



用户指南

# AWS IoT SiteWise



# AWS IoT SiteWise: 用户指南

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

# Table of Contents

什么是 AWS IoT SiteWise ? .....	1
工作方式 .....	1
摄取工业数据 .....	2
对资产进行建模，将收集的数据情境化 .....	3
使用查询、警报和预测进行分析 .....	3
可视化操作 .....	4
存储数据 .....	4
与其他 服务集成 .....	4
概念 .....	4
使用案例 .....	8
制造业 .....	8
食品和饮料 .....	9
能源与公用事业 .....	9
开始使用 .....	10
要求 .....	10
设置一个 AWS 账户 .....	11
注册获取 AWS 账户 .....	11
创建具有管理访问权限的用户 .....	11
使用快速入门演示 .....	12
创建演 AWS IoT SiteWise 示 .....	13
删除演 AWS IoT SiteWise 示 .....	15
教程 .....	16
计算 OEE .....	16
先决条件 .....	16
如何计算 OEE .....	17
从事物中摄取数据 AWS IoT .....	18
先决条件 .....	19
步骤 1：创建策略 .....	20
第 2 步：创建 AWS IoT 事物 .....	22
步骤 3：创建设备资产模型 .....	24
步骤 4：创建设备队列 .....	25
第 5 步：表示设备 .....	26
步骤 6：代表设备群 .....	27
步骤 7：向设备发送数据 .....	28

步骤 8：设备客户端脚本 .....	30
步骤 9：清理资源 .....	37
在 Monitor 中 SiteWise 可视化和共享数据 .....	38
先决条件 .....	39
步骤 1：创建门户 .....	40
步骤 2：登录门户 .....	44
步骤 3：创建项目 .....	45
步骤 4：创建仪表盘 .....	48
第 5 步：浏览门户 .....	55
步骤 6：清理资源 .....	56
将属性值更新发布到 Amazon DynamoDB .....	58
先决条件 .....	59
步骤 1：配置 AWS IoT SiteWise 以发布属性值更新 .....	59
步骤 2：创建规则 .....	61
步骤 3：创建 DynamoDB 表 .....	64
步骤 4：配置规则操作 .....	66
步骤 5：浏览数据 .....	67
步骤 6：清理资源 .....	68
正在将数据摄取到 AWS IoT SiteWise .....	72
管理数据流 .....	72
管理数据流 .....	73
使用 AWS IoT SiteWise API .....	80
使用 AWS IoT Core 规则 .....	83
授予所需的访问权限 .....	83
配置 规则操作 .....	84
通过基本提取降低成本 .....	92
使用 AWS IoT Events 动作 .....	93
使用 AWS IoT Greengrass 直播管理器 .....	93
使用 CreateBulkImportJob API .....	94
创建批量导入作业(AWS CLI) .....	95
描述批量导入作业(AWS CLI) .....	98
列出批量导入作业(AWS CLI) .....	100
使用 SiteWise 边缘网关 .....	101
要求 .....	101
要求 .....	102
创建 SiteWise Edge 网关 .....	105

创建 SiteWise Edge 网关 .....	105
在本地设备上安装 SiteWise Edge 网关软件 .....	106
启用边缘数据处理 .....	109
设置边缘功能 .....	109
在边缘处理数据 .....	111
配置发布器 .....	112
配置数据源 .....	115
配置 OPC-UA 源 .....	116
配置数据来源身份验证 .....	136
为源服务器数据选择目标 .....	139
添加合作伙伴数据来源 .....	142
安全性 .....	142
添加合作伙伴数据来源 .....	143
在你的 SiteWise 边缘网关上设置 docker .....	144
合作伙伴数据来源 .....	144
使用包 .....	145
升级包 .....	145
管理 SiteWise Edge 网关 .....	146
使用 AWS IoT SiteWise 控制台管理您的 SiteWise Edge 网关 .....	146
使用 AWS OpsHub 管理 SiteWise 边缘网关 AWS IoT SiteWise .....	147
使用本地操作系统凭据访问您的 SiteWise Edge 网关 .....	148
管理 SiteWise Edge 网关证书 .....	150
更改 SiteWise Edge 网关组件包的版本 .....	151
在西门子工业 SiteWise 边缘上运行边缘 .....	151
先决条件 .....	152
安全性 .....	152
创建配置文件 .....	153
故障排除 .....	154
联系我们 .....	155
筛选资产 .....	155
设置边缘筛选 .....	155
使用 API .....	156
适用于 AWS IoT SiteWise 边缘设备的所有可用 API .....	156
仅限边缘的 API .....	157
教程：获取资产模型列表 .....	160
备份和恢复 SiteWise Edge 网关 .....	169

指标数据的每日备份 .....	169
恢复 SiteWise Edge 网关 .....	170
恢复 AWS IoT SiteWise 数据 .....	171
验证备份和恢复是否成功 .....	172
设置 SiteWise 边缘网关 (AWS IoT Greengrass Version 1) .....	173
选择 AWS IoT Greengrass V1 SiteWise Edge 网关设备 .....	174
配置 AWS IoT Greengrass V1 SiteWise Edge 网关 .....	175
在 AWS IoT Greengrass V1 SiteWise Edge 网关上配置数据源 .....	193
工业资产建模 .....	212
资产和模型状态 .....	213
检查资产的状态 .....	214
检查资产模型或组件模型的状态 .....	216
自定义复合模型 ( 组件 ) .....	217
内联自定义复合模型 .....	218
Component-model-based 自定义复合模型 .....	220
使用路径引用自定义复合模型属性 .....	221
使用对象 ID .....	223
使用对象 UUID .....	223
使用外部 ID .....	224
创建资产模型和组件模型 .....	225
创建资产模型 .....	226
创建组件模型 .....	240
定义数据属性 .....	243
创建自定义复合模型 ( 组件 ) .....	317
创建资产 .....	320
创建资产 ( 控制台 ) .....	321
创建资产 (AWS CLI) .....	322
配置新资产 .....	323
搜索资产 .....	323
先决条件 .....	323
高级搜索开启 AWS IoT SiteWise 控制台 .....	323
将工业数据流映射到资产属性 .....	327
设置属性别名 ( 控制台 ) .....	328
设置属性别名 (AWS CLI) .....	329
更新属性值 .....	331
关联资产和取消资产关联 .....	334

关联资产和取消资产关联 ( 控制台 ) .....	334
关联和取消关联资产 ( )AWS CLI .....	335
更新资产和模型 .....	337
更新资产 .....	337
更新资产模型和组件模型 .....	338
更新自定义复合模型 ( 组件 ) .....	342
删除资产和模型 .....	345
删除资产 .....	345
删除资产模型 .....	347
使用资产和模型进行批量操作 .....	349
关键概念和术语 .....	349
支持的功能 .....	350
批量操作的先决条件 .....	350
运行批量导入任务 .....	353
运行批量导出任务 .....	355
作业进度跟踪和错误处理 .....	358
导入元数据示例 .....	363
导出元数据示例 .....	377
AWS IoT SiteWise 元数据传输任务架构 .....	379
通过警报监控数据 .....	398
警报类型 .....	398
警报状态 .....	399
警报状态属性 .....	400
在资产模型上定义警报 .....	403
定义 AWS IoT Events 警报 .....	405
定义外部警报 .....	438
为资产配置警报 .....	439
配置阈值 ( 控制台 ) .....	440
配置阈值 (AWS CLI) .....	440
配置通知设置 ( 控制台 ) .....	442
配置通知设置 (CLI) .....	443
响应警报 .....	444
响应警报 ( 控制台 ) .....	445
响应警报 (API) .....	448
获取外部警报状态 .....	448
映射外部警报状态流 .....	449

获取警报状态数据 .....	451
使用 Web 门户监控数据 .....	453
SiteWise 监控角色 .....	454
SAML 联合身份验证 .....	455
SiteWise 显示器概念 .....	456
开始使用 .....	457
创建门户 .....	458
配置门户 .....	459
邀请管理员 .....	463
添加门户用户 .....	465
创建控制面板 (CLI) .....	469
为门户启用警报 .....	474
在边缘启用您的门户 .....	477
管理门户 .....	477
更改入口的属性 .....	478
添加或删除门户管理员 .....	479
向门户管理员发送电子邮件邀请 .....	482
添加或删除门户用户 .....	482
删除门户 .....	485
使用 IoT 控制面板应用程序监控数据 .....	487
从中查询数据 AWS IoT SiteWise .....	488
查询当前资产价值 .....	488
查询资产属性的当前值 (控制台) .....	489
查询资产属性的当前价值 (AWS CLI) .....	489
查询历史资产属性值 .....	490
查询资产属性的价值历史记录 (AWS CLI) .....	491
查询资产属性聚合 .....	492
资产属性的聚合 (API) .....	492
资产属性的聚合 ()AWS CLI .....	493
AWS IoT SiteWise 查询语言 .....	494
先决条件 .....	495
查询语言参考 .....	495
与其他服务交互 .....	503
了解资产属性的 MQTT 主题 .....	503
处理资产属性通知 .....	504
启用资产属性通知 (控制台) .....	504



启用资产属性通知 (AWS CLI) .....	505
查询资产属性通知消息 .....	507
将数据导出到 Amazon S3 .....	509
创建堆 AWS CloudFormation 栈 .....	510
在 Amazon S3 中查看您的数据 .....	512
分析导出的数据 .....	513
模板资源已创建 .....	520
与 Grafana 集成 .....	523
集成 AWS IoT TwinMaker .....	525
启用集成 .....	526
集成 AWS IoT SiteWise 和 AWS IoT TwinMaker .....	526
检测设备异常 .....	527
添加预测定义 (控制台) .....	528
训练预测 (控制台) .....	531
启动或停止对预测的推理 (控制台) .....	532
添加预测定义 (CLI) .....	533
训练预测并开始推理 (CLI) .....	536
训练预测 (CLI) .....	537
开始或停止对预测的推理 (CLI) .....	539
管理数据存储 .....	542
配置存储设置 .....	542
数据保留的影响 .....	543
配置暖层 (控制台) 的存储设置 .....	543
为温层配置存储设置 (AWS CLI) .....	544
配置冷层 (控制台) 的存储设置 .....	548
配置冷层的存储设置 (AWS CLI) .....	550
存储设置疑难解答 .....	555
错误：存储桶不存在 .....	555
错误：对 Amazon S3 路径的访问被拒绝 .....	555
错误：无法代入角色 ARN .....	556
错误：无法访问跨区域 Amazon S3 存储桶 .....	556
冷层中保存的数据的文件路径和架构 .....	556
设备数据 (测量) .....	556
指标、转换和聚合 .....	560
资产元数据 .....	563
资产层次结构元数据 .....	567

存储数据索引文件 .....	569
安全性 .....	570
数据保护 .....	570
互连网络流量隐私保护 .....	571
数据加密 .....	572
静态加密 .....	572
传输中加密 .....	574
密钥管理 .....	576
Identity and Access Management .....	577
受众 .....	578
使用身份进行身份验证 .....	578
如何 AWS IoT SiteWise 与 IAM 配合使用 .....	581
托管策略 .....	596
服务相关角色 .....	599
为警示设置权限 .....	612
防止跨服务混淆代理 .....	618
故障排除 .....	619
合规性验证 .....	621
韧性 .....	621
基础设施安全性 .....	622
配置和漏洞分析 .....	622
VPC 端点 .....	623
支持的 API 操作 .....	623
创建接口 VPC 终端节点 .....	626
AWS IoT SiteWise 通过接口 VPC 终端节点进行访问 .....	626
创建 VPC 端点策略 .....	628
安全最佳实操 .....	629
在 OPC-UA 服务器上使用身份验证凭证 .....	629
为 OPC-UA 服务器使用加密通信模式 .....	629
保持组件更新 .....	629
加密您的 SiteWise Edge 网关的文件系统 .....	629
安全访问您的边缘配置 .....	630
向 SiteWise 监控器用户授予可能的最低权限 .....	630
不要暴露敏感信息 .....	630
遵循 AWS IoT Greengrass 安全最佳实践 .....	630
另请参阅 .....	630

日记账记录和监控 .....	632
监控服务日志 .....	632
管理登录 AWS IoT SiteWise .....	634
示例：AWS IoT SiteWise 日志文件条目 .....	635
监控 SiteWise 边缘网关日志 .....	636
使用 Amazon CloudWatch 日志 .....	636
使用服务日志 .....	638
使用事件日志 .....	639
使用 Amazon CloudWatch 指标进行监控 .....	642
AWS IoT Greengrass Version 2 网关指标 .....	642
AWS IoT Greengrass Version 1 网关指标 .....	649
使用 AWS CloudTrail 记录 API 调用 .....	654
AWS IoT SiteWise 信息在 CloudTrail .....	654
AWS IoT SiteWise 中的数据事件 CloudTrail .....	655
AWS IoT SiteWise 中的管理事件 CloudTrail .....	657
示例：AWS IoT SiteWise 日志文件条目 .....	657
标记资源 .....	659
在中使用标签 AWS IoT SiteWise .....	659
用 tagging AWS Management Console .....	659
使用 API 进行标 AWS IoT SiteWise 记 .....	659
在 IAM 策略中使用标签 .....	660
故障排除 .....	662
批量导入和导出疑难解答 .....	662
对门户进行故障排除 .....	663
用户和管理员无法访问 AWS IoT SiteWise 门户 .....	663
网关故障排除 .....	664
配置和访问 SiteWise Edge 网关日志 .....	664
排除 SiteWise Edge 网关问题 .....	665
疑难解答 AWS IoT Greengrass 问题 .....	667
对 AWS IoT SiteWise 规则操作进行故障排除 .....	667
配置 AWS IoT Core 日志 .....	668
配置重新发布错误操作 .....	669
排查问题 .....	671
排查规则问题 .....	672
排查规则问题 .....	673
端点和限额 .....	678

---

端点 .....	678
data.iotsitewise.region.amazonaws.com .....	678
api.iotsitewise.region.amazonaws.com .....	678
iotsitewise.region.amazonaws.com .....	679
model.iotsitewise.region.amazonaws.com .....	679
edge.iotsitewise.region.amazonaws.com .....	679
monitor.iotsitewise.region.amazonaws.com .....	679
配额 .....	679
异常检测配额 .....	690
文档历史记录 .....	691
AWS 术语表 .....	704
.....	dccv

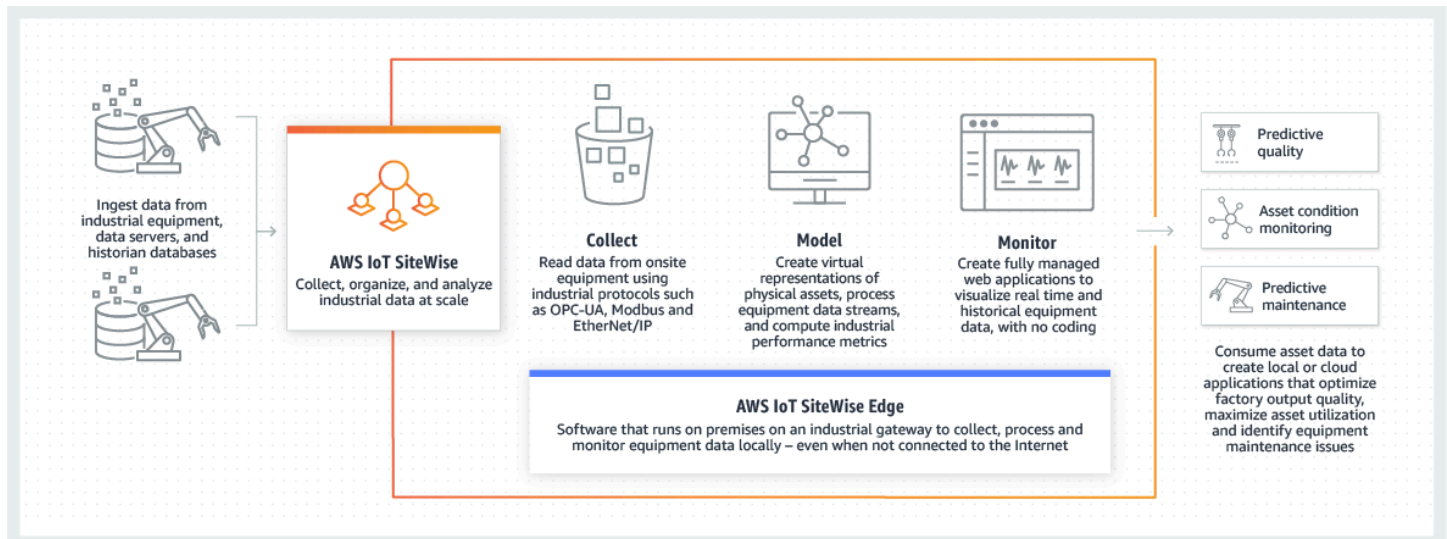
# 什么是 AWS IoT SiteWise ?

AWS IoT SiteWise 是一项托管服务，可让您轻松地大规模收集、存储、组织和监控来自工业设备的数据，从而帮助您做出更好的数据驱动型决策。您可以使用 AWS IoT SiteWise 监控各设施的操作，快速计算常见的工业性能指标，并创建用于分析工业设备数据的应用程序，以防止出现代价高昂的设备问题并缩小生产缺口。

AWS IoT SiteWise Monitor 使您的操作用户能够快速创建 Web 应用程序，以实时查看和分析您的工业数据。您可以通过配置和监控指标（如两次故障之间的平均时间和整体设备效率 (OEE)）了解您的工业运营情况。

AWS IoT SiteWise Edge 是其中的一个组件 AWS IoT SiteWise，它允许在本地设备上收集、存储和处理数据。如果您的互联网访问权限有限或需要对数据保密，则此功能非常有用。

下图显示了的基本架构 AWS IoT SiteWise：



## 主题

- [如何 AWS IoT SiteWise 运作](#)
- [AWS IoT SiteWise 概念](#)
- [的用例 AWS IoT SiteWise](#)

## 如何 AWS IoT SiteWise 运作

AWS IoT SiteWise 提供了一个资源建模框架，可用于创建工业设备、过程和设施的表示形式。您的设备和流程的表示形式在中称为资产模型 AWS IoT SiteWise。使用资产模型，您可以定义要使用的原始

数据以及如何将其处理为有用的指标。在[AWS IoT SiteWise 控制台](#)中为您的工业运营构建和可视化资产和模型。您还可以配置资产模型，以便在边缘或 AWS 云端收集和處理数据。

## 主题

- [摄取工业数据](#)
- [对资产进行建模，将收集的数据情境化](#)
- [使用查询、警报和预测进行分析](#)
- [可视化操作](#)
- [存储数据](#)
- [与其他 服务集成](#)

## 摄取工业数据

AWS IoT SiteWise 通过摄取工业数据开始使用。提取数据的方式有以下几种：

- 直接从现场服务器获取：利用 OPC-UA 等协议直接从现场设备读取数据。在各种平台（例如常见的工业网关或虚拟服务器）上部署与 AWS IoT Greengrass V2 之兼容的 SiteWise Edge 网关软件。您最多可以将 100 台 OPC-UA 服务器连接到单个 AWS IoT SiteWise 网关。有关更多信息，请参阅 [SiteWise 边缘网关要求](#)。

请注意，在 Modbus TCP 和以太网/IP (EIP) 等协议的背景下，我们与 Domatica 之合作。AWS IoT Greengrass V2

- 使用包进行边缘数据处理：通过添加包来增强您的 SiteWise 边缘网关，以启用全面的边缘功能。启用 SiteWise Edge 后 AWS IoT Greengrass V2，数据处理直接在现场执行，然后使用数据 AWS IoT Greengrass 流安全地传输到 AWS 云端。有关更多信息，请参阅 [使用包](#)。
- 通过 @@ 批量操作通过 Amazon S3 进行自适应摄取：处理大量资产或资产模型时，使用批量操作从 Amazon S3 存储桶中批量导入和导出资源。有关更多信息，请参阅 [使用资产和模型进行批量操作](#)。
- 带有 AWS IoT 核心规则的 MQTT 消息：对于连接到 AWS IoT Core 发送 MQTT 消息的设备，请使用 AWS IoT 核心规则引擎将这些消息定向到 AWS IoT SiteWise。如果您有设备连接到 AWS IoT Core 发送 [MQTT](#) 消息，请使用 AWS IoT 核心规则引擎将这些消息路由到。AWS IoT SiteWise 有关更多信息，请参阅 [使用规则摄取数据 AWS IoT Core](#)。
- 事件触发的数据提取：使用 AWS IoT Events 操作在中配置物联 SiteWise 网操作，AWS IoT Events 以便在事件发生时向其发送数据。AWS IoT SiteWise 有关更多信息，请参阅 [正在从中提取数据 AWS IoT Events](#)。

- AWS IoT SiteWise API：您在边缘或云端的应用程序可以直接将数据发送到 AWS IoT SiteWise。有关更多信息，请参阅 [使用 API 摄取数据 AWS IoT SiteWise](#)。

## 对资产进行建模，将收集的数据情境化

摄取数据后，您可以使用这些数据通过构建物理操作模型来创建资产、流程和设施的虚拟表示形式。代表设备或进程的资产将数据流传输到 AWS 云端。资产也可以表示逻辑设备分组。层次结构是通过关联资产以镜像复杂操作而形成的。这些层次结构允许资产访问关联子资产中的数据。资产是根据资产模型创建的。资产模型是标准化资产格式的声明性结构。重复使用资产组件，以实现模型的组织性和可维护性。有关更多信息，请参阅 [工业资产建模](#)。

借 AWS IoT SiteWise 助，您可以配置资产，将传入的数据转换为上下文指标并进行转换。

- 接收设备数据时，变换会起作用。
- 指标按您定义的时间间隔进行计算。

指标和转换既适用于单个资产，也适用于多个资产。AWS IoT SiteWise 自动计算与您的设备数据、指标和变换相关的不同时间范围内的常用统计聚合，例如平均值、总和和计数。

可以使用同步资产 AWS IoT TwinMaker。有关更多信息，请参阅 [集成 AWS IoT SiteWise 和 AWS IoT TwinMaker](#)。

## 使用查询、警报和预测进行分析

AWS IoT SiteWise 通过运行查询和设置警报来分析收集的日期。您还可以使用 Amazon Lookout 自动检测指标中的异常并确定其根本原因。

- 设置特定的警报，以便在设备或流程偏离最佳性能时提醒您的团队，确保快速识别和解决问题。有关更多信息，请参阅 [通过警报监控数据](#)。
- 使用 AWS IoT SiteWise API 操作查询资产属性的当前值、历史值以及特定时间间隔内的聚合。有关更多信息，请参阅 [从中查询数据 AWS IoT SiteWise](#)。
- 在 Amazon Lookout for Equipment 中使用异常检测来识别和可视化设备或操作条件的变化。通过异常检测，您可以确定运营的预防性维护措施。这种集成允许客户在 Amazon Lookout for E AWS IoT SiteWise equipment 之间同步数据。有关更多信息，请参阅 [使用 Amazon Lookout for Equipment 检测设备异常](#)。

## 可视化操作

设置 M SiteWise onitor，为您的运营员工创建 Web 应用程序。Web 应用程序可帮助员工将您的操作可视化。使用 IAM 身份中心或 IAM 为员工处理不同级别的访问权限。为每位员工配置唯一的登录名和权限，以查看整个工业运营的特定子集。AWS IoT SiteWise 为这些员工提供了学习如何使用 Monito SiteWise r 的[应用指南](#)。

有关可视化操作的更多信息，请参阅[使用监控数据 AWS IoT SiteWise Monitor](#)。

## 存储数据

您可以将时序存储与工业数据湖集成。AWS IoT SiteWise 有三个用于存储工业数据的存储层：

- 针对实时应用程序进行了优化的热存储层。
- 针对分析工作负载进行了优化的温存储层。
- 客户管理的冷存储层，使用 Amazon S3，用于具有高延迟容忍度的操作数据应用程序。

AWS IoT SiteWise 通过将最新数据保存在热存储层中来帮助您管理存储成本。然后，您可以定义数据保留策略，将历史数据移动到温层或冷层存储。有关更多信息，请参阅[管理数据存储](#)。

您还可以导入和导出资产元数据。有关更多信息，请参阅[资产元数据](#)。

## 与其他 服务集成

AWS IoT SiteWise 与多项 AWS 服务集成，可在 AWS 云端开发完整的 AWS IoT 解决方案。有关更多信息，请参阅[与其他 AWS 服务互动](#)。

## AWS IoT SiteWise 概念

以下是以下的核心概念 AWS IoT SiteWise：

### 聚合

聚合是 AWS IoT SiteWise 自动计算所有时间序列数据的基本指标或度量值。有关更多信息，请参阅[查询资产属性聚合](#)。

### 资产

当你 AWS IoT SiteWise 从工业设备中输入或摄取数据时，你的设备、设备和流程都显示为资产。每项资产都有相关的数据。例如，一台设备可能有序列号、位置、品牌和型号以及安装日期。它还



可能具有可用性、性能、质量、温度、压力等的时间序列值。将资产分组为层次结构，允许资产访问存储在其子资产中的数据。有关更多信息，请参阅 [工业资产建模](#)。

## 资产层次结构

设置资产层次结构以创建工业运营的逻辑表示形式。为此，请在资产模型中定义一个层次结构，并将根据该模型创建的资产与指定的层次结构相关联。父资产中的指标可以组合来自子资产属性的数据，从而允许您计算指标，从而深入了解您的整体运营或其中的特定部分。有关更多信息，请参阅 [定义资产模型层次结构](#)。

## 资产模型

每项资产都是使用资产模型制作的。资产模型是定义和标准化资产格式的结构。它们可确保同一类型的多个资产之间信息的一致性，使您能够处理代表设备组的资产中的数据。在每个资产模型中，您可以定义[属性](#)、时间序列输入（[测量值](#)）、时间序列转换（[转换](#)）、时间序列聚合（[指标](#)）和[资产层次结构](#)。有关更多信息，请参阅 [工业资产建模](#)。

通过为边缘配置资产模型，决定资产模型属性的处理位置。利用此功能来处理 and 监控本地设备上的资产数据。

## 资产属性

资产属性是每项资产中保存工业数据的结构。每个属性都有一种数据类型，也可以有一个单位。特性可以是[属性](#)、[测量值](#)、[转换](#)或[指标](#)。有关更多信息，请参阅 [定义数据属性](#)。

配置要在边缘计算的资产属性。有关在边缘处理数据的更多信息，请参阅 [the section called “启用边缘数据处理”](#)。

## 属性

属性是资产的属性，通常保持不变，例如设备制造商或设备位置。属性可以有预设值。根据资产模型创建的每项资产都包含该模型中定义的属性的默认值。有关更多信息，请参阅 [定义静态数据（属性）](#)。

## 控制面板

每个项目都包含一组控制面板。控制面板为一组资产的值提供一组可视化效果。项目所有者创建控制面板及其包含的可视化效果。当项目所有者准备好共享控制面板集时，所有者可以邀请查看者加入项目，从而使他们能够访问项目中的所有控制面板。如果希望为不同的控制面板使用一组不同的查看器，则必须在项目之间划分控制面板。当查看者查看仪表板时，他们可以自定义时间范围以查看特定数据。

## 数据流

AWS IoT SiteWise 甚至在创建资产模型和资产之前，就输入或摄取工业数据。AWS IoT SiteWise 自动生成数据流以从您的设备收集原始数据流。

### 数据流别名

数据流别名可帮助您轻松识别数据流。例如，别名 `server1-windfarm/3/turbine/7/temperature` 表示来自风力发电场 #3 中涡轮机 #7 的温度值。该术语 `server1` 是帮助识别 OPC-UA 服务器的数据源名称，`server1-` 也是附加到从此 OPC-UA 服务器报告的所有数据流的前缀。

### 数据流关联

创建资产模型和资产后，将数据流与资产中定义的资产属性相关联以构建数据。AWS IoT SiteWise 然后可以使用资产模型和资产来处理来自数据流的传入数据。您还可以取消数据流与资产属性的关联。有关更多信息，请参阅 [管理数据流](#)。

### 公式

每个 [转换](#) 和 [指标](#) 属性都附带一个公式，该公式概述了该属性如何转换或聚合数据。这些公式包括由提供的属性输入、运算符和函数 AWS IoT SiteWise。有关更多信息，请参阅 [使用公式表达式](#)。

### 测量值

测量是资产的属性，用于描述来自设备或设备的原始传感器时间序列数据流。有关更多信息，请参阅 [定义来自设备的数据流（测量）](#)。

### 指标

指标是表示聚合时间序列数据的资产的属性。每个指标都附有一个 [数学表达式（公式）](#)，该表达式概述了如何聚合数据点以及计算该聚合的时间间隔。指标在每个指定的时间间隔内生成一个数据点。有关更多信息，请参阅 [聚合来自属性和其他资产（指标）的数据](#)。

### 包

SiteWise 边缘网关使用数据包来确定如何收集、处理和路由数据。目前，AWS IoT SiteWise 支持数据采集包和数据处理包。有关 SiteWise Edge 网关可用包的更多信息，请参阅 [the section called “使用包”](#)。

#### 数据收集包

使用数据收集包，这样您的 SiteWise 边缘网关就可以收集您的工业数据并将其路由到您选择的 AWS 目的地。此包会自动添加到您的 SiteWise Edge 网关，无法删除。

