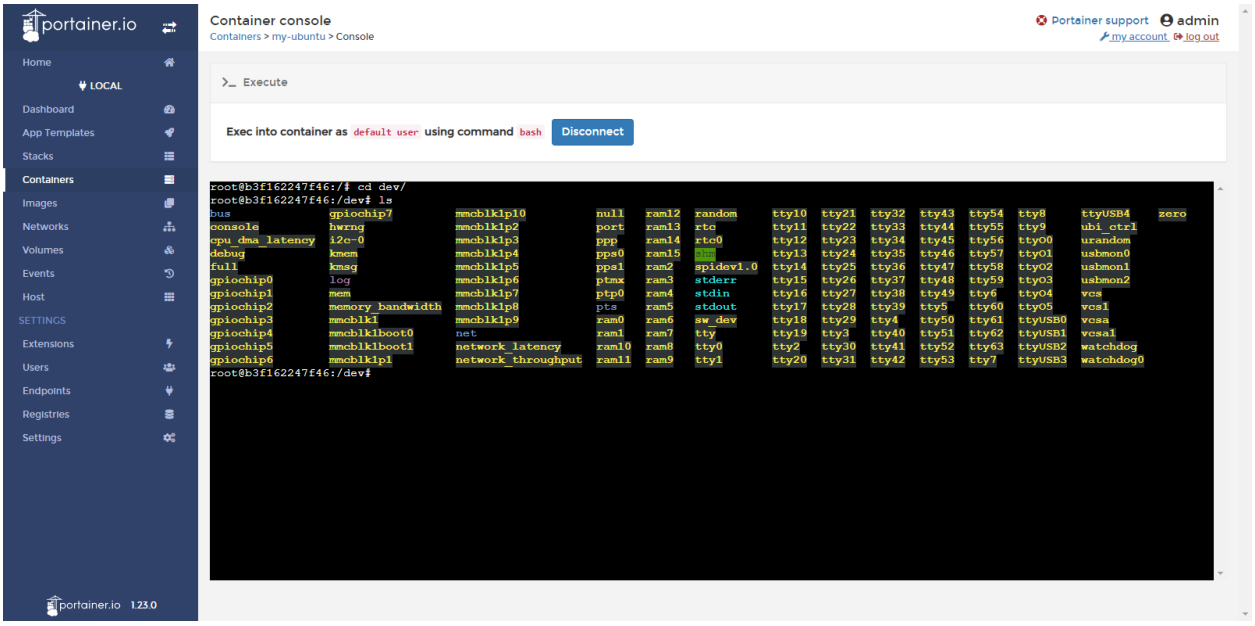
部署容器时，在 Portainer 的“Advanced container settings>>Volumes”页面添加一条“Volume mapping”。下图将网关的 dev 目录中的文件映射到了容器中的 dev 目录下（网关的 dev 目录中包含了相应的接口文件）：



在 Portainer 的“Advanced container settings>>Runtime & Resouces”页面启用“Privileged mode”（未启用时使用串口会提示没有操作权限）

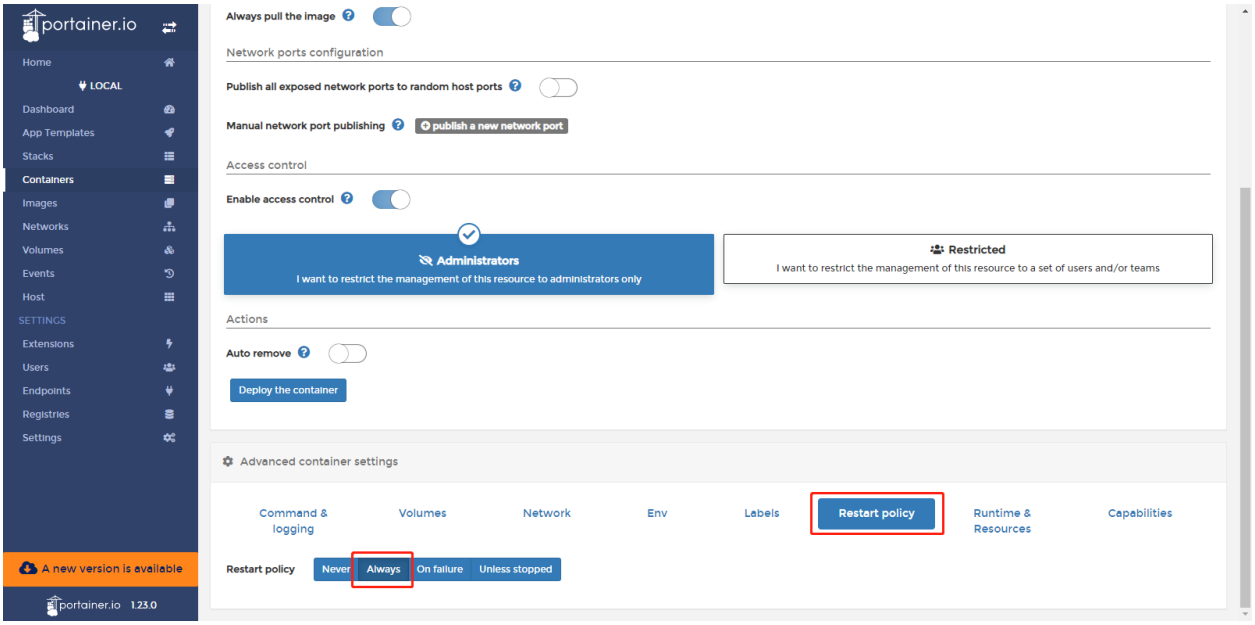


设置完毕后部署容器，随后在容器的 console 中进入 dev 目录可以看到 `ttyO1`，`ttyO3` 等接口文件。



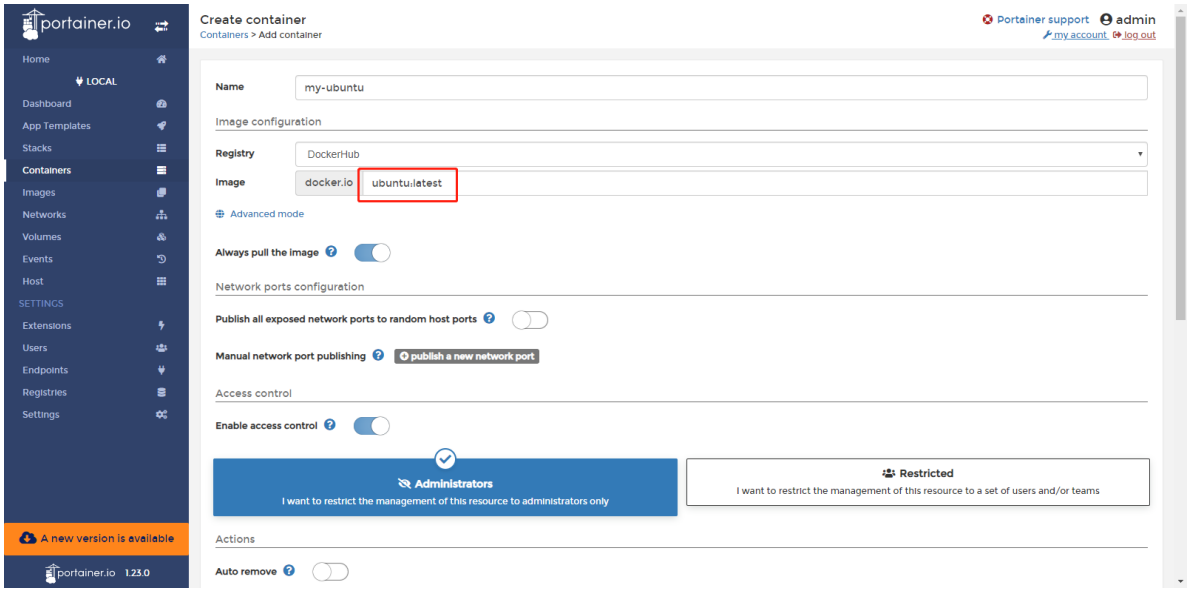
设置容器永久运行

部署容器时，在 Portainer 的 “Advanced container settings>>Restart policy” 页面选择 “Restart policy” 为 “Always”，设置为 “Always” 后容器只要停止运行则会自动重启容器。

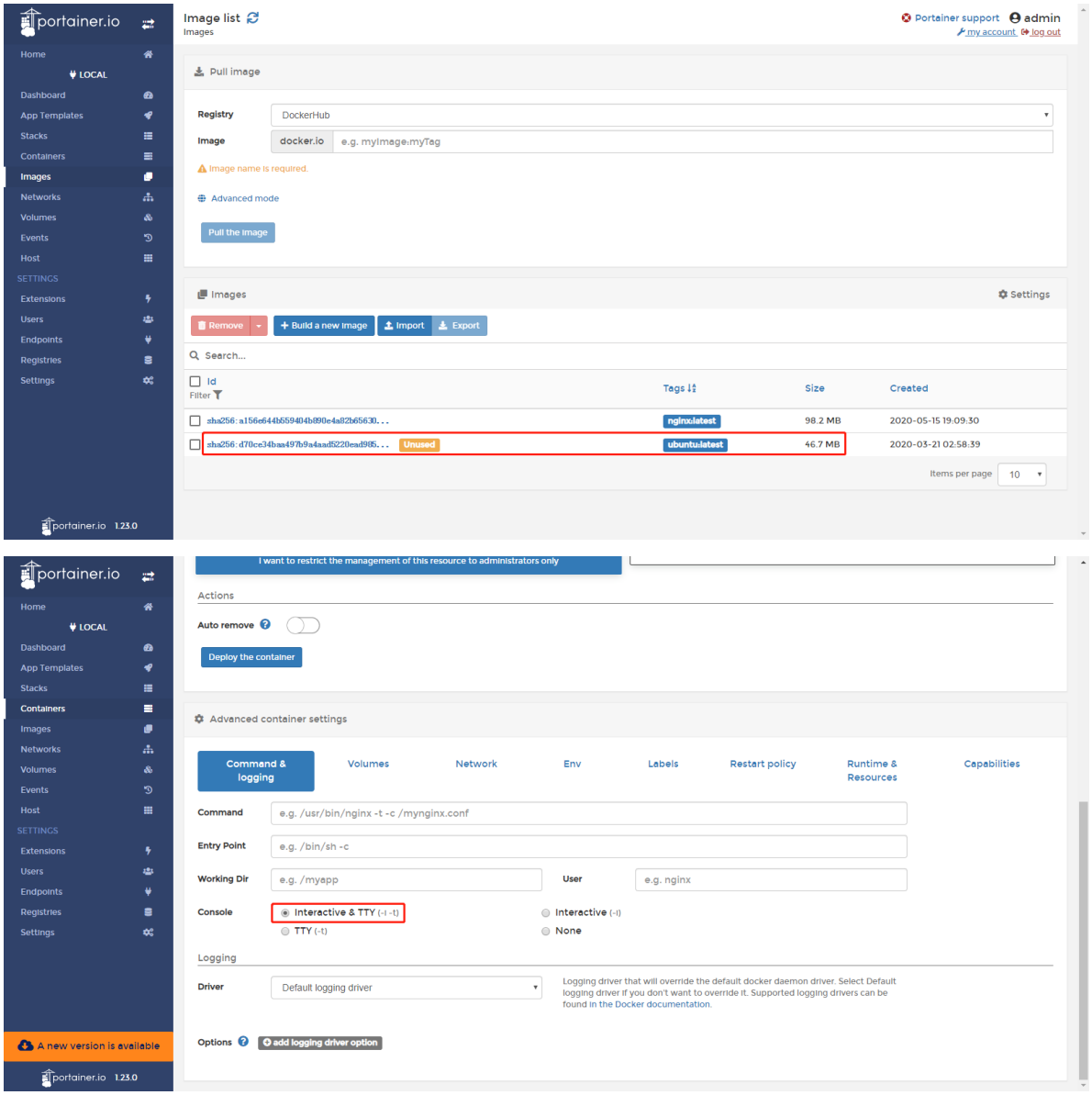


在网关中运行 Ubuntu

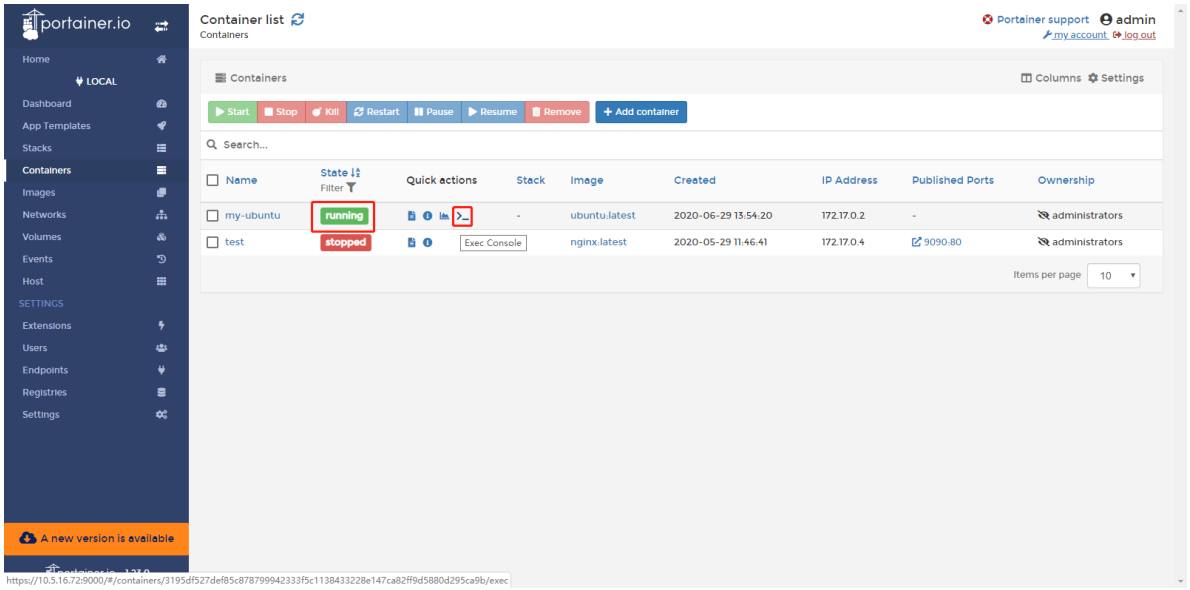
- 步骤 1：在 Portainer 的 “Local>>Images” 页面拉取 Ubuntu 镜像，如下图所示：



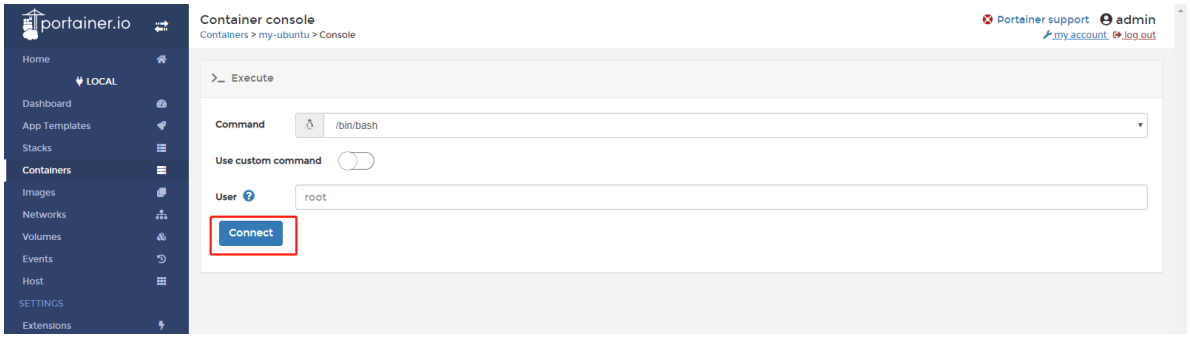
- 步骤 2：进入 Portainer 的 “Local>>Containers” 页面，点击 “Add container” 以添加一个新容器。容器所使用的镜像选择上一步下载的 Ubuntu 镜像，同时在 “Advanced container settings>>Command &logging” 中勾选 “Console” 中的 “Interactive & TTY (-i -t)” 项。配置完成后点击 “Deploy the container” 部署容器。