#### 数据处理包

使用数据处理包在边缘处理您的数据，并将其保留 30 天以供本地应用程序使用。

## 门户

AWS IoT SiteWise Monitor 门户是一个 Web 应用程序，可用于可视化和共享 AWS IoT SiteWise 数据。门户具有一个或多个管理员，并且包含零个或多个项目。

### 门户管理员

每个 SiteWise 监控器门户都有一个或多个门户管理员。门户管理员使用门户创建包含资产和控制面板集合的项目。门户管理员随后会将资产和拥有者分配给每个项目。通过控制对项目的访问，门户管理员可以指定项目拥有者和查看者可以查看哪些资产。

### 项目

每个 M SiteWise onitor 门户都包含一组项目。每个项目都有一个与之关联的 AWS IoT SiteWise 资产子集。项目拥有者创建一个或多个控制面板，以提供一致的方式来查看与这些资产关联的数据。项目拥有者可以邀请查看者加入项目，以允许他们查看项目中的资产和控制面板。该项目是 Monitor SiteWise r 内部共享的基本单元。项目所有者可以邀请 AWS 管理员授予门户访问权限的用户。用户必须具有门户访问权限，然后才能与该用户共享该门户中的项目。

### 项目拥有者

每个 M SiteWise onitor 项目都有所有者。项目所有者以控制面板形式创建可视化效果，来一致地呈现您的操作数据。当控制面板准备好共享时，项目所有者可以邀请查看者加入项目。项目所有者还可以将其他所有者分配给项目。项目所有者可以为警报配置阈值和通知设置。

### 项目查看者

每个 M SiteWise onitor 项目都有查看器。项目查看者可以连接到门户以查看项目拥有者创建的控制面板。在每个控制面板中，项目查看者可以调整时间范围以更好地了解运营数据。项目查看者只能查看其具有访问权限的项目中的控制面板。项目查看者可以确认和暂停警报。

### 属性别名

您可以选择在资产属性上创建别名，例如 OPC-UA 服务器数据流路径（例如 /company/windfarm/3/turbine/7/temperature），从而简化了在摄取或检索资产数据期间资产属性的识别。当您使用 [SiteWise Edge 网关](#) 从服务器提取数据时，您的属性别名必须与原始数据流的路径相匹配。有关更多信息，请参阅 [将工业数据流映射到资产属性](#)。

### 属性通知

当您为资产属性启用属性通知时，AWS IoT Core 每当该资产收到新值时，都会向其 AWS IoT SiteWise 发布一条 MQTT 消息。消息负载包含有关该属性值更新的详细信息。使用财产价值通知创建解决方案，将您的工业数据 AWS IoT SiteWise 与其他 AWS 服务连接起来。有关更多信息，请参阅 [与其他 AWS 服务互动](#)。

## SiteWise 边缘网关

边 SiteWise 边缘网关位于客户所在地，用于收集、处理和指挥数据。SiteWise 边缘网关通过 [OPC-UA](#) 协议连接到您的工业数据源，以收集和处理数据，然后将其发送到云端。AWS SiteWise 边缘网关还可以连接到[合作伙伴的数据源](#)。SiteWise 边缘网关使用数据包进行数据收集、边缘处理等。有关可用包的更多信息，请参阅[the section called “使用包”](#)。

您可以灵活地在任何能够运行的设备或平台上创建 SiteWise Edge 网关 AWS IoT Greengrass。有关更多信息，请参阅 [使用 SiteWise 边缘网关](#)。

## 转换

变换是表示变换后的时间序列数据的资产的属性。每个转换都附带一个数学表达式（公式），用于指定如何将数据点从一种形式转换为另一种形式。转换后的数据点与输入数据点有关 one-to-one 系。有关更多信息，请参阅 [转换数据（转换）](#)。

## 可视化

在每个控制面板中，项目所有者决定如何显示与项目关联的资产的属性和警报。可用性可能表示为折线图，而其他值可能显示为条形图或关键绩效指标 (KPI)。警报最好以状态网格和状态时间轴的形式显示。项目所有者可以自定义每个可视化效果，以便更好地了解该资产的数据。

# 的用例 AWS IoT SiteWise

AWS IoT SiteWise 用于各行各业，用于许多工业数据收集和分析应用程序。

始终如一地从所有来源收集数据，以帮助快速解决问题。AWS IoT SiteWise 提供远程监控，可直接在现场收集数据，或从多个设施的多个来源收集数据。AWS IoT SiteWise 为工业物联网数据解决方案提供了必要的灵活性。

## 制造业

AWS IoT SiteWise 可以简化从设备中收集和利用数据的过程，从而精确定位并最大限度地减少效率低下现象，从而增强工业运营。AWS IoT SiteWise 帮助您从生产线和设备收集数据。借 AWS IoT SiteWise 助，您可以将数据传输到 AWS 云端，并为您的特定设备和流程建立性能指标。您可以使用生成的指标来了解运营的整体有效性，并确定创新和改进的机会。您还可以查看您的制造过程，并识别设备和工艺缺陷、生产缺口或产品缺陷。

## 食品和饮料

食品和饮料工业设施可处理各个种类的食品加工，包括将谷物磨成面粉、屠宰和包装肉类，以及组装、烹饪和冷冻适于微波炉烧煮的餐食。食品加工厂通常跨越多个地点，工厂和设备操作员集中在同一地点，以监控过程和设备。例如，制冷装置评估原料的处理和有效期。他们监控各设施的废物产生，以确保运营效率。借 AWS IoT SiteWise 助，您可以按生产线和设施对来自多个位置的传感器数据流进行分组，以便您的过程工程师可以更好地了解各个设施并进行改进。

## 能源与公用事业

借 AWS IoT SiteWise 助，您可以更轻松、更高效地解决设备问题。您可以远程实时监控资产性能。随时随地访问历史设备数据，以查明潜在问题，调度准确的资源，并更快地预防和修复问题。

# 入门 AWS IoT SiteWise

借助 AWS IoT SiteWise，您可以收集、组织、分析和可视化您的数据。

AWS IoT SiteWise 提供了一个演示，您无需配置真实数据源即可使用该演示来探索服务。有关更多信息，请参阅 [使用演示 AWS IoT SiteWise](#)。

您可以完成以下教程来探索以下各项的某些功能 AWS IoT SiteWise：

- [从事物中摄取数据 AWS IoT](#)
- [在 Monitor 中可视化和共享风电场 SiteWise 数据](#)
- [将属性值更新发布到 Amazon DynamoDB](#)

要了解更多信息，请参阅以下主题 AWS IoT SiteWise：

- [正在将数据摄取到 AWS IoT SiteWise](#)
- [工业资产建模](#)
- [启用边缘数据处理](#)
- [使用监控数据 AWS IoT SiteWise Monitor](#)
- [从中查询数据 AWS IoT SiteWise](#)
- [与其他 AWS 服务互动](#)

## 主题

- [要求](#)
- [设置一个 AWS 账户](#)
- [使用演示 AWS IoT SiteWise](#)

## 要求

您必须有一个 AWS 账户 可以开始使用 AWS IoT SiteWise。如果没有，请参阅[设置一个 AWS 账户](#)。

使用可 AWS IoT SiteWise 用的区域。有关更多信息，请参阅 [AWS IoT SiteWise 终端节点和限额](#)。您可以使用中的区域选择器切换 AWS Management Console 到其中一个区域。

# 设置一个 AWS 账户

## 主题

- [注册获取 AWS 账户](#)
- [创建具有管理访问权限的用户](#)

## 注册获取 AWS 账户

如果您没有 AWS 账户，请完成以下步骤来创建一个。

### 报名参加 AWS 账户

1. 打开 <https://portal.aws.amazon.com/billing/signup>。
2. 按照屏幕上的说明进行操作。

在注册时，将接到一通电话，要求使用电话键盘输入一个验证码。

当您注册时 AWS 账户，就会创建 AWS 账户根用户一个。根用户有权访问该账户中的所有 AWS 服务和资源。作为安全最佳实践，请为用户分配管理访问权限，并且只使用根用户来执行[需要根用户访问权限的任务](#)。

AWS 注册过程完成后会向您发送一封确认电子邮件。在任何时候，您都可以通过转至 <https://aws.amazon.com/> 并选择我的账户来查看当前的账户活动并管理您的账户。

## 创建具有管理访问权限的用户

注册后，请保护您的安全 AWS 账户 AWS 账户根用户 AWS IAM Identity Center，启用并创建管理用户，这样您就不会使用 root 用户执行日常任务。

### 保护你的 AWS 账户根用户

1. 选择 Root 用户并输入您的 AWS 账户 电子邮件地址，以账户所有者的身份登录。[AWS Management Console](#)在下一页上，输入您的密码。

要获取使用根用户登录方面的帮助，请参阅《AWS 登录 用户指南》中的[以根用户身份登录](#)。

2. 为您的根用户启用多重身份验证 (MFA)。

有关说明，请参阅 [IAM 用户指南中的为 AWS 账户 根用户启用虚拟 MFA 设备 \(控制台\)](#)。

## 创建具有管理访问权限的用户

### 1. 启用 IAM Identity Center

有关说明，请参阅《AWS IAM Identity Center 用户指南》中的[启用 AWS IAM Identity Center](#)。

### 2. 在 IAM Identity Center 中，为用户授予管理访问权限。

有关使用 IAM Identity Center 目录 作为身份源的教程，请参阅《[用户指南](#)》IAM Identity Center 目录中的[使用默认设置配置AWS IAM Identity Center 用户访问权限](#)。

## 以具有管理访问权限的用户身份登录

- 要使用您的 IAM Identity Center 用户身份登录，请使用您在创建 IAM Identity Center 用户时发送到您的电子邮件地址的登录网址。

有关使用 IAM Identity Center 用户[登录的帮助](#)，请参阅[AWS 登录 用户指南中的登录 AWS 访问门户](#)。

## 将访问权限分配给其他用户

### 1. 在 IAM Identity Center 中，创建一个权限集，该权限集遵循应用最低权限的最佳做法。

有关说明，请参阅《AWS IAM Identity Center 用户指南》中的[创建权限集](#)。

### 2. 将用户分配到一个组，然后为该组分配单点登录访问权限。

有关说明，请参阅《AWS IAM Identity Center 用户指南》中的[添加组](#)。

## 使用演示 AWS IoT SiteWise 示

您可以使用 AWS IoT SiteWise 演示 AWS IoT SiteWise 轻松探索。AWS IoT SiteWise 将演示作为 AWS CloudFormation 模板提供，您可以部署该模板来创建资产模型、资产和 M SiteWise onitor 门户，并生成长达一周的示例数据。

### Important

创建演示后，您将开始为该演示创建和使用的资源付费。



## 主题

- [创建演 AWS IoT SiteWise 示](#)
- [删除演 AWS IoT SiteWise 示](#)

## 创建演 AWS IoT SiteWise 示

您可以从 AWS IoT SiteWise 控制台创建 AWS IoT SiteWise 演示。

### Note

该演示创建了 Lambda 函数、一个 CloudWatch 事件规则和演示所需的 AWS Identity and Access Management (IAM) 角色。您可能在中看到这些资源 AWS 账户。我们建议您保留这些资源，直到您完成演示。如果您删除这些资源，演示可能会停止正常工作。

在 AWS IoT SiteWise 控制台中创建演示

1. 导航到[AWS IoT SiteWise 控制台](#)，在页面的右上角找到 SiteWise 演示。
2. (可选) 在 SiteWise demo 下，更改保留演示资产的天数字段，以指定在删除演示之前保留演示的天数。
3. (可选) 要创建用于 SiteWise 监控示例数据的监控器入口，请执行以下操作。

### Note

您需要为本演示创建和使用的 SiteWise 监控器资源付费。有关更多信息，请参阅“AWS IoT SiteWise 定价”中的“[SiteWise 监控](#)”。

- a. 请选择监控资源。
- b. 请选择权限。
- c. 请选择一个现有 IAM 角色来授予您的联合 IAM 用户访问门户的权限。

### Important

您的 IAM 角色必须包括以下权限。

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "iotsitewise:Describe*",
      "iotsitewise:List*",
      "iotsitewise:Get*",
      "cloudformation:DescribeStacks",
      "iam:GetPolicyVersion",
      "iam:GetPolicy",
      "iam:ListAttachedRolePolicies",
      "sso:DescribeRegisteredRegions",
      "organizations:DescribeOrganization"
    ],
    "Resource": "*"
  }
]
```

有关如何使用 SiteWise 监控器的更多信息，请参阅[什么是 AWS IoT SiteWise Monitor？](#) 在《AWS IoT SiteWise Monitor 应用指南》中。

#### 4. 选择 创建演示。

创建演示约需 3 分钟时间。如果创建演示失败，说明您的账户可能没有足够的权限。切换到具有管理权限的账户，或使用以下步骤删除演示并重试：

##### a. 选择 删除演示。

删除演示约需 15 分钟时间。

b. 如果演示未删除，请打开[AWS CloudFormation 控制台](#)，选择名为 IoT 的堆栈 SiteWiseDemoAssets，然后在右上角选择删除。

c. 如果演示无法再次删除，请按照 AWS CloudFormation 控制台中的步骤跳过删除失败的资源，然后重试。

#### 5. 演示创建成功后，您可以浏览 [AWS IoT SiteWise 控制台](#) 中的演示资产和数据。

## 删除演示 AWS IoT SiteWise 示

AWS IoT SiteWise 演示将在一周后自行删除，或者如果您从 AWS CloudFormation 控制台创建演示堆栈，则该演示将在您选择的天数后自行删除。如果您已完成演示资源的使用，则可以提前删除演示。如果演示创建失败，您也可以删除演示。使用以下步骤手动删除演示。

### 要删除演示 AWS IoT SiteWise 示

1. 导航到 [AWS CloudFormation 控制台](#)。
2. 从堆栈列表中选择 IoTSiteWiseDemoAssets。
3. 选择 Delete (删除)。

当您删除堆栈时，为演示创建的所有资源都将被删除。

4. 在确认对话框中，选择 删除堆栈。

删除堆栈约需 15 分钟时间。如果演示无法删除，请再次选择右上角的 删除。如果演示无法再次删除，请按照 AWS CloudFormation 控制台中的步骤跳过删除失败的资源，然后重试。

# AWS IoT SiteWise 教程

欢迎来到教 AWS IoT SiteWise 程页面。不断增加的教程集使您掌握驾驭错综复杂之处所需的知识和技能。AWS IoT SiteWise 这些教程提供了各种各样的基本主题，以满足您的需求。在深入研究教程时，发现有关各个方面的宝贵见解。AWS IoT SiteWise

每个教程都使用特定的装备示例。这些教程适用于测试环境，它们使用虚构的公司名称、模型、资产、属性等。其目的在于提供一般性指导。未经仔细审查和调整以满足您组织的独特需求，这些教程不适合在生产环境中直接使用。

## 主题

- [计算 OEE AWS IoT SiteWise](#)
- [从事物中摄取数据 AWS IoT](#)
- [在 Monitor 中可视化和共享风电场 SiteWise 数据](#)
- [将属性值更新发布到 Amazon DynamoDB](#)

## 计算 OEE AWS IoT SiteWise

本教程提供了一个示例，讲解如何计算制造过程的整体设备效率 (OEE)。因此，您的 OEE 计算或公式可能与此处显示的不同。通常，OEE 被定义为  $\text{Availability} * \text{Quality} * \text{Performance}$ 。要了解有关计算 OEE 的更多信息，请参阅维基百科上的[整体设备效率](#)。

## 先决条件

要完成本教程，您必须为具有以下三个数据流的设备配置数据引入：

- `Equipment_State` – 代表机器状态的数字代码，如闲置、故障、计划停止或正常运行。
- `Good_Count` – 一个数据流，其中每个数据点都包含自上一个数据点以来的成功操作数。
- `Bad_Count` – 一个数据流，其中每个数据点都包含自上一个数据点以来的不成功操作数。

要配置数据引入，请参阅[正在将数据摄取到 AWS IoT SiteWise](#)。如果您没有可用的工业运营，您可以编写一个脚本，通过 AWS IoT SiteWise API 生成并上传示例数据。

## 如何计算 OEE

在本教程中，您将创建一个资产模型，该模型通过三个数据输入流计算

OEE：Equipment\_State、Good\_Count 和 Bad\_Count。在此示例中，请考虑通用包装机，如用于包装糖、薯片或油漆的包装机。在[AWS IoT SiteWise 控制台](#)中，使用以下度量、变换和指标创建 AWS IoT SiteWise 资产模型。然后，您可以创建一个资产来表示包装机器，并观察如何 AWS IoT SiteWise 计算 OEE。

定义以下[测量值](#)来表示来自包装机的原始数据流。

### 测量值

- Equipment\_State – 以数字代码提供包装机当前状态的数据流（或测量值）：
  - 1024 – 机器处于闲置状态。
  - 1020 – 故障，如错误或延迟。
  - 1000 – 计划的停止。
  - 1111 – 正常运行。
- Good\_Count – 一个数据流，其中每个数据点都包含自上一个数据点以来的成功操作数。
- Bad\_Count – 一个数据流，其中每个数据点都包含自上一个数据点以来的不成功操作数。

使用 Equipment\_State 测量数据流及其包含的代码，定义以下[转换](#)（或派生的测量值）。变换与原始测量值有关 one-to-one 系。

### 转换

- Idle = eq(Equipment\_State, 1024) – 包含机器闲置状态的转换后的数据流。
- Fault = eq(Equipment\_State, 1020) – 包含机器故障状态的转换后的数据流。
- Stop = eq(Equipment\_State, 1000) – 包含机器计划停止状态的转换后的数据流。
- Running = eq(Equipment\_State, 1111) – 包含机器正常运行状态的转换后的数据流。

使用原始测量值和转换的测量值，定义在指定时间间隔内聚合机器数据的以下[指标](#)。定义本节介绍的指标时，为每个指标选择相同的时间间隔。

### 指标

- Successes = sum(Good\_Count) – 在指定时间间隔内成功填充的包数。

- $Failures = \text{sum}(\text{Bad\_Count})$  – 在指定时间间隔内未成功填充的包数。
- $Idle\_Time = \text{statetime}(\text{Idle})$  – 机器在每个指定时间间隔内的总闲置时间（以秒为单位）。
- $Fault\_Time = \text{statetime}(\text{Fault})$  – 机器在每个指定时间间隔内的总故障时间（以秒为单位）。
- $Stop\_Time = \text{statetime}(\text{Stop})$  – 机器在每个指定时间间隔内的总计划停止时间（以秒为单位）。
- $Run\_Time = \text{statetime}(\text{Running})$  – 机器在每个指定时间间隔内的总正常运行时间（以秒为单位）。
- $Down\_Time = Idle\_Time + Fault\_Time + Stop\_Time$  – 机器在指定时间间隔内的总停机时间（以秒为单位），计算为除  $Run\_Time$  以外的机器状态的时间总计。
- $Availability = Run\_Time / (Run\_Time + Down\_Time)$  – 机器在指定时间间隔内可以运行的正常运行时间，或计划运行时间的百分比。
- $Quality = \text{Successes} / (\text{Successes} + \text{Failures})$  – 机器在指定时间间隔内成功填充包的百分比。
- $Performance = ((\text{Successes} + \text{Failures}) / Run\_Time) / \text{Ideal\_Run\_Rate}$  – 在指定时间间隔内，您的进程以超出理想运行速率（以秒为单位）的百分比表示的机器性能。

例如，您的  $\text{Ideal\_Run\_Rate}$  可能是每分钟 60 个包（每秒 1 个包）。如果您的  $\text{Ideal\_Run\_Rate}$  以分钟或小时为单位，则需要除以适当的单位转换系数，因为  $Run\_Time$  以秒为单位。

- $OEE = Availability * Quality * Performance$  – 机器在指定时间间隔内的整体设备效率。此公式将 OEE 计算为 1 的比例。

## 从事物中摄取数据 AWS IoT

在本教程中，学习如何使用设备影子 AWS IoT SiteWise 从一组设备中提取数据。AWS IoT 设备影子是存储 AWS IoT 设备当前状态信息的 JSON 对象。有关更多信息，请参阅 AWS IoT 开发人员指南中的[设备影子服务](#)。

完成本教程后，可以在中 AWS IoT SiteWise 根据 AWS IoT 内容设置操作。通过使用 AWS IoT 东西，您可以将您的操作与的其他有用功能集成 AWS IoT。例如，您可以配置 AWS IoT 功能来执行以下任务：

- 配置其他规则以将数据流式传输到[AWS IoT Events](#)、[Amazon DynamoDB](#) 等。AWS 服务有关更多信息，请参阅 AWS IoT 开发人员指南 中的[规则](#)。

- 使用 AWS IoT 队列索引服务对您的设备数据进行索引、搜索和汇总。有关更多信息，请参阅 AWS IoT 开发人员指南中的[实例集索引服务](#)。
- 使用来审核和保护您的设备 AWS IoT Device Defender。有关更多信息，请参阅 AWS IoT 开发人员指南中的[AWS IoT Device Defender](#)。

在本教程中，您将学习如何将数据从设备影子提取到 AWS IoT SiteWise 中的资产。AWS IoT SiteWise 为此，您需要创建一件或多件 AWS IoT 事物，然后运行一个脚本，该脚本使用 CPU 和内存使用数据更新每件事物的设备影子。在本教程中，您可以使用 CPU 利用率和内存使用量数据来模拟真实传感器数据。然后，您创建一条规则，其中包含一个 AWS IoT SiteWise 操作，该操作会在 AWS IoT SiteWise 每次事物的设备影子更新时将这些数据发送到资产。有关更多信息，请参阅[使用规则摄取数据 AWS IoT Core](#)。

## 主题

- [先决条件](#)
- [步骤 1：创建 AWS IoT 策略](#)
- [步骤 2：创建和配置 AWS IoT 事物](#)
- [步骤 3：创建设备资产模型](#)
- [步骤 4：创建设备队列资产模型](#)
- [步骤 5：创建和配置设备资产](#)
- [步骤 6：创建和配置设备队列资产](#)
- [步骤 7：在 AWS IoT Core 中创建向设备资产发送数据的规则](#)
- [步骤 8：运行设备客户端脚本](#)
- [步骤 9：在完成教程后清理资源](#)

## 先决条件

要完成本教程，您需要：

- 一个 AWS 账户。如果没有，请参阅[设置一个 AWS 账户](#)。
- 一台运行 Windows、macOS/Linux、或 Unix 访问的开发计算机 AWS Management Console。有关更多信息，请参阅[AWS Management Console 入门](#)。
- 具有管理员权限的 AWS Identity and Access Management (IAM) 用户。
- Python3 安装在您的开发计算机上或安装在要注册为 AWS IoT 事物的设备上。

## 步骤 1：创建 AWS IoT 策略

在此过程中，创建一个允许您的设备访问 AWS IoT 本教程中使用的资源的 AWS IoT 策略。

### 创建 AWS IoT 策略

1. 登录到 [AWS Management Console](#)。
2. 查看支持 [AWS IoT SiteWise 的地区](#)。如有必要，切换到其中一个受支持的区域。
3. 导航到 [AWS IoT 控制台](#)。如果出现“Connect 设备”按钮，请选择该按钮。
4. 在左侧导航窗格中，选择安全，然后选择策略。
5. 选择创建。
6. 输入 AWS IoT 策略的名称（例如，**SiteWiseTutorialDevicePolicy**）。
7. 在策略文档下，选择 JSON，以 JSON 形式输入以下政策。将 *region* 和 *account-id* 替换为您的区域和账户 ID，例如 **us-east-1** 和 **123456789012**。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:Connect",
      "Resource": "arn:aws:iot:region:account-id:client/SiteWiseTutorialDevice*"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Publish",
      "Resource": [
        "arn:aws:iot:region:account-id:topic/$aws/things/
        ${iot:Connection.Thing.ThingName}/shadow/update",
        "arn:aws:iot:region:account-id:topic/$aws/things/
        ${iot:Connection.Thing.ThingName}/shadow/delete",
        "arn:aws:iot:region:account-id:topic/$aws/things/
        ${iot:Connection.Thing.ThingName}/shadow/get"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iot:Receive",
      "Resource": [
```



```

    "arn:aws:iot:region:account-id:topic/$aws/things/
    ${iot:Connection.Thing.ThingName}/shadow/update/accepted",
    "arn:aws:iot:region:account-id:topic/$aws/things/
    ${iot:Connection.Thing.ThingName}/shadow/delete/accepted",
    "arn:aws:iot:region:account-id:topic/$aws/things/
    ${iot:Connection.Thing.ThingName}/shadow/get/accepted",
    "arn:aws:iot:region:account-id:topic/$aws/things/
    ${iot:Connection.Thing.ThingName}/shadow/update/rejected",
    "arn:aws:iot:region:account-id:topic/$aws/things/
    ${iot:Connection.Thing.ThingName}/shadow/delete/rejected"
  ]
},
{
  "Effect": "Allow",
  "Action": "iot:Subscribe",
  "Resource": [
    "arn:aws:iot:region:account-id:topicfilter/$aws/things/
    ${iot:Connection.Thing.ThingName}/shadow/update/accepted",
    "arn:aws:iot:region:account-id:topicfilter/$aws/things/
    ${iot:Connection.Thing.ThingName}/shadow/delete/accepted",
    "arn:aws:iot:region:account-id:topicfilter/$aws/things/
    ${iot:Connection.Thing.ThingName}/shadow/get/accepted",
    "arn:aws:iot:region:account-id:topicfilter/$aws/things/
    ${iot:Connection.Thing.ThingName}/shadow/update/rejected",
    "arn:aws:iot:region:account-id:topicfilter/$aws/things/
    ${iot:Connection.Thing.ThingName}/shadow/delete/rejected"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "iot:GetThingShadow",
    "iot:UpdateThingShadow",
    "iot>DeleteThingShadow"
  ],
  "Resource": "arn:aws:iot:region:account-id:thing/SiteWiseTutorialDevice*"
}
]
}

```

此策略使您的 AWS IoT 设备能够使用 MQTT 消息建立连接并与设备影子通信。有关 MQTT 消息的更多信息，请参阅[什么是 MQTT?](#)。要与设备影子进行交互，您的 AWS IoT 事物会发布和接收有关以开头的主题的 MQTT 消息。`$aws/things/thing-name/shadow/`此策略包含一个称

为“事物”策略变量 `${iot:Connection.Thing.ThingName}`。此变量在每个主题中替换关联事物的名称。该 `iot:Connect` 语句限制了哪些设备可以建立连接，从而确保事物策略变量只能替换以开头的名称 `SiteWiseTutorialDevice`。

有关更多信息，请参阅 AWS IoT 开发人员指南中的 [事物策略变量](#)。

### Note

此策略适用于名称以 `SiteWiseTutorialDevice` 开头的事物。要为您的事物使用不同的名称，您必须相应地更新策略。

## 8. 选择创建。

## 步骤 2：创建和配置 AWS IoT 事物

在此过程中，您将创建和配置一个 AWS IoT 事物。你可以将你的开发计算机指定为一个 AWS IoT 东西。随着你的进步，请记住，你在这里学习的原则可以应用于实际项目。你可以灵活地在任何能够运行 AWS IoT SDK 的设备上制作和设置 AWS IoT 东西，包括 AWS IoT Greengrass 和 FreeRTOS。有关更多信息，请参阅 AWS IoT 开发人员指南中的 [AWS IoT SDK](#)。

### 创建和配置 AWS IoT 事物

1. 打开命令行并运行以下命令，以便为本教程创建目录。

```
mkdir iot-sitewise-rule-tutorial
cd iot-sitewise-rule-tutorial
```

2. 运行以下命令为您的事物证书创建一个目录。

```
mkdir device1
```

如果您正在创建其他事物，请相应地增加目录名称中的数字，以跟踪哪些证书属于哪个事物。

3. 导航到 [AWS IoT 控制台](#)。
4. 在左侧导航窗格中，选择管理部分中的所有设备。然后选择 Things (事物)。
5. 如果显示您还没有任何事物对话框，请选择 创建事物。否则，请选择创建事物。
6. 在创建事物页面上，选择创建单个事物，然后选择下一步。

- 在指定事物属性页面上，为 AWS IoT 事物输入名称（例如 **SiteWiseTutorialDevice1**），然后选择下一步。如果您正在创建其他事物，请相应地增加事物名称中的数字。

**⚠ Important**

事物名称必须与您在步骤 1：创建策略中创建的 AWS IoT 策略中使用的名称相匹配。否则，您的设备将无法连接到 AWS IoT。

- 在配置证书 - 可选页面上，选择自动生成新证书（推荐），然后选择下一步。证书可以安全 AWS IoT 地识别您的设备。
- 在“将策略附加到证书-可选”页面上，选择您在步骤 1：创建策略中创建的 AWS IoT 策略，然后选择创建事物。
- 在下载证书与密钥对话框中，执行以下操作：
  - 选择 **下载** 链接以下载事物的证书、公钥和私钥。将所有三个文件保存到您为事物证书创建的目录（例如，`iot-sitewise-rule-tutorial/device1`）。

**⚠ Important**

这是您唯一一次可以下载在设备成功连接 AWS IoT 时所需的事物证书和密钥。

- 选择 **下载** 链接，以下载根 CA 证书。将根 CA 证书保存到 `iot-sitewise-rule-tutorial`。我们建议下载 Amazon 根 CA 1。
- 选择完成。