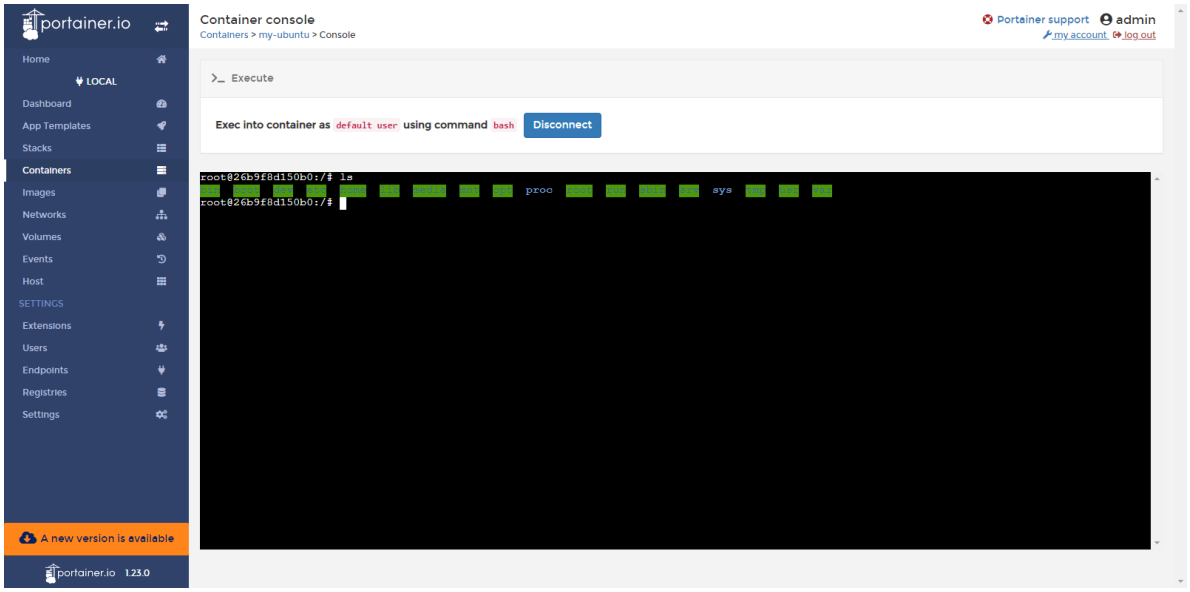


- 步骤 3：部署后在 Portainer 的“Local>>Containers”页面可看到容器已运行，点击“Exec Console”登录控制台。



在 Exceute 中点击“Connect”后进入容器内部运行相应命令。

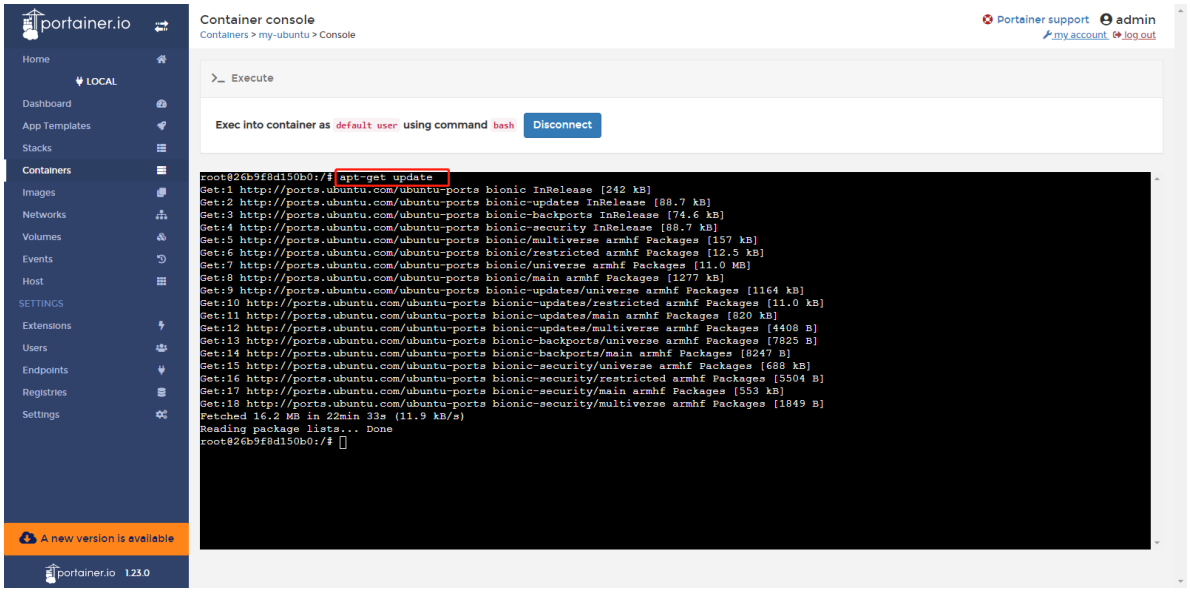


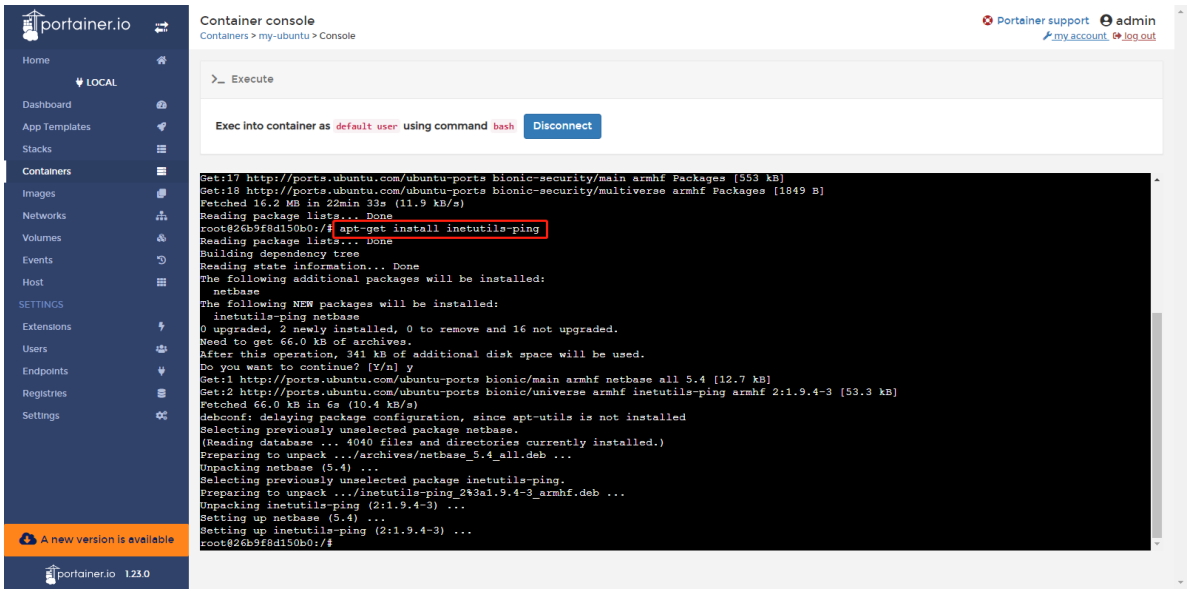


通过容器构建镜像（创建镜像保存容器配置）

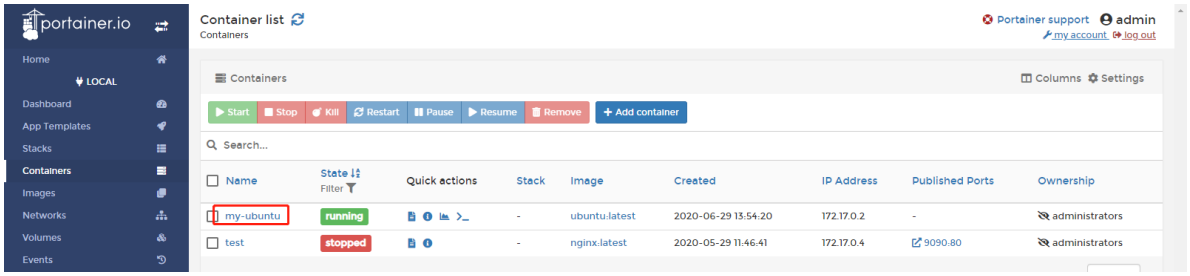
当容器中已配置好相应的开发或运行环境，如果需要保存容器中的环境配置，可以根据容器的更改创建一个新的镜像。方法如下（以 Ubuntu 容器安装 ping 工具为例）：

- 步骤 1：配置容器的开发或运行环境分别运行 `apt-get update` 和 `apt-get install inetutils-ping` 命令安装 ping 工具。

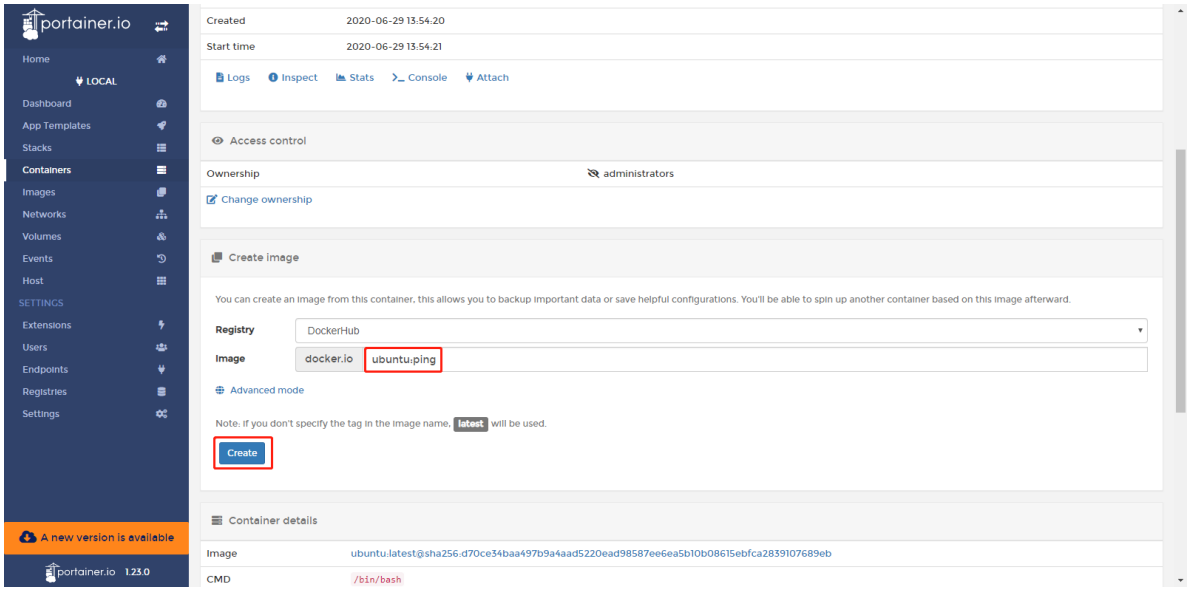




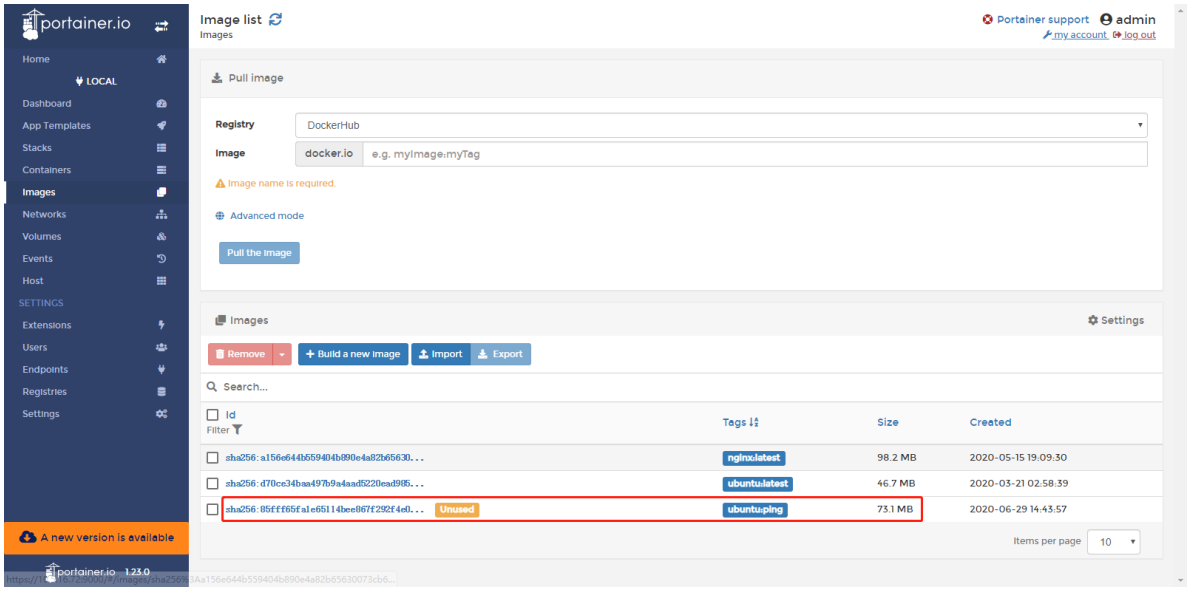
- 步骤 2：根据容器创建镜像点击容器名称进入容器的详情页面



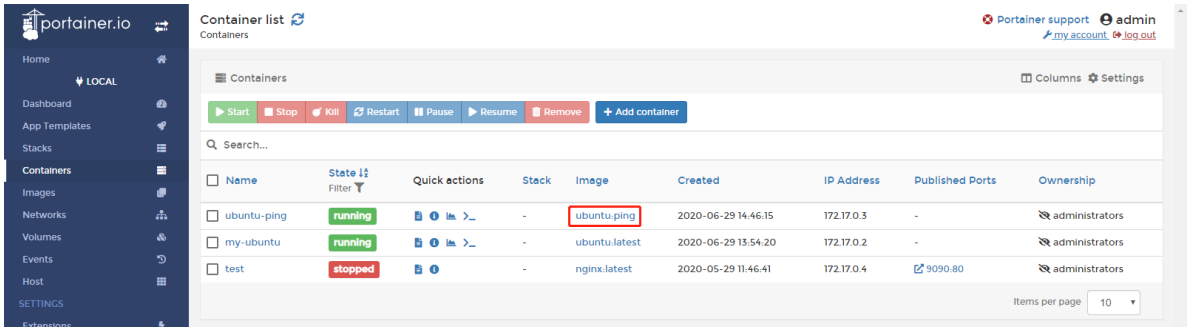
在详情页面的“Create image”中配置镜像的名称并点击“Creat”



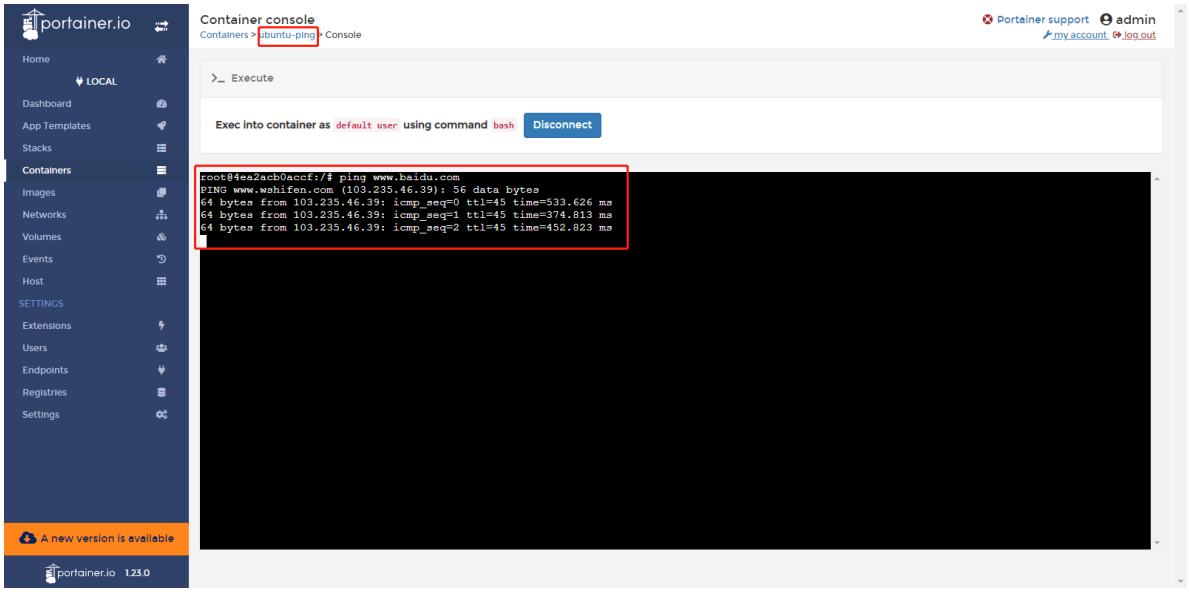
- 步骤 3：使用创建的镜像部署容器镜像创建完成后可以在 Portainer 的“Local>>Images”页面查看



随后在 Portainer 的“Local>>Containers”页面通过创建的镜像来部署一个 ubuntu 容器，如下图所示：

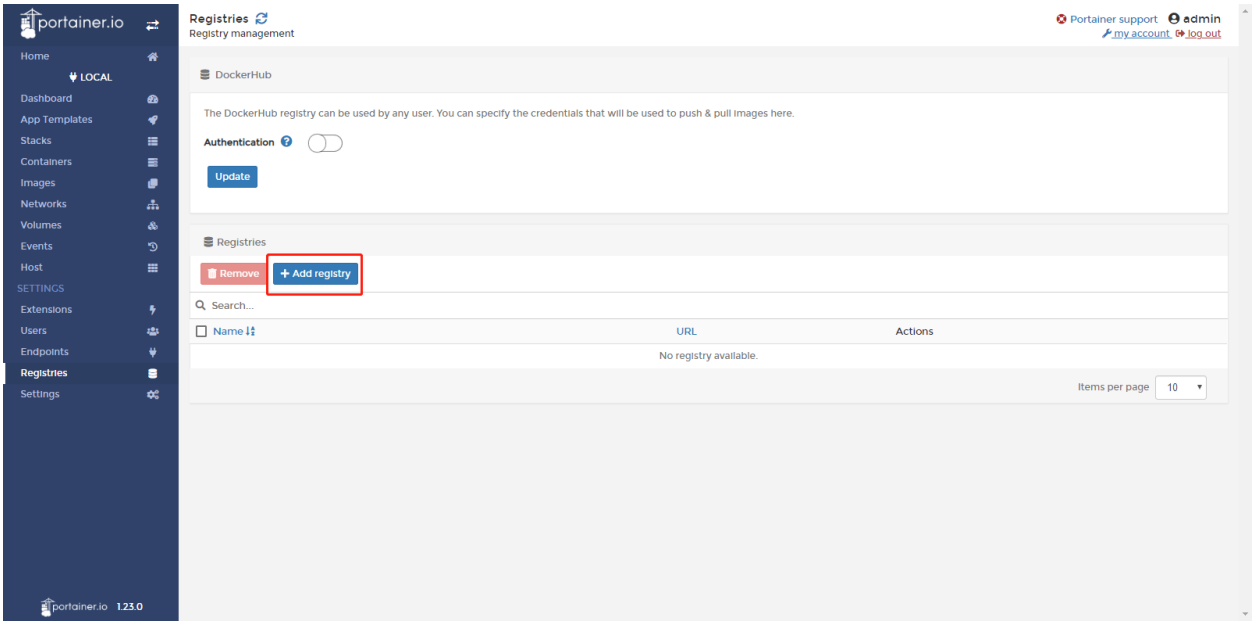


登录该容器的控制台，可以正常使用 ping 命令

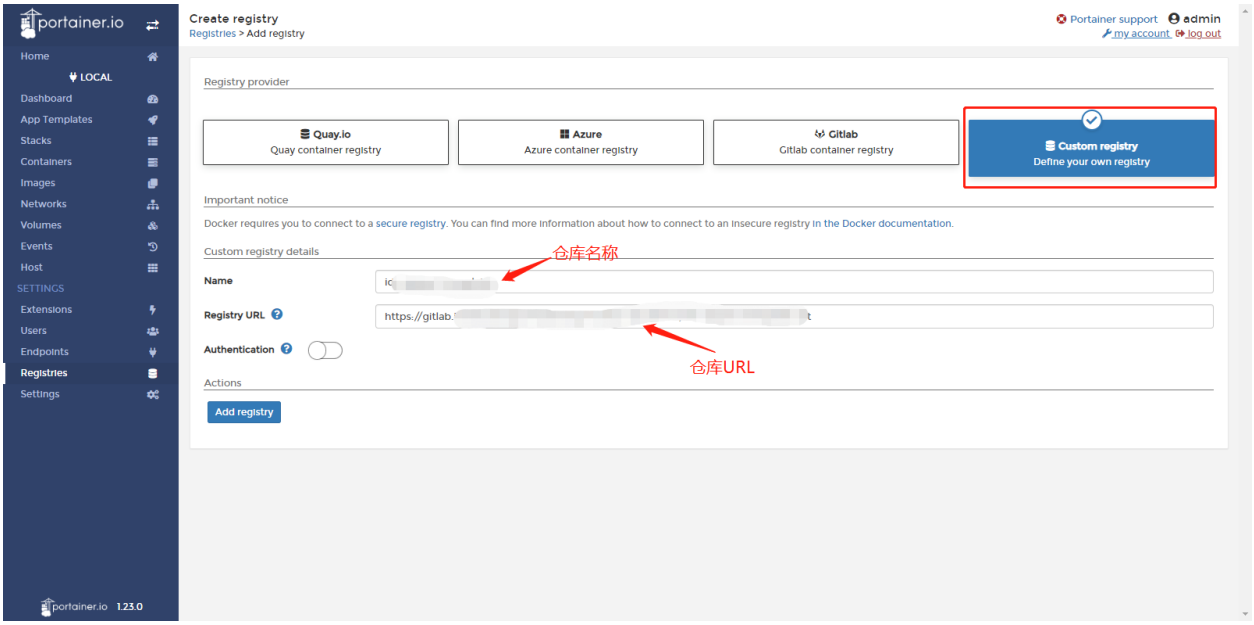


如何从 gitlab/github 上下载 docker 镜像

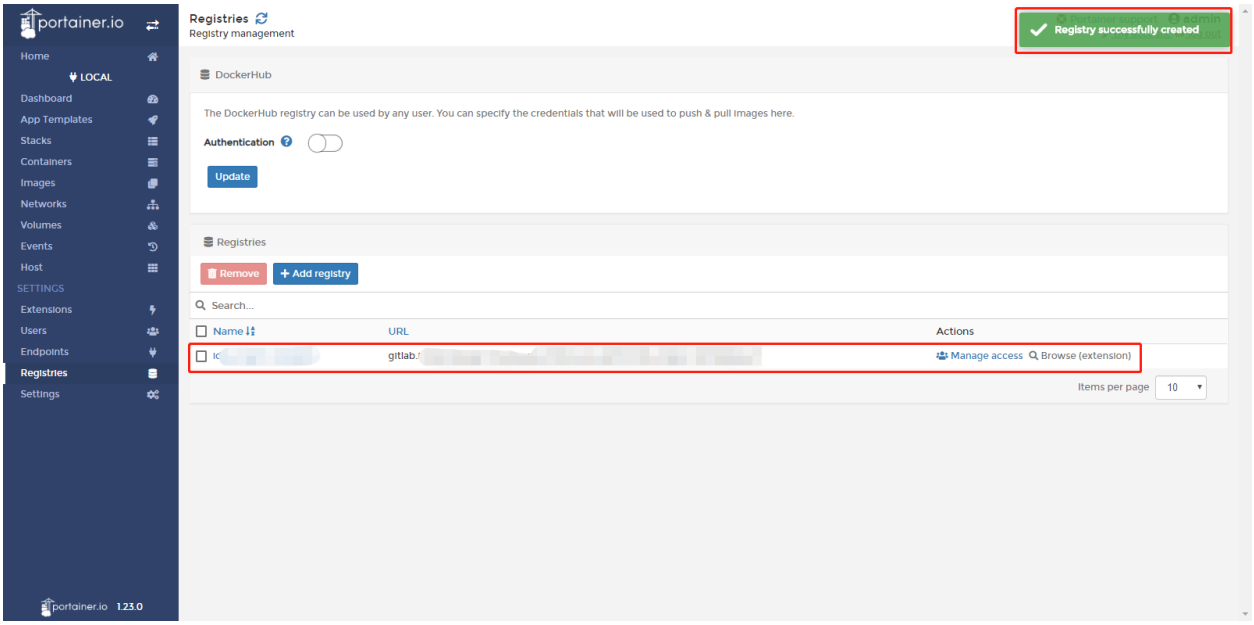
在 Portainer 的“Local>>Registries”页面点击“Add registry”以添加 docker 镜像仓库（必须为公开的仓库）。



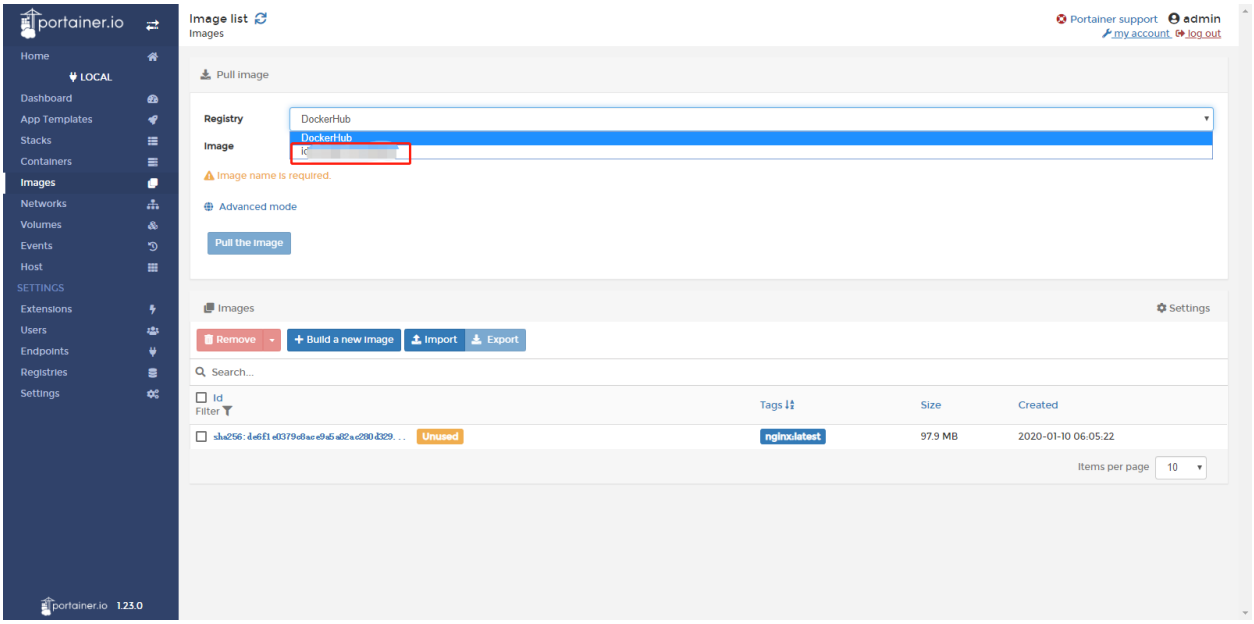
随后选择“Custom registry”并配置镜像仓库信息，配置完毕后点击“Add registry”。



镜像仓库添加成功后如下图所示：



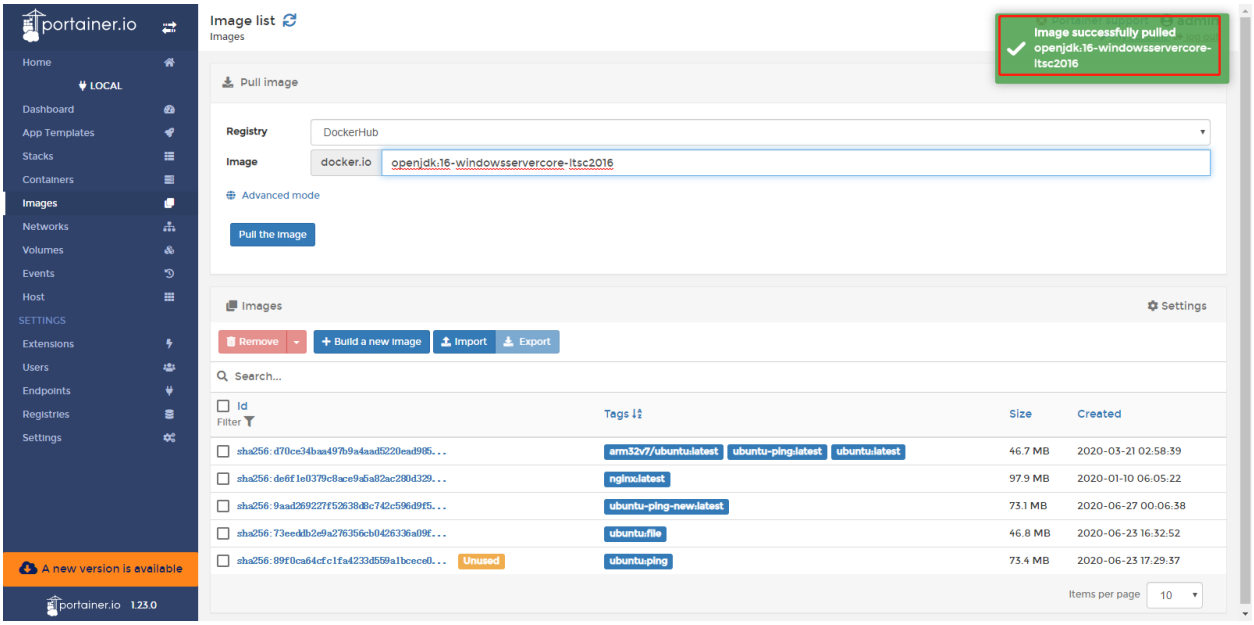
添加成功后，在拉取 docker 镜像时可以选择已配置的镜像仓库。

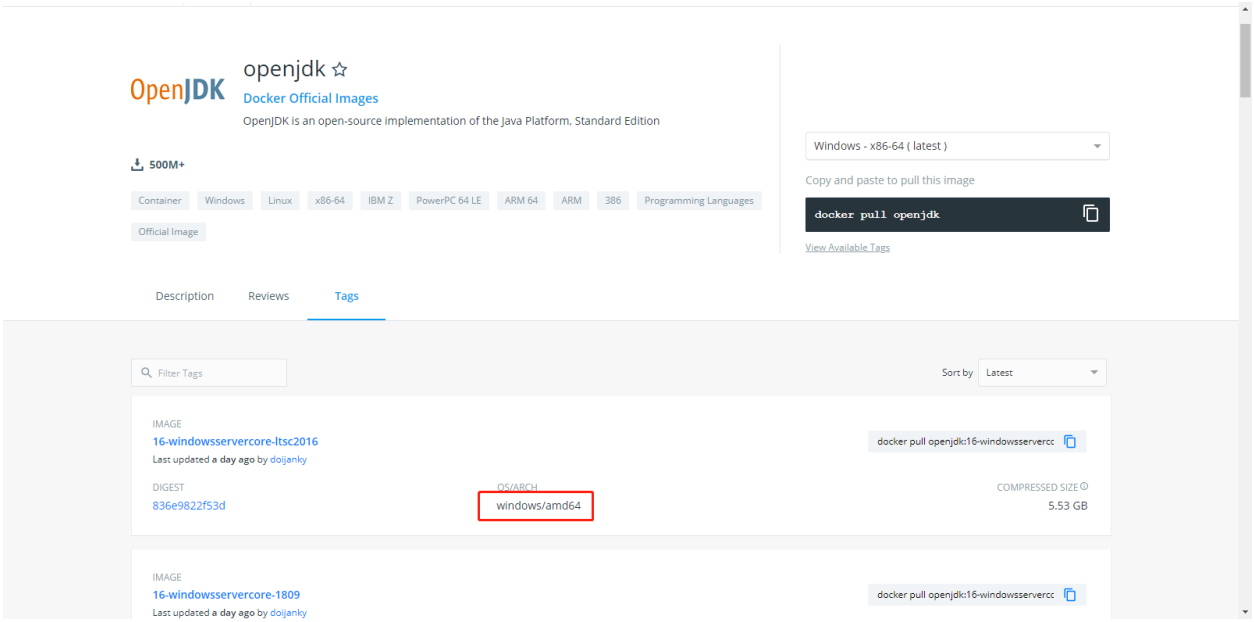


1.1.4 FAQ

Q1：在“Images”页面拉取镜像提示成功，但是在“Images”页面中未显示拉取到的镜像。

A1：因为网关的 CPU 架构为 linux/arm/v7，因此只有支持 linux/arm/v7 架构的镜像可以正常在网关中运行，其他如 window/amd64 等架构的镜像可能无法正常导入、拉取或在网关中运行。请确认拉取的镜像是否支持 linux/arm/v7。





1.2 Azure IoT Edge 用户手册

Azure IoT Edge 将云分析和自定义业务逻辑移到设备，这样你的组织就可以专注于业务见解而非数据管理。通过将业务逻辑打包到标准容器中，横向扩展 IoT 解决方案，然后可以将这些容器部署到任何设备，并从云中监视所有这些设备。关于 Azure IoT Edge 的详细介绍请参考了解 Azure IoT Edge 模块。映翰通 IG902 系列产品提供 Azure IoT Edge SDK 以支持 Azure IoT Edge，使你能够快速开发并完成任务，安全和高效地部署相关服务。该 SDK 主要通过管理 Azure IoT Edge runtime 来管理从 Azure 云平台部署并运行在 IoT Edge 设备（即 IG902）上的 IoT Edge 模块（docker image）。本文档将为你说明如何基于 Azure IoT Edge SDK 实现通过 Azure 云平台在 IG902 上部署并运行一个模拟遥测数据并发送到 IoT Hub 的 IoT Edge 模

块。

- *1.* 环境准备
  - *1.1* 配置 *Azure IoT* 环境
  - *1.2* 配置 *IG902* 环境
    - *1.2.1* 配置 *IG902* 连接 *Internet*
    - *1.2.2* 更新 *IG902* 软件版本
  - *1.3* 修改 *Azure IoT Edge* 配置文件
- *2.* 运行 *Azure IoT Edge*
- *3.* 配置并部署模块

1.2.1 1. 环境准备

开始之前，你需要准备以下事项（你可以访问资源中心获取软件版本）：

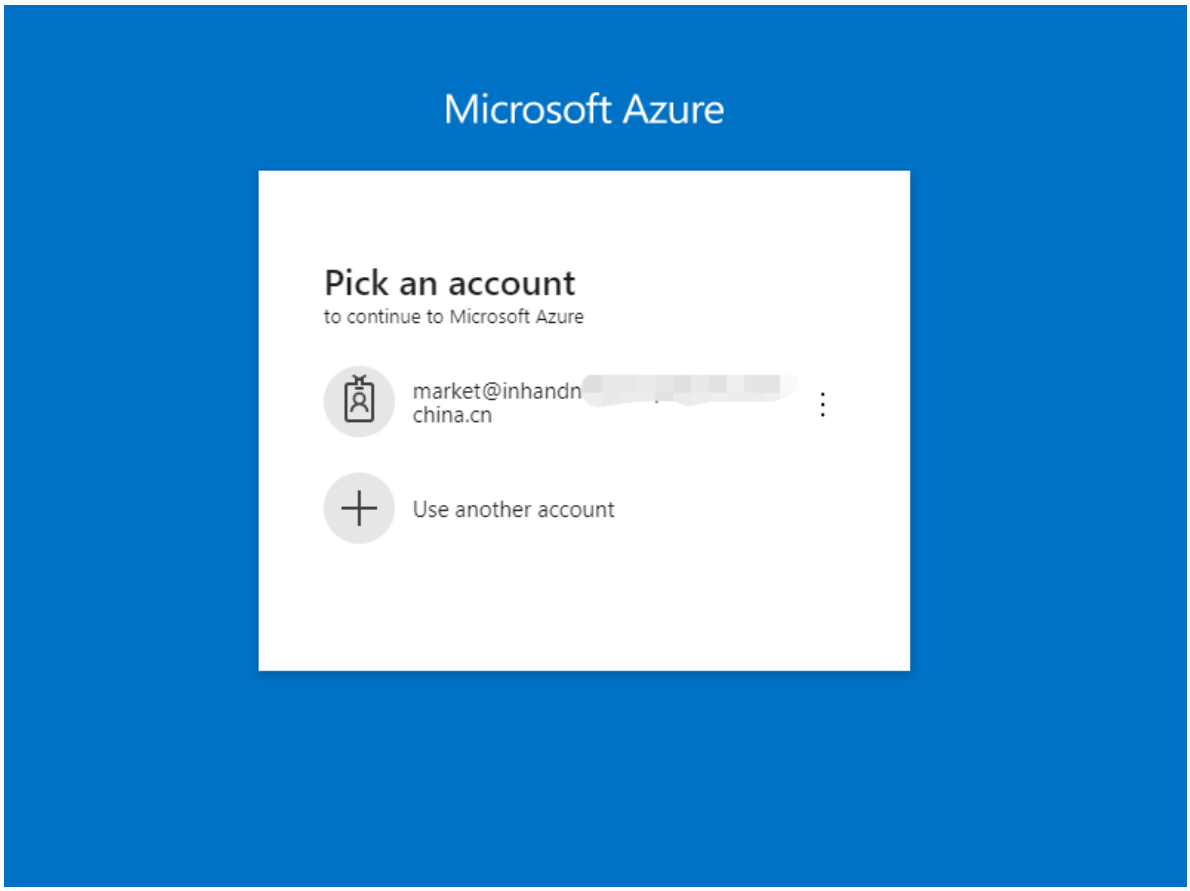
- Azure IoT 账户
- IG902 固件版本：`v2.0.0.r12644` 及以上
- Docker SDK 版本：`18.06.3-ce` 及以上
- Azure IoT Edge SDK 版本：`1.0.4` 及以上
- IG902 系列产品