现在，你已经在电脑上注册了一个 AWS IoT 东西。采取以下步骤之一：

- 继续步骤 3：在不创建其他 AWS IoT 内容的情况下创建设备资产模型。您可以只使用一个事物完成本教程。
- 在另一台计算机或设备上重复本部分中的步骤以创建更多 AWS IoT 事物。在本教程中，我们建议您按照此选项操作，以便从多个设备提取唯一的 CPU 利用率和内存使用量数据。
- 在同一台设备（您的计算机）上重复此部分中的步骤以创建更多 AWS IoT 事物。每个 AWS IoT 设备都会从您的计算机接收相似的 CPU 和内存使用数据，因此请使用这种方法来演示从多个设备摄取非唯一数据。

## 步骤 3：创建设备资产模型

在此过程中，您将在中创建一个资产模型 AWS IoT SiteWise 来表示流式传输 CPU 和内存使用情况数据的设备。为了处理代表设备组的资产中的数据，资产模型要求同一类型的多个资产之间保持一致的信息。有关更多信息，请参阅 [工业资产建模](#)。

### 创建表示设备的资产模型

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在左侧导航窗格中，选择 **模型**。
3. 选择 **创建模型**。
4. 在模型详细信息下，输入您的模型的名称。例如，**SiteWise Tutorial Device Model**。
5. 在 **测量定义** 下，执行以下操作：
  - a. 在名称中，输入 **CPU Usage**。
  - b. 在单位中，输入 **%**。
  - c. 将数据类型保留为 **双精度**。

测量属性表示设备的原始数据流。有关更多信息，请参阅 [定义来自设备的数据流 \(测量\)](#)。

6. 选择 **添加测量** 以添加第二个测量属性。
7. 在 **测量定义** 下的第二行中，执行以下操作：
  - a. 在名称中，输入 **Memory Usage**。
  - b. 在单位中，输入 **%**。
  - c. 将数据类型保留为 **双精度**。
8. 在 **指标定义** 下，执行以下操作：
  - a. 在名称中，输入 **Average CPU Usage**。
  - b. 在公式中，输入 **avg(CPU Usage)**。自动完成列表出现时，从该列表中选择 **CPU Usage**。
  - c. 在时间间隔中，输入 **5 minutes**。

指标属性可定义聚合计算，用于处理一个时间间隔内的所有输入数据点，并针对每个时间间隔输出一个数据点。此指标属性每 5 分钟计算一次每台设备的平均 CPU 利用率。有关更多信息，请参阅 [聚合来自属性和其他资产 \(指标\) 的数据](#)。

9. 选择 **添加指标** 以添加第二个指标属性。

10. 在 指标定义 下的第二行中，执行以下操作：
  - a. 在名称中，输入 **Average Memory Usage**。
  - b. 在公式中，输入 **avg(Memory Usage)**。自动完成列表出现时，从该列表中选择 Memory Usage。
  - c. 在时间间隔中，输入 **5 minutes**。

此指标属性每 5 分钟计算一次每台设备的平均内存使用量。

11. ( 可选 ) 添加您对每个设备进行计算时感兴趣的其他指标。一些有趣的函数包括 min 和 max。有关更多信息，请参阅 [使用公式表达式](#)。在步骤 4：创建一个设备实例集资产模型，您将创建一个父资产，该资产可以使用来自整个设备队列的数据来计算指标。
12. 选择 创建模型。

## 步骤 4：创建设备队列资产模型

在此过程中，您将制作一个资产模型 AWS IoT SiteWise 来符号化您的设备集合。在此资产模型中，您可以建立一种结构，允许您将大量设备资产链接到一个总体机队资产。之后，您将概述车队资产模型中的指标，以整合来自所有联网设备资产的数据。这种方法可让您全面了解整个车队的集体表现。

### 创建表示设备队列的资产模型

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在左侧导航窗格中，选择 模型。
3. 选择 创建模型。
4. 在模型详细信息下，输入您的模型的名称。例如，**SiteWise Tutorial Device Fleet Model**。
5. 在 层次结构定义 下，执行以下操作：
  - a. 在 层次结构名称 中，输入 **Device**。
  - b. 在 层次结构模型 中，选择您的设备资产模型 (**SiteWise Tutorial Device Model**)。

层次结构定义父 ( 队列 ) 资产模型和子 ( 设备 ) 资产模型之间的关系。父资产可以访问子资产的属性数据。以后创建资产时，需要根据父资产模型中的层次结构定义将子资产与父资产关联。有关更多信息，请参阅 [定义资产模型层次结构](#)。

6. 在 指标定义 下，执行以下操作：

- a. 在名称中，输入 **Average CPU Usage**。
- b. 在公式中，输入 **avg(Device | Average CPU Usage)**。显示自动完成列表时，选择 Device 以选择层次结构，然后选择 Average CPU Usage，从之前创建的设备资产中选择指标。
- c. 在时间间隔中，输入 **5 minutes**。

此指标属性通过 **Device** 层次结构计算与队列资产关联的所有设备资产的平均 CPU 利用率。

7. 选择添加指标以添加第二个指标属性。
8. 在指标定义下的第二行中，执行以下操作：
  - a. 在名称中，输入 **Average Memory Usage**。
  - b. 在公式中，输入 **avg(Device | Average Memory Usage)**。显示自动完成列表时，选择 Device 以选择层次结构，然后选择 Average Memory Usage，从之前创建的设备资产中选择指标。
  - c. 在时间间隔中，输入 **5 minutes**。

此指标属性通过 **Device** 层次结构计算与队列资产关联的所有设备资产的平均内存使用量。

9. (可选) 添加您有兴趣跨设备队列计算的其他指标。
10. 选择 创建模型。

## 步骤 5：创建和配置设备资产

在此过程中，您将根据您的设备资产模型生成设备资产。然后，为每个测量属性定义属性别名。属性别名是标识资产属性的唯一字符串。稍后，您可以使用别名而不是资产 ID 和属性 ID 来识别要上传数据的属性。有关更多信息，请参阅 [将工业数据流映射到资产属性](#)。

### 创建设备资产并定义属性别名

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在左侧导航窗格中，选择 资产。
3. 选择 创建资产。
4. 在模型信息下，选择您的设备资产模型，**SiteWise Tutorial Device Model**。
5. 在资产信息下，输入资产的名称。例如，**SiteWise Tutorial Device 1**。

6. 选择 创建资产。
7. 对于新设备资产，请选择 编辑。
8. 在 CPU Usage 下，输入 `/tutorial/device/SiteWiseTutorialDevice1/cpu` 作为属性别名。您可以在属性别名中包含 AWS IoT 事物的名称，这样您就可以使用单个 AWS IoT 规则从所有设备提取数据。
9. 在 Memory Usage 下，输入 `/tutorial/device/SiteWiseTutorialDevice1/memory` 作为属性别名。
10. 选择保存。

如果您之前创建了多个 AWS IoT 内容，请对每台设备重复步骤 3 到 10，并相应地增加资产名称和属性别名中的数字。例如，第二个设备资产的名称应为 **SiteWise Tutorial Device 2**，其属性别名应为 `/tutorial/device/SiteWiseTutorialDevice2/cpu` 和 `/tutorial/device/SiteWiseTutorialDevice2/memory`。

## 步骤 6：创建和配置设备队列资产

在此过程中，您将形成从您的设备队列资产模型派生的设备队列资产。然后，将您的个人设备资产关联到舰队资产。这种关联使舰队资产的指标属性能够编译和分析来自多个设备的数据。这些数据为您提供整个机队集体绩效的综合视图。

### 创建设备队列资产并关联设备资产

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在左侧导航窗格中，选择 资产。
3. 选择 创建资产。
4. 在模型信息中，选择您的设备实例集资产模型 **SiteWise Tutorial Device Fleet Model**。
5. 在资产信息下，输入资产的名称。例如，**SiteWise Tutorial Device Fleet 1**。
6. 选择 创建资产。
7. 对于新设备队列资产，请选择 编辑。
8. 在“与此资产关联的资产”下，选择“添加关联资产”，然后执行以下操作：
  - a. 在 层次结构 下，选择 Device。此层次结构标识设备和设备队列资产之间的层次关系。您在本教程前面的设备队列资产模型中定义了此层次结构。
  - b. 在 资产 下，选择您的设备资产 SiteWise Tutorial Device 1。
9. ( 可选 ) 如果您之前创建了多个设备资产，请对您创建的每个设备资产重复步骤 8 到 10。

## 10. 选择保存。

现在，您应该可以看到设备资产按层次结构进行组织。

## 步骤 7：在 AWS IoT Core 中创建向设备资产发送数据的规则

在此过程中，您需要在中建立规则 AWS IoT Core。该规则旨在解释来自设备影子的通知消息，并将数据传输到您的设备资产 AWS IoT SiteWise。每次设备影子更新时，都会 AWS IoT 发送一条 MQTT 消息。您可以创建一个规则，以便在设备影子基于 MQTT 消息发生变化时执行操作。在这种情况下，目标是处理更新消息，提取属性值，然后将其传输到您的设备资产 AWS IoT SiteWise。

使用 AWS IoT SiteWise 操作创建规则

1. 导航到 [AWS IoT 控制台](#)。
2. 在左侧导航窗格中，选择消息路由，然后选择规则。
3. 选择 创建规则。
4. 输入规则的名称和说明，然后选择下一步。
5. 输入以下 SQL 语句并选择下一步。

```
SELECT
  *
FROM
  '$aws/things/+/shadow/update/accepted'
WHERE
  startsWith(topic(3), "SiteWiseTutorialDevice")
```

此规则查询语句起作用，因为 Device Shadow 服务将影子更新发布到 `$aws/things/thingName/shadow/update/accepted`。有关设备影子的更多信息，请参阅 AWS IoT 开发人员指南中的 [设备影子服务](#)。

在 WHERE 子句中，此规则查询语句使用 `topic(3)` 函数从主题的第三段获取事物名称。然后，该语句筛选掉名称与教程设备名称不匹配的设备。有关 AWS IoT SQL 的更多信息，请参阅《AWS IoT 开发人员指南》中的 [AWS IoT SQL 参考](#)。

6. 在规则操作下，选择向 AWS IoT SiteWise 中的资产属性发送消息数据，然后执行以下操作：
  - a. 选择 按属性别名。
  - b. 在属性别名中，输入 `/tutorial/device/${topic(3)}/cpu`。



`${...}` 语法是一个替换模板。AWS IoT 评估大括号内的内容。此替换模板从主题中提取事物名称，以创建对每个事物唯一的别名。有关更多信息，请参阅 AWS IoT 开发人员指南中的 [替换模板](#)。

#### Note

由于替换模板中的表达式与 SELECT 语句分开评估，因此您不能使用替换模板来引用使用 AS 子句创建的别名。除了支持的函数和运算符之外，您只能引用原始负载中存在的信息。

- c. 在条目 ID -可选中，输入 ``${concat(topic(3), "-cpu-", floor(state.reported.timestamp))}``。

条目 ID 唯一标识尝试输入的每个值。如果条目返回错误，您可以在错误输出中找到条目 ID 以对问题进行故障排除。此条目 ID 中的替换模板将事物名称与设备报告的时间戳组合起来。例如，生成的条目 ID 可能看起来像 `SiteWiseTutorialDevice1-cpu-1579808494`。

- d. 在时间（以秒为单位）中，输入 ``${floor(state.reported.timestamp)}``。

此替换模板从设备报告的时间戳计算时间（以秒为单位）。在本教程中，设备将 Unix 纪元时间中的时间戳（以秒为单位）报告为浮点数。

- e. 在偏移量（以纳秒为单位）-可选中，输入 ``${floor((state.reported.timestamp % 1) * 1E9)}``。

此替换模板通过转换设备报告时间戳的小数部分来计算时间（以秒为单位）的纳秒偏移量。

#### Note

AWS IoT SiteWise 要求您的数据具有以 Unix 纪元时间为单位的当前时间戳。如果您的设备没有准确报告时间，您可以使用 [timestamp\(\)](#) 从 AWS IoT 规则引擎获取当前时间。此函数以毫秒为单位报告时间，因此您必须将规则操作的时间参数更新为以下值：

- 在时间（以秒为单位）中，输入 ``${floor(timestamp() / 1E3)}``。
- 在偏移量（以纳秒为单位）中，输入 ``${(timestamp() % 1E3) * 1E6}``。

- f. 在数据类型中，选择双精度。

此数据类型必须与您在资产模型中定义的资产属性的数据类型相匹配。

- g. 在值中，输入 `${state.reported.cpu}`。在替换模板中，您可以使用 `.` 运算符从 JSON 结构中检索值。
  - h. 选择 添加条目 为内存使用量属性添加新条目，然后再次为该属性完成以下步骤：
    - i. 选择 按属性别名。
    - ii. 在属性别名中，输入 `/tutorial/device/${topic(3)}/memory`。
    - iii. 在条目 ID - 可选项中，输入 `${concat(topic(3), "-memory-", floor(state.reported.timestamp))}`。
    - iv. 在时间 (以秒为单位) 中，输入 `${floor(state.reported.timestamp)}`。
    - v. 在偏移量 (以纳秒为单位) - 可选项中，输入 `${floor((state.reported.timestamp % 1) * 1E9)}`。
    - vi. 在数据类型中，选择 双精度。
    - vii. 在值中，输入 `${state.reported.memory}`。
  - i. 在 IAM 角色下，选择创建新角色，以便为此规则操作创建 IAM 角色。此角色允许将数据推送 AWS IoT 到设备队列资产及其资产层次结构中的属性。
  - j. 输入角色名称，然后选择创建。
7. (可选) 配置可用于对规则进行故障排除的错误操作。有关更多信息，请参阅 [排查规则问题](#)。
  8. 选择 下一步。
  9. 查看设置并选择创建以创建规则。

## 步骤 8：运行设备客户端脚本

在本教程中，您没有使用实际设备来报告数据。取而代之的是，您运行一个脚本来使用 CPU 和内存使用情况更新您的 AWS IoT 设备影子，以模仿真实的传感器数据。要运行该脚本，必须先安装所需的 Python 软件包。在此过程中，您将安装所需的 Python 软件包，然后运行设备客户端脚本。

### 配置和运行设备客户端脚本

1. 导航到 [AWS IoT 控制台](#)。
2. 在左侧导航窗格底部，选择 设置。
3. 保存您的自定义终端节点以便与设备客户端脚本一起使用。您可以使用此终端节点与事物影子进行交互。此终端节点对您在当前区域中的账户是唯一的。

您的自定义终端节点应与以下示例类似。

```
identifier.iot.region.amazonaws.com
```

4. 打开命令行并运行以下命令以导航到之前创建的教程目录。

```
cd iot-sitewise-rule-tutorial
```

5. 运行以下命令来安装 AWS IoT Device SDK for Python。

```
pip3 install AWSIoTPythonSDK
```

有关更多信息，请参阅 AWS IoT 开发人员指南中的 [AWS IoT Device SDK for Python](#)。

6. 运行以下命令来安装 psutil，这是一个跨平台的进程和系统实用程序库。

```
pip3 install psutil
```

有关更多信息，请参阅 Python Package 索引中的 [psutil](#)。

7. 在 `iot-sitewise-rule-tutorial` 目录中创建一个名为 `thing_performance.py` 的文件，然后将以下 Python 代码复制到该文件中。

```
import AWSIoTPythonSDK.MQTTLib as AWSIoTPyMQTT

import json
import psutil
import argparse
import logging
import time

# Configures the argument parser for this program.
def configureParser():
    parser = argparse.ArgumentParser()
    parser.add_argument(
        "-e",
        "--endpoint",
        action="store",
        required=True,
        dest="host",
        help="Your AWS IoT custom endpoint",
    )
    parser.add_argument(
```

```
        "-r",
        "--rootCA",
        action="store",
        required=True,
        dest="rootCAPath",
        help="Root CA file path",
    )
    parser.add_argument(
        "-c",
        "--cert",
        action="store",
        required=True,
        dest="certificatePath",
        help="Certificate file path",
    )
    parser.add_argument(
        "-k",
        "--key",
        action="store",
        required=True,
        dest="privateKeyPath",
        help="Private key file path",
    )
    parser.add_argument(
        "-p",
        "--port",
        action="store",
        dest="port",
        type=int,
        default=8883,
        help="Port number override",
    )
    parser.add_argument(
        "-n",
        "--thingName",
        action="store",
        required=True,
        dest="thingName",
        help="Targeted thing name",
    )
    parser.add_argument(
        "-d",
        "--requestDelay",
        action="store",
```

```
        dest="requestDelay",
        type=float,
        default=1,
        help="Time between requests (in seconds)",
    )
    parser.add_argument(
        "-v",
        "--enableLogging",
        action="store_true",
        dest="enableLogging",
        help="Enable logging for the AWS IoT Device SDK for Python",
    )
    return parser

# An MQTT shadow client that uploads device performance data to AWS IoT at a
regular interval.
class PerformanceShadowClient:
    def __init__(
        self,
        thingName,
        host,
        port,
        rootCAPath,
        privateKeyPath,
        certificatePath,
        requestDelay,
    ):
        self.thingName = thingName
        self.host = host
        self.port = port
        self.rootCAPath = rootCAPath
        self.privateKeyPath = privateKeyPath
        self.certificatePath = certificatePath
        self.requestDelay = requestDelay

    # Updates this thing's shadow with system performance data at a regular
interval.
    def run(self):
        print("Connecting MQTT client for {}".format(self.thingName))
        mqttClient = self.configureMQTTClient()
        mqttClient.connect()
        print("MQTT client for {} connected".format(self.thingName))
        deviceShadowHandler = mqttClient.createShadowHandlerWithName(
```

```

        self.thingName, True
    )

    print("Running performance shadow client for {}...
\n".format(self.thingName))
    while True:
        performance = self.readPerformance()
        print("[{}]" .format(self.thingName))
        print("CPU:\t{}%".format(performance["cpu"]))
        print("Memory:\t{}%\n".format(performance["memory"]))
        payload = {"state": {"reported": performance}}
        deviceShadowHandler.shadowUpdate(
            json.dumps(payload), self.shadowUpdateCallback, 5
        )
        time.sleep(args.requestDelay)

# Configures the MQTT shadow client for this thing.
def configureMQTTClient(self):
    mqttClient = AWSIoTPyMQTT.AWSIoTMQTTShadowClient(self.thingName)
    mqttClient.configureEndpoint(self.host, self.port)
    mqttClient.configureCredentials(
        self.rootCAPath, self.privateKeyPath, self.certificatePath
    )
    mqttClient.configureAutoReconnectBackoffTime(1, 32, 20)
    mqttClient.configureConnectDisconnectTimeout(10)
    mqttClient.configureMQTTOperationTimeout(5)
    return mqttClient

# Returns the local device's CPU usage, memory usage, and timestamp.
def readPerformance(self):
    cpu = psutil.cpu_percent()
    memory = psutil.virtual_memory().percent
    timestamp = time.time()
    return {"cpu": cpu, "memory": memory, "timestamp": timestamp}

# Prints the result of a shadow update call.
def shadowUpdateCallback(self, payload, responseStatus, token):
    print("[{}]" .format(self.thingName))
    print("Update request {} {}\n".format(token, responseStatus))

# Configures debug logging for the AWS IoT Device SDK for Python.
def configureLogging():
    logger = logging.getLogger("AWSIoTPythonSDK.core")

```

```
logger.setLevel(logging.DEBUG)
streamHandler = logging.StreamHandler()
formatter = logging.Formatter(
    "%(asctime)s - %(name)s - %(levelname)s - %(message)s"
)
streamHandler.setFormatter(formatter)
logger.addHandler(streamHandler)

# Runs the performance shadow client with user arguments.
if __name__ == "__main__":
    parser = configureParser()
    args = parser.parse_args()
    if args.enableLogging:
        configureLogging()
    thingClient = PerformanceShadowClient(
        args.thingName,
        args.host,
        args.port,
        args.rootCAPath,
        args.privateKeyPath,
        args.certificatePath,
        args.requestDelay,
    )
    thingClient.run()
```

8. 在命令行中运行 `thing_performance.py` 及以下参数：

- `-n, --thingName` - 您的事物名称，例如 **SiteWiseTutorialDevice1**。
- `-e, --endpoint`— 您在本过程前面保存的自定义 AWS IoT 终端节点。
- `-r, --rootCA`— AWS IoT 根 CA 证书的路径。
- `-c, --cert`— 你的 AWS IoT 事物证书的路径。
- `-k, --key`— 你的 AWS IoT 东西证书私钥的路径。
- `-d, --requestDelay` - ( 可选 ) 每个设备影子更新之间等待的时间 ( 以秒为单位 )。默认值为 1 秒。
- `-v, --enableLogging` - ( 可选 ) 如果存在此参数，则脚本将从 AWS IoT Device SDK for Python 打印调试消息。

您的命令应类似于以下示例。

```
python3 thing_performance.py \  
  --thingName SiteWiseTutorialDevice1 \  
  --endpoint identifier.iot.region.amazonaws.com \  
  --rootCA AmazonRootCA1.pem \  
  --cert device1/thing-id-certificate.pem.crt \  
  --key device1/thing-id-private.pem.key
```

如果您正在为其他 AWS IoT 内容运行脚本，请相应地更新事物名称和证书目录。

9. 尝试打开和关闭设备上的程序，了解 CPU 利用率和内存使用量如何变化。脚本会打印每个 CPU 利用率和内存使用量读数。如果脚本成功将数据上传到 Device Shadow 服务，则脚本的输出应与以下示例类似。

```
[SiteWiseTutorialDevice1]  
CPU:    24.6%  
Memory: 85.2%  
  
[SiteWiseTutorialDevice1]  
Update request e6686e44-fca0-44db-aa48-3ca81726f3e3 accepted
```

10. 请按照以下步骤操作，验证脚本是否正在更新设备影子：
  - a. 导航到 [AWS IoT 控制台](#)。
  - b. 在左侧导航窗格中选择所有设备，然后选择事物。
  - c. 选择你的东西，SiteWiseTutorialDevice。
  - d. 选择设备影子选项卡，选择经典影子，然后核实影子状态是否与以下示例类似。

```
{  
  "reported": {  
    "cpu": 24.6,  
    "memory": 85.2,  
    "timestamp": 1579567542.2835066  
  }  
}
```

如果你的东西的影子状态为空或者看起来不像前面的示例，请检查脚本是否正在运行并成功连接 AWS IoT。如果脚本在连接时继续超时 AWS IoT，请检查您的[事物策略](#)是否已根据本教程进行配置。

11. 请按照以下步骤操作，验证规则操作是否正在将数据发送到 AWS IoT SiteWise：



- a. 导航到 [AWS IoT SiteWise 控制台](#)。
- b. 在左侧导航窗格中，选择 资产。
- c. 选择设备队列资产 (SiteWise Tutorial Device Fleet 1) 旁边的箭头以展开其资产层次结构，然后选择设备资产 (SiteWise Tutorial Device 1)。
- d. 选择 测量值。
- e. 验证 最新值 单元格是否具有 CPU Usage 和 Memory Usage 属性的值。

Measurements				
Name	Alias	Notification status	Notification topic	Latest value
CPU Usage	/tutorial/device/SiteWiseTutorialDevice1/cpu	⊖ Disabled	-	24.6
Memory Usage	/tutorial/device/SiteWiseTutorialDevice1/memory	⊖ Disabled	-	85.2

- f. 如果 CPU Usage 和 Memory Usage 属性没有最新值，请刷新页面。如果在几分钟后没有显示值，请参阅[排查规则问题](#)。
12. 您已完成本教程。如果要探索数据的实时可视化效果，可以在 AWS IoT SiteWise Monitor 中配置门户。有关更多信息，请参阅 [使用监控数据 AWS IoT SiteWise Monitor](#)。否则，您可以在命令提示符下按 Ctrl+C 停止设备客户端脚本。该 Python 程序发送的消息应该不足以产生费用，但完成后最好将其停止。

## 步骤 9：在完成教程后清理资源

完成有关从 AWS IoT 事物中提取数据的教程后，请清理资源以免产生额外费用。

要删除中的分层资产 AWS IoT SiteWise

1. 导航到 [AWS IoT SiteWise 控制台](#)
2. 在左侧导航窗格中，选择 资产。
3. 在中删除资源时 AWS IoT SiteWise，必须先取消与它们的关联。

请完成以下步骤以取消设备资产与设备队列资产之间的关联：

- a. 选择您的设备实例集资产 (SiteWise Tutorial Device Fleet 1)。
- b. 选择编辑。
- c. 在 与此资产关联的资产 下，针对与此设备队列资产关联的每个设备资产，选择 取消关联。
- d. 选择保存。

现在，您应该可以看到设备资产不再按层次结构进行组织。

4. 选择您的设备资产 (SiteWise Tutorial Device 1)。
5. 选择 删除。
6. 在确认对话框中，输入 **Delete**，然后选择 删除。
7. 对每个设备资产和设备实例集资产 (SiteWise Tutorial Device Fleet 1) 重复步骤 4 到 6。

#### 要在中删除分层资产模型 AWS IoT SiteWise

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 如果还没有删除，请删除设备和设备队列资产。有关更多信息，请参阅[前一过程](#)。如果您拥有从某个模型创建的资产，则无法删除该模型。
3. 在左侧导航窗格中，选择 模型。
4. 选择您的设备队列资产模型 (SiteWise Tutorial Device Fleet Model)。

删除分层资产模型时，首先要删除父资产模型。

5. 选择 删除。
6. 在确认对话框中，输入 **Delete**，然后选择 删除。
7. 对设备资产模型 (SiteWise Tutorial Device Model) 重复步骤 4 到 6。

#### 禁用或删除中的规则 AWS IoT Core

1. 导航到 [AWS IoT 控制台](#)。
2. 在左侧导航窗格中，选择消息路由，然后选择规则。
3. 选择您的规则，然后选择删除。
4. 在确认对话框中，键入规则名称，然后选择删除。

## 在 Monitor 中可视化和共享风电场 SiteWise 数据

本教程介绍 AWS IoT SiteWise Monitor 如何使用托管 Web 应用程序（称为门户）来可视化和共享工业数据。每个门户都包含项目，使您可以灵活地选择每个项目中哪些数据可以访问。然后，指定组织中可以访问每个门户的人员。您的用户使用 AWS IAM Identity Center 帐户登录门户，因此您可以使用现有的身份存储或由管理的存储 AWS。

您以及具有足够权限的用户可以在各个项目中创建控制面板，以有意义的方式可视化工业数据。然后，用户可以查看这些控制面板，以快速了解数据并监控操作。您可以为公司中的每个用户，针对每个项目配置管理权限或只读权限。有关更多信息，请参阅 [使用监控数据 AWS IoT SiteWise Monitor](#)。

在整个教程中，您将增强 AWS IoT SiteWise 演示，提供风力发电场的示例数据集。您可以在 Monitor SiteWise r 中配置门户、创建项目和仪表板来可视化风电场数据。本教程还介绍了如何创建其他用户，以及分配拥有或查看项目及其相关仪表板的权限。

### Note

使用 M SiteWise onitor 时，按每位登录门户的用户付费（每月）。在本教程中，您将创建三个用户，但只需使用一个用户登录。完成本教程后，这一个用户将产生费用。有关更多信息，请参阅 [AWS IoT SiteWise 定价](#)。

## 主题

- [先决条件](#)
- [步骤 1：在 SiteWise 监控器中创建入口](#)
- [步骤 2：登录门户](#)
- [第 3 步：创建风电场项目](#)
- [第 4 步：创建仪表板以可视化风电场数据](#)
- [第 5 步：浏览门户](#)
- [第 6 步：完成教程后清理资源](#)

## 先决条件

要完成本教程，您需要：

- 一个 AWS 账户。如果没有，请参阅 [设置一个 AWS 账户](#)。
- 一台运行 Windows、macOS Linux、或 Unix 访问的开发计算机 AWS Management Console。有关更多信息，请参阅 [AWS Management Console 入门](#)。
- 具有管理员权限的 AWS Identity and Access Management (IAM) 用户。
- 正在运行的 AWS IoT SiteWise 风力发电场演示。当你设置演示时，它会定义模型和资产，AWS IoT SiteWise 并将数据流式传输到它们以表示风力发电场。有关更多信息，请参阅 [使用演 AWS IoT SiteWise 示](#)。

- 如果您在账户中启用了 IAM Identity Center，请登录您的 AWS Organizations 管理账户。有关更多信息，请参阅 [AWS Organizations 术语和概念](#)。如果您尚未启用 IAM Identity Center，则需要在本教程中将其启用，并将您的帐户设置为管理帐户。

如果您无法登录 AWS Organizations 管理账户，则只要您的组织中有 IAM Identity Center 用户，就可以部分完成本教程。在这种情况下，您可以创建门户和控制面板，但无法创建新的 IAM Identity Center 用户来分配项目。

## 步骤 1：在 SiteWise 监控器中创建入口

在此过程中，您在 AWS IoT SiteWise Monitor 中创建门户。每个门户都是一个托管 Web 应用程序，您和您的用户可以使用 AWS IAM Identity Center 帐户登录该应用程序。借助 IAM Identity Center，您可以使用贵公司的现有身份存储，也可以创建由管理的身份存储 AWS。贵公司的员工无需单独创建，即可登录 AWS 账户。

### 创建门户

1. 登录 [AWS IoT SiteWise 控制台](#)。
2. 查看支持的 [AWS IoT SiteWise 终端节点和配额](#)，并在需要时切换区域。AWS IoT SiteWise 您必须在同一地区运行 AWS IoT SiteWise 演示。
3. 在左侧导航窗格中，选择 门户。
4. 选择 创建门户。
5. 如果您已经启用了 IAM Identity Center，请跳至步骤 6。否则，请完成以下步骤以启用 IAM Identity Center。
  - a. 在启用 AWS IAM Identity Center (SSO) 页面上，输入您的电子邮件地址、名字和姓氏，为自己创建一个 IAM Identity Center 用户，使其成为门户管理员。请使用您可以访问的电子邮件地址，这样您会收到一封为新 IAM Identity Center 用户设置密码的电子邮件。