1.1 配置 Azure IoT 环境

如果你已经在 Azure IoT 上配置了相应的 IoT Hub 和 IoT Edge 设备，可以跳过这一小节。

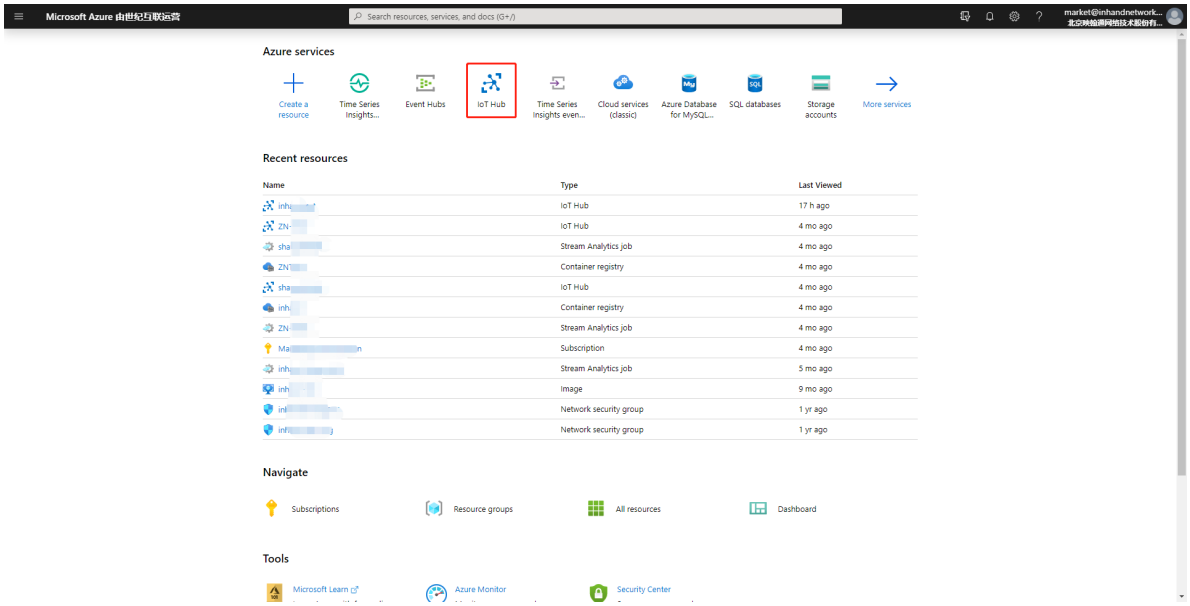
- 步骤 1：登录 Azure IoT

  访问https://portal.azure.cn/登录 Azure。

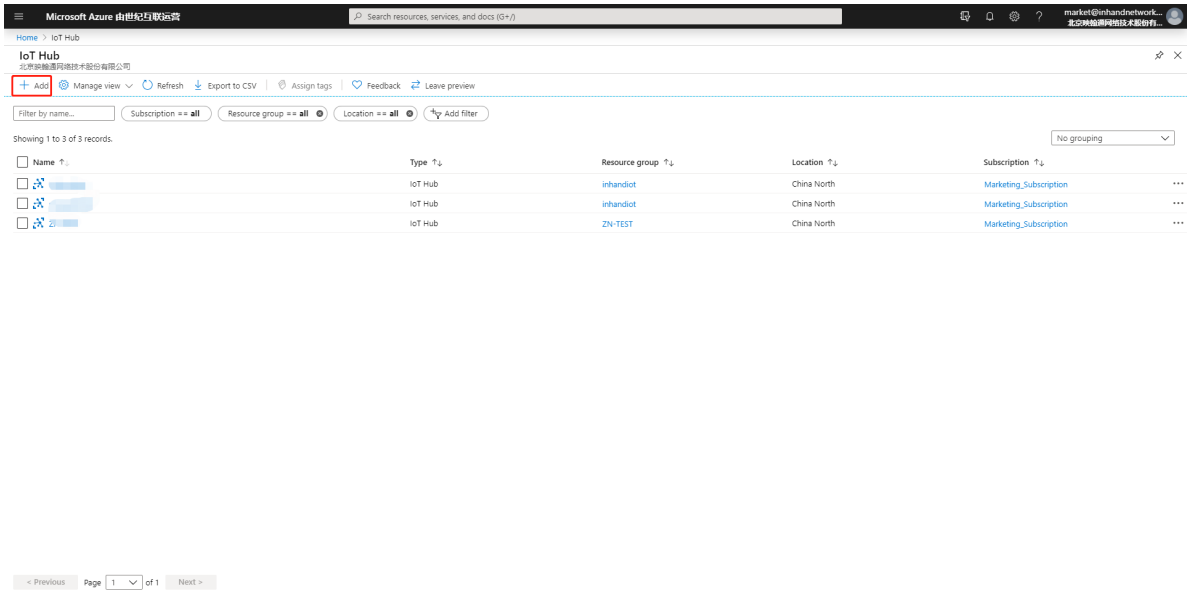


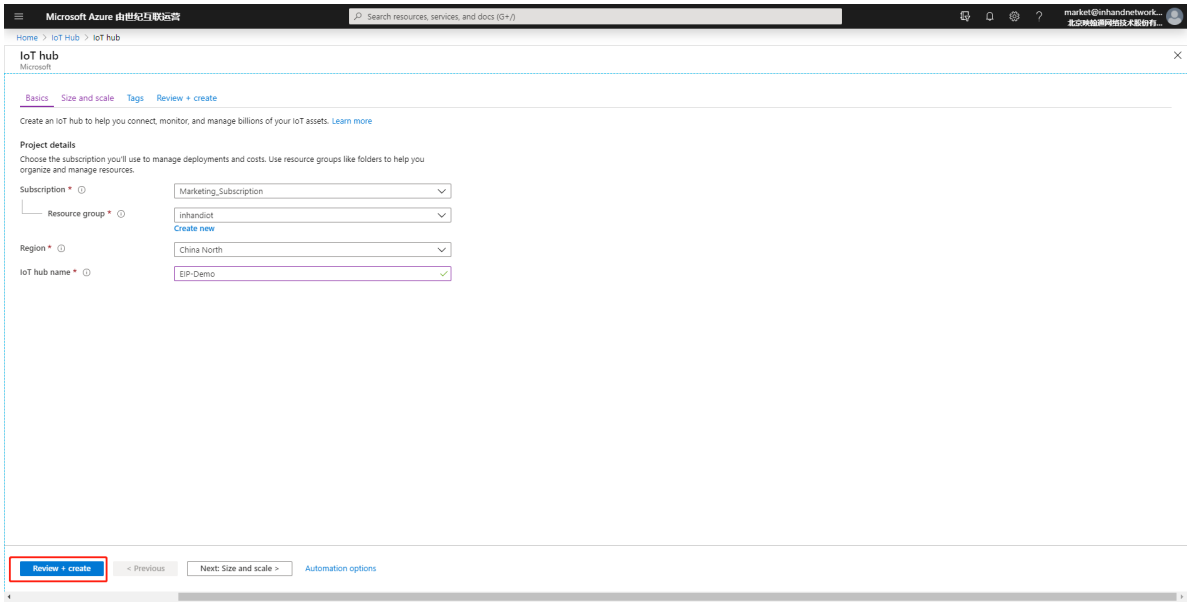
- 步骤 2：添加 IoT Hub

  登录成功后如下图所示，选择“IoT Hub”。

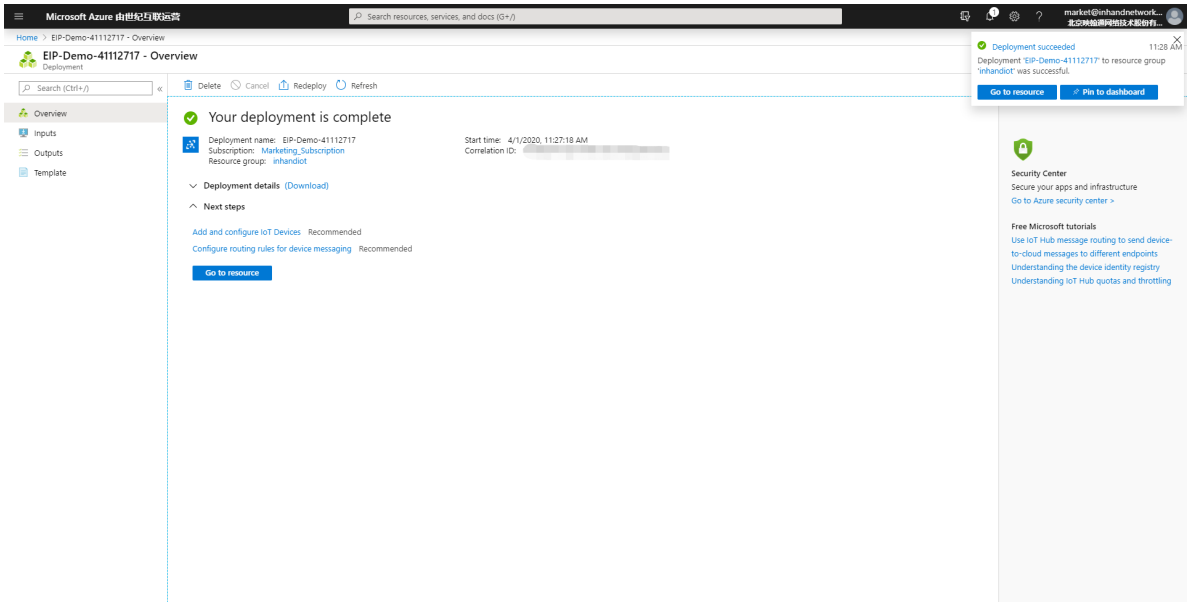


  点击“Add”创建一个 IoT Hub。


Microsoft Azure 由世纪互联运营
Search resources, services, and docs (G+/)
Home > IoT Hub
IoT Hub
Add
Manage view
Refresh
Export to CSV
Assign tags
Feedback
Leave preview
Filter by name...
Subscription == all
Resource group == all
Location == all
Add filter
Showing 1 to 3 of 3 records.
No grouping
Name
Type
Resource group
Location
Subscription
IoT Hub
inhandiot
China North
Marketing_Subscription
IoT Hub
inhandiot
China North
Marketing_Subscription
IoT Hub
ZN-TEST
China North
Marketing_Subscription
< Previous
Page 1 of 1
Next >

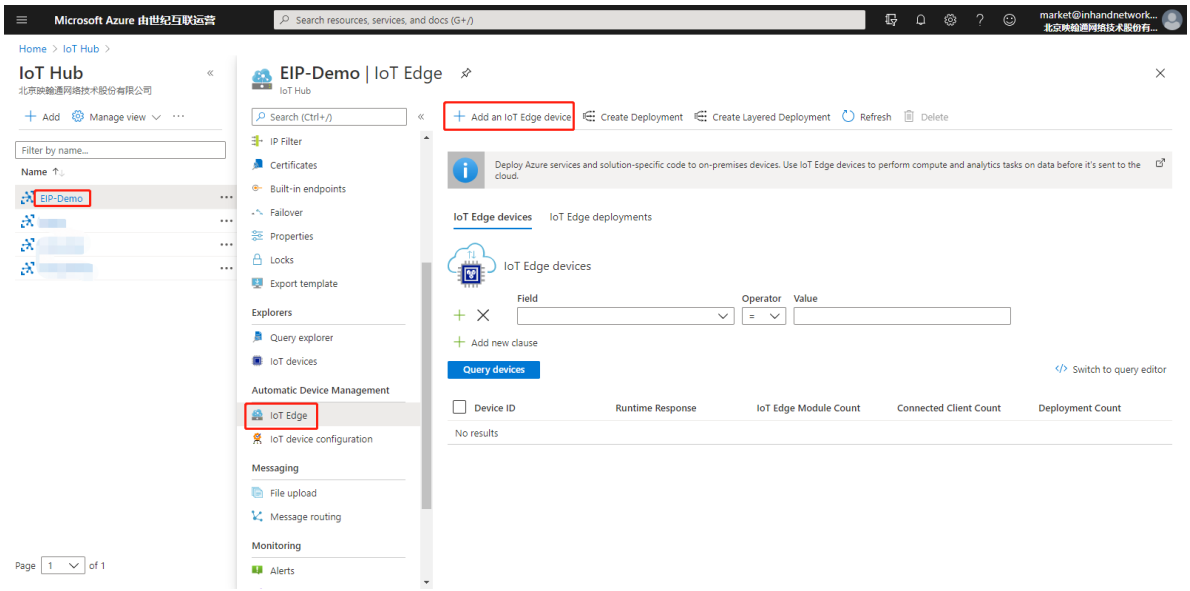

Microsoft Azure 由世纪互联运营
Search resources, services, and docs (G+/)
Home > IoT Hub > IoT hub
IoT hub
Microsoft
Basics
Size and scale
Tags
Review + create
Create an IoT hub to help you connect, monitor, and manage billions of your IoT assets. Learn more
Project details
Choose the subscription you'll use to manage deployments and costs. Use resource groups like folders to help you organize and manage resources.
Subscription *
Marketing_Subscription
Resource group *
inhandiot
Create new
Region *
China North
IoT hub name *
EIP-Demo
Review + create
< Previous
Next: Size and scale >
Automation options

创建成功后如下图所示：

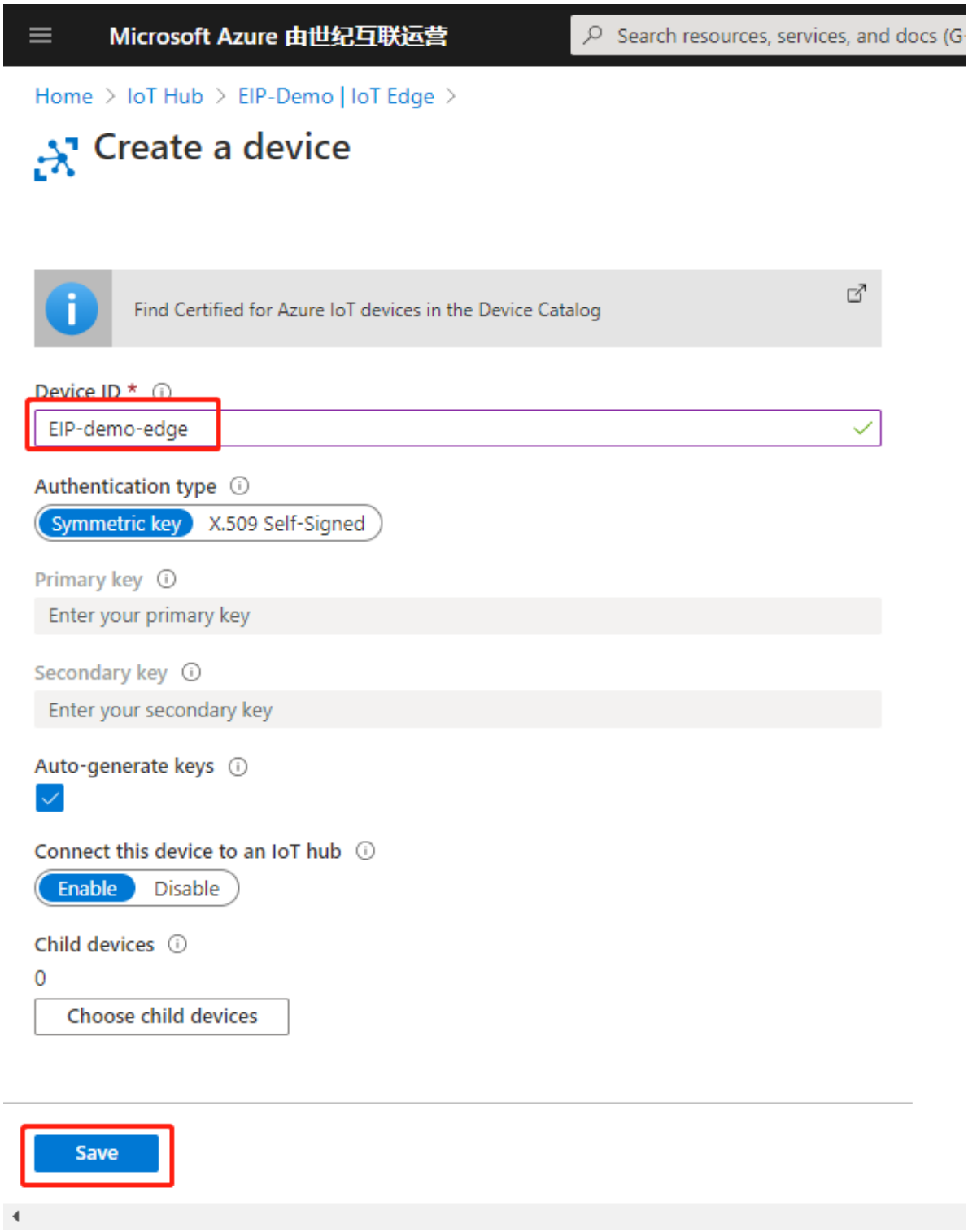


- 步骤 3：添加 IoT Edge 设备

  在“IoT Hub”中点击相应的 IoT Hub，并进入该 IoT Hub 的“IoT Edge”页面，点击“Add an IoT Edge device”。

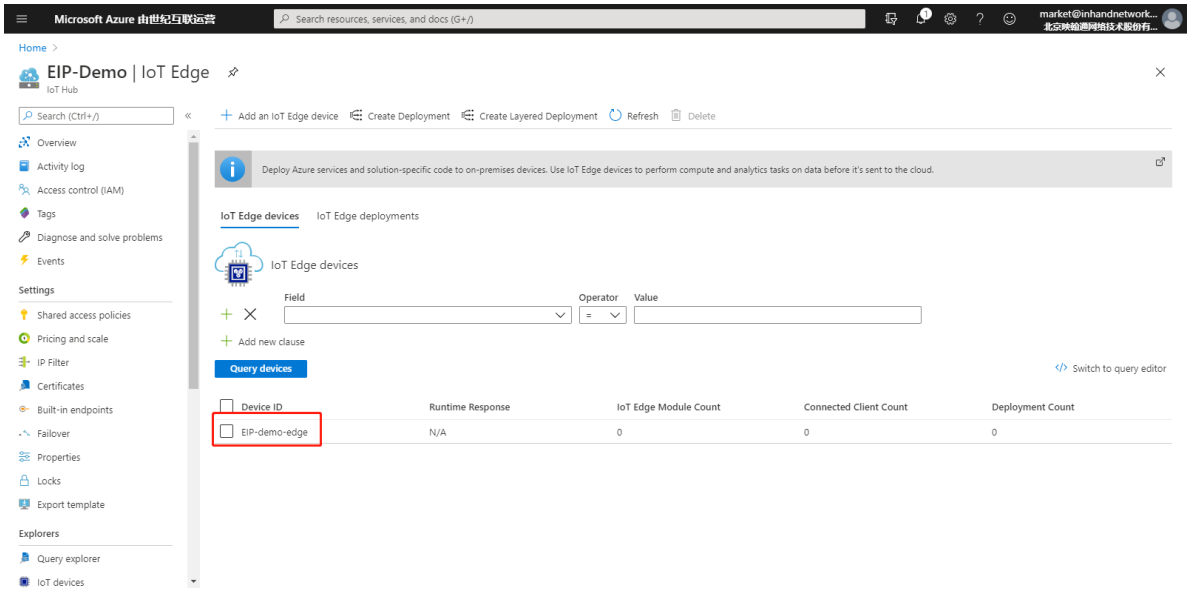


  配置相应的参数并点击“Save”。

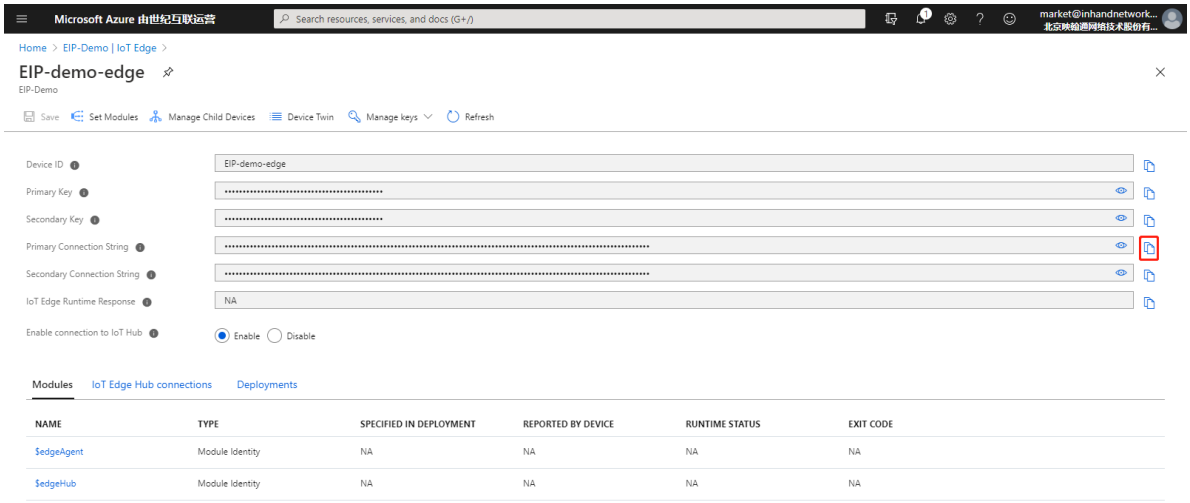


- 步骤 4：复制 IoT Edge 设备的连接字符串

  IoT Edge 设备创建成功后如下图所示：



点击 IoT Edge 设备的“Device ID”进入该 IoT Edge 设备的详情页面，复制“Primary Connection String”参数以备后续使用。



1.2 配置 IG902 环境

1.2.1 配置 IG902 连接 Internet

配置 IG902 连接 Internet，请参考IG902 连接 Internet。

1.2.2 更新 IG902 软件版本

- 更新 IG902 固件版本

  如何更新 IG902 固件版本请参考更新 IG902 软件版本。

- 更新 IG902 Docker SDK

  如何更新 IG902 Docker SDK 请参考安装 Docker SDK

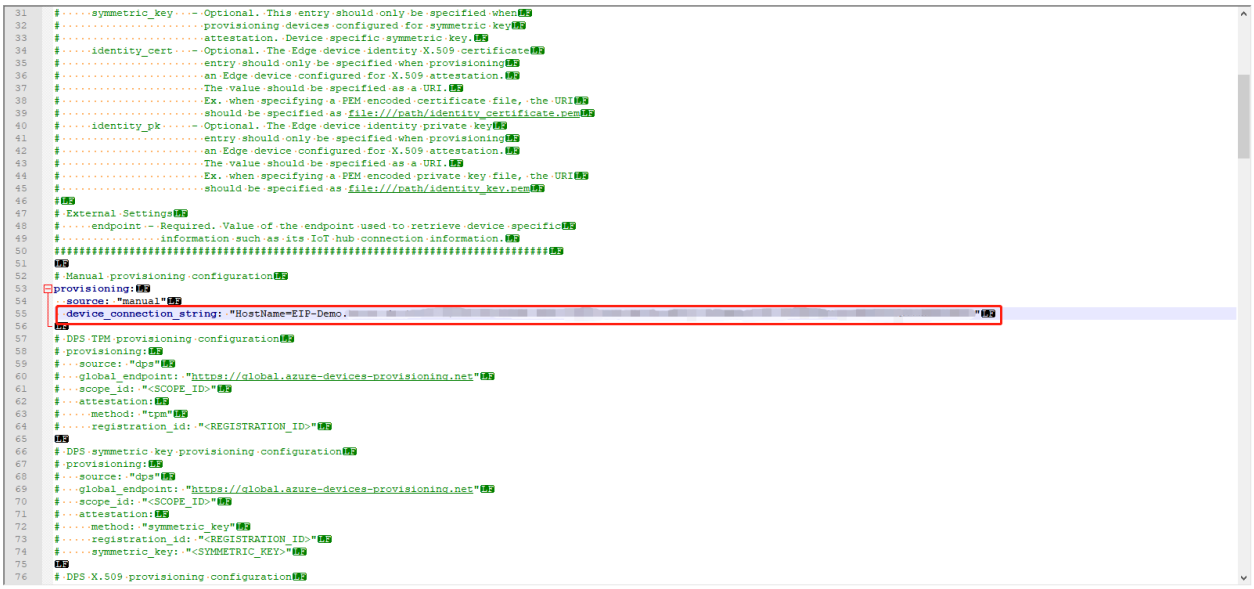
- 更新 IG902 Azure IoT Edge SDK

  进入“边缘计算 >> 云厂商边缘计算”页面，取消勾选“启用安全守护程序”后，点击“升级”按钮，选择 Azure IoT Edge SDK 文件并点击“确定”。

1.3 修改 Azure IoT Edge 配置文件

在“边缘计算 >> 云厂商边缘计算”页面点击“导出”以导出 Azure IoT Edge 配置文件。

修改 Azure IoT Edge 配置文件中的“device_connection_string”参数并保存，该字符串为复制 *IoT Edge* 设备的连接字符串获取的 IoT Edge 设备的“Primary Connection String”参数。



随后导入修改后的 Azure IoT Edge 配置文件。

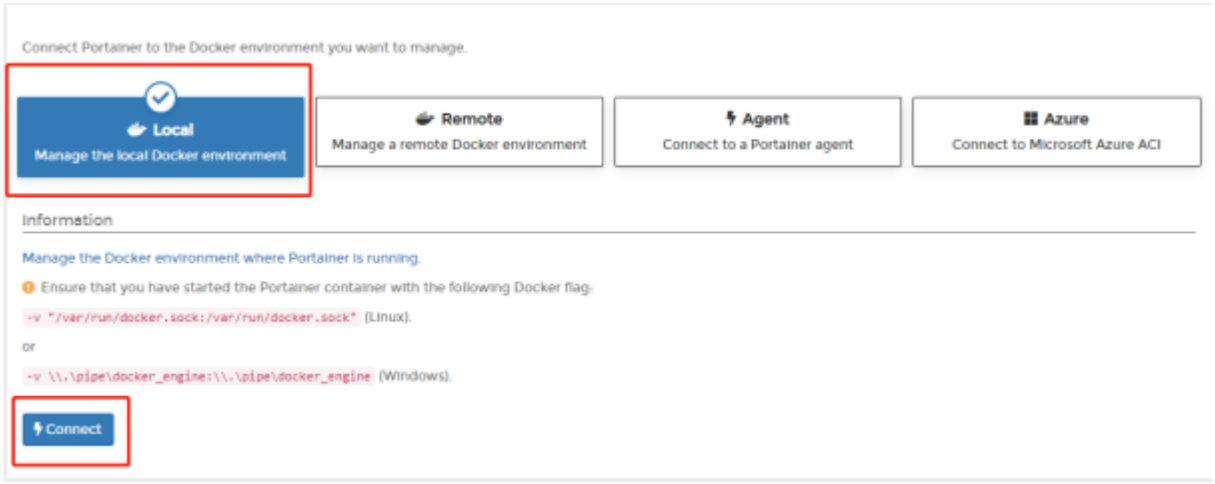
1.2.2 2. 运行 Azure IoT Edge

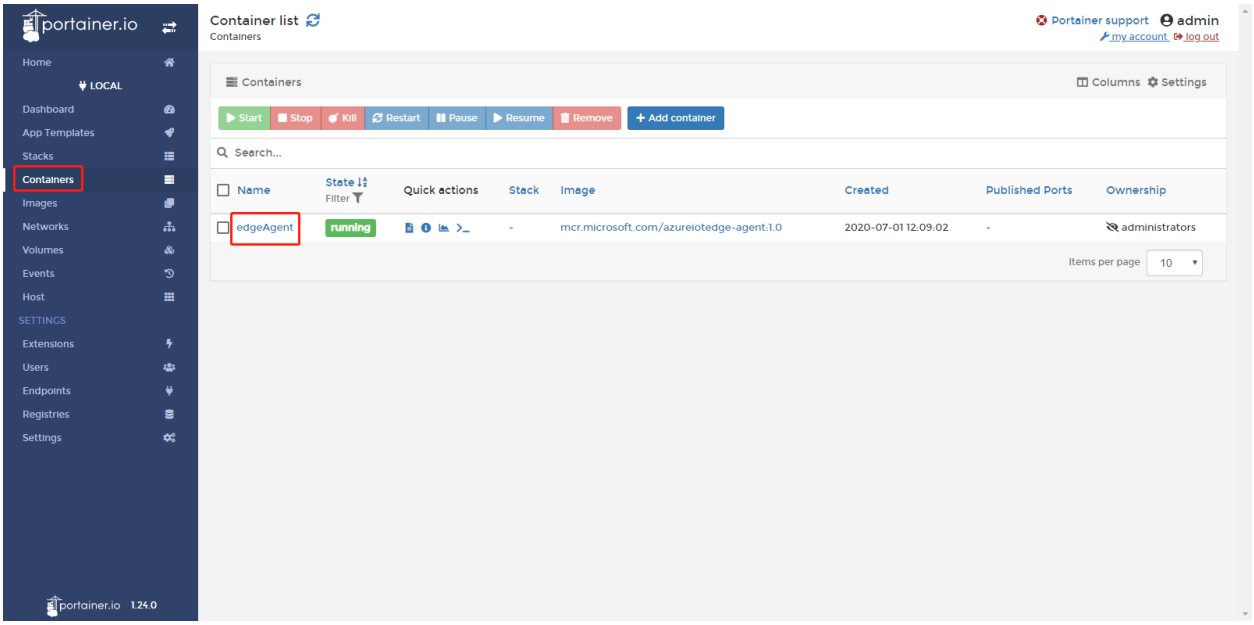
进入“边缘计算 >> 启动 Docker 管理”页面，勾选“启动 Docker 管理器”。

进入“边缘计算 >> 云厂商边缘计算”页面，勾选“启用安全守护程序”。

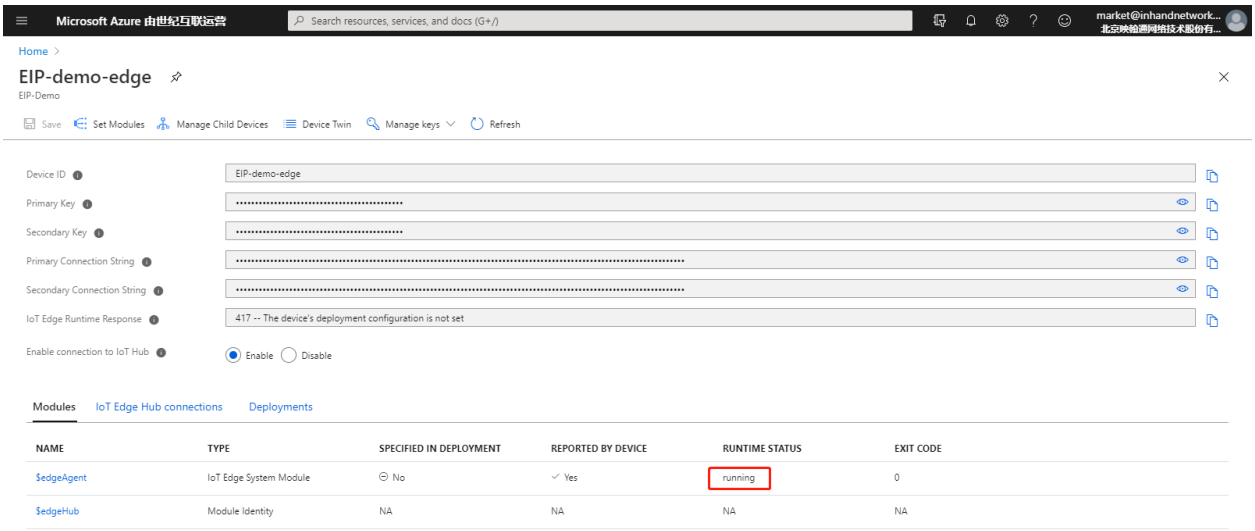
启用安全守护程序后，Azure IoT Edge 守护程序会拉取镜像并创建 `edgeAgent` 容器，镜像文件较大，这需要一段时间，请耐心等待大约 20 分钟。你可以进入 portainer 的“LOCAL >> Containers”页面查看是否有 `edgeAgent` 容器运行，当 `edgeAgent` 容器运行时，说明 Azure IoT Edge 已经处于正常工作状态了。


概览 / 边缘计算 / Docker 管理
启用Docker管理器:
Docker版本:
18.06.3-ce
升级
用户名:
admin
* 密码:
* 端口号:
9000
进入Docker管理页面
提 交
重 置


Connect Portainer to the Docker environment you want to manage.
Local
Manage the local Docker environment
Remote
Manage a remote Docker environment
Agent
Connect to a Portainer agent
Azure
Connect to Microsoft Azure ACI
Information
Manage the Docker environment where Portainer is running.
Ensure that you have started the Portainer container with the following Docker flag:
-v "/var/run/docker.sock:/var/run/docker.sock" (Linux).
or
-v \\.\pipe\docker_engine:\\.\pipe\docker_engine (Windows).
Connect



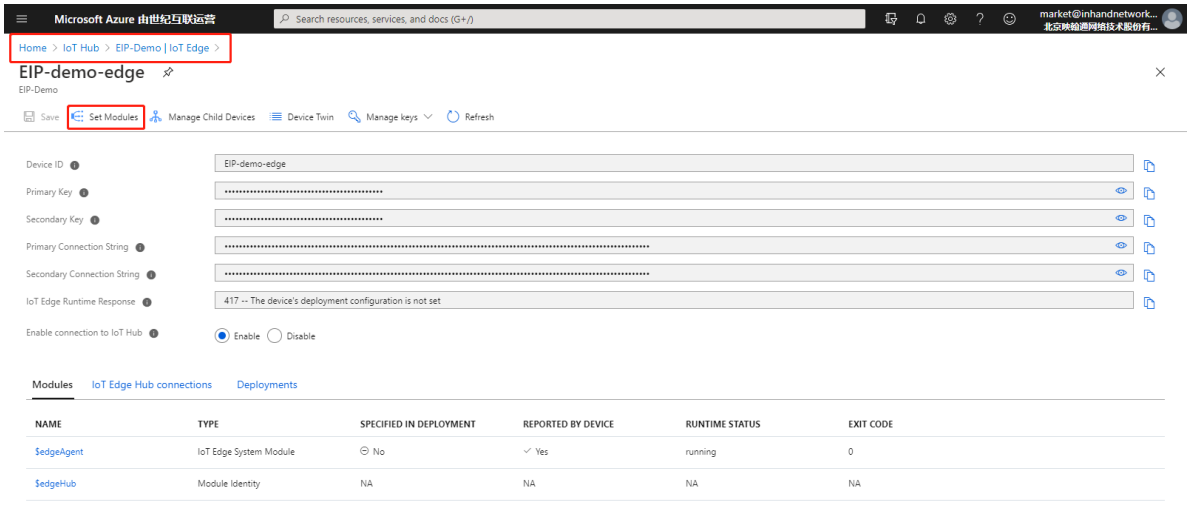
此时，在 IoT Edge 设备的详情页面可以看到 $edgeAgent 的 “RUNTIME STATUS” 为 `running`。



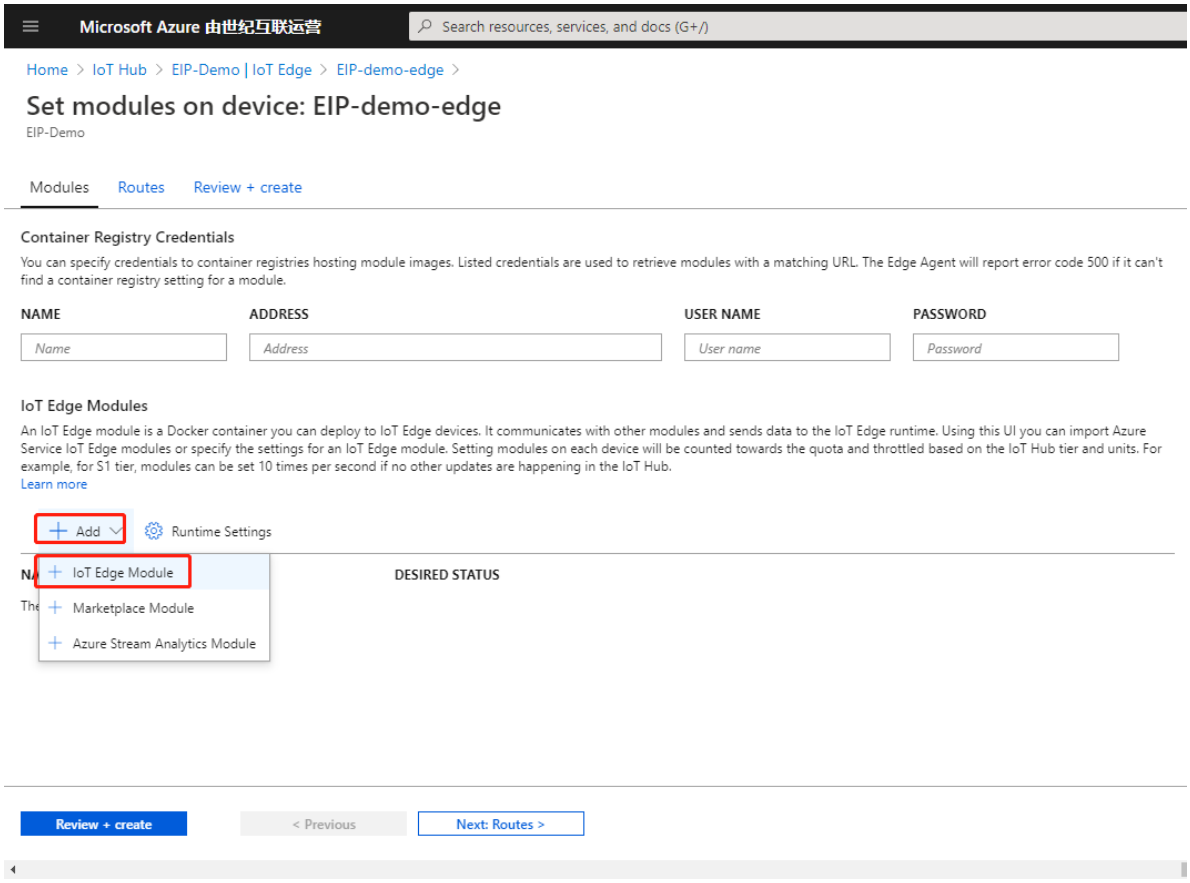
1.2.3 3. 配置并部署模块

- 步骤 1：添加 IoT Edge 模块

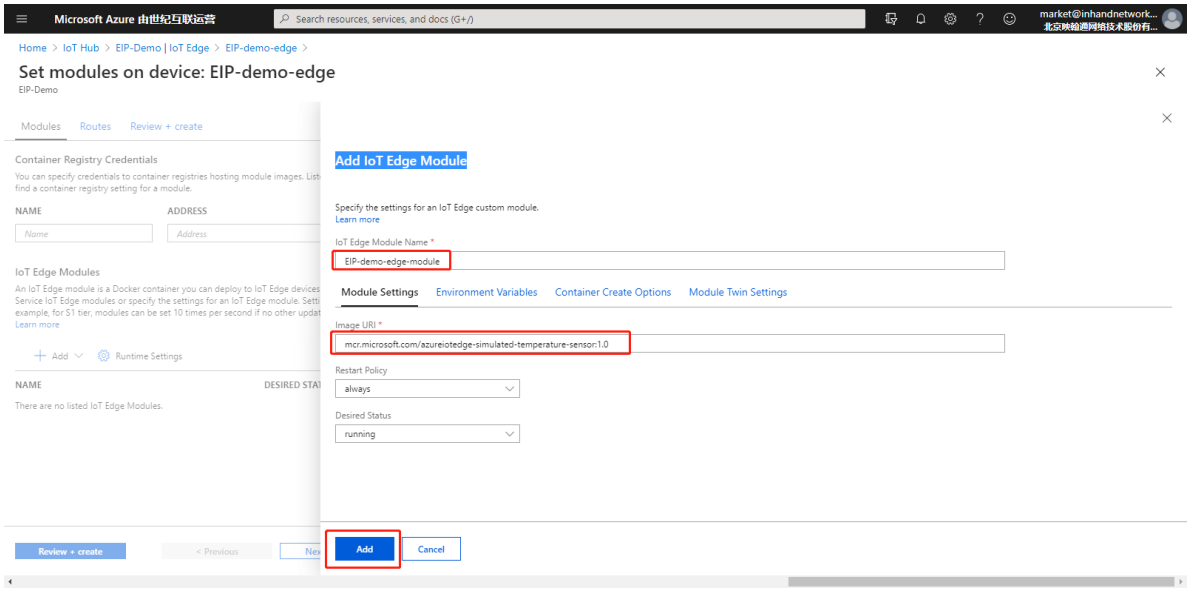
  在 IoT Edge 设备的详情页面点击 “Set Modules”