在门户中，门户管理员会创建项目并为项目分配用户。之后您可以创建更多用户。

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1  
Enable SSO

Step 2  
Portal configuration

Step 3  
Invite administrators

Step 4  
Assign users

## Enable AWS Single Sign-On (SSO)

AWS IoT SiteWise Monitor requires SSO to create a portal and invite users. Create your first user below to enable AWS Single-Sign On. Later in this process, you'll have the opportunity to create other users by using the AWS SSO console. [Learn more](#)

### Create a user

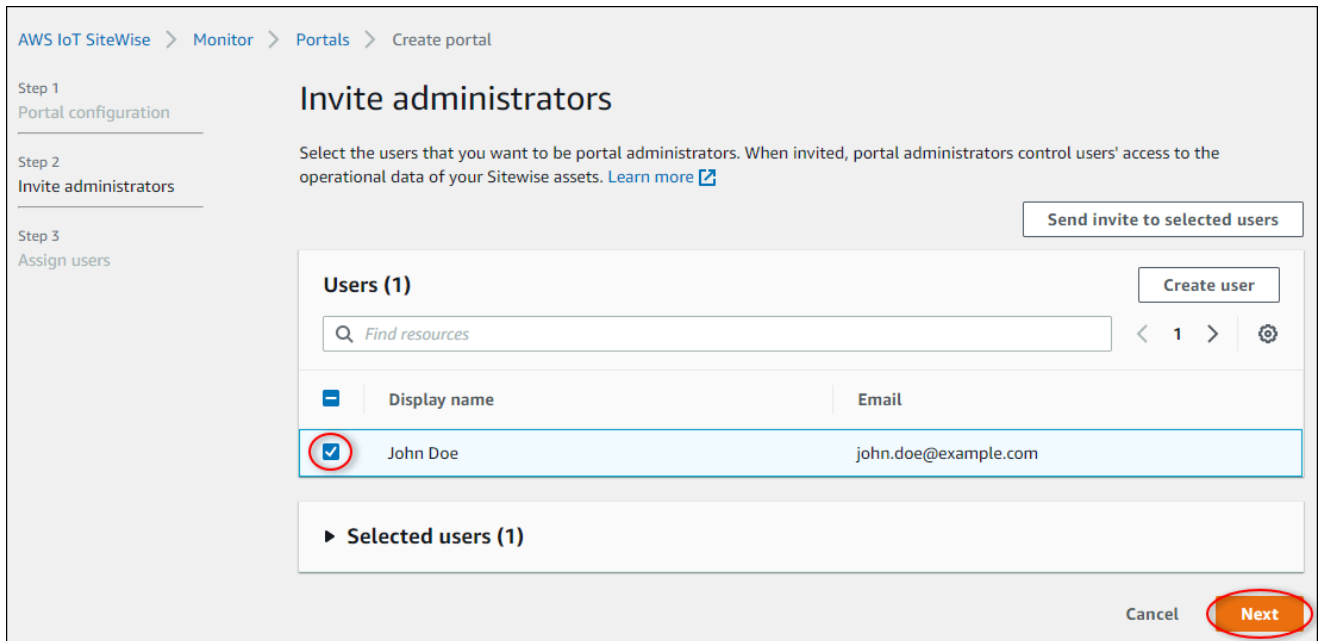
Email address

First name  Last name

Upon creation this application will enable AWS Organizations and Single Sign-On. [Learn more](#)

Cancel

- b. 选择 创建用户。
6. 在 门户配置 页面上，完成以下步骤：
    - a. 输入门户的名称，例如 **WindFarmPortal**。
    - b. （可选）输入门户的描述。如果您有多个门户，请使用有意义的描述来跟踪每个门户包含的内容。
    - c. （可选）上传图像以在门户中显示。
    - d. 输入一个电子邮件地址，当门户用户遇到门户问题并需要贵公司 AWS 管理员的帮助才能解决问题时，可以联系到该地址。
    - e. 选择 创建门户。
  7. 在邀请管理员页面上，您可以将 IAM Identity Center 用户作为管理员分配到门户。门户管理员管理门户内的权限和项目。在此页上，执行以下操作：
    - a. 选择作为门户管理员的用户。如果您在本教程前面部分启用了 IAM Identity Center，请选择您创建的用户。



- b. (可选) 选择 向选定用户发送邀请。您的电子邮件客户端将打开，邮件正文中会显示一个邀请。在将电子邮件发送给门户管理员之前您可以对其进行自定义。您也可以稍后再向门户管理员发送电子邮件。如果您是首次尝试使用 M SiteWise onitor，并且将成为门户管理员，则无需给自己发送电子邮件。
  - c. 选择下一步。
8. 在分配用户页面上，您可以将 IAM Identity Center 用户分配给门户。门户管理员稍后可以将这些用户分配为项目所有者或查看者。项目所有者可以在项目中创建控制面板。项目查看者对分配给他们的项目具有只读访问权限。在此页面上，您可以创建要添加到门户的 IAM Identity Center 用户。

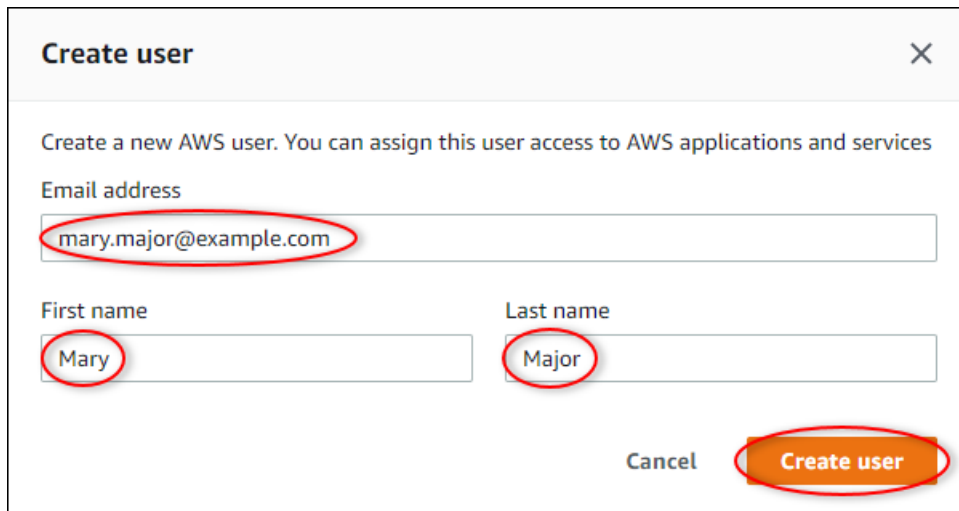
#### Note

如果您未登录 AWS Organizations 管理账户，则无法创建 IAM Identity Center 用户。选择分配用户以创建没有门户用户的门户，然后跳过此步骤。

在此页上，执行以下操作：

- a. 完成以下步骤两次即可创建两个 IAM Identity Center 用户：
  - i. 选择创建用户以打开一个对话框，您可以在其中输入新用户的详细信息。
  - ii. 输入要注册的用户的名字、姓氏和电子邮件地址。IAM Identity Center 会向用户发送一封电子邮件，让他们设置密码。如果您想以这些用户的身份登录门户，请选择您可以访问的

电子邮件地址。每个电子邮件地址必须唯一。您的用户使用其电子邮件地址作为用户名登录门户。



**Create user** [X]

Create a new AWS user. You can assign this user access to AWS applications and services

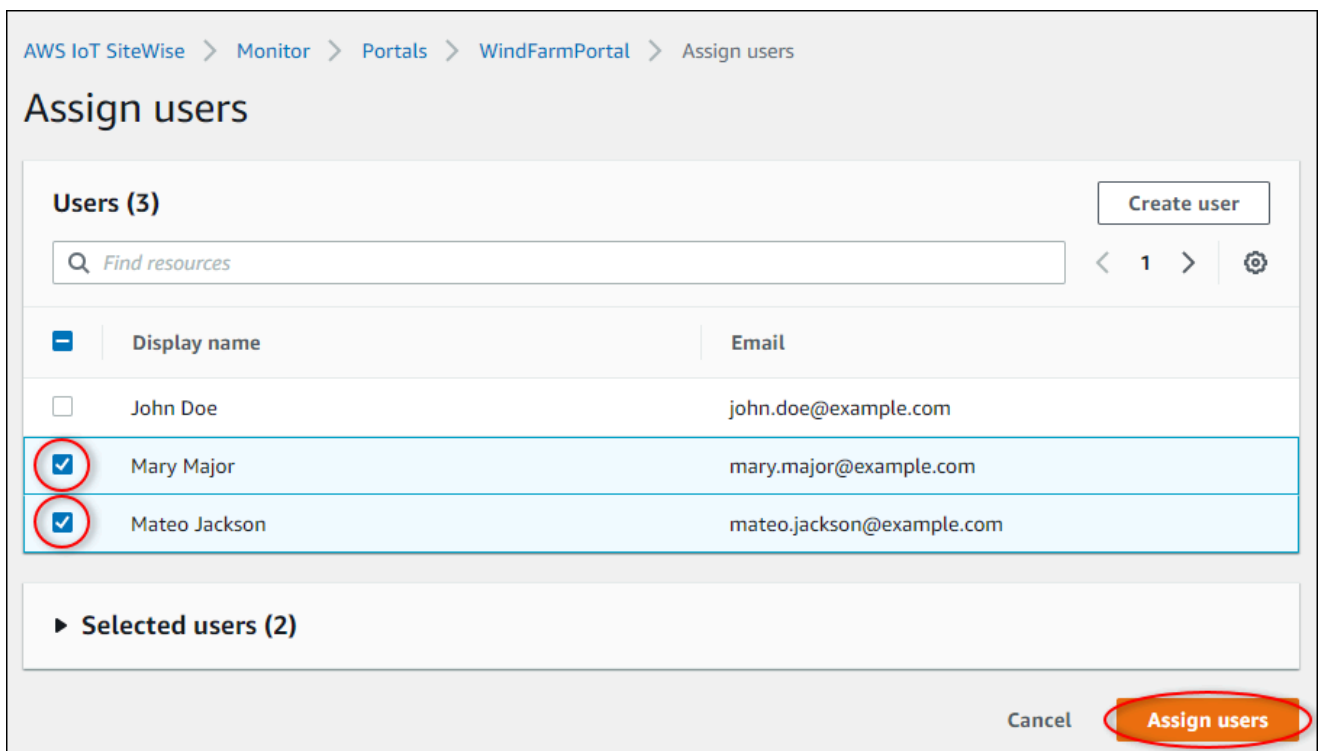
Email address  
mary.major@example.com

First name: Mary      Last name: Major

Cancel      **Create user**

iii. 选择 创建用户。

b. 选择您在上一步中创建的两个 IAM Identity Center 用户。



AWS IoT SiteWise > Monitor > Portals > WindFarmPortal > Assign users

## Assign users

Users (3) [Create user]

Find resources < 1 > [Settings]

<input type="checkbox"/>	Display name	Email
<input type="checkbox"/>	John Doe	john.doe@example.com
<input checked="" type="checkbox"/>	Mary Major	mary.major@example.com
<input checked="" type="checkbox"/>	Mateo Jackson	mateo.jackson@example.com

▶ Selected users (2)

Cancel      **Assign users**

c. 选择分配用户以将这些用户添加到门户。

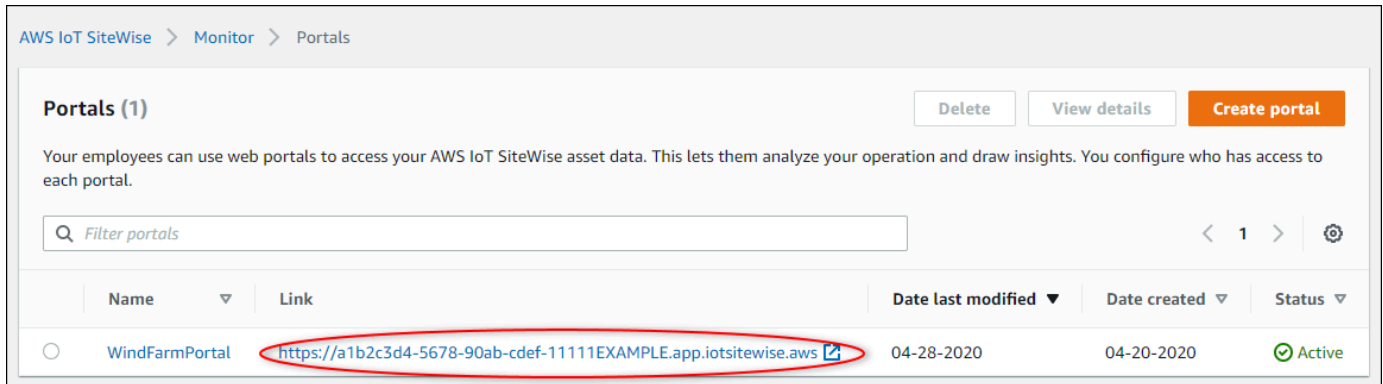
此时将打开门户页面，其中列出了您的新门户。

## 步骤 2：登录门户

在此过程中，您将使用添加到新门户的 AWS IAM Identity Center 用户登录新门户。

### 登录门户

1. 在门户页面上，选择新门户的链接以在新选项卡中打开门户。



2. 如果您在本教程前面部分中创建了第一个 IAM Identity Center 用户，请使用以下步骤为该用户创建密码：
  - a. 查看主题为 Invitation to join AWS IAM Identity Center 的电子邮件。
  - b. 打开邀请电子邮件并选择 Accept invitation。
  - c. 在新窗口中，为 IAM Identity Center 用户设置密码。

如果您稍后想以之前创建的第二和第三个 IAM Identity Center 用户身份登录门户，也可以完成这些步骤，为这些用户设置密码。

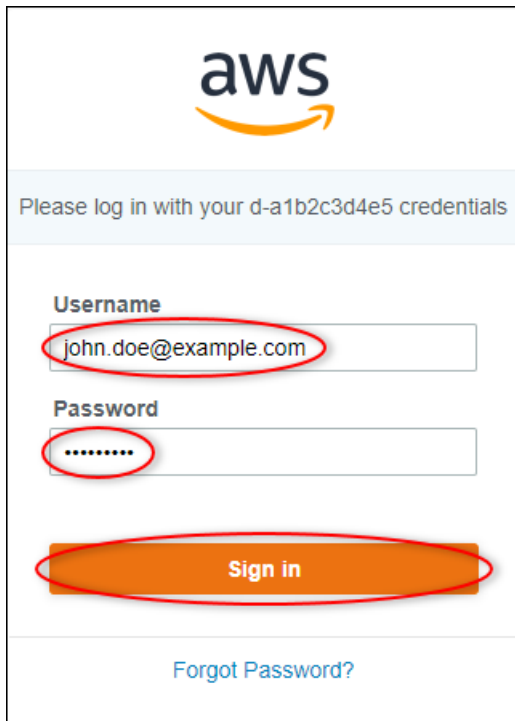
#### Note

如果没有收到电子邮件，您可以在 IAM Identity Center 控制台中为用户生成密码。有关更多信息，请参阅 AWS IAM Identity Center 用户指南中的[重置用户密码](#)。

3. 进入您的 IAM Identity Center Username，然后 Password。如果您在本教程前面部分中启用了 IAM Identity Center，则您的 Username 是您创建的门户管理员用户的电子邮件地址。

所有门户用户（包括门户管理员）都必须使用其 IAM Identity Center 用户凭证登录。这些凭证通常不是用于登录 AWS Management Console 的那些凭证。





aws

Please log in with your d-a1b2c3d4e5 credentials

Username  
john.doe@example.com

Password  
.....

Sign in

[Forgot Password?](#)

#### 4. 选择Sign in。

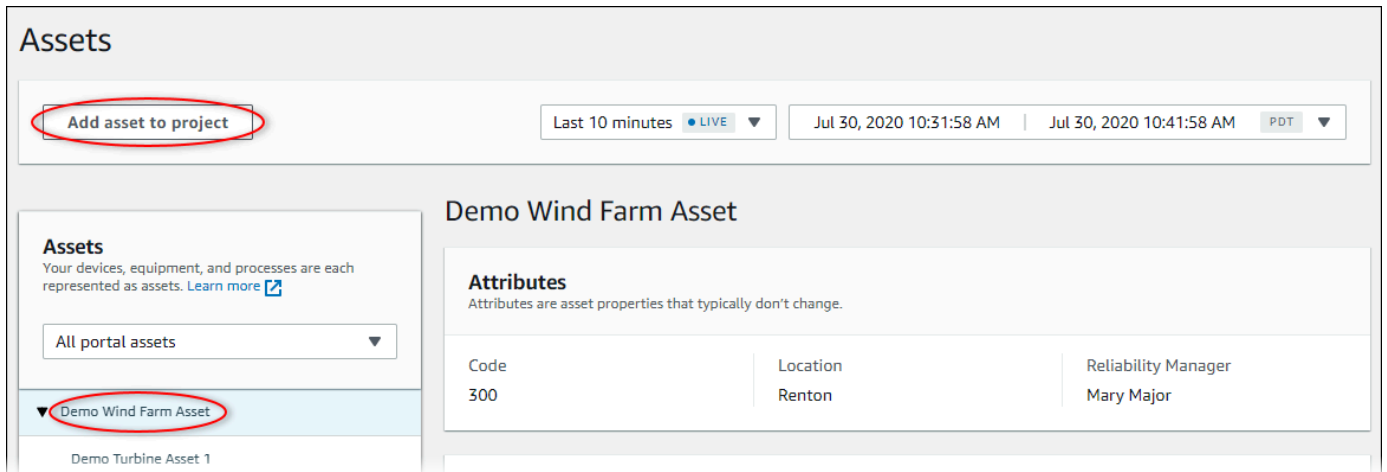
您的门户打开。

## 第 3 步：创建风电场项目

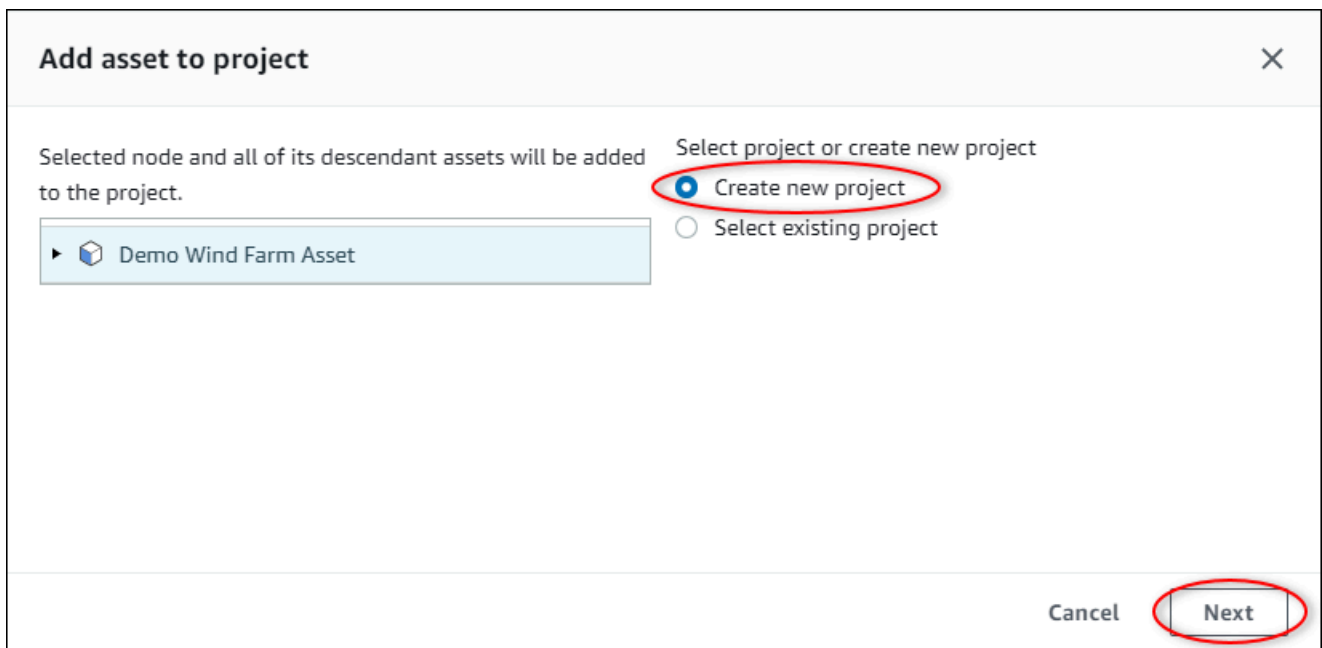
在此过程中，您将在门户中创建项目。项目是定义一组权限、资产和仪表板的资源，您可以对其进行配置以可视化该项目中的资产数据。通过项目，您可以定义谁有权访问操作的哪些子集，以及如何可视化这些子集的数据。您可以将门户用户分配作为各个项目的拥有者或查看者。项目所有者可以通过创建控制面板来可视化数据，并与其他用户共享项目。项目查看者可以查看控制面板，但不能进行编辑。有关 SiteWise 监控器中角色的更多信息，请参阅[SiteWise 监控角色](#)。

### 创建风电场项目

1. 在门户左侧的导航窗格中，选择资产选项卡。在资产页面，您可以浏览门户中的所有可用资产并将资产添加到项目中。
2. 在资产浏览器中，选择 Demo Wind Farm Asset。选择资产时，您可以浏览该资产的实时数据和历史数据。您也可以按Shift下选择多个资源并比较其数据 side-by-side。
3. 选择左上角的向项目添加资产。项目包含控制面板，门户用户可以在其中进行查看来浏览数据。每个项目都可以访问您的资源子集 AWS IoT SiteWise。将资产添加到项目时，对该项目具有访问权限的所有用户也可以访问该资产及其子集的数据。



4. 在向项目添加资产对话框中，选择创建新项目，然后选择下一步。



5. 在创建新项目对话框中，为项目输入项目名称和项目描述，然后选择向项目添加资产。



**Create new project** ✕

Project name  
Wind Farm 1  
The project name can have up to 256 characters.

Project description  
A project that contains dashboards for wind farm #1.  
The project description can have up to 2048 characters.

Cancel Previous **Add asset to project**

此时将打开新项目的页面。

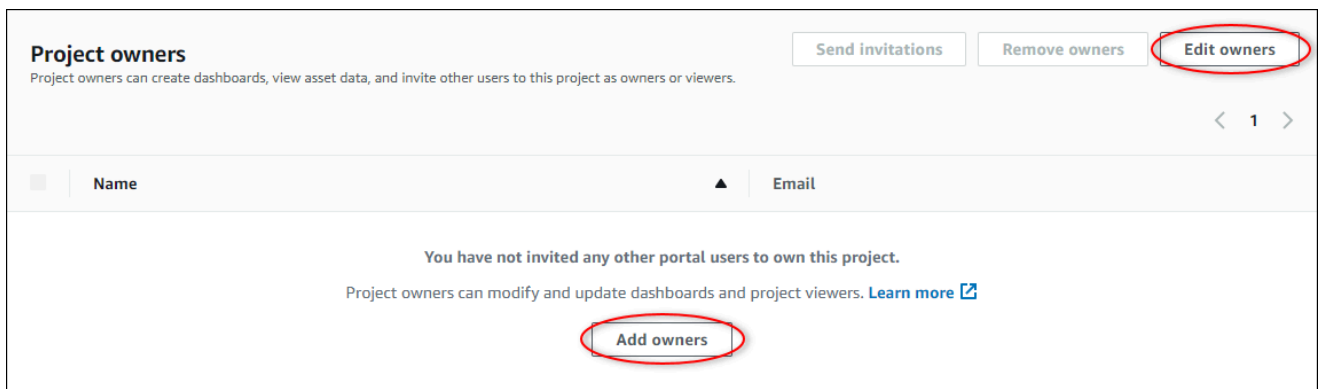
- 在项目页面上，您可以将门户用户添加为该项目的所有者或查看者。

**Note**

如果您尚未登录 AWS Organizations 管理账户，则可能没有门户用户可以分配给此项目，因此您可以跳过此步骤。

在此页上，执行以下操作：

- 在项目所有者下，选择添加所有者或编辑用户。



**Project owners** Send invitations Remove owners **Edit owners**

Project owners can create dashboards, view asset data, and invite other users to this project as owners or viewers.

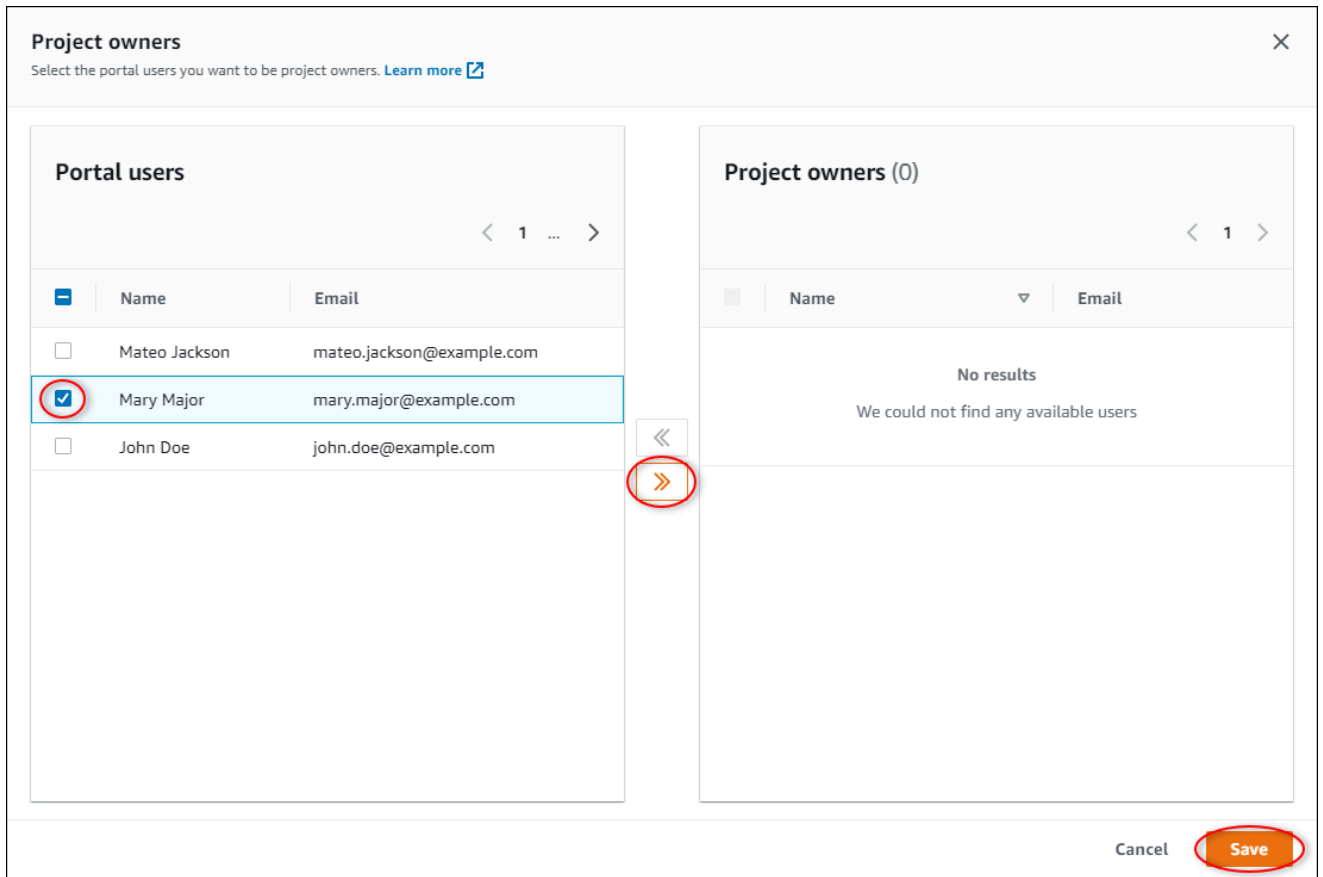
< 1 >

Name	Email
------	-------

You have not invited any other portal users to own this project.  
Project owners can modify and update dashboards and project viewers. [Learn more](#)

**Add owners**

- 选择要添加为项目拥有者的用户（例如 Mary Major），然后选择 >> 图标。



- c. 选择保存。

您的 IAM Identity Center 用户 Mary Major 可以登录该门户以编辑此项目中的控制面板，并与该门户中的其他用户共享此项目。

- d. 在项目查看者下，选择添加查看者或编辑用户。
- e. 选择要添加为项目查看者的用户（例如 Mateo Jackson），然后选择 >> 图标。
- f. 选择保存。