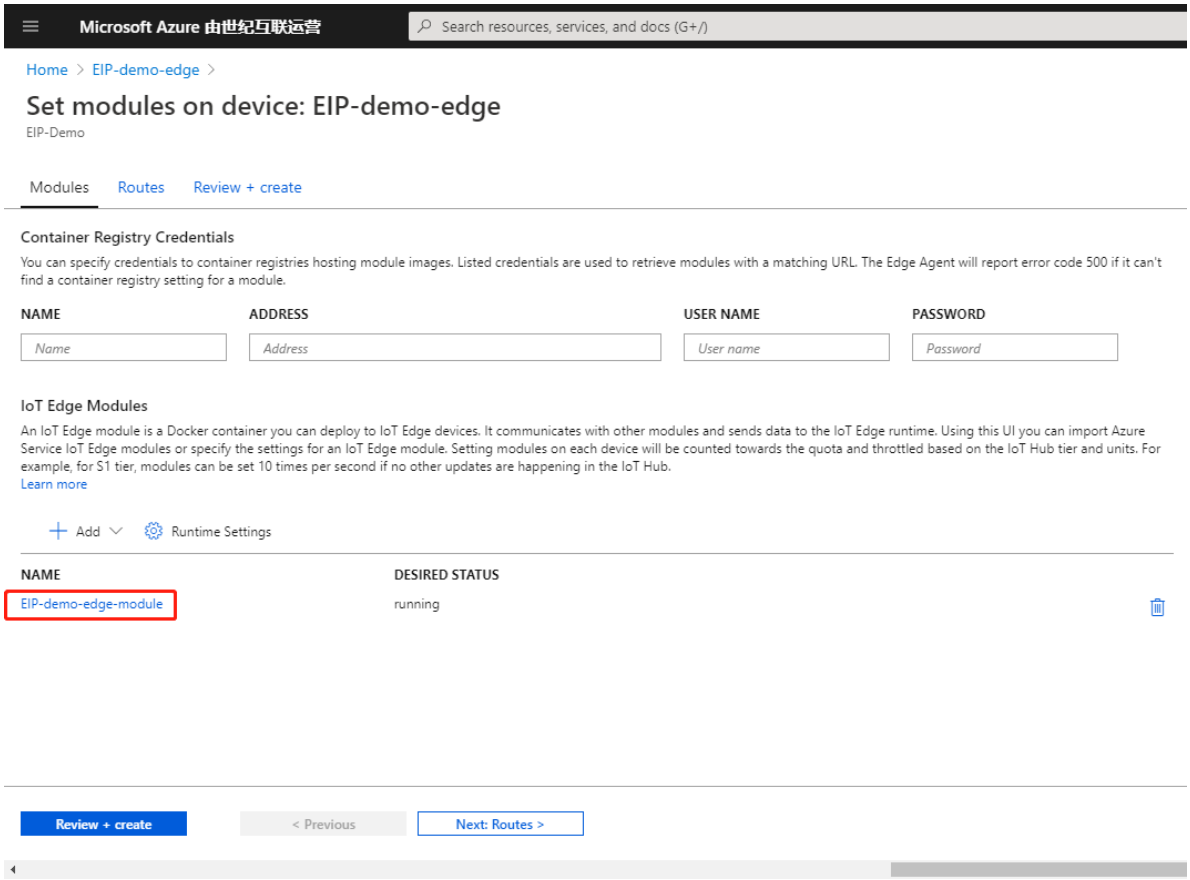
在“Set modules on device”页面点击“Add”按钮并选择“IoT Edge Module”以添加 IoT Edge 模块。



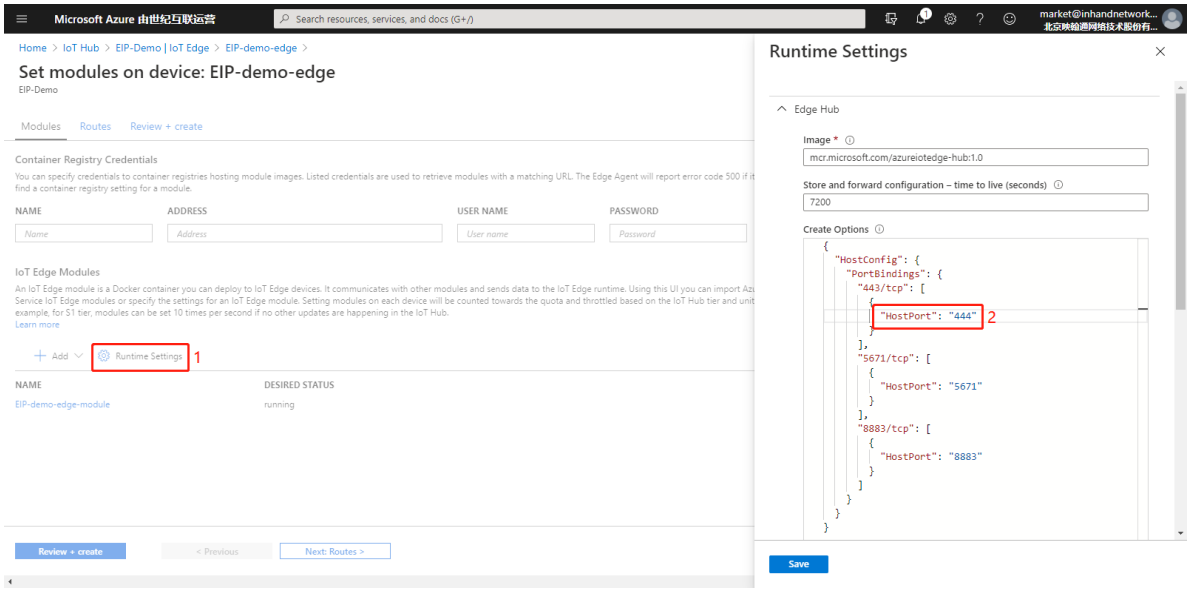
在“Add IoT Edge Module”弹出框中配置模块名称和镜像 URL，这里以 `mcr.microsoft.com/azureiotedge-simulated-temperature-sensor:1.0` 镜像为例，该镜像为微软官方提供，模拟遥测数据并发送到 IoT Hub。如何开发模块请参考为 Linux 设备开发和部署 Python IoT Edge 模块。



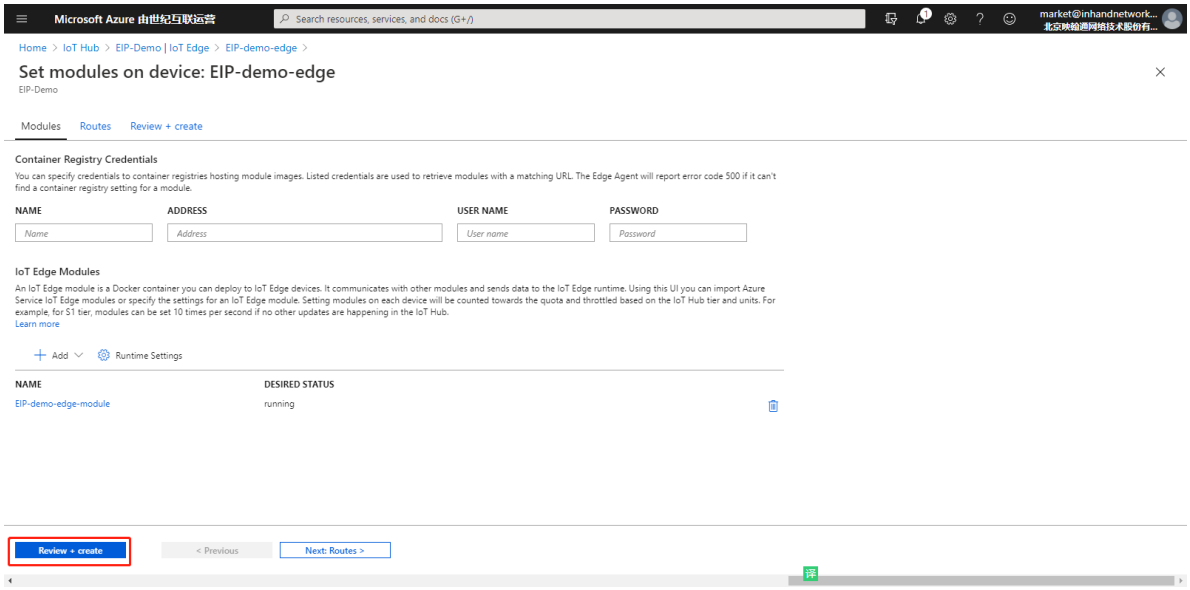
模块添加后如下图所示：



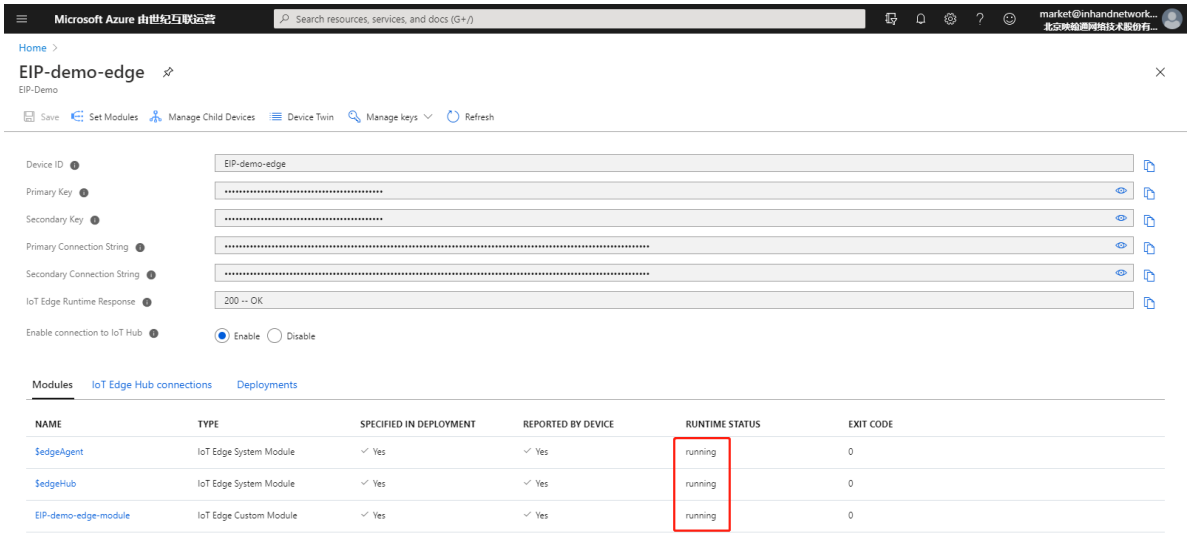
edgeHub 容器监听 443 端口并默认映射到宿主机（即 IG902）的 443 端口。通常 IG902 的 443 端口会被其他程序监听占用，所以需要修改 edgeHub 容器的映射端口以保证 edgeHub 可以启动。点击“Runtime Settings”按钮，在“Runtime Settings”弹出框中修改“HostPort”为其他端口，如：444。修改完成后点击“Save”



随后点击“Review + create”，确认无误后点击“Create”，完成 IoT Edge 模块添加。

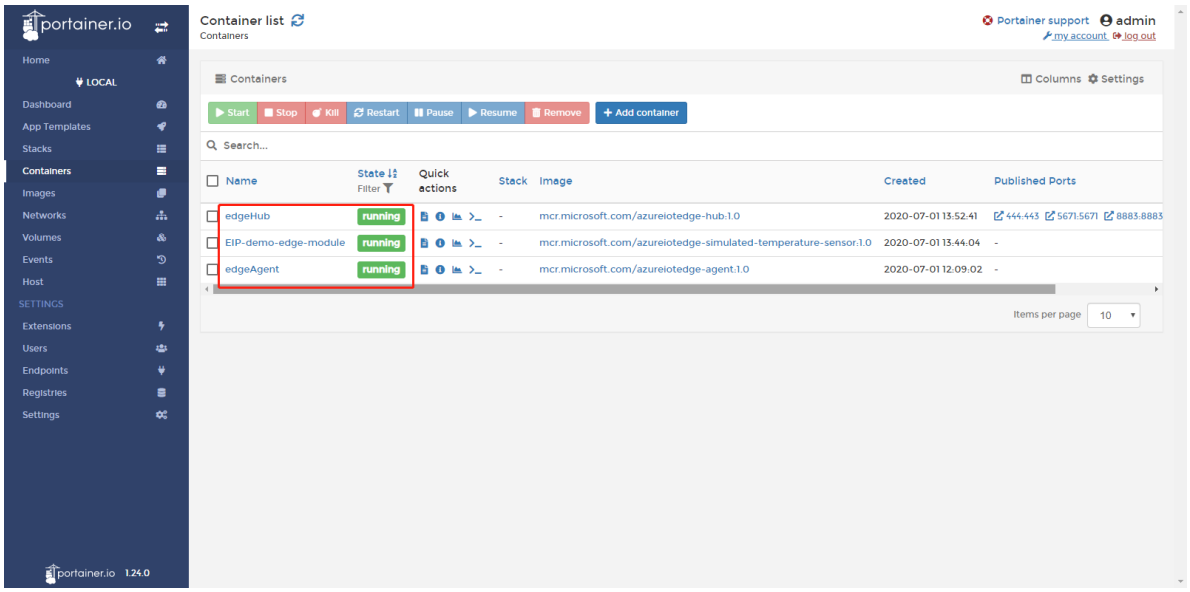


随后 `edgeHub` 和 `mcr.microsoft.com/azureiotedge-simulated-temperature-sensor:1.0` 将自行部署并运行在 IG902 上，你可以在 IoT Edge 设备的详情页面查看各个容器的部署情况。当“RUNTIME STATUS”为 `running` 时说明容器已部署并正常运行。（两个镜像大小之和约为 400MB，部署时间约为 20 分钟或更长，请耐心等待）

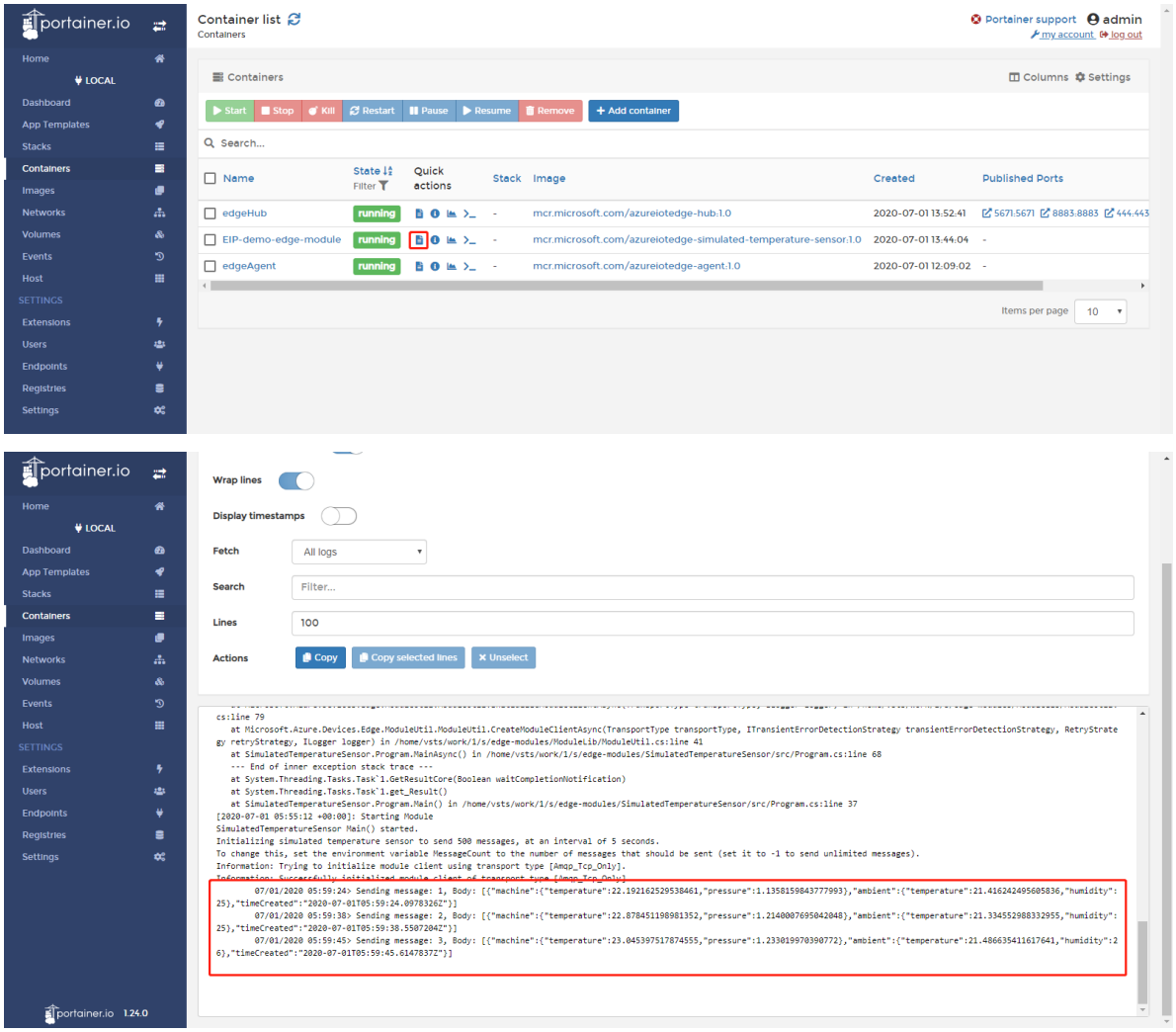


- 步骤 2：查看容器运行情况

  访问 portainer 的“LOCAL >> Containers”页面，可以看到有三个容器已经在运行起来了。



点击 `EIP-demo-edge-module` 容器的“Logs”按钮查看容器的运行日志，当日志如下图所示说明容器已正常运行：模拟模拟遥测数据并发送到 IoT Hub。