您的 IAM Identity Center 用户 Mateo Jackson 可以登录此门户以查看风电场项目中的控制面板，但不能进行编辑。

## 第 4 步：创建仪表板以可视化风电场数据

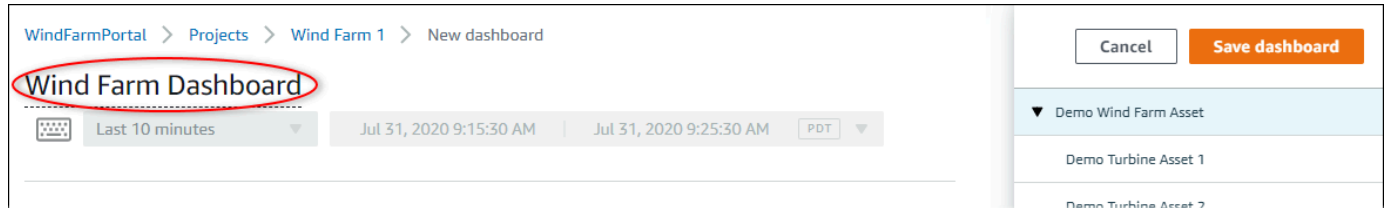
在此过程中，您将创建控制面板以可视化演示风电场数据。控制面板包含项目资产数据可自定义的可视化。每个可视化可以有不同的类型，例如折线图、条形图或关键绩效指标 (KPI) 显示。您可以选择最适合您数据的可视化类型。项目所有者可以编辑仪表板，而项目查看者只能查看仪表板以获得见解。

## 创建具有可视化的控制面板

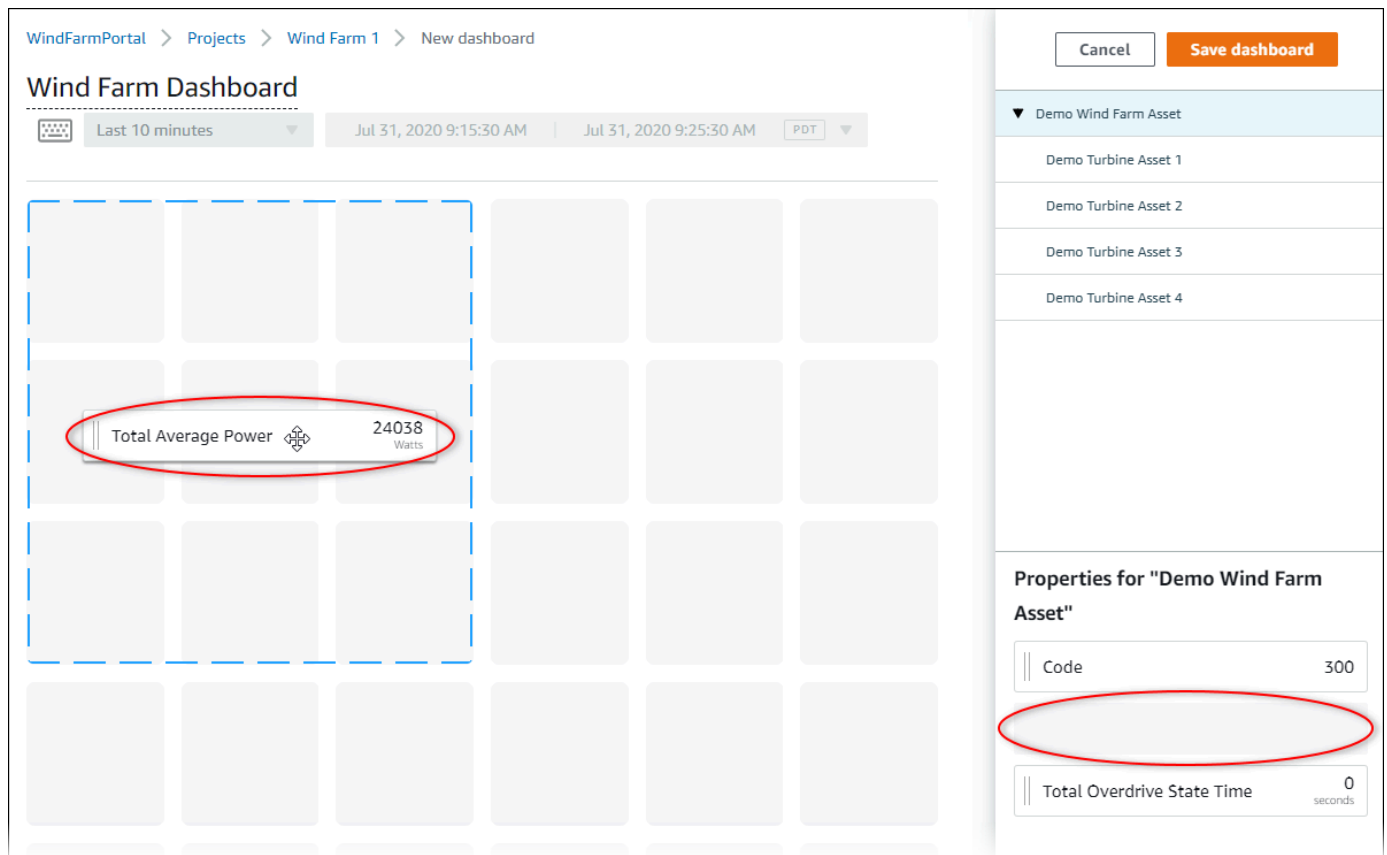
1. 在新项目的页面上，选择创建控制面板以创建控制面板并打开其编辑页面。

在控制面板的编辑页面中，您可以将资产属性从资产层次结构拖动到控制面板以创建可视化。然后，您可以在控制面板中编辑每个可视化的标题、图例标题、类型、大小和位置。

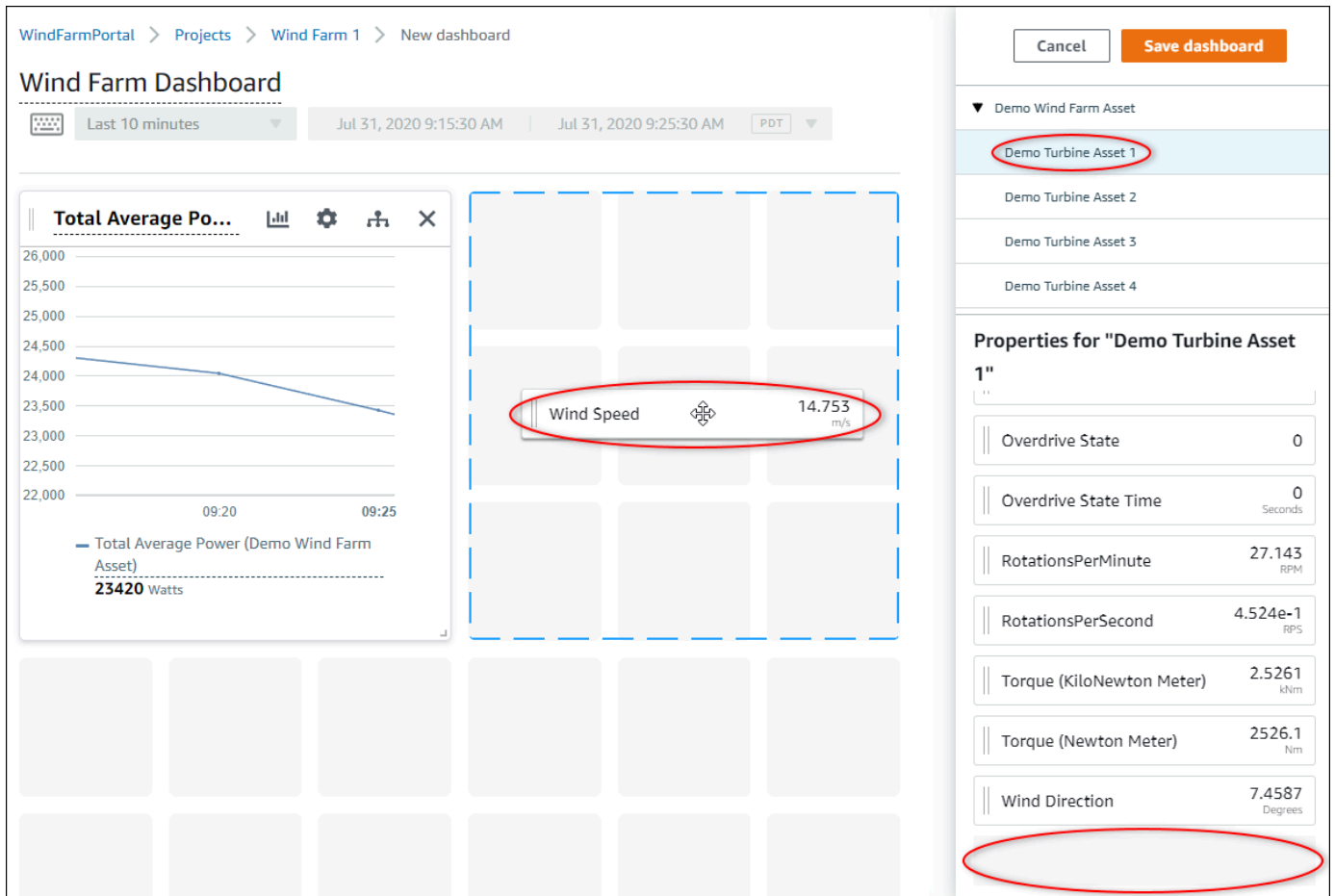
2. 输入您的控制面板的名称。



3. 将 Total Average Power 从 Demo Wind Farm Asset 拖动到控制面板以创建可视化。

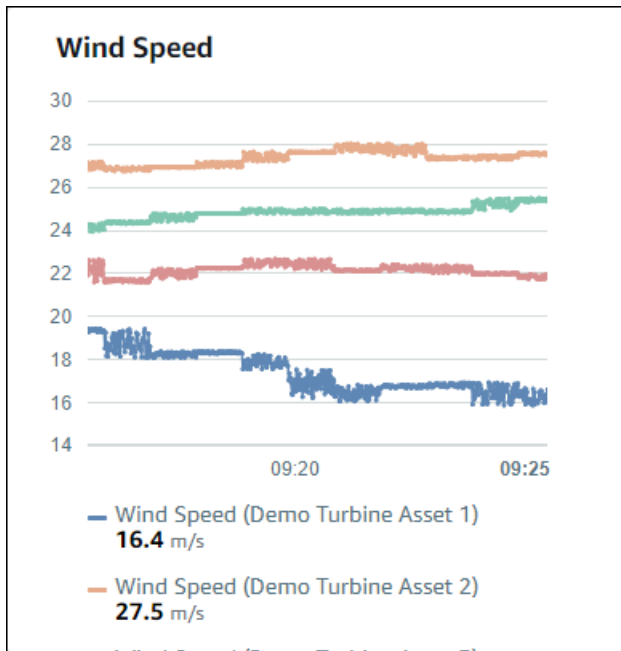


4. 选择 Demo Turbine Asset 1 显示该资产的属性，然后拖动 Wind Speed 到控制面板，创建风速可视化效果。

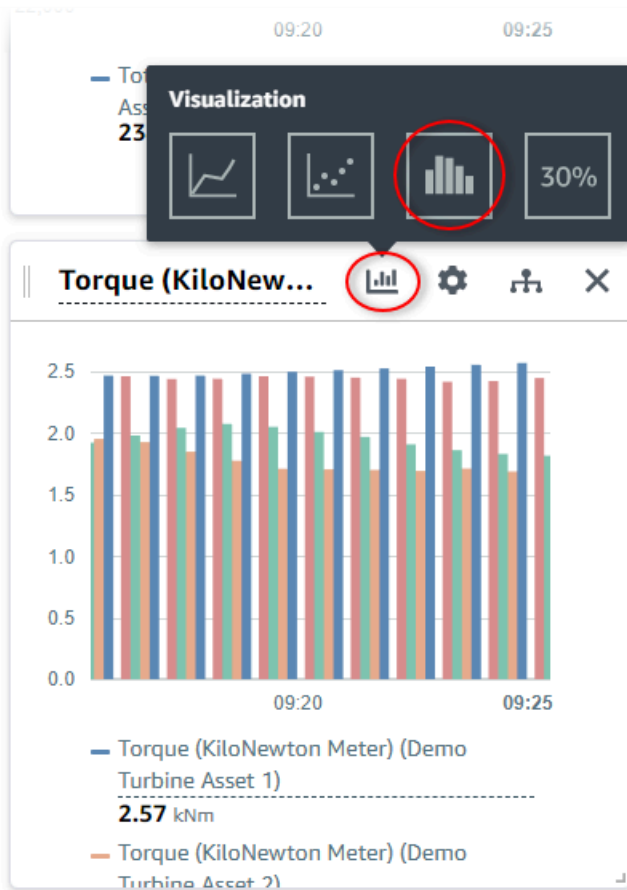


5. 为每个Demo Turbine Asset 2、3 和 4 (按该顺序排列) 添加 Wind Speed 到新的风速可视化效果中。

您的 Wind Speed 可视化应类似于以下屏幕截图。

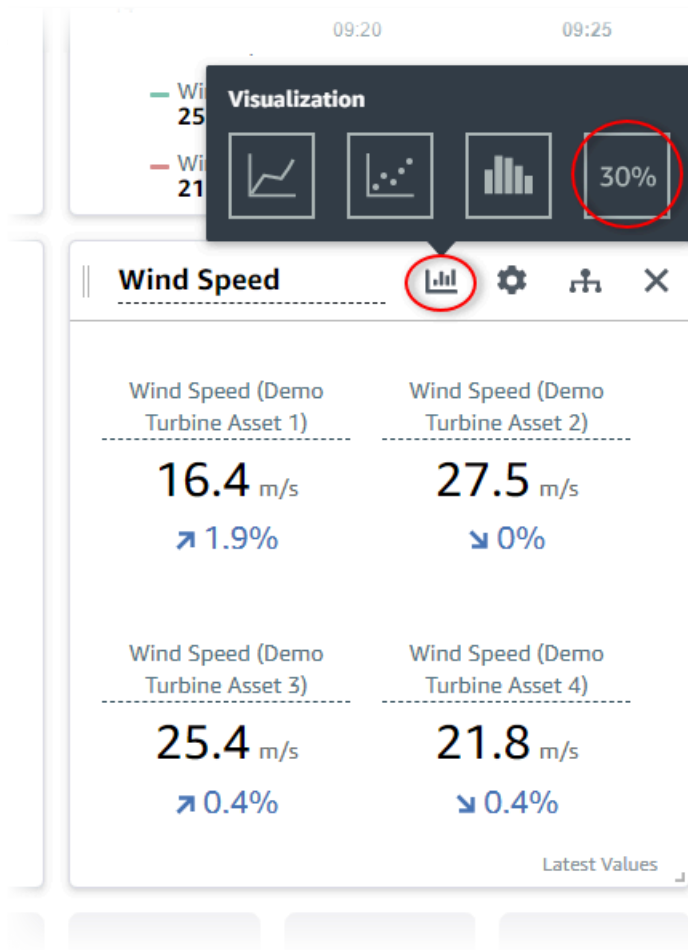


- 对风力涡轮机的 Torque (KiloNewton Meter) 属性重复步骤 4 和 5，以创建风力涡轮机扭矩的可视化。
- 为 Torque (KiloNewton Meter) 可视化选择可视化类型图标，然后选择条形图图标。



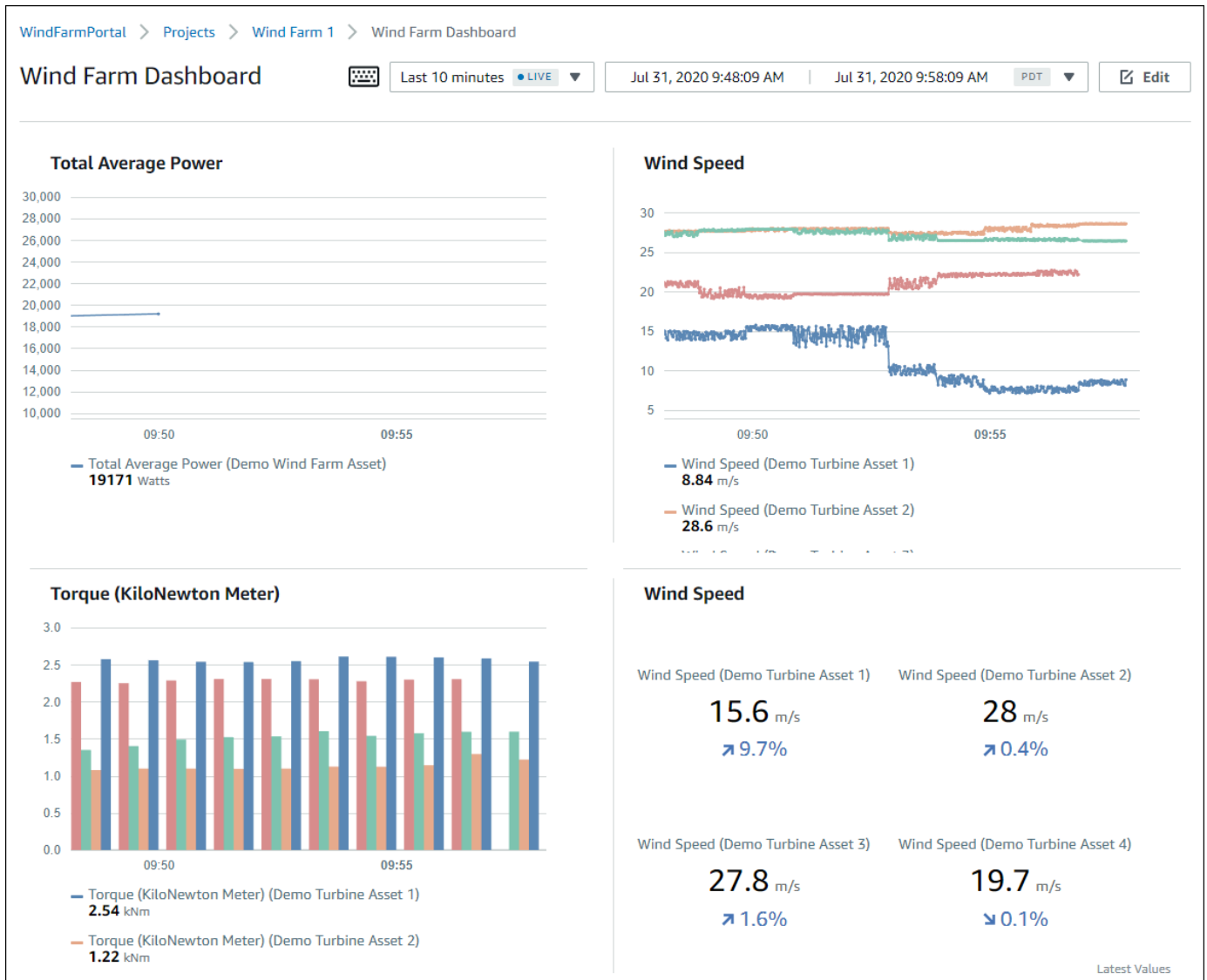
8. 对风力涡轮机的 Wind Direction 属性重复步骤 4 和 5，以创建风向的可视化。
9. 为 Wind Direction 可视化选择可视化类型图标，然后选择 KPI 图表图标 (30%)。





10. (可选) 根据需要对每个可视化的标题、图例标题、类型、大小和位置进行其他更改。
11. 选择右上角的保存控制面板以保存控制面板。

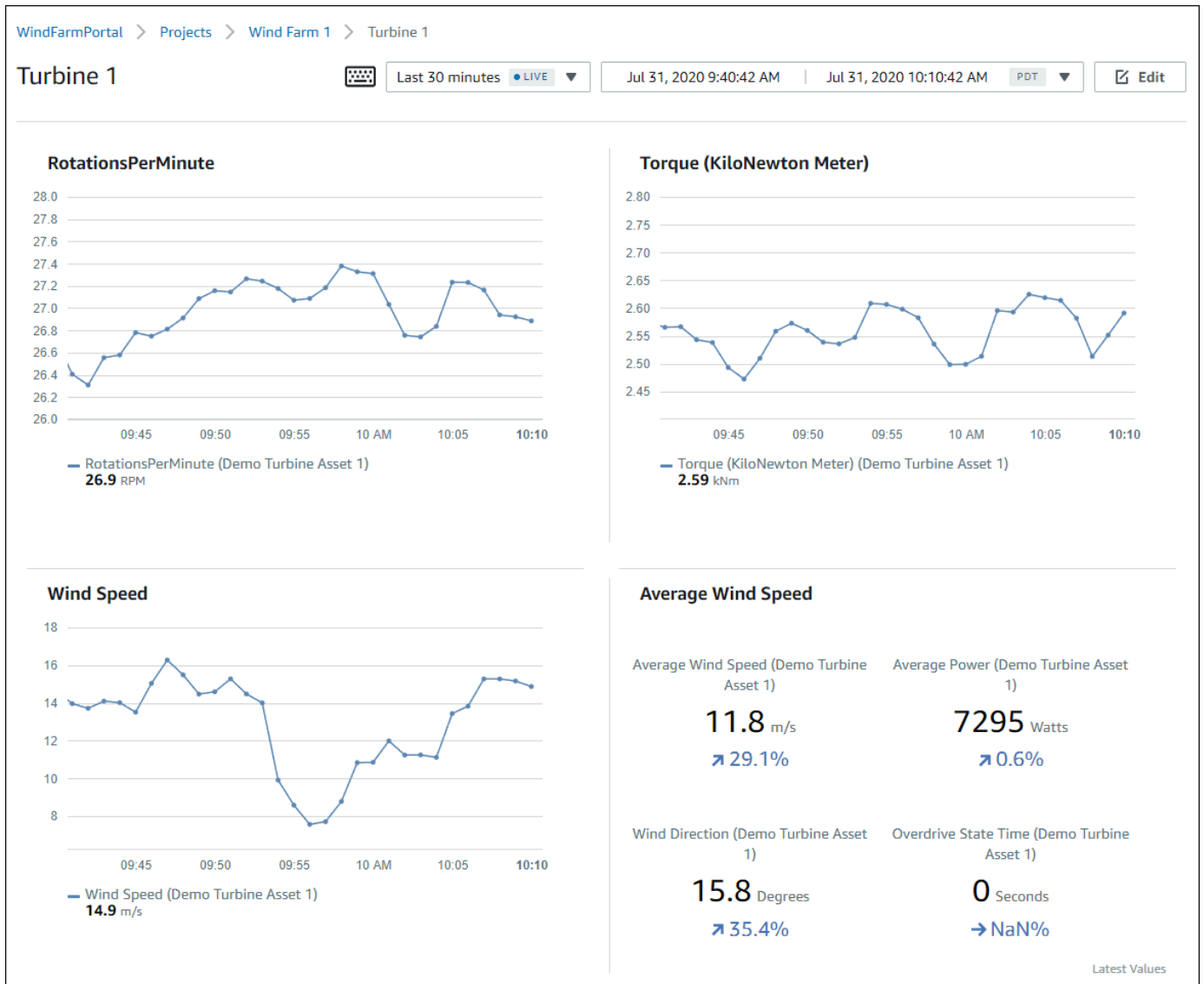
您的控制面板应类似于以下屏幕截图。



## 12. (可选) 为每个风力涡轮机资产创建额外的控制面板。

作为最佳实践，我们建议您为每个资产创建控制面板，以便项目查看者可以调查每个单独资产的任何问题。您只能向每个可视化添加最多 5 个资产，因此在许多情况下必须为分层资产创建多个控制面板。

演示风力涡轮机的控制面板类似于以下屏幕截图。



13. (可选) 在可视化中更改时间轴或选择数据点，以浏览控制面板中的数据。有关更多信息，请参阅 AWS IoT SiteWise Monitor 应用指南中的[查看控制面板](#)。

## 第 5 步：浏览门户

在此过程中，您可以以比门户管理员更少权限的用户身份浏览 AWS IoT SiteWise 门户。

浏览门户网站并完成教程

- (可选) 如果您将其他用户作为所有者或查看者添加到项目中，则可以以这些用户的身份登录门户。这样，您就能够以比门户管理员权限更少的用户身份浏览门户。

**⚠ Important**

您需要为每位登录门户的用户付费。有关更多信息，请参阅[AWS IoT SiteWise 定价](#)。

要以其他用户身份浏览门户，请执行以下操作：

- a. 选择门户左下角的注销退出 Web 应用程序。
- b. 在 IAM Identity Center 应用程序门户的右上角选择注销，退出您的 IAM Identity Center 用户。
- c. 以您分配为项目所有者或项目查看者的 IAM Identity Center 用户身份登录门户。有关更多信息，请参阅 [步骤 2：登录门户](#)。

您已完成本教程。在 Monito SiteWise r 中完成对演示风电场的探索后，请按照下一个步骤清理资源。

## 第 6 步：完成教程后清理资源

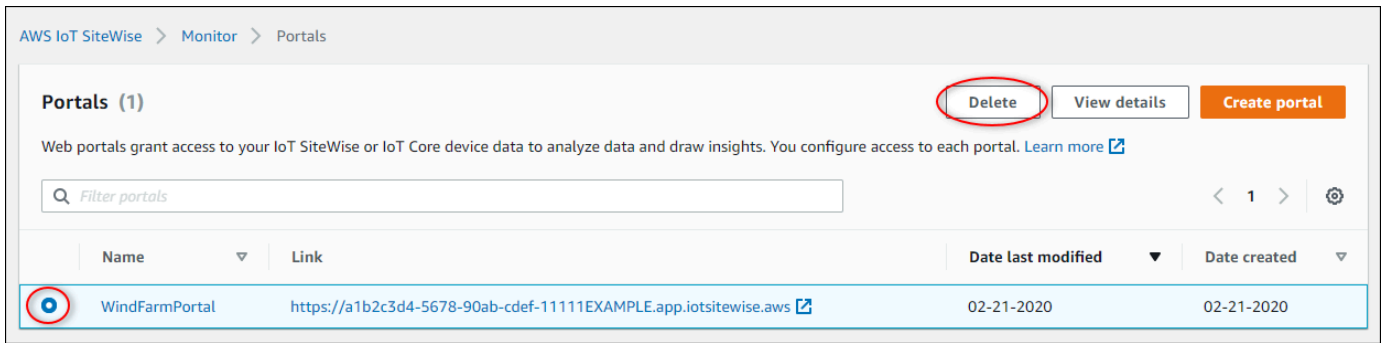
完成本教程后，您可以清除资源。如果用户未登录您的门户，则您无需为 AWS IoT SiteWise 付费，但您可以删除门户和 AWS IAM Identity Center 目录用户。您的演示风电场资产将在您创建演示时选择的时段结束时间删除，或者您也可以手动删除演示。有关更多信息，请参阅 [删除演 AWS IoT SiteWise 示](#)。

使用以下过程删除门户和 IAM Identity Center 用户。

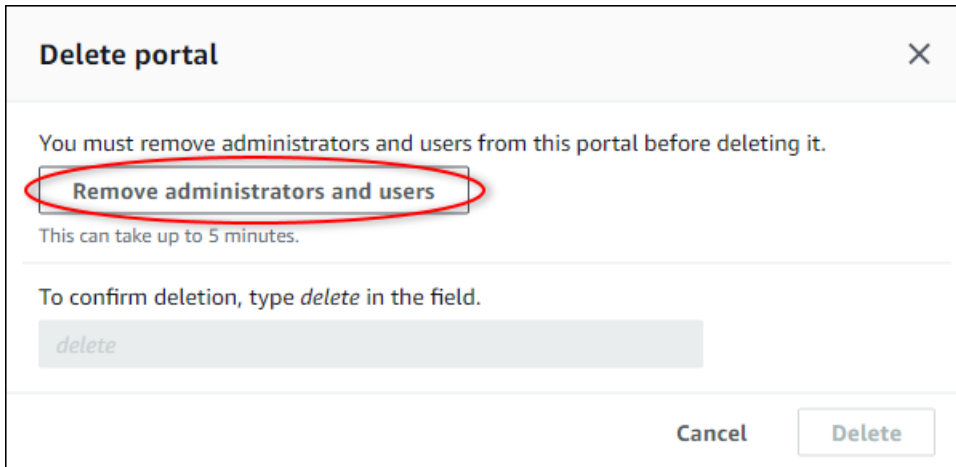
### 删除门户

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在左侧导航窗格中，选择门户。
3. 选择您的门户 WindFarmPortal，然后选择删除。