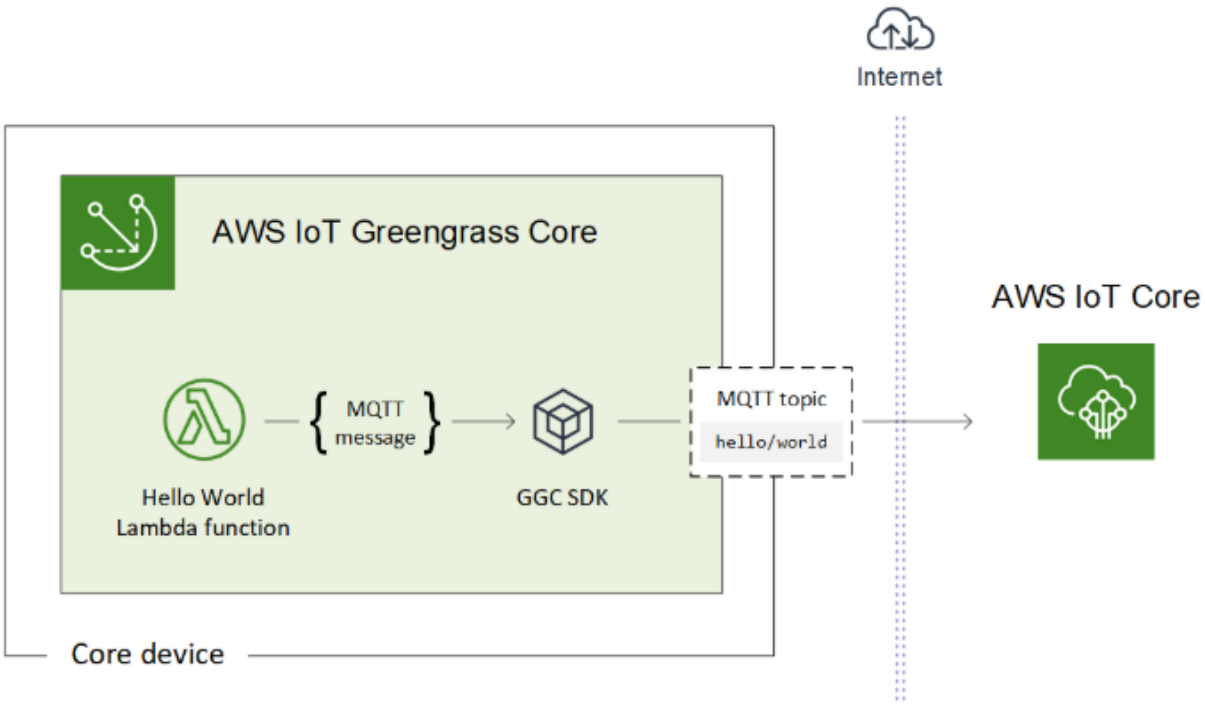


至此，完成了通过 Azure 云平台在 IG902 上部署并运行一个模拟遥测数据并发送到 IoT Hub 的 IoT Edge 模块。

1.3 AWS IoT Greengrass 用户手册

AWS IoT Greengrass 是将云功能扩展到本地设备的软件。这使得设备可以更靠近信息源来收集和分析数据，自主响应本地事件，同时在本地网络上彼此安全地通信。本地设备还可以与 AWS IoT Core 安全通信并将 IoT 数据导出到 AWS 云。AWS IoT Greengrass 开发人员可以使用 AWS Lambda 函数和预构建的连接器来创建无服务器应用程序，这些应用程序将部署到设备上以进行本地执行。

AWS IoT Greengrass SDK 支持 IG902 作为 AWS IoT Greengrass 核心（以下简称 Greengrass 核心）实现 Lambda 函数的部署与本地执行、使用托管订阅在 AWS IoT 与设备、连接器和 Lambda 函数之间进行的 MQTT 消息传递等功能，使您能够快速开发并完成任务，安全，高效地部署相关服务。本文档将为你说明如何基于 AWS IoT Greengrass SDK 实现通过 AWS 云平台在 IG902 上部署并运行 Hello World Lambda 函数，该函数会通过 MQTT 协议将 Hello World 消息发送到 AWS IoT，整体架构如下图所示：



- 先决条件
- *1.* 环境准备
  - *1.1* 配置 *AWS IoT*
  - *1.2* 配置 *IG902*
    - *1.2.1* 基础配置
    - *1.2.2* 连接 *AWS IoT Greengrass*
- *2.* 部署并验证 *Lambda* 函数
  - *2.1* 部署 *Lambda* 函数
  - *2.2* 验证 *Lambda* 函数是否在核心设备上运行
- 附录
  - *Device Supervisor*（*Greengrass* 设备）与 *Greengrass* 核心通讯
    - *AWS IoT Greengrass* 组配置
    - *Device Supervisor* 配置

1.3.1 先决条件

开始之前，你需要准备以下事项（你可以访问资源中心获取软件版本）：

- AWS IoT 账户

- IG902 固件版本：`v2.0.0.r13054` 及以上
- AWS IoT Greengrass SDK 版本：`1.10.2-RC1` 及以上
- Python SDK 版本：`py3sdk-V1.4.0_Edge-IG9` 及以上
- IG902 系列产品

1.3.2 1. 环境准备

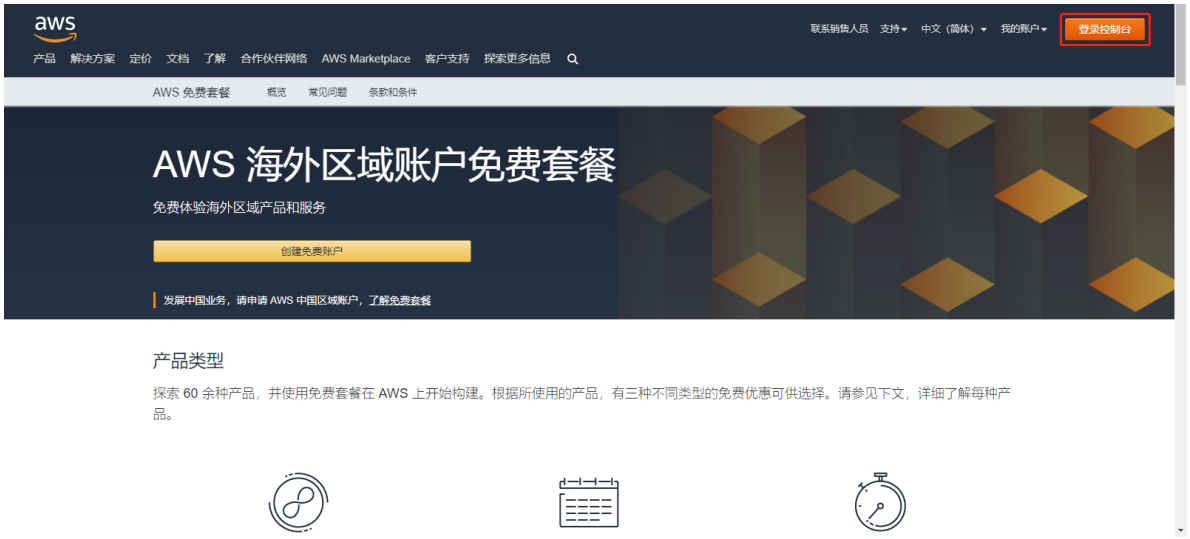
- *1.1 配置 AWS IoT*
- *1.2 配置 IG902*

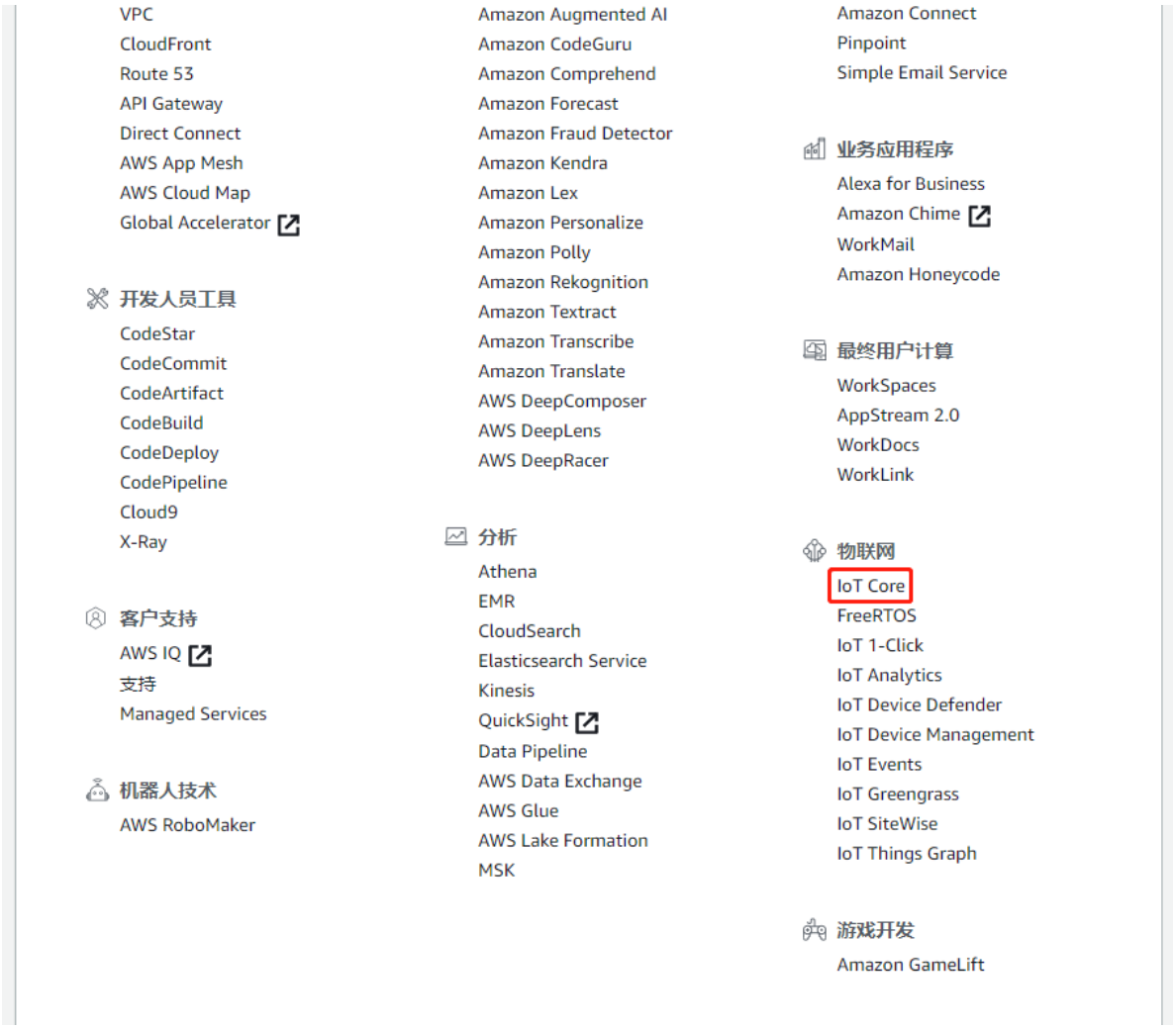
1.1 配置 AWS IoT

如果你已经在 AWS IoT Greengrass 上配置了相应的 Greengrass 组和 Greengrass 核心，可以跳过这一小节。

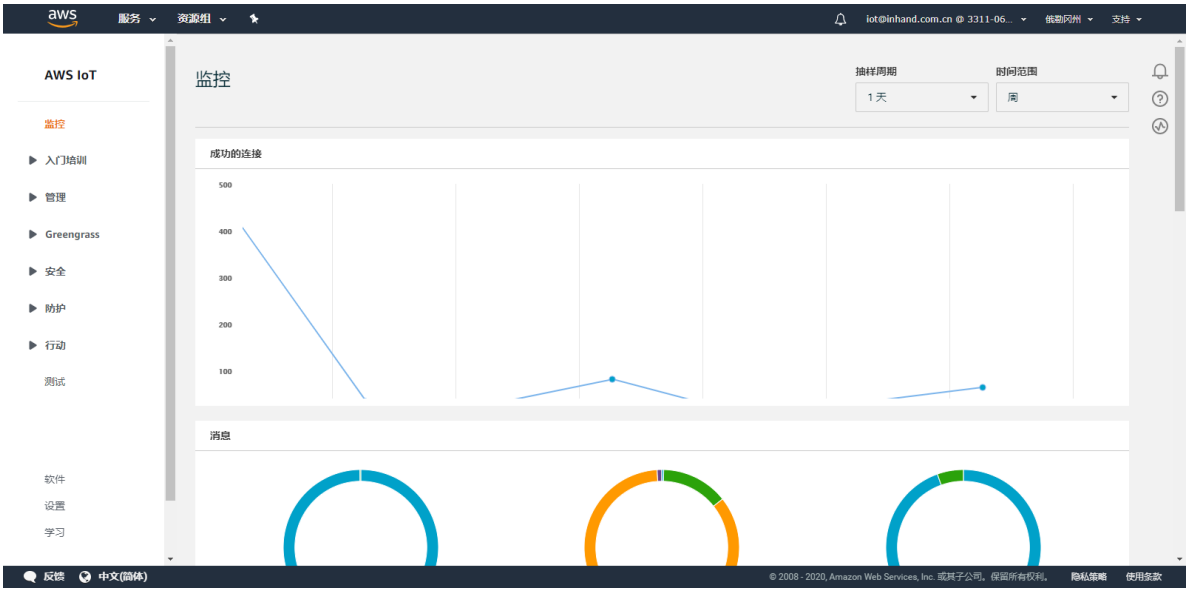
- 步骤 1：登录 AWS IoT Core

  访问 AWS 官网 https://aws.amazon.com/ 并登录控制台，登录控制台后选择 “IoT Core”。





登录“IoT Core”后页面如下所示：



- 步骤 2：配置 AWS IoT Greengrass 组

  参考在 AWS IoT 上配置 AWS IoT Greengrass创建一个 Greengrass 组并下载核心的安全资源以备后续使用（无须下载 AWS IoT Greengrass Core 软件安装包）。

- 步骤 3：创建并打包 Lambda 函数

  参考创建并打包 Lambda 函数创建并打包一个 Hello World Lambda 函数。注意：IG902 仅支持运行时为 `Python3.7` 的 Lambda 函数。

- 步骤 4：为 AWS IoT Greengrass 组配置 Lambda 函数

  参考为 AWS IoT Greengrass 配置 Lambda 函数为 Greengrass 组配置 Lambda 函数。注意：IG902 不支持 Java 8 运行时，因此需要在 Greengrass 组设置页面上禁用流管理器。

1.2 配置 IG902

1.2.1 基础配置

如何配置 IG902 联网、更新软件版本等操作请参考IG902 快速使用手册。更新 AWS IoT Greengrass SDK 时，取消勾选“启用 AWS IoT Greengrass”。

随后点击“升级”按钮并选择相应的升级文件，选择后点击“确认”即可。

1.2.2 连接 AWS IoT Greengrass

进入 IG902 的“边缘计算 > 云厂商边缘计算 > AWS IoT Greengrass”页面，导入相应文件后勾选“启用 AWS IoT Greengrass”。示例配置如下：

各项参数说明如下：

- 导入 AWS IoT 根 CA：导入用于服务器身份验证的 CA 证书。AWS IoT Greengrass SDK 中已经包含 VeriSign Class 3 Public Primary G5 根 CA 证书”和”Amazon Root CA 1”，通常你无须再导入 CA 证书即可正常连接 AWS IoT Greengrass。如需导入其他 CA 证书，可以从这里下载相应的 CA 证书并导入（建议使用 `Amazon Root CA 1` 或 `Starfield 根 CA` 证书）。目前不支持“Amazon Root CA 3”证书。
- 导入证书和密钥：导入在 AWS IoT 上配置 AWS IoT Greengrass中创建 Greengrass 组时下载的核心的安全资源。
- 导入配置文件：你可以参考配置 AWS IoT Greengrass 核心修改 AWS IoT Greengrass 核心软件的配置文件。导入证书和密钥后会自动生成 Greengrass 核心的配置文件，通常您可以直接使用而无须修改配置文件。
- 导出配置文件：你可以通过导出配置文件导出 Greengrass 核心的配置文件。
- 导出日志文件：你可以导出 Greengrass 核心的运行日志以分析 Greengrass 核心的运行情况。

配置完成后，勾选“启用 AWS IoT Greengrass”以连接 AWS IoT Greengrass。

1.3.3 2. 部署并验证 Lambda 函数

- *2.1 部署 Lambda 函数*
- *2.2 验证 Lambda 函数是否在核心设备上运行*

2.1 部署 Lambda 函数

参考将云配置部署到核心设备从第 3 步开始将 AWS IoT Greengrass 配置部署到 IG902 上。

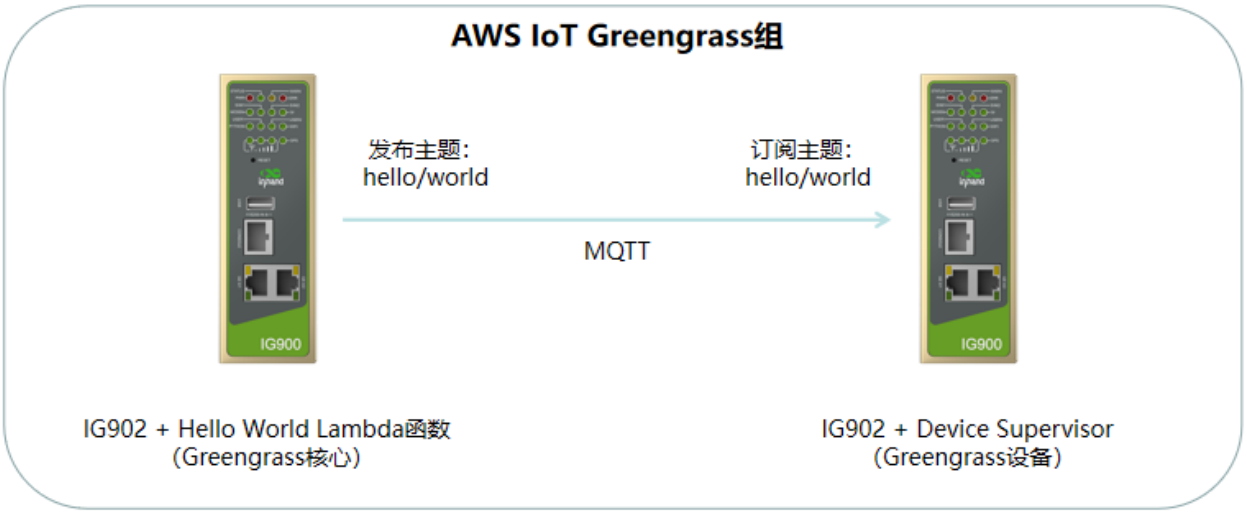
2.2 验证 Lambda 函数是否在核心设备上运行

参考验证 Lambda 函数是否在核心设备上运行测试 Lambda 函数是否已正常部署并运行在 IG902 上。

1.3.4 附录

Device Supervisor（Greengrass 设备）与 Greengrass 核心通讯

IG902 的 AWS IoT Greengrass 功能使 IG902 作为 Greengrass 核心，而运行在 IG501/902 上的 Device Supervisor 可以使 IG501/902 作为 Greengrass 设备 (需要 Device Supervisor 1.2.6 及以后版本)。配置后的 Greengrass 设备和 Greengrass 核心可以进行 MQTT 通讯以实现核心与设备的数据交互，详细说明请参考AWS IoT Greengrass 组中的设备交互。以下以一台 IG902 作为 Greengrass 核心，一台 IG902 运行 Device Supervisor 作为 Greengrass 设备并与 Greengrass 核心通讯为例进行说明（两台 IG902 处于同一局域网），整体拓扑如下：