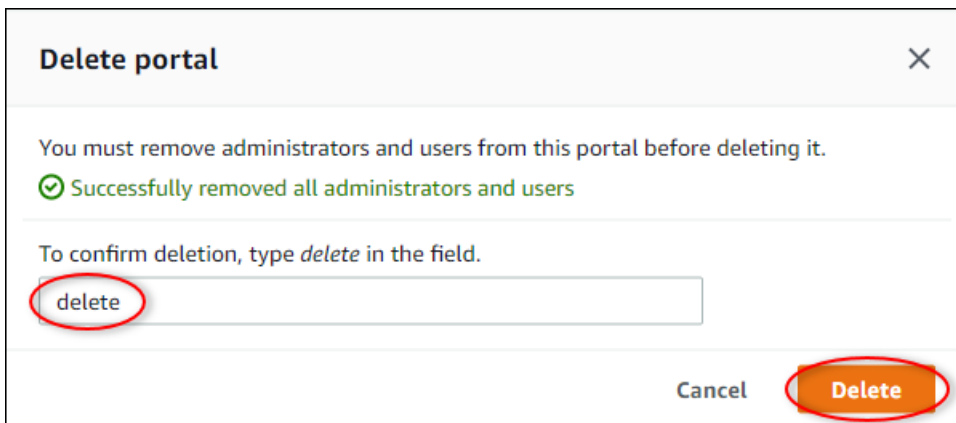
删除门户或项目时，与已删除项目关联的资产不受影响。



- 在删除门户对话框中，选择删除管理员和用户。

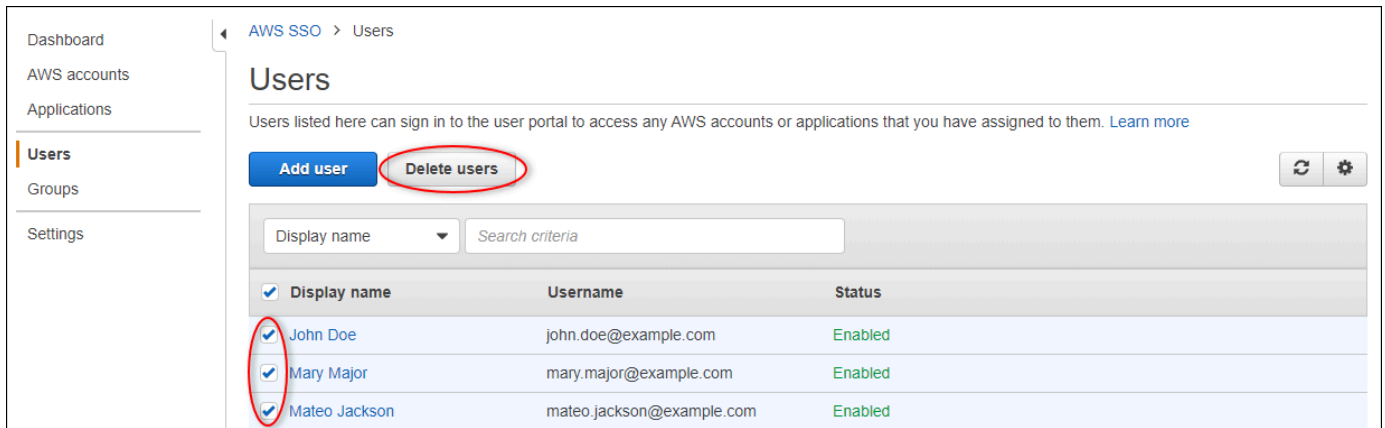


- 输入 **delete** 以确认删除，然后选择删除。

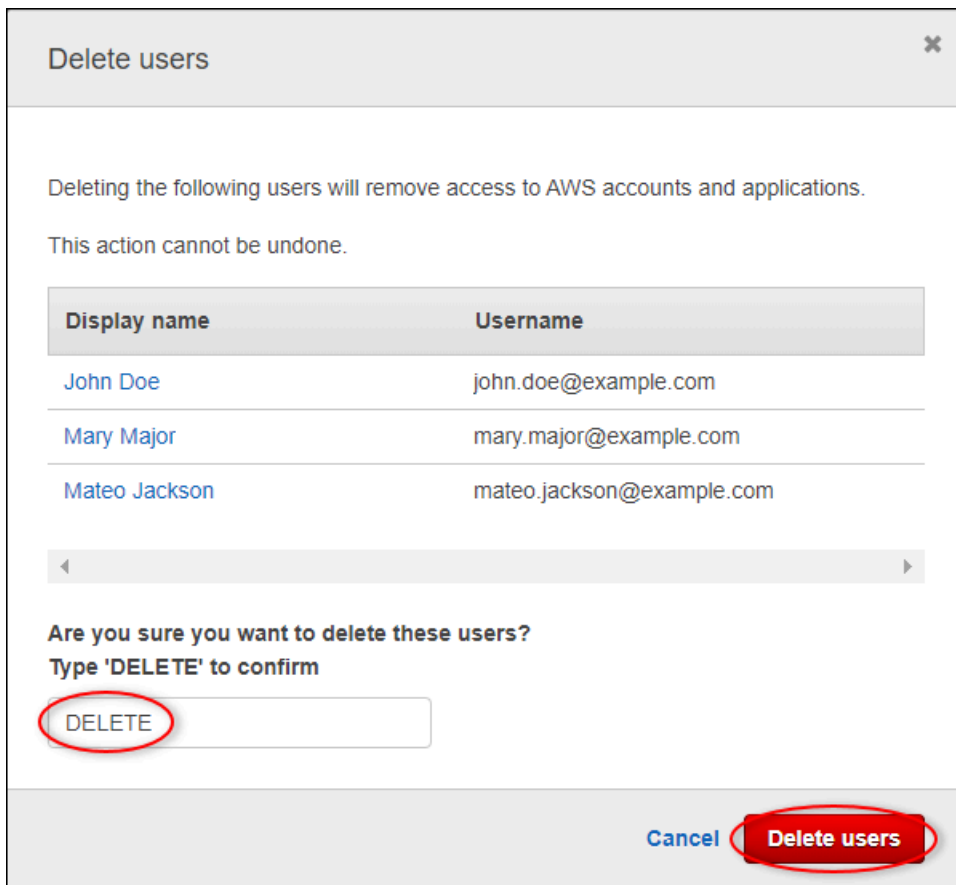


## 删除 IAM Identity Center 用户

- 导航到 [IAM Identity Center 控制台](#)。
- 在左侧导航窗格中，选择 用户。
- 选中要删除的各个用户旁的复选框，然后选择 删除用户。



4. 在删除用户对话框中，输入 **DELETE**，然后选择删除用户。



## 将属性值更新发布到 Amazon DynamoDB

本教程介绍了一种使用 [Amazon Dynamo](#) DB 存储数据的便捷方法，无需重复查询 API 即可更轻松地访问历史资产数据。AWS IoT SiteWise 完成本教程后，您可以创建使用您的资产数据的自定义软件，例如整个风电场的风速和风向的实时地图。如果您想在不实施自定义软件解决方案的情况下监控和可视化数据，请参阅[使用监控数据 AWS IoT SiteWise Monitor](#)。

在本教程中，您将在提供风力发电场示例数据集的 AWS IoT SiteWise 演示基础上进行构建。您可以配置从风电场演示的属性值更新，通过 AWS IoT Core 规则将数据发送到您创建的 DynamoDB 表。启用属性值更新后，AWS IoT SiteWise 会以 MQTT 消息的 AWS IoT Core 形式将您的数据发送到。然后，根据这些消息的内容定义执行操作（例如 DynamoDB 操作）的 AWS IoT 核心规则。有关更多信息，请参阅 [与其他 AWS 服务互动](#)。

## 主题

- [先决条件](#)
- [步骤 1：配置 AWS IoT SiteWise 以发布属性值更新](#)
- [第 2 步：在 C AWS IoT Core 中创建规则](#)
- [步骤 3：创建 DynamoDB 表](#)
- [步骤 4：配置 DynamoDB 规则操作](#)
- [步骤 5：在 DynamoDB 中浏览数据](#)
- [第 6 步：完成教程后清理资源](#)

## 先决条件

要完成本教程，您需要：

- 一个 AWS 账户。如果没有，请参阅[设置一个 AWS 账户](#)。
- 一台运行 Windows、macOS、Linux 或 Unix 的开发计算机，用于访问。AWS Management Console 有关更多信息，请参阅 [AWS Management Console 入门](#)。
- 具有管理员权限的 IAM 用户。
- 正在运行的 AWS IoT SiteWise 风力发电场演示。当你设置演示时，它会定义模型和资产，AWS IoT SiteWise 并将数据流式传输到它们以表示风力发电场。有关更多信息，请参阅 [使用演示 AWS IoT SiteWise 示](#)。

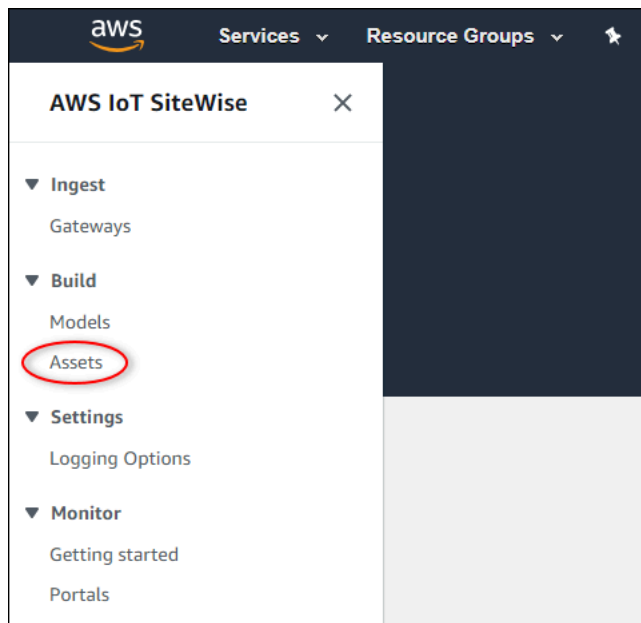
## 步骤 1：配置 AWS IoT SiteWise 以发布属性值更新

在该过程中，您在演示涡轮机资产的 Wind Speed 属性上启用属性值通知。启用属性值通知后，在 MQTT 消息中将每个值更新 AWS IoT SiteWise 发布到 AWS IoT Core。

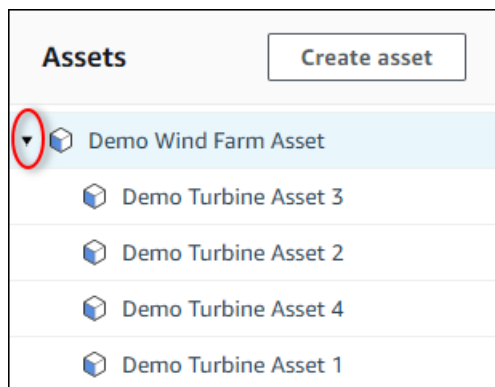
在资产属性上启用属性值更新通知

1. 登录 [AWS IoT SiteWise 控制台](#)。

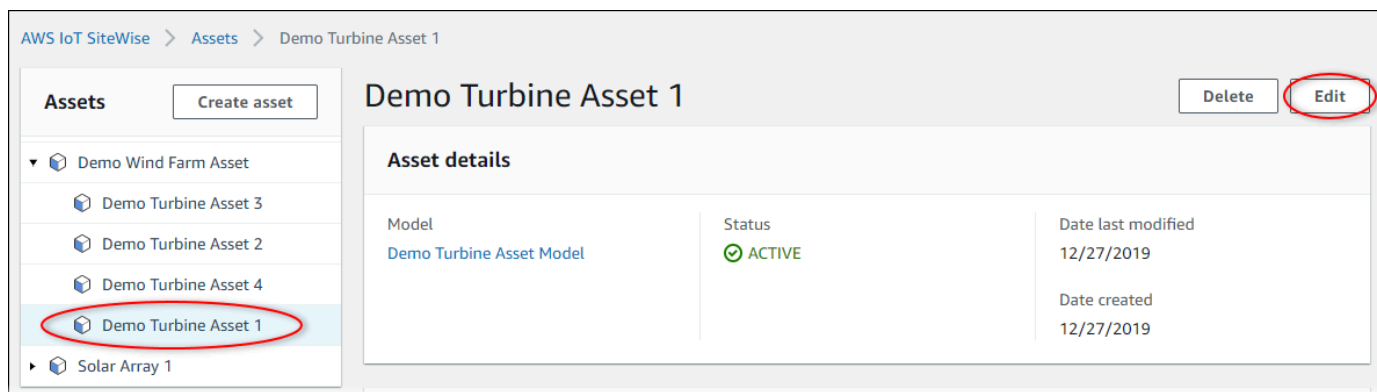
2. 查看支持的[AWS IoT SiteWise 终端节点和配额](#)，必要时切换 AWS 区域。AWS IoT SiteWise 切换到你正在运行 AWS IoT SiteWise 演示的区域。
3. 在左侧导航窗格中，选择 资产。



4. 选择 Demo Wind Farm Asset 旁边的箭头，展开风电场资产的层次结构。



5. 选择一个演示涡轮机，然后选择 编辑。





## 6. 将 Wind Speed 属性的 通知状态 更新为 已启用。

## 7. 选择页面底部的 保存资产。

## 8. 对于每个演示涡轮机资产，重复步骤 5 至 7。

## 9. 选择演示涡轮机（例如，Demo Turbine Asset 1）。

## 10. 选择 测量值。

## 11. 选择 Wind Speed 属性旁边的复制图标，以将通知主题复制到剪贴板。保存通知主题以在本教程后面使用。您只需记录一个涡轮机中的通知主题。

Torque (KiloNewton Meter)	-	⊖ Disabled	-	2.128123
Wind Speed	-	✔ Enabled	\$aws/sitewise/asset-models/d8f8f...	26.49812

通知主题应类似于以下示例。

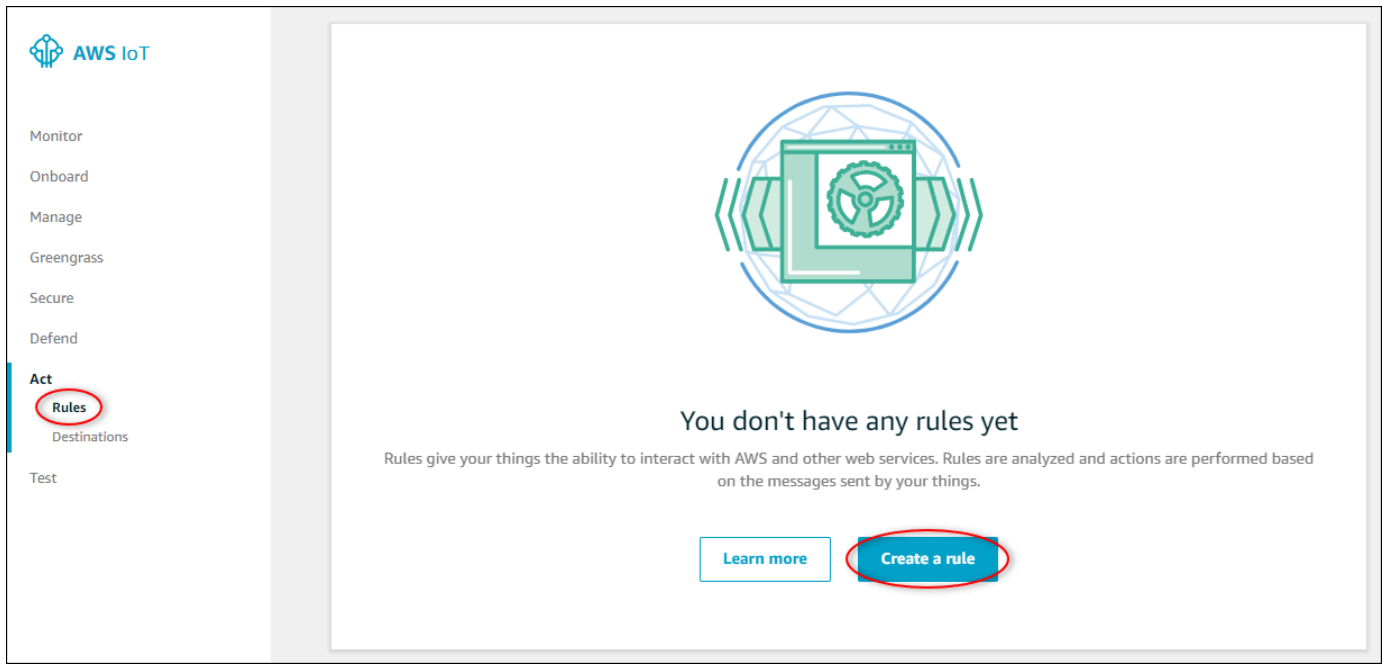
```
$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/
assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE
```

## 第 2 步：在 C AWS IoT Core 中创建规则

在此过程中，您将在 AWS IoT Core 中创建一条规则，用于解析属性值通知消息并将数据插入到 Amazon DynamoDB 表中。AWS IoT Core 核心规则解析 MQTT 消息，并根据每条消息的内容和主题执行操作。然后，您创建一个带有 DynamoDB 操作的规则，用于将数据插入您在本教程中创建的 DynamoDB 表。

### 使用 DynamoDB 操作创建规则

1. 导航到 [AWS IoT 控制台](#)。如果显示 开始 按钮，请选择该按钮。
2. 在左侧导航窗格中，选择 行动，然后选择规则。



3. 如果显示您还没有任何规则对话框，请选择创建规则。否则，选择 创建。
4. 为规则输入名称和描述。

The screenshot shows the "Create a rule" form. At the top is a blue header with the text "Create a rule". Below the header is a paragraph: "Create a rule to evaluate messages sent by your things and specify what to do when a message is received (for example, write data to a DynamoDB table or invoke a Lambda function).". There are two input fields: "Name" with the value "WindSpeedRule" and "Description" with the value "A DynamoDBv2 rule that records wind data from wind turbine assets in AWS IoT SiteWise." Both input fields are circled in red.

5. 查找您在本教程前面保存的通知主题。

```
$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/  
assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-  
cdef-33333EXAMPLE
```

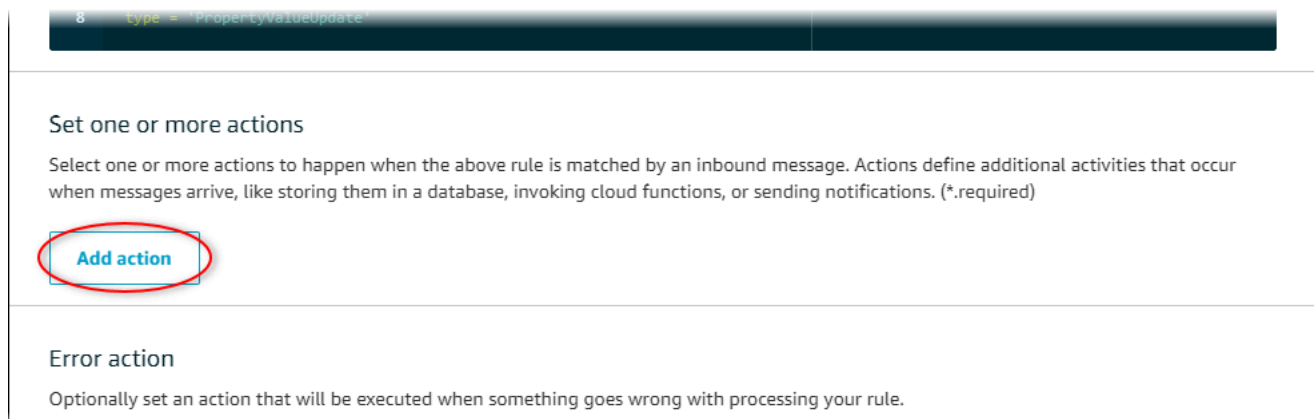
将主题中的资产 ID ( 后面的 IDassets/ ) 替换为+。这将为所有演示风力涡轮机资产选择风速属性。+ 主题筛选器接受主题中的单个级别的所有节点。您的主题应类似于以下示例。

```
$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/+/  
properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

6. 输入以下规则查询语句。将 FROM 部分中的主题替换为您的通知主题。

```
SELECT  
  payload.assetId AS asset,  
  (SELECT VALUE (value.doubleValue) FROM payload.values) AS windspeed,  
  timestamp() AS timestamp  
FROM  
  '$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/+/  
properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE'  
WHERE  
  type = 'PropertyValueUpdate'
```

7. 在设置一个或多个操作下，选择添加操作。



8 type = 'PropertyValueUpdate'

**Set one or more actions**

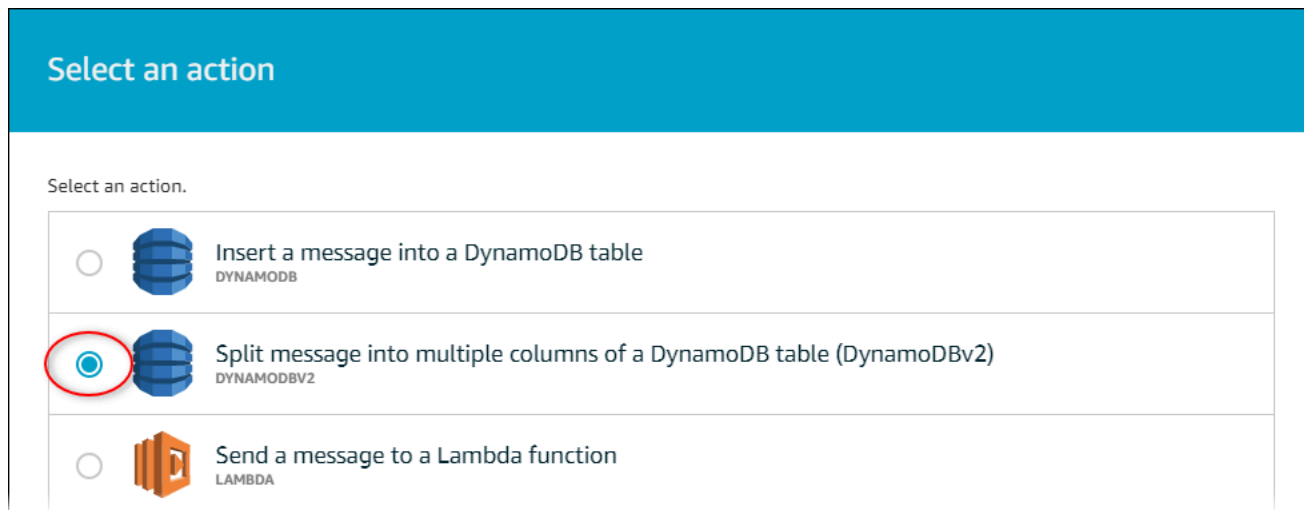
Select one or more actions to happen when the above rule is matched by an inbound message. Actions define additional activities that occur when messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. (\*.required)

**Add action**

**Error action**

Optionally set an action that will be executed when something goes wrong with processing your rule.

8. 在选择操作页面上，选择将消息拆分为 DynamoDB 表 (DynamoDBv2) 的多个列。



9. 选择页面底部的 配置操作。
10. 在 配置操作 页面，选择 创建新资源。

DynamoDB 控制台将打开新的选项卡。在完成以下过程时，请将规则操作选项卡保持打开状态。

### 步骤 3：创建 DynamoDB 表

在此过程中，您将创建一个 Amazon DynamoDB 表来接收来自规则操作的风速数据。

#### 创建 DynamoDB 表

1. 在 DynamoDB 控制台控制面板中，选择创建表。
2. 输入表的名称。

## Create DynamoDB table Tutorial ?

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

**Table name\***  ⓘ

**Primary key\*** Partition key

ⓘ

Add sort key

ⓘ

### Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

Use default settings

- No secondary indexes.
- Provisioned capacity set to 5 reads and 5 writes.
- Basic alarms with 80% upper threshold using SNS topic "dynamodb".
- Encryption at Rest with DEFAULT encryption type.

ⓘ You do not have the required role to enable Auto Scaling by default. Please refer to [documentation](#).

+ Add tags **NEW!**

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

Cancel

3. 对于主键，请执行以下操作：

- 输入 **timestamp** 以作为分区键。
- 选择 **数字** 类型。
- 选中 **添加排序键** 复选框。
- 输入 **asset** 以作为排序键，并保留默认排序键类型 **字符串**。

4. 选择**创建**。

在正在创建表通知消失时，您的表将准备就绪。

5. 返回到具有 **配置操作** 页面的选项卡。在完成以下过程时，请将 **DynamoDB** 选项卡保持打开状态。

## 步骤 4：配置 DynamoDB 规则操作

在此过程中，您将配置 Amazon DynamoDB 规则操作，以将属性值更新中的数据插入到新的 DynamoDB 表中。

### 配置 DynamoDB 规则操作

1. 在配置操作页面上，刷新表名称列表，然后选择新的 DynamoDB 表。

2. 选择创建角色以创建一个 IAM 角色，该角色向 AWS IoT 核心授予执行规则操作的访问权限。
3. 输入角色名称，然后选择创建角色。

4. 选择添加操作。
5. 选择页面底部的 创建规则 以完成创建规则的过程。

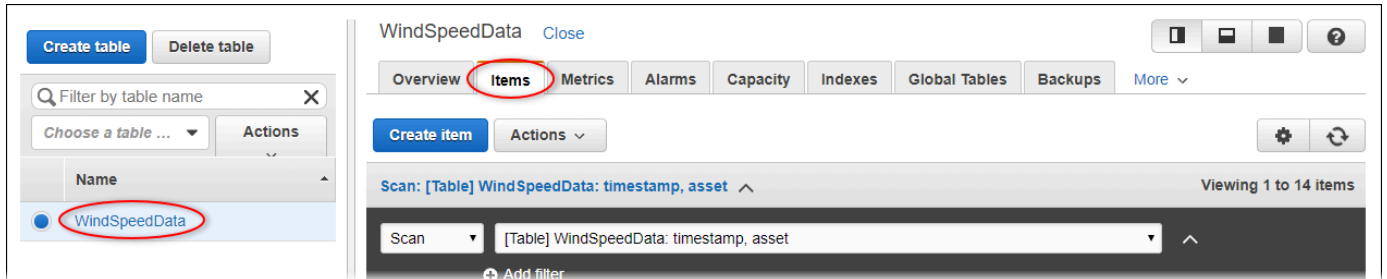
将开始在 DynamoDB 表中显示您的演示资产数据。

## 步骤 5：在 DynamoDB 中浏览数据

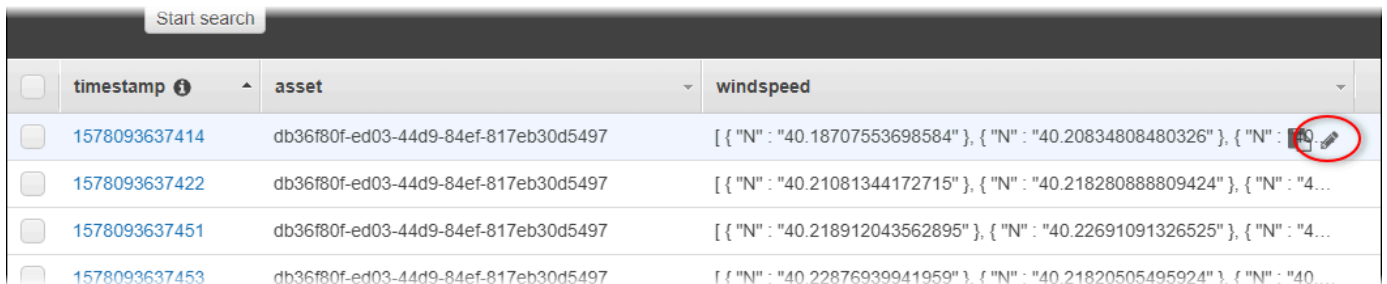
在此过程中，您将在新的 Amazon DynamoDB 表中浏览演示资产的风速数据。

在 DynamoDB 中浏览资产数据

1. 返回到打开了 DynamoDB 表的选项卡。
2. 在您以前创建的表中，选择 **项目** 选项卡以查看表中的数据。如果您没有看到表中的行，请刷新页面。如果在几分钟后没有显示行，请参阅[排查规则问题](#)。

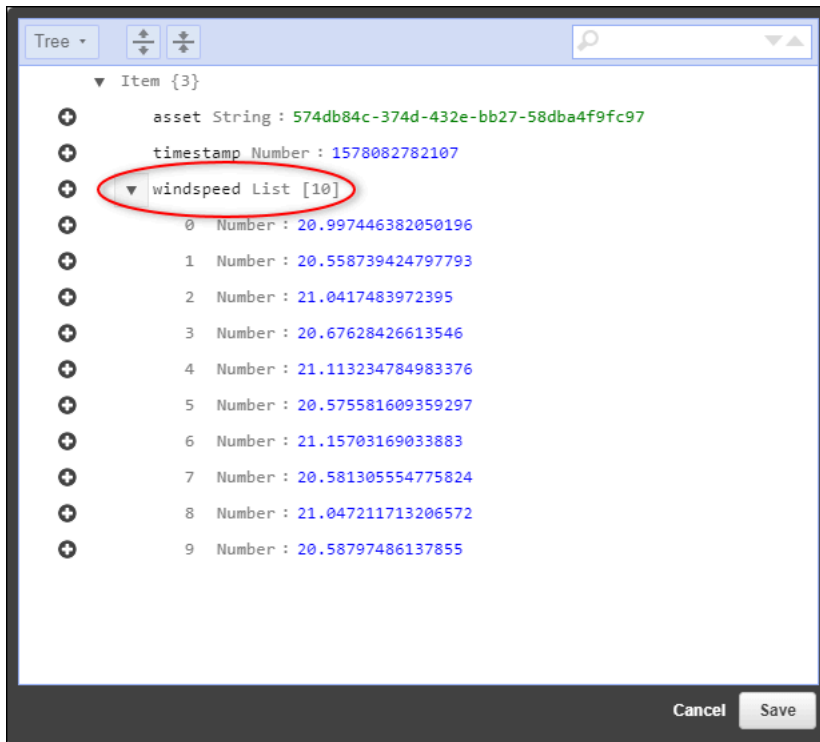


3. 在表的一行中，选择编辑图标以展开数据。



timestamp	asset	windspeed
1578093637414	db36f80f-ed03-44d9-84ef-817eb30d5497	[{"N": "40.18707553698584"}, {"N": "40.20834808480326"}, {"N": "40.21876939941959"}]
1578093637422	db36f80f-ed03-44d9-84ef-817eb30d5497	[{"N": "40.21081344172715"}, {"N": "40.218280888809424"}, {"N": "40.22691091326525"}]
1578093637451	db36f80f-ed03-44d9-84ef-817eb30d5497	[{"N": "40.218912043562895"}, {"N": "40.22691091326525"}, {"N": "40.22876939941959"}]
1578093637453	db36f80f-ed03-44d9-84ef-817eb30d5497	[{"N": "40.22876939941959"}, {"N": "40.21820505495924"}, {"N": "40.21876939941959"}]

4. 选择 windspeed 结构旁边的箭头，以展开风速数据点列表。每个列表都反映了风电场演示发送到 AWS IoT SiteWise 的一批风速数据点。如果您设置规则操作以供自己使用，则可能需要使用不同的数据格式。有关更多信息，请参阅[查询资产属性通知消息](#)。



现在您已经完成了本教程，请禁用或删除该规则并删除您的 DynamoDB 表，以免产生额外费用。要清理您的资源，请参阅[第 6 步：完成教程后清理资源](#)。

## 第 6 步：完成教程后清理资源

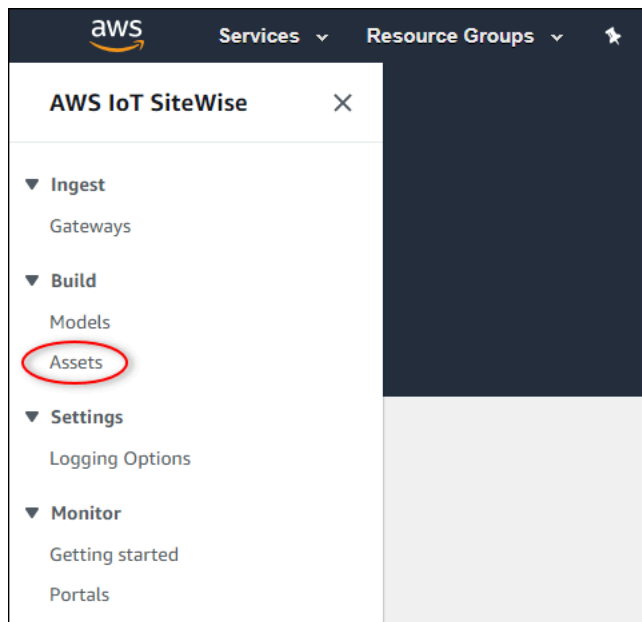
完成本教程后，清理资源以避免产生额外费用。您的演示风电场资产将在您在创建演示时选择的持续时间结束时删除。您也可以手动删除演示。有关更多信息，请参阅[删除演示 AWS IoT SiteWise 示](#)。

使用以下过程禁用属性值更新通知（如果您没有删除演示）、禁用或删除您的 AWS IoT 规则，以及删除您的 DynamoDB 表。

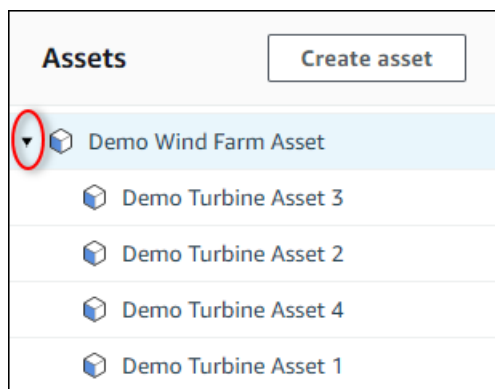
禁用资产属性上的属性值更新通知

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在左侧导航窗格中，选择 资产。

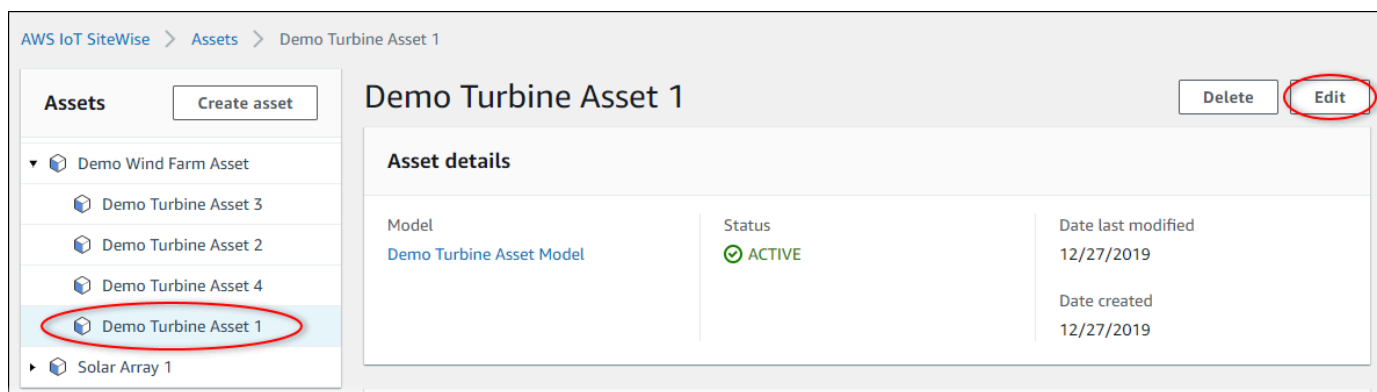




3. 选择 Demo Wind Farm Asset 旁边的箭头，展开风电场资产的层次结构。



4. 选择一个演示涡轮机，然后选择 编辑。



5. 将 Wind Speed 属性的 通知状态 更新为 已禁用。

"Wind Speed"

Enter a property alias

Must be less than 2048 characters.

Notification status

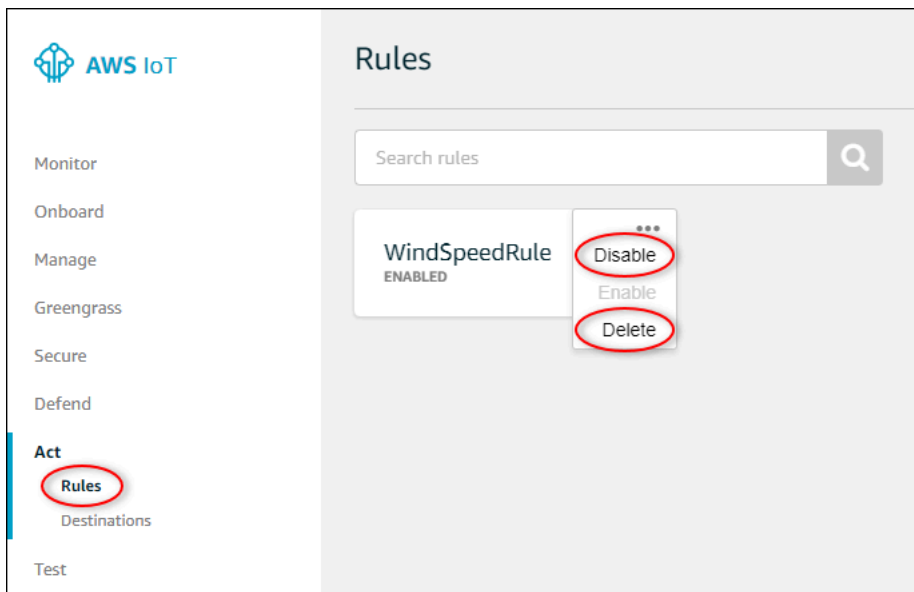
DISABLED

Notification will be published to topic \$aws/sitewise/asset-models/d8f8f20a-4d3a-491c-a9c5-352736979bdb/assets/db36f80f-ed03-44d9-84ef-817eb30d5497/properties/ca5b9e21-f19c-4ea1-8472-0e9400fc12bf

6. 选择页面底部的 保存资产。
7. 对于每个演示涡轮机资产，重复步骤 4 至 6。

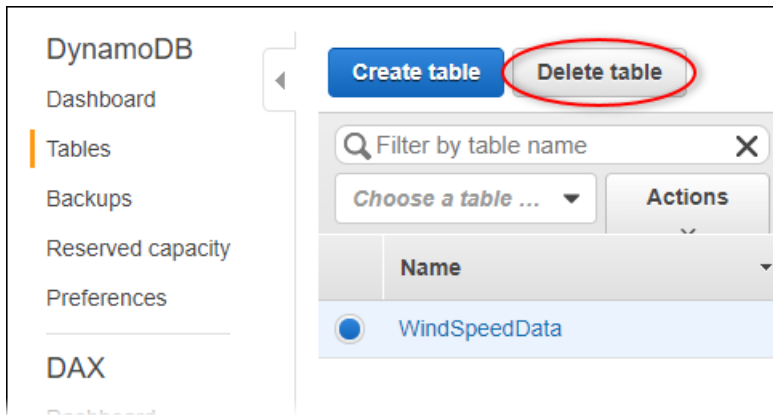
## 禁用或删除中的规则 AWS IoT Core

1. 导航到 [AWS IoT 控制台](#)。
2. 在左侧导航窗格中，选择行动，然后选择规则。
3. 选择您的规则上的菜单，然后选择 禁用 或 删除。

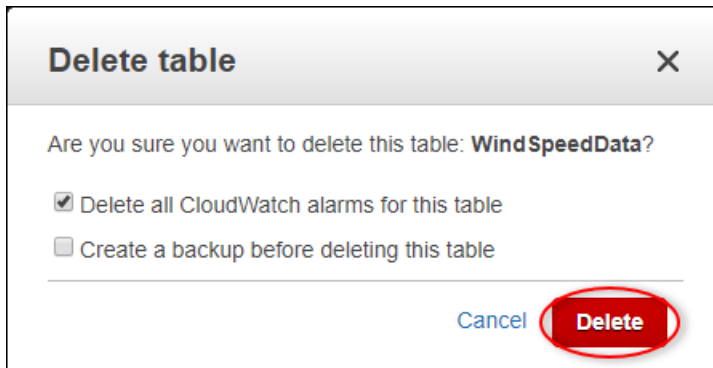


## 删除 DynamoDB 表

1. 导航到 [DynamoDB 控制台](#)。
2. 在左侧导航窗格中，选择 表。
3. 选择您之前创建的表 WindSpeedData。
4. 选择 删除表。



5. 在删除表对话框中，选择删除。



# 正在将数据摄取到 AWS IoT SiteWise

AWS IoT SiteWise 旨在有效地收集工业数据并将其与相应的资产相关联，这些资产代表了工业运营的各个方面。本文档重点介绍将数据摄入的实际方面 AWS IoT SiteWise，提供了针对不同工业用例量身定制的多种方法。有关构建虚拟工业运营的说明，请参阅[工业资产建模](#)。

您可以使用以下任一选项将 AWS IoT SiteWise 工业数据发送到：

- AWS IoT SiteWise 边缘-使用 [SiteWise Edge 网关](#) 作为 AWS IoT SiteWise 和您的数据服务器之间的中介。AWS IoT SiteWise 提供了可以在任何平台上部署的 AWS IoT Greengrass 组件，这些组件可以运行 AWS IoT Greengrass 以设置 SiteWise Edge 网关。此选项支持与 [OPC-UA 服务器协议](#) 链接。
- AWS IoT SiteWise API-使用 [AWS IoT SiteWise API](#) 上传来自任何其他来源的数据。使用我们的流式传输 [BatchPutAssetPropertyValueAPI](#) 在几秒钟内进行摄取，或者使用面向批量的 [CreateBulkImportJobAPI](#) 来促进更具成本效益的大批量摄取。
- AWS IoT 核心规则-使用 [AWS IoT 核心规则](#) 上传 AWS IoT 事物或其他 AWS 服务发布的 MQTT 消息中的数据。
- AWS IoT Events actions —使用中由特定事件触发的 [AWS IoT Events AWS IoT Events 操作](#)。此方法适用于数据上传与事件发生相关的场景。
- AWS IoT Greengrass 流管理器-使用 [AWS IoT Greengrass 流管理器](#) 使用边缘设备从本地数据源上传数据。此选项适用于数据来自本地或边缘位置的情况。

这些方法为管理来自不同来源的数据提供了一系列解决方案。深入研究每个选项的详细信息，以全面了解所提供的数据摄取功能。AWS IoT SiteWise

## 管理数据流

在深入研究创建资产模型和资产之前 AWS IoT SiteWise，请先设置数据源，将信息直接从工业设备发送到平台。AWS IoT SiteWise 旨在自动生成用于收集原始数据的数据流。每个数据流都由唯一的别名标识，因此可以更轻松地跟踪每条数据的来源。

例如，假设一个风力发电场使用 AWS IoT SiteWise Edge 网关将空气温度、螺旋桨旋转速度和功率输出时间序列数据从 OPC-UA 服务器发送到。AWS IoT SiteWise server1-windfarm/3/turbine/7/temperature 数据流别名标识来自风力发电场 #3 中涡轮机 #7 的温度值。server1 是 OPC-UA 数据源的名称。该 server1 前缀用于来自此服务器的所有数据流，有助于按其来源组织数据。

创建资产模型和资产后，通过将每个数据流与特定的资产属性关联来组织数据的涌入。这种关联不仅 AWS IoT SiteWise 允许收集，还可以根据您的资产结构处理数据。如有必要，您还可以删除数据流和资产属性之间的链接。

目前，您只能将数据流与测量值相关联。测量值是一种资产属性类型，表示设备的原始传感器数据流，如带有时间戳的温度值或带有时间戳的每分钟转数 (RPM) 值。

当这些测量值定义指标或转换时，传入的数据会触发特定的计算。值得注意的是，一个资产属性一次只能链接到一个数据流。

#### Note

一个资产属性不能同时与多个数据流相关联。

AWS IoT SiteWise 使用 TimeSeries Amazon 资源名称 (ARN) 资源来确定您的存储费用。有关更多信息，请参阅[AWS IoT SiteWise 定价](#)。

以下各节介绍如何使用 AWS IoT SiteWise 控制台或 API 来管理数据流。

主题

- [管理数据流](#)

## 管理数据流

要开始管理数据流，请完成以下操作。

#### Note

如果您是 2021 年 11 月 24 日 AWS IoT SiteWise 之后的新手，则可以跳过此部分。在此日期 AWS IoT SiteWise 之前开始使用的客户需要配置服务设置，AWS IoT SiteWise 以允许在没有资产模型和资产的情况下采集数据。

- 确保您的 IAM 角色具有以下示例中显示的权限。

Example IAM 用户策略

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "PutAssetPropertyValuesAssetPropertyOnly",
    "Effect": "Allow",
    "Action": "iotsitewise:BatchPutAssetPropertyValue",
    "Resource": "arn:aws:iotsitewise:*:*:asset/*"
  },
  {
    "Sid": "PutAssetPropertyValuesPropertyAliasAllowed",
    "Effect": "Allow",
    "Action": "iotsitewise:BatchPutAssetPropertyValue",
    "Resource": "arn:aws:iotsitewise:*:*:time-series/*"
  }
]
}
```

### Important

在将数据摄取到数据流之前，请执行以下操作。

- 如果您使用属性别名来标识数据流，则必须对 `time-series` 资源进行授权。
- 如果您使用资产 ID 来识别包含关联资产属性的资产，则必须对 `asset` 资源进行授权。

有关配置 IAM policy 的更多信息，请参阅 IAM 用户指南中的 [管理 IAM policies](#)。

- 配置数据提取设置 AWS IoT SiteWise 以允许接受与资产属性无关的数据流。

## 主题

- [配置数据提取设置](#)
- [管理数据流](#)

## 配置数据提取设置

### Console

使用 AWS IoT SiteWise 控制台配置 AWS IoT SiteWise 为接受与资产属性无关的数据流。

## 要配置数据摄取设置 ( 控制台 )

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中的 **设置** 下选择 **数据摄取**。
3. 在 **数据摄取**页面上，选择 **编辑**。
4. 在 **取消关联的数据摄取**部分，选择 **为与资产属性无关的数据流启用数据摄取**。

### Important

在配置 AWS IoT SiteWise 为接受与资产属性无关的数据流之后，您无法关闭此设置。

5. 选择 **保存**。
6. 在 **启用取消关联的数据摄取**中，选择 **更新**。取消关联的数据摄取的状态变为 **活动**。此过程可能需要数分钟完成。

## AWS CLI

使用 [PutStorageConfiguration](#) API 操作配置 AWS IoT SiteWise 为接受与资产属性无关的数据流。下一节使用 AWS CLI。

## 配置数据摄取设置 ( AWS CLI )

1. AWS IoT SiteWise 要配置为接收与资产属性无关的数据流，请运行以下命令。

### Important

在配置 AWS IoT SiteWise 为接受与资产属性无关的数据流之后，您无法关闭此设置。

```
aws iotsitewise put-storage-configuration \  
                -\--storage-type SITEWISE_DEFAULT_STORAGE \  
                -\--disassociated-data-storage ENABLED
```

您可以将 `storageType` 配置到 `MULTI_LAYER_STORAGE`。有关更多信息，请参阅 [管理数据存储](#)。

## Example 响应

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "disassociatedDataStorage": "ENABLED",
  "configurationStatus": {
    "state": "UPDATE_IN_PROGRESS"
  }
}
```

此过程可能需要数分钟完成。

- 若要检索存储配置信息，请运行以下命令。

```
aws iotsitewise describe-storage-configuration
```

## Example 响应

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "disassociatedDataStorage": "ENABLED",
  "configurationStatus": {
    "state": "ACTIVE"
  },
  "lastUpdateDate": "2021-11-16T15:54:14-07:00"
}
```

## 管理数据流

使用 AWS IoT SiteWise 控制台 或管理您的数据流 AWS CLI。

### Console

使用 AWS IoT SiteWise 控制台管理您的数据流。

要管理数据流 (控制台)

- 导航到 [AWS IoT SiteWise 控制台](#)。
- 在导航窗格中，选择 数据流。
- 或者，要添加或更新标签，请选择要编辑的数据流，然后选择 管理标签。



在 **编辑标签** 页面上，选择 **添加标签**。在 **密钥** 字段中，输入要使用的标签的名称。

选择保存。

4. 或者，在 **数据流表** 中，您可以通过以下方式筛选数据流。
  - 在第一个下拉菜单中，选择 **别名前缀或资产 ID**。
    - **别名前缀**- 数据流的别名前缀。如果您的目标数据流具有别名前缀，则可以选择此选项。
    - **资产 ID** - 在其中创建资产属性的资产 ID。如果您的目标数据流与资产属性相关联，则可以选择此选项。
  - 在第二个下拉菜单中，选择 **所有数据流、关联的数据流或 取消关联的数据流**。
    - **所有数据流**- 与资产属性关联或不关联的数据流。
    - **关联的数据流**- 与资产属性关联的数据流。
    - **取消关联的数据流**- 未与资产属性关联的数据流。
5. 选择您要管理的数据流。AWS IoT SiteWise 在页面底部的图表中显示您选择的数据流。如果选择超过 10 个，则图表将仅显示前 10 个。
6. (可选) 通过以下方式配置图表。
  - a. 在 **聚合函数** 中，选择以下选项之一。
    - **数据点计数** – 返回给定变量在当前时间间隔内的数据点总数。
    - **平均值** – 返回给定变量在当前时间间隔内的平均值。
    - **总和** – 返回给定变量在当前时间间隔内的值总和。
    - **最小值** – 返回给定变量在当前时间间隔内的最小值。
    - **最大值** – 返回给定变量在当前时间间隔内的最大值。

有关更多信息，请参阅 [在公式表达式中使用聚合函数](#)。

- b. 对于 **时间范围**，请选择以下选项之一。
  - **过去 1 小时** - 图表显示过去一小时的聚合数据。
  - **过去 2 小时** - 图表显示过去两个小时的聚合数据。
  - **过去 3 小时** - 图表显示过去三个小时的聚合数据。
  - **过去 4 小时** - 图表显示过去四个小时的聚合数据。
- c. 对于 **集成类型**，选择下列选项之一。

- 1 分钟 - 在指定时间范围内每分钟聚合一次数据。
  - 1 小时 - 在指定时间范围内每小时聚合一次数据。
7. 选择 管理数据流。
  8. 在 更新数据流关联部分的测量名称列中，执行以下任一操作。
    - 如果数据流与测量相关联，请通过选择关闭图标来删除关联。
    - 如果数据流未与测量值关联，请选择选择测量值。
  9. 在选择测量值表中，导航到目标资产，然后选择要关联的测量值。
  10. 或者，在更新资产属性别名部分中，为每个测量输入唯一的别名。
  11. 选择更新。

状态 列可以展示以下值之一。

- 待处理 - 您正在更新数据流关联或资产属性别名。
- 提交 - 保存您对关联或资产属性别名的更改。
- 错误 - AWS IoT SiteWise 无法处理您更新数据流关联或测量别名的请求。
- 成功 - 您成功更新了测量的数据流关联或别名。

## AWS CLI

使用以下 API 操作来管理您的数据流。代码示例使用 AWS CLI。

- [AssociateTimeSeriesToAssetProperty](#)— 将数据流（时间序列）与资产属性相关联。
- [DisassociateTimeSeriesFromAssetProperty](#)— 取消数据流与资产属性的关联。
- [DeleteTimeSeries](#)— 删除数据流。
- [DescribeTimeSeries](#)— 检索有关数据流的信息。
- [ListTimeSeries](#)— 检索数据流的分页列表。

### AssociateTimeSeriesToAssetProperty

要将数据流与资产属性相关联，请运行以下命令。

**⚠ Important**

指定的资产属性当前不得与任何数据流相关联。

- *data-stream-alias* 替换为要关联的数据流的别名。
- 将 *asset-ID* 替换为在其中创建资产属性的资产 ID。
- 将 *property-ID* 替换为资产属性的 ID。

```
aws iotsitewise associate-time-series-to-asset-property \  
    --alias data-stream-alias \  
    --assetId asset-ID \  
    --propertyId property-ID
```

### DisassociateTimeSeriesFromAssetProperty

要取消数据流与资产属性之间的关联，请运行以下命令。

- *data-stream-alias* 替换为要取消关联的数据流的别名。
- 将 *asset-ID* 替换为在其中创建资产属性的资产 ID。
- 将 *property-ID* 替换为资产属性的 ID。

```
aws iotsitewise disassociate-time-series-from-asset-property \  
    --alias data-stream-alias \  
    --assetId asset-ID \  
    --propertyId property-ID
```

### DeleteTimeSeries

要删除数据流，运行以下命令。

*data-stream-alias* 替换为要删除的数据流的别名。

```
aws iotsitewise delete-time-series --alias data-stream-alias
```

要识别数据流，请执行以下操作之一：

- 如果数据流未与资产属性关联，请指定该数据流的 `alias`。
- 如果数据流与资产属性相关联，请指定以下选项之一：
  - 数据流的 `alias`。
  - 标识资产属性的 `assetId` 和 `propertyId`。

## DescribeTimeSeries

使用 `DescribeTimeSeries` API 操作验证您是成功关联还是取消了数据流的关联。

要检索有关数据流的信息，请运行以下命令。

```
aws iotsitewise describe-time-series --alias data-stream-alias
```

要识别数据流，请执行以下操作之一：

- 如果数据流未与资产属性关联，请指定该数据流的 `alias`。
- 如果数据流与资产属性相关联，请指定以下选项之一：
  - 数据流的 `alias`。
  - 标识资产属性的 `assetId` 和 `propertyId`。

## ListTimeSeries

使用 `ListTimeSeries` API 操作验证是否成功删除了数据流。

要检索数据流的分页列表，请运行以下命令。

```
aws iotsitewise list-time-series
```

# 使用 API 摄取数据 AWS IoT SiteWise

使用 AWS IoT SiteWise API 将带有时间戳的工业数据发送到您的资产属性和测量属性。API 接受包含 `timestamp-quality-value (TQV)` 结构的有效负载。

使用 [BatchPutAssetPropertyValue](#) 操作上传您的数据。通过此操作，您可以一次上传多个数据条目，从多个设备收集数据，并在单个请求中将其全部发送。

**⚠ Important**

该 [BatchPutAssetPropertyValue](#) 操作受以下配额的约束：

- 每个请求最多 10 个 [条目](#)。
- 每个条目最多 10 个 [属性值](#) ( TQV 数据点 )。
- AWS IoT SiteWise 拒绝任何时间戳超过过去 7 天或将来 10 分钟以上的数据。

有关这些配额的更多信息，请参阅 AWS IoT SiteWise API 参考 [BatchPutAssetPropertyValue](#) 中的。

要标识资产属性，请指定以下选项之一：

- 数据发送到 `propertyId` 的资产属性的 `assetId` 和。
- `propertyAlias`，这是一个数据流别名（例如，`/company/windfarm/3/turbine/7/temperature`）。要使用此选项，您必须首先设置资产属性的别名。要设置属性别名，请参见 [将工业数据流映射到资产属性](#)。

以下示例演示了如何从存储在 JSON 文件中的负载发送风力涡轮机的温度和每分钟转数 (RPM) 读数。

```
aws iotsitewise batch-put-asset-property-value --cli-input-json file://batch-put-payload.json
```

`batch-put-payload.json` 中的示例负载包含以下内容。

```
{
  "entries": [
    {
      "entryId": "unique entry ID",
      "propertyAlias": "/company/windfarm/3/turbine/7/temperature",
      "propertyValues": [
        {
          "value": {
            "integerValue": 38
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          }
        }
      ]
    }
  ]
}
```

```

    }
  ]
},
{
  "entryId": "unique entry ID",
  "propertyAlias": "/company/windfarm/3/turbine/7/rpm",
  "propertyValues": [
    {
      "value": {
        "doubleValue": 15.09
      },
      "timestamp": {
        "timeInSeconds": 1575691200
      },
      "quality": "GOOD"
    }
  ]
}
]
}
}

```

负载中的每个条目都包含一个 `entryId`，您可以将其定义为任何唯一的字符串。如果任何请求条目失败，则每个错误都将包含相应请求的 `entryId`，以便您知道要重试的请求。

列表中的每个结构都 `propertyValues` 是一个 timestamp-quality-value (TQV) 结构，其中包含 a value timestamp、a 和 ( 可选 )。quality

- value – 包含下列字段之一的结构，具体取决于要设置的属性的类型：
  - booleanValue
  - doubleValue
  - integerValue
  - stringValue
- timestamp – 包含当前 Unix 纪元时间 ( 以秒为单位 ) 的结构，timeInSeconds如果您有时间精确的数据，也可以在timestamp结构中设置offsetInNanos密钥。AWS IoT SiteWise 拒绝任何时间戳在过去超过 7 天或将来超过 10 分钟的数据点。
- quality – ( 可选 ) 以下质量字符串之一：
  - GOOD – ( 默认 ) 数据不受任何问题的影响。
  - BAD – 数据受传感器故障等问题的影响。
  - UNCERTAIN – 数据受传感器不准确等问题的影响。

有关如何在计算中 AWS IoT SiteWise 处理数据质量的更多信息，请参阅[公式表达式中的数据质量](#)。

## 使用规则摄取数据 AWS IoT Core

使用中的规则将数据 AWS IoT SiteWise 从 AWS IoT 事物和其他 AWS 服务发送到 AWS IoT Core。规则转换 MQTT 消息并执行与 AWS 服务交互的操作。AWS IoT SiteWise 规则操作将消息数据从 AWS IoT SiteWise API 转发给[BatchPutAssetPropertyValue](#)操作。有关更多信息，请参阅 AWS IoT 开发人员指南中的[规则](#)和 [AWS IoT SiteWise 操作](#)。

要按照教程完成设置通过设备影子提取数据的规则所需的步骤，请参阅[从事物中摄取数据 AWS IoT](#)。

您也可以将数据从发送 AWS IoT SiteWise 到其他 AWS 服务。有关更多信息，请参阅 [与其他 AWS 服务互动](#)。

### 主题

- [授 AWS IoT 予所需的访问权限](#)
- [配置 AWS IoT SiteWise 规则操作](#)
- [通过基本提取降低成本](#)

## 授 AWS IoT 予所需的访问权限

您可以使用 IAM 角色来控制每条规则可以访问的 AWS 资源。在创建规则之前，您必须创建一个 IAM 角色，其策略允许该规则对所需 AWS 资源执行操作。AWS IoT 在运行规则时担任此角色。

如果您在 AWS IoT 控制台中创建规则操作，则可以选择根资产来创建有权访问所选资产层次结构的角色。有关如何为规则手动定义角色的更多信息，请参阅《AWS IoT 开发人员指南 AWS IoT》中的[授予所需访问权限和传递角色权限](#)。

对于 AWS IoT SiteWise 规则操作，您必须定义一个允许 `iotsitewise:BatchPutAssetPropertyValue` 访问规则向其发送数据的资产属性的角色。为了提高安全性，您可以在 `Condition` 属性中指定 AWS IoT SiteWise 资产层次结构路径。

以下示例信任策略允许访问特定资产及其子级。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```

    "Effect": "Allow",
    "Action": "iotsitewise:BatchPutAssetPropertyValue",
    "Resource": "*",
    "Condition": {
      "StringLike": {
        "iotsitewise:assetHierarchyPath": [
          "/root node asset ID",
          "/root node asset ID/*"
        ]
      }
    }
  }
]
}

```

Condition从政策中删除，以允许访问您的所有资产。以下示例信任策略允许访问您在当前区域中的所有资产。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "*"
    }
  ]
}

```

## 配置 AWS IoT SiteWise 规则操作

AWS IoT SiteWise 规则操作将启动规则的 MQTT 消息中的数据发送到中的资产属性。AWS IoT SiteWise您可以同时将多个数据条目上传到不同的资产属性，以便在一条消息中发送设备所有传感器的更新。您还可以为每个数据条目一次上传多个数据点。