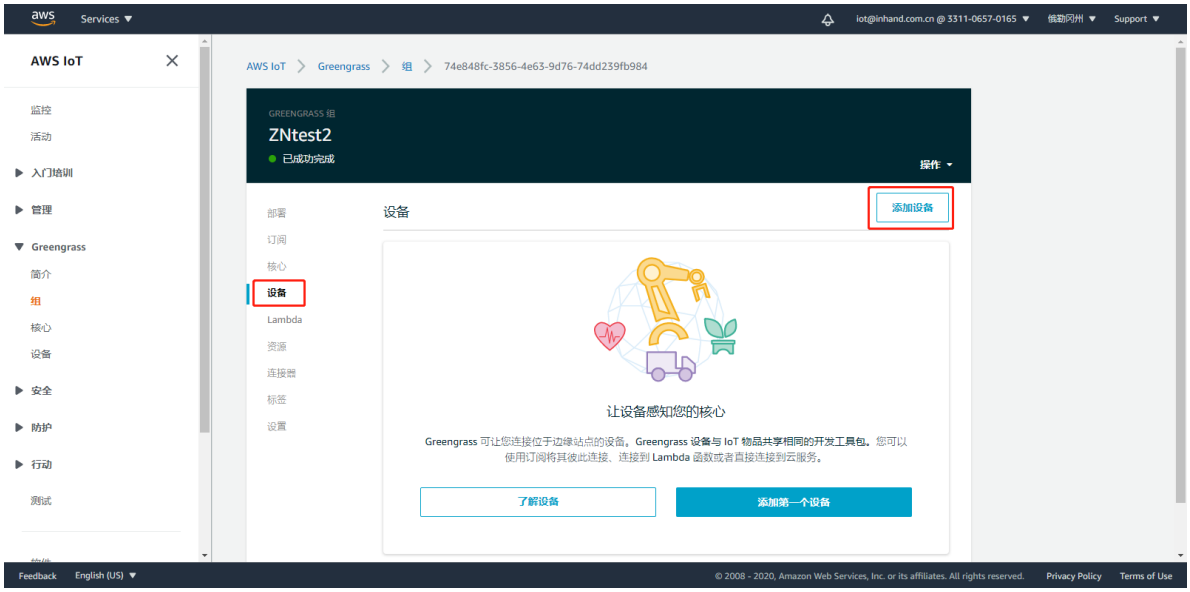


AWS IoT Greengrass 组配置

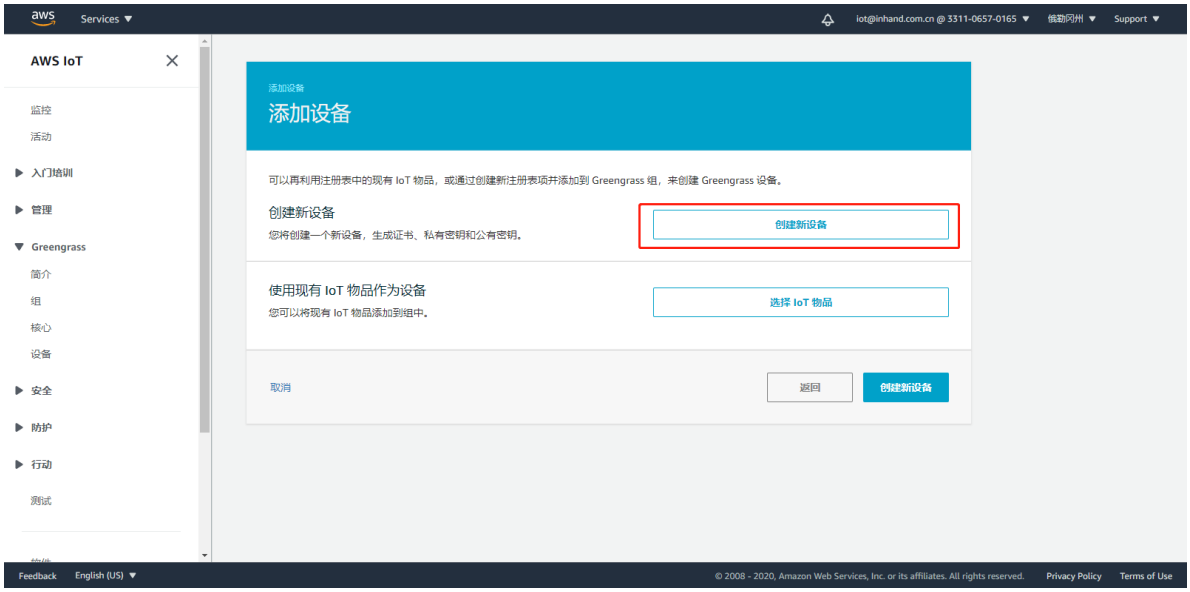
参照 *1.1 配置 AWS IoT* 配置 Greengrass 组，配置完成参考如下流程后为 Greengrass 组添加设备和相应订阅信息：

- 添加设备

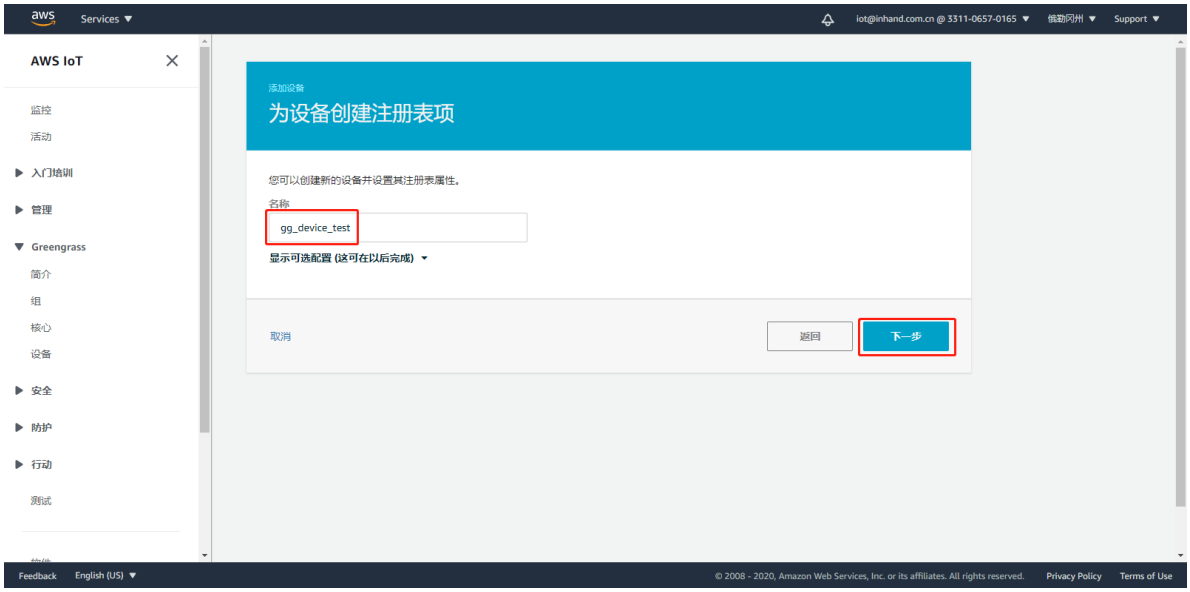
  进入“Greengrass > 组”页面，选择相应的 Greengrass 组并进入“设备”页面，点击“添加设备”。



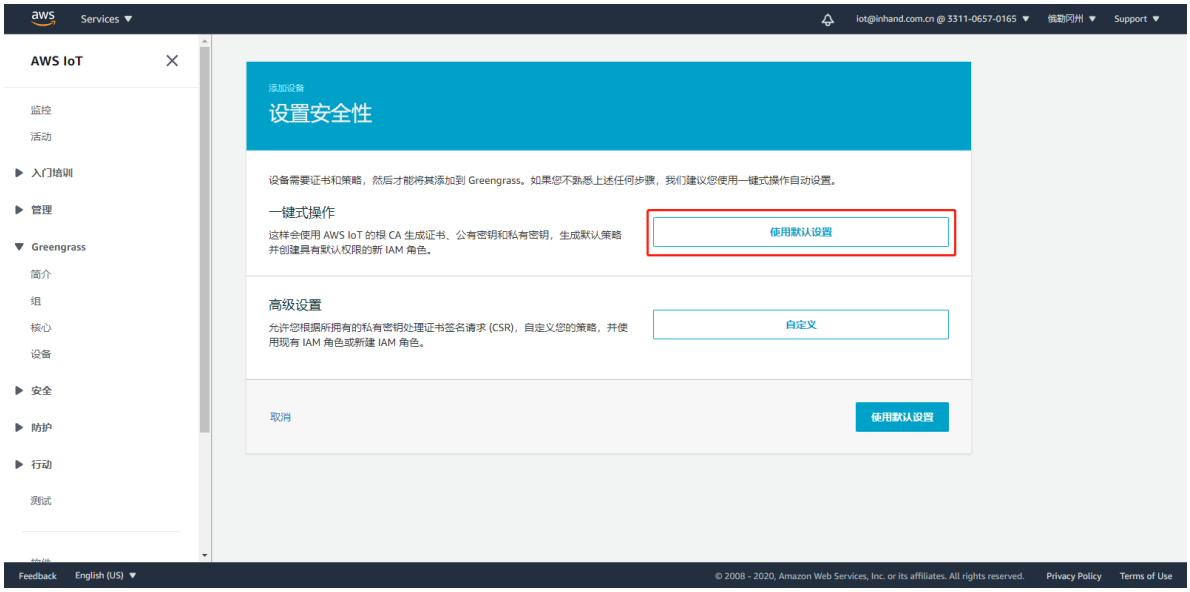
选择“创建新设备”。(不建议使用“选择 IoT 物品”，可能会导致 Greengrass 核心和 Greengrass 设备无法正常通讯)

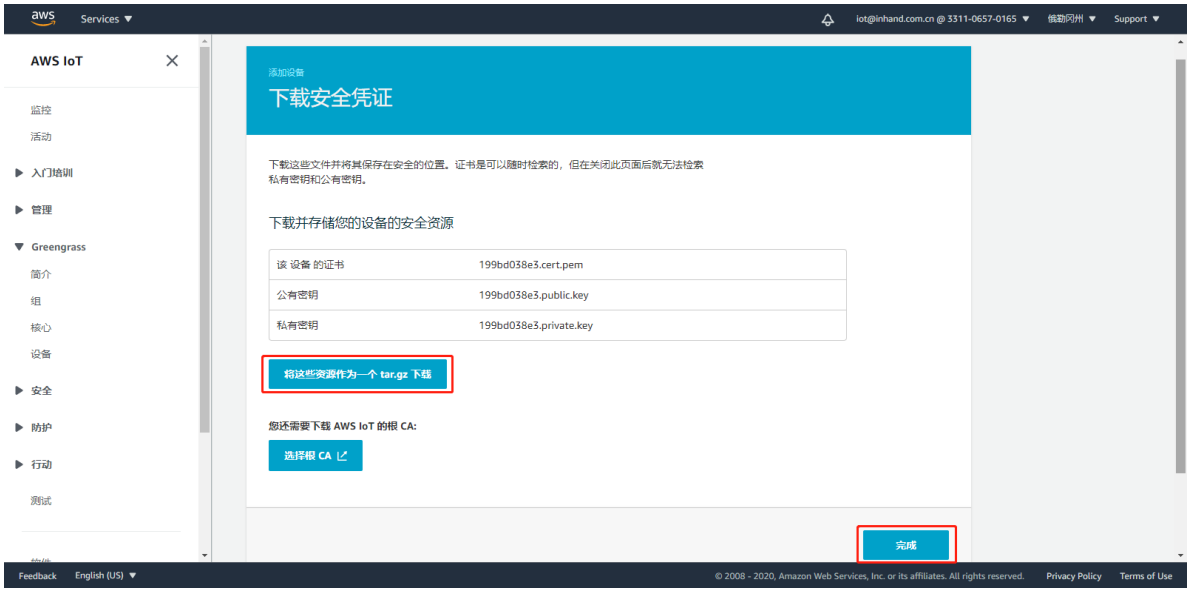


配置设备名称并点击“下一步”。



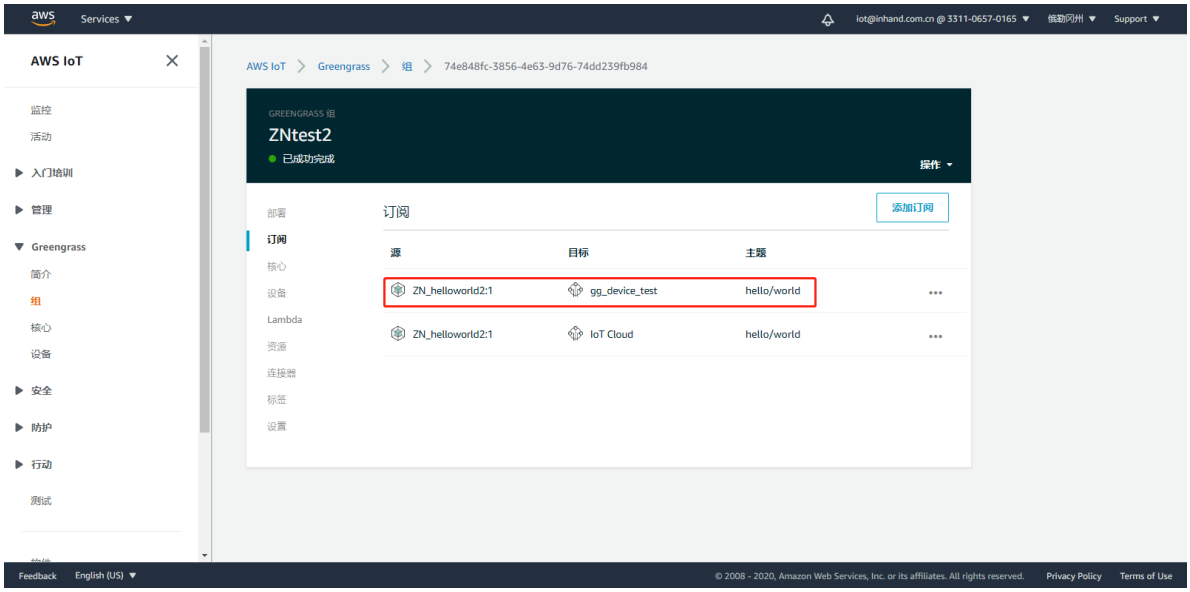
点击“使用默认设置”并下载安全凭证。





- 配置订阅消息

  在 Greengrass 组的“订阅”页面配置 Lambda 函数发送给 Greengrass 设备的订阅消息，如何配置请参考为 AWS IoT Greengrass 配置 Lambda 函数。



配置完成后，参考将云配置部署到核心设备从第 3 步开始将 AWS IoT Greengrass 配置部署到 IG902 上。

Device Supervisor 配置

- 配置与 Greengrass 核心的连接

  参考AWS IoT 使用说明中的第一章在 IG902 配置 Device Supervisor 正常运行。随后进入“边缘计算 > 设备监控 > 云服务”页面配置与 AWS IoT Greengrass 核心的连接。示例配置如下：

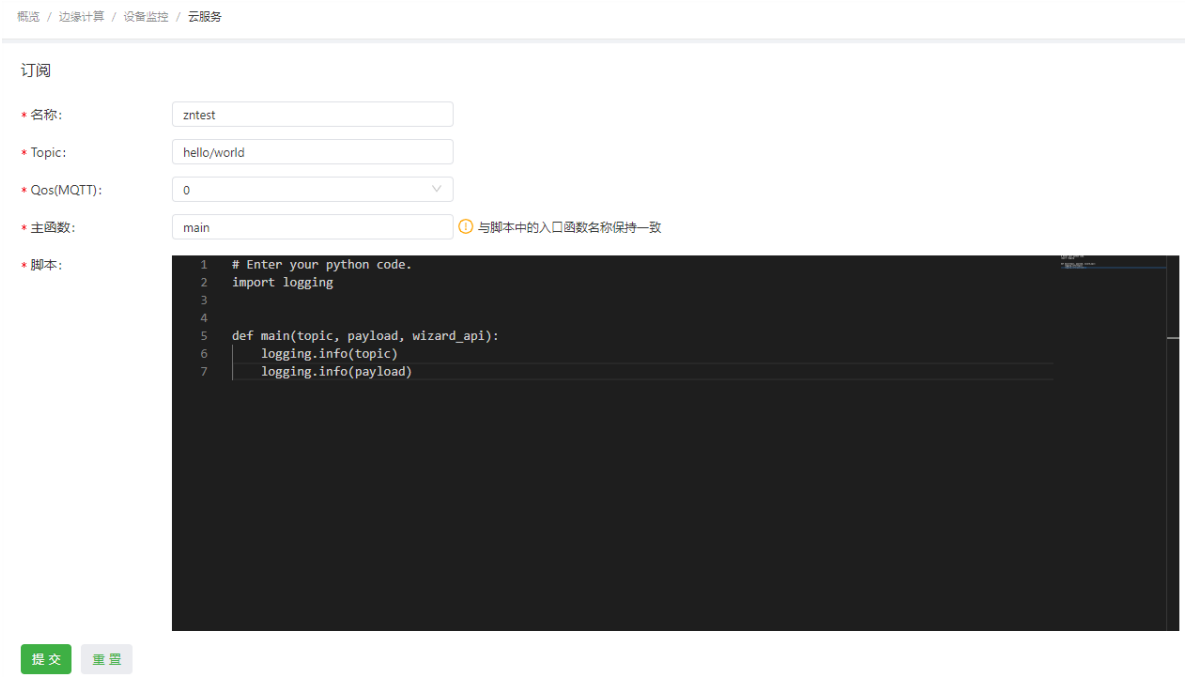
各项参数说明如下：

- 类型：连接 AWS IoT 时，选择“AWS IoT”
- 终端节点类型：与 Greengrass 核心通讯时选择“Greengrass Core”
- 终端节点：AWS IoT 的终端节点地址，可从 AWS IoT 的“设置”页面获取。使用“VeriSign Class 3 Public Primary G5 根 CA 证书”时，请删除地址中的“-ats”

- MQTT 客户端 ID：Greengrass 组中的设备名称
- 物品的证书：导入创建物品时下载的压缩包中的物品证书
- 私有密钥：导入创建物品时下载的压缩包中的私有秘钥（后缀为“private.key”）
- CA 证书：导入用于服务器身份验证的 CA 证书，你可以从这里下载相应的 CA 证书（建议使用 Amazon Root CA 1 或 Starfield 根 CA 证书）。目前不支持“Amazon Root CA 3”证书
- 其余项使用默认配置即可

- 配置与 Greengrass 核心的通讯

  在订阅消息中增加一条如下订阅并将接收到的数据打印到 App 日志中。



在“Python 边缘计算”页面点击查看“device_supervisor”的运行日志，可以看到日志中打印 Hello World

Lambda 函数发布的“hello/world”主题的负载信息。