### Note

当您 AWS IoT SiteWise 使用规则操作向发送数据时，您的数据必须满足BatchPutAssetPropertyValue操作的所有要求。例如，您的数据的时间戳不能早于当前 Unix 纪元时间 7 天。有关更多信息，请参阅[使用 AWS IoT SiteWise API 提取数据](#)。



对于规则操作中的每个数据条目，您可以标识资产属性，并为该资产属性指定每个数据点的时间戳、质量和值。规则操作要求所有参数均为字符串。

要在条目中标识资产属性，可以指定以下项之一：

- 资产 ID (assetId) 和 属性 ID (propertyId)，这是您要发送数据到的资产属性的值。您可以使用查找资产 ID 和财产 ID AWS IoT SiteWise 控制台。如果您知道资产 ID，则可以使用 to call AWS CLI [DescribeAsset](#)来查找资产 ID。
- 属性别名 (propertyAlias)，这是一个数据流别名（例如，/company/windfarm/3/turbine/7/temperature）。要使用此选项，您必须首先设置资产属性的别名。要了解如何设置属性别名，请参阅[将工业数据流映射到资产属性](#)。

对于每个条目中的时间戳，请使用您的设备报告的时间戳或提供的时间戳。AWS IoT Core时间戳有两个参数：

- 时间（以秒为单位）(timeInSeconds) – 传感器或设备报告数据的 Unix 纪元时间（以秒为单位）。
- 偏移量（以纳秒为单位）(offsetInNanos) –（可选）以秒为单位的时间的纳秒偏移量。

#### Important

如果您的时间戳是一个字符串、包含小数部分或不是以秒为单位，则 AWS IoT SiteWise 会拒绝该请求。您必须将时间戳转换为秒和纳秒偏移量。使用 AWS IoT 规则引擎的功能来转换时间戳。有关更多信息，请参阅下列内容：

- [获取未报告准确时间的设备的时间戳](#)
- [转换字符串格式的时间戳](#)

在执行计算、调用函数以及从消息负载中提取值的操作中，您可以为多个参数使用替换模板。有关更多信息，请参阅 AWS IoT 开发人员指南中的[替换模板](#)。

#### Note

由于替换模板中的表达式与 SELECT 语句分开评估，因此您不能使用替换模板来引用使用 AS 子句创建的别名。除了支持的函数和运算符之外，您只能引用原始负载中存在的信息。

## 主题

- [获取未报告准确时间的设备的时间戳](#)
- [转换字符串格式的时间戳](#)
- [转换纳秒精度的时间戳字符串](#)
- [示例规则配置](#)
- [规则操作问题排查](#)

## 获取未报告准确时间的设备的时间戳

如果您的传感器或设备未报告准确的时间数据，请使用 `time_stamp()` 从 [AWS IoT 规则引擎](#) 获取当前 [Unix 纪元时间](#)。此函数以毫秒为单位输出时间，因此您必须将值转换为以秒为单位的时间，并将偏移量转换为以纳秒为单位。为此，请使用以下转换：

- 对于 时间 (以秒为单位) (`timeInSeconds`)，使用 `${floor(timestamp() / 1E3)}` 将时间从毫秒转换为秒。
- 对于 偏移量 (以纳秒为单位) (`offsetInNanos`)，使用 `${(timestamp() % 1E3) * 1E6}` 计算时间戳的纳秒偏移量。

## 转换字符串格式的时间戳

如果您的传感器或设备以字符串格式报告时间数据 (例如，`2020-03-03T14:57:14.699Z`)，请使用 `time_to_epoch()` (字符串、字符串)。此函数将时间戳和格式模式作为参数输入，并以毫秒为单位输出时间。然后，您必须将时间转换为以秒为单位的时间和以纳秒为单位的偏移量。为此，请使用以下转换：

- 对于时间 (以秒为单位) (`timeInSeconds`)，使用 `${floor(time_to_epoch("2020-03-03T14:57:14.699Z", "yyyy-MM-dd'T'HH:mm:ss'Z'") / 1E3)}` 将时间戳字符串转换为毫秒，然后转换为秒。
- 对于偏移量 (以纳秒为单位) (`offsetInNanos`)，使用 `${(time_to_epoch("2020-03-03T14:57:14.699Z", "yyyy-MM-dd'T'HH:mm:ss'Z'") % 1E3) * 1E6}` 计算时间戳字符串的纳秒偏移量。

**Note**

`time_to_epoch` 函数支持高达毫秒精度的时间戳字符串。要转换精度为微秒或纳秒的字符串，请配置规则调 AWS Lambda 用的函数以将时间戳转换为数值。有关更多信息，请参阅 [转换纳秒精度的时间戳字符串](#)。

## 转换纳秒精度的时间戳字符串

如果您的设备以精度为纳秒的字符串格式发送时间戳信息（例如 `2020-03-03T14:57:14.699728491Z`），请使用以下过程配置规则操作。你可以创建一个 AWS Lambda 函数，将时间戳从字符串转换为以秒为单位的时间 (`timeInSeconds`) 和以 nanos 为单位的偏移量 (`offsetInNanos`)。然后，在规则操作参数中使用 [aws\\_lambda \( FunctionArn、InputJson \)](#) 来调用该 Lambda 函数并在规则中使用输出。

**Note**

本节包含高级说明（假设您已熟悉如何创建以下资源）：

- Lambda 函数。有关更多信息，请参阅 AWS Lambda 开发人员指南中的 [使用控制台创建 Lambda 函数](#) 或 [将 Lambda 与 AWS CLI 结合使用](#)。
- AWS IoT 带有规则操作的 AWS IoT SiteWise 规则。有关更多信息，请参阅 [使用规则摄取数据 AWS IoT Core](#)。

## 创建解析时间戳字符串的 AWS IoT SiteWise 规则操作

### 1. 创建具有以下属性的 Lambda 函数：

- 函数名称 – 使用描述性函数名称（例如 `ConvertNanosecondTimestampFromString`）。
- 运行时 — 使用 Python 3 运行时，例如 Python 3.11 (`python3.11`)。
- 权限-创建具有基本 Lambda 权限的角色 (`AWSLambdaBasicExecutionRole`)。
- 层 — 添加 AWS SDKPandas-Python311 层以供 Lambda 函数使用。numpy
- 函数代码 – 使用以下函数代码，该函数使用名为 `timestamp` 的字符串参数，并为该时间戳输出 `timeInSeconds` 和 `offsetInNanos` 值。

```
import json
import math
```

```
import numpy

# Converts a timestamp string into timeInSeconds and offsetInNanos in Unix epoch
time.
# The input timestamp string can have up to nanosecond precision.
def lambda_handler(event, context):
    timestamp_str = event['timestamp']
    # Parse the timestamp string as nanoseconds since Unix epoch.
    nanoseconds = numpy.datetime64(timestamp_str, 'ns').item()
    time_in_seconds = math.floor(nanoseconds / 1E9)
    # Slice to avoid precision issues.
    offset_in_nanos = int(str(nanoseconds)[-9:])
    return {
        'timeInSeconds': time_in_seconds,
        'offsetInNanos': offset_in_nanos
    }
```

此 Lambda 函数使用来自 `datetime64` 输入 ISO 8601 格式的时间戳字符串。NumPy

#### Note

如果您的时间戳字符串不是 ISO 8601 格式，则可以使用定义时间戳格式的 pandas 实施解决方案。有关更多信息，请参阅 [pandas.to\\_datetime](#)。

- 为规则配置 AWS IoT SiteWise 操作时，请使用以下替换模板来替换时间（以秒为单位 `timeInSeconds`）和以纳秒为单位的偏移量（`offsetInNanos`）。这些替换模板假定您的消息负载包含 `timestamp` 格式的时间戳字符串。`aws_lambda` 函数为其第二个参数使用 JSON 结构，因此您可以根据需要修改以下替换模板。

- 对于时间（以秒为单位）(`timeInSeconds`)，请使用以下替换模板。

```
{aws_lambda('arn:aws:lambda:region:account-id:function:ConvertNanosecondTimestampFromString', {'timestamp': timestamp}).timeInSeconds}
```

- 对于以偏移量（以纳秒为单位）(`offsetInNanos`)，请使用以下替换模板。

```
{aws_lambda('arn:aws:lambda:region:account-id:function:ConvertNanosecondTimestampFromString', {'timestamp': timestamp}).offsetInNanos}
```

- 对于每个参数，将##### ID 替换为您的地区和 AWS 账户 ID。如果您为 Lambda 函数使用了不同的名称，请相应更改。
3. 授予使用该 AWS IoT 权限调用您的函数的 `lambda:InvokeFunction` 权限。有关更多信息，请参阅 [aws\\_lambda\(functionArn, inputJson\)](#)。
  4. 测试您的规则（例如，使用 AWS IoT MQTT 测试客户端），并验证它 AWS IoT SiteWise 接收了您发送的数据。

如果您的规则未按预期工作，请参阅 [对 AWS IoT SiteWise 规则操作进行故障排除](#)。

### Note

此解决方案为每个时间戳字符串调用 Lambda 函数两次。如果您的规则处理各个负载中具有相同时间戳的多个数据点，则可以创建另一个规则来减少 Lambda 函数的调用次数。为此，请使用重新发布操作创建规则，该操作会调用 Lambda 并发布原始负载，其中时间戳字符串转换为 `timeInSeconds` 和 `offsetInNanos`。然后，使用规则操作创建 AWS IoT SiteWise 规则以使用转换后的有效负载。使用这种方法，您可以减少规则调用 Lambda 的次数，但可以增加规则操作的 AWS IoT 运行次数。如果将此解决方案应用于您的使用案例，请考虑每项服务的定价。

## 示例规则配置

本节包含使用 AWS IoT SiteWise 操作创建规则的示例规则配置。

Example 使用属性别名作为消息主题的示例规则操作

以下示例创建了一个规则，其 AWS IoT SiteWise 操作使用主题（通过 `topic()`）作为属性别名来标识资产属性。使用此示例定义一条规则，用于将双类型数据采集到所有风力发电场中的所有风力涡轮机。此示例要求您为所有涡轮机资产的属性定义属性别名。您需要定义另一个类似的规则来摄取整数型数据。

```
aws iot create-topic-rule \  
  --rule-name SiteWiseWindFarmRule \  
  --topic-rule-payload file://sitewise-rule-payload.json
```

`sitewise-rule-payload.json` 中的示例负载包含以下内容。

```

{
  "sql": "SELECT * FROM '/company/windfarm/+/turbine/+/+' WHERE type = 'double'",
  "description": "Sends data to the wind turbine asset property with the same alias as the topic",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [
    {
      "iotSiteWise": {
        "putAssetPropertyValueEntries": [
          {
            "propertyAlias": "${topic()}",
            "propertyValues": [
              {
                "timestamp": {
                  "timeInSeconds": "${timeInSeconds}"
                },
                "value": {
                  "doubleValue": "${value}"
                }
              }
            ]
          }
        ],
        "roleArn": "arn:aws:iam::account-id:role/MySiteWiseActionRole"
      }
    }
  ]
}

```

使用此规则操作，将以下消息作为采集数据的主题发送到风力涡轮机属性别名（例如/company/windfarm/3/turbine/7/temperature）。

```

{
  "type": "double",
  "value": "38.3",
  "timeInSeconds": "1581368533"
}

```

## Example 使用 timestamp() 确定时间的示例规则操作

以下示例创建了一个规则，其 AWS IoT SiteWise 操作通过 ID 标识资产属性，并使用 [timestamp\(\)](#) 来确定当前时间。

```
aws iot create-topic-rule \  
  --rule-name SiteWiseAssetPropertyRule \  
  --topic-rule-payload file://sitewise-rule-payload.json
```

sitewise-rule-payload.json 中的示例负载包含以下内容。

```
{  
  "sql": "SELECT * FROM 'my/asset/property/topic'",  
  "description": "Sends device data to an asset property",  
  "ruleDisabled": false,  
  "awsIotSqlVersion": "2016-03-23",  
  "actions": [  
    {  
      "iotSiteWise": {  
        "putAssetPropertyValueEntries": [  
          {  
            "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",  
            "propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",  
            "propertyValues": [  
              {  
                "timestamp": {  
                  "timeInSeconds": "${floor(timestamp() / 1E3)}",  
                  "offsetInNanos": "${(timestamp() % 1E3) * 1E6}"  
                },  
                "value": {  
                  "doubleValue": "${value}"  
                }  
              }  
            ]  
          }  
        ],  
        "roleArn": "arn:aws:iam::account-id:role/MySiteWiseActionRole"  
      }  
    }  
  ]  
}
```

使用此规则操作，向发送以下消息my/asset/property/topic以获取数据。

```
{
  "type": "double",
  "value": "38.3"
}
```

## 规则操作问题排查

要对中的 AWS IoT SiteWise 规则操作进行故障排除 AWS IoT Core，请为您的规则配置 CloudWatch 日志或配置重新发布错误操作。有关更多信息，请参阅 [对 AWS IoT SiteWise 规则操作进行故障排除](#)。

## 通过基本提取降低成本

AWS IoT Core 提供了一项名为 Basic Ingest 的功能，您可以使用该功能发送数据，AWS IoT Core 而不会产生 [AWS IoT 消息传送费用](#)。基本提取功能通过从提取路径中删除发布/订阅消息代理来优化大量数据提取工作负载的数据流。如果您知道消息应该路由到哪些规则，则可以使用“基本提取”。

要使用基本提取，您可以使用特殊主题 `$aws/rules/rule-name`，直接将消息发送到特定规则。例如，要向名为 SiteWiseWindFarmRule 的规则发送消息，您可以向主题 `$aws/rules/SiteWiseWindFarmRule` 发送消息。

如果您的规则操作使用包含 [topic\(Decimal\)](#) 的替换模板，则可以在“基本提取”特殊主题的末尾传递原始主题，例如 `$aws/rules/rule-name/original-topic`。例如，要将基本提取与之前部分中的风电场属性别名示例结合使用，您可以向以下主题发送消息。

```
$aws/rules/SiteWiseWindFarmRule//company/windfarm/3/turbine/7/temperature
```

### Note

上面的示例包含第二个斜杠 (//)，因为从规则操作可见的主题中 AWS IoT 删除了 Basic Ingest 前缀 (`$aws/rules/rule-name`)。在此示例中，规则接收主题 `/company/windfarm/3/turbine/7/temperature`。

有关更多信息，请参阅 AWS IoT 开发人员指南中的 [借助基本摄取功能降低消息收发成本](#)。



## 正在从中提取数据 AWS IoT Events

借助 AWS IoT Events 助，您可以在 AWS 云端为物联网队列构建复杂的事件监控应用程序。使用中的 IoT SiteWise 操作在 AWS IoT Events 事件发生 AWS IoT SiteWise 时向资产属性发送数据。

AWS IoT Events 旨在简化 AWS 云端物联网设备和系统的事件监控应用程序的开发。使用 AWS IoT Events，您可以：

- 检测并响应物联网机群中的变化、异常或特定情况。
- 提高您的运营效率并实现对物联网生态系统的主动管理。

AWS IoT SiteWise 通过与 AWS IoT SiteWise 操作集成，AWS IoT Events 扩展了其功能，允许您自动更新资产属性 AWS IoT SiteWise 以响应特定事件。这种交互可以简化数据摄取和管理。它还可以为您提供切实可行的见解。

有关更多信息，请参阅 AWS IoT Events 开发人员指南中的以下主题：

- [什么是 AWS IoT Events？](#)
- [AWS IoT Events 操作](#)
- [物联网 SiteWise 行动](#)

## 使用 AWS IoT Greengrass 直播管理器

AWS IoT Greengrass 流管理器是一项集成功能，可便于将数据流从本地源传输到 AWS 云端。它充当管理数据流的中间层，使在边缘运行的设备能够在数据发送到之前收集和存储数据 AWS IoT SiteWise，以进行进一步的分析和处理。

通过在 AWS IoT SiteWise 控制台上配置本地源来添加数据目标。您还可以在自定义 AWS IoT Greengrass 解决方案中使用流管理器将数据采集到。AWS IoT SiteWise

### Note

要从 OPC-UA 源摄取数据，请配置在上运行的 AWS IoT SiteWise Edge 网关。AWS IoT Greengrass 有关更多信息，请参阅 [使用 SiteWise 边缘网关](#)。

有关如何为本地源数据配置目标的更多信息，请参阅 [配置数据源](#)。

有关如何在自定义 AWS IoT Greengrass 解决方案中使用流管理器提取数据的更多信息，请参阅《AWS IoT Greengrass Version 2 开发者指南》中的以下主题：

- [什么是 AWS IoT Greengrass ?](#)
- [在 AWS IoT Greengrass Core 上管理数据流](#)
- [将数据导出到 AWS IoT SiteWise 资产属性](#)

## 使用 API 摄取数据 CreateBulkImportJob

使用 CreateBulkImportJob API 从 Amazon S3 导入大量数据。您的数据必须以 CSV 格式保存在 Amazon S3 中。数据文件可以包含以下列。

### Note

要标识资产属性，请指定以下选项之一。

- 您要向其发送数据的资产属性的 ASSET\_ID 和 PROPERTY\_ID。
  - ALIAS，这是一个数据流别名（例如，/company/windfarm/3/turbine/7/temperature）。要使用此选项，您必须首先设置资产属性的别名。要了解如何设置属性别名，请参阅[the section called “将工业数据流映射到资产属性”](#)。
- 
- ALIAS – 标识属性的属性别名，例如 OPC-UA 服务器数据流路径（例如，/company/windfarm/3/turbine/7/temperature）。有关更多信息，请参阅[将工业数据流映射到资产属性](#)。
  - ASSET\_ID – 资产的 ID。
  - PROPERTY\_ID – 资产属性的 ID。
  - DATA\_TYPE – 属性的数据类型可以是以下类型之一。
    - STRING – 最多 1024 字节的字符串。
    - INTEGER – 有符号的 32 位整数，范围是 [-2,147,483,648, 2,147,483,647]。
    - DOUBLE – 具有 [-10<sup>100</sup>, 10<sup>100</sup>] 范围和 IEEE 754 双精度的浮点数。
    - BOOLEAN – true 或 false。
  - TIMESTAMP\_SECONDS – 数据点的时间戳，以 Unix 纪元时间表示。
  - TIMESTAMP\_NANO\_OFFSET – 从 TIMESTAMP\_SECONDS 转换时的纳秒偏移量。

- QUALITY – ( 可选 ) 资产属性值的质量。值可以是以下值之一。
  - GOOD – ( 默认 ) 数据不受任何问题的影响。
  - BAD – 数据受传感器故障等问题的影响。
  - UNCERTAIN – 数据受传感器不准确等问题的影响。

有关如何在计算中 AWS IoT SiteWise 处理数据质量的更多信息，请参阅[公式表达式中的数据质量](#)。

- VALUE – 资产属性的值。

### Example .csv 格式的数据文件

```
asset_id,property_id,DOUBLE,1635201373,0,GOOD,1.0
asset_id,property_id,DOUBLE,1635201374,0,GOOD,2.0
asset_id,property_id,DOUBLE,1635201375,0,GOOD,3.0
```

```
unmodeled_alias1,DOUBLE,1635201373,0,GOOD,1.0
unmodeled_alias1,DOUBLE,1635201374,0,GOOD,2.0
unmodeled_alias1,DOUBLE,1635201375,0,GOOD,3.0
unmodeled_alias1,DOUBLE,1635201376,0,GOOD,4.0
unmodeled_alias1,DOUBLE,1635201377,0,GOOD,5.0
unmodeled_alias1,DOUBLE,1635201378,0,GOOD,6.0
unmodeled_alias1,DOUBLE,1635201379,0,GOOD,7.0
unmodeled_alias1,DOUBLE,1635201380,0,GOOD,8.0
unmodeled_alias1,DOUBLE,1635201381,0,GOOD,9.0
unmodeled_alias1,DOUBLE,1635201382,0,GOOD,10.0
```

AWS IoT SiteWise 提供了以下 API 操作来创建批量导入任务和获取有关现有任务的信息。

- [CreateBulkImportJob](#)— 创建新的批量导入任务。
- [DescribeBulkImportJob](#)— 检索有关批量导入任务的信息。
- [ListBulkImportJob](#)— 检索所有批量导入任务摘要的分页列表。

## 创建批量导入作业(AWS CLI)

使用 [CreateBulkImportJob](#) API 操作将数据从 Amazon S3 传输到 AWS IoT SiteWise。使用 [CreateBulkImportJob](#) API 以经济实惠的方式小批量摄取数据。下面的示例使用了 AWS CLI。

**⚠ Important**

在创建批量导入任务之前，必须启用 AWS IoT SiteWise 温层或 AWS IoT SiteWise 冷层。有关更多信息，请参阅 [配置存储设置](#)。

批量导入旨在将历史数据存储到 AWS IoT SiteWise。它不会在 AWS IoT SiteWise 温层或 AWS IoT SiteWise 冷层上启动计算或通知。

运行以下命令。将 *file-name* 替换为包含批量导入作业配置的文件名称。

```
aws iotsitewise create-bulk-import-job --cli-input-json file://file-name.json
```

**Example 批量导入任务配置**

以下是配置设置的示例：

- 将 *adaptive-ingestion-flag* 替换为 true 或 false。
  - 如果设置为 false，则批量导入任务会将 AWS IoT SiteWise 历史数据提取到。
  - 如果设置为 true，则批量导入任务将执行以下操作：
    - 将新数据摄入。AWS IoT SiteWise
    - 计算指标和转换，并支持对时间戳在七天以内的数据发出通知。
- 将 *delete-files-after-import-flag* 替换 true 为可在将数据摄入到 AWS IoT SiteWise 温层存储后从 S3 数据存储桶中删除。
- 将 *error-bucket* 替换为 Amazon S3 存储桶的名称，与此批量导入作业相关的错误将发送到该存储桶。
- *error-bucket-prefix* 替换为发送与此批量导入任务相关的错误的 Amazon S3 存储桶的前缀。

Amazon S3 使用前缀作为文件夹名称来整理存储桶中的数据。每个 Amazon S3 对象都有一个键，作为对象在存储桶中的唯一标识符。桶内的每个对象都只能有一个键。前缀必须以正斜杠 (/) 结尾。有关更多信息，请参阅 Amazon Simple Storage Service 用户指南中的 [使用前缀整理对象](#)。

- 将 *data-bucket* 替换为从中导入数据的 Amazon S3 存储桶的名称。
- *data-bucket-key* 替换为包含您的数据的 Amazon S3 对象的密钥。每个对象都有一个作为唯一标识符的键。每个对象有且只有一个键。
- *data-bucket-version-id* 替换为版本 ID，以识别包含您的数据的 Amazon S3 对象的特定版本。此参数为可选的。
- 将 *column-name* 替换为 .csv 文件中指定的列名。

- 将 *job-name* 替换为标识批量导入作业的唯一名称。
- *job-role-arn* 替换为允许 AWS IoT SiteWise 读取 Amazon S3 数据的 IAM 角色。

### Note

确保您的角色具有以下示例中显示的权限。将#####替换为包含您的数据的 Amazon S3 存储桶的名称。此外，将#####替换为发送与此批量导入任务相关的错误的 Amazon S3 存储桶的名称。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::data-bucket",
        "arn:aws:s3:::data-bucket/*",
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::error-bucket",
        "arn:aws:s3:::error-bucket/*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

```
{
```

```
"adaptiveIngestion": adaptive-ingestion-flag,
"deleteFilesAfterImport": delete-files-after-import-flag,
"errorReportLocation": {
  "bucket": "error-bucket",
  "prefix": "error-bucket-prefix"
},
"files": [
  {
    "bucket": "data-bucket",
    "key": "data-bucket-key",
    "versionId": "data-bucket-version-id"
  }
],
"jobConfiguration": {
  "fileFormat": {
    "csv": {
      "columnNames": [ "column-name" ]
    }
  }
},
"jobName": "job-name",
"jobRoleArn": "job-role-arn"
}
```

## Example 响应

```
{
  "jobId": "f8c031d0-01d1-4b94-90b1-afe8bb93b7e5",
  "jobStatus": "PENDING",
  "jobName": "myBulkImportJob"
}
```

## 描述批量导入作业(AWS CLI)

使用 [DescribeBulkImportJob](#) API 操作检索有关批量导入任务的信息。以下示例使用 AWS CLI。

将 *job-ID* 替换为您要检索的批量导入作业的 ID。

```
aws iotsitewise describe-bulk-import-job --job-id job-ID
```

## Example 响应

```
{
  "files":[
    {
      "bucket":"test-bucket",
      "key":"100Tags12Hours.csv"
    },
    {
      "bucket":"test-bucket",
      "key":"BulkImportData1MB.csv"
    },
    {
      "bucket":"test-bucket",
      "key":"UnmodeledBulkImportData1MB.csv"
    }
  ],
  "errorReportLocation":{
    "prefix":"errors/",
    "bucket":"test-error-bucket"
  },
  "jobConfiguration":{
    "fileFormat":{
      "csv":{
        "columnNames":[
          "ALIAS",
          "DATA_TYPE",
          "TIMESTAMP_SECONDS",
          "TIMESTAMP_NANO_OFFSET",
          "QUALITY",
          "VALUE"
        ]
      }
    }
  },
  "jobCreationDate":1645745176.498,
  "jobStatus":"COMPLETED",
  "jobName":"myBulkImportJob",
  "jobLastUpdateDate":1645745279.968,
  "jobRoleArn":"arn:aws:iam::123456789012:role/DemoRole",
  "jobId":"f8c031d0-01d1-4b94-90b1-afe8bb93b7e5"
}
```

## 列出批量导入作业(AWS CLI)

使用 [ListBulkImportJobs](#) API 操作检索所有批量导入任务摘要的分页列表。以下示例使用 AWS CLI。

```
aws iotsitewise list-bulk-import-jobs --filter COMPLETED
```

### Example 响应

```
{
  "jobSummaries":[
    {
      "id":"bdbbfa52-d775-4952-b816-13ba1c7cb9da",
      "name":"myBulkImportJob",
      "status":"COMPLETED"
    },
    {
      "id":"15ffc641-dbd8-40c6-9983-5cb3b0bc3e6b",
      "name":"myBulkImportJob2",
      "status":"RUNNING"
    }
  ]
}
```



# 使用 SiteWise 边缘网关

边 AWS IoT SiteWise 边缘网关充当您的工业设备和之间的中介 AWS IoT SiteWise。E SiteWise Edge 网关在上运行 AWS IoT Greengrass V2，支持本地数据收集和处理。您可以使用 AWS IoT SiteWise 来管理您 AWS OpsHub 的 SiteWise Edge 网关和监控现场操作。

您可以使用本地设备上的 Mon SiteWise 门户在设施中本地监控数据。有关更多信息，请参阅 [在边缘启用您的门户](#)。

## 主题

- [SiteWise 边缘网关要求](#)
- [创建 SiteWise Edge 网关](#)
- [在本地设备上安装 SiteWise Edge 网关软件](#)
- [启用边缘数据处理](#)
- [在边缘处理数据](#)
- [配置 AWS IoT SiteWise 发布者组件](#)
- [配置数据源](#)
- [将合作伙伴数据源添加到 SiteWise Edge 网关](#)
- [使用包](#)
- [管理 SiteWise Edge 网关](#)
- [在西门子工业 SiteWise 边缘上运行边缘](#)
- [在 SiteWise Edge 网关上筛选资产](#)
- [在边缘使用 AWS IoT SiteWise API](#)
- [备份和恢复 SiteWise Edge 网关](#)
- [设置 SiteWise 边缘网关 \(AWS IoT Greengrass Version 1\)](#)

## SiteWise 边缘网关要求

AWS IoT SiteWise 边缘网关 AWS IoT Greengrass V2 作为一组 AWS IoT Greengrass 组件运行，支持本地数据收集、处理和发布。要配置在上运行的 SiteWise Edge 网关 AWS IoT Greengrass V2，您必须在中创建网关 AWS Cloud 并运行 SiteWise Edge 网关软件来设置您的本地设备。

## 要求

本地设备必须满足以下要求才能安装和运行 SiteWise Edge 网关软件。

- 支持 AWS IoT Greengrass V2 核心软件版本 [v2.3.0](#) 或更高版本。有关更多信息，请参阅 AWS IoT Greengrass Version 2 开发人员指南中的[要求](#)。
- 以下支持平台之一：
  - 操作系统：Ubuntu 20.04 或更高版本  
架构：x86\_64 (AMD64) 或 ARMv8 (Aarch64)
  - OS：Red Hat Enterprise Linux (RHEL) 8  
架构：x86\_64 (AMD64) 或 ARMv8 (Aarch64)
  - OS：Amazon Linux 2  
架构：x86\_64 (AMD64) 或 ARMv8 (Aarch64)
  - OS：Debian 11  
架构：x86\_64 (AMD64) 或 ARMv8 (Aarch64)
  - OS：Windows Server 2019 及更高版本  
架构：x86\_64 (AMD64)

### Note

ARM 平台仅支持带有数据收集包的 SiteWise Edge 网关。不支持数据处理包。


- 至少 4 GB 内存。
- SiteWise Edge 网关软件至少可用 10 GB 的磁盘空间。
- 如果您计划在边缘使用处理数据 AWS IoT SiteWise，则您的本地设备还必须满足以下要求：
  - 配备 x86 64 位四核处理器。
  - 至少有 16 GB 内存。
  - 如果使用 Windows，则至少有 32 GB 的内存空间。
  - 至少有 256 GB 可用磁盘空间。
- 最低磁盘空间和计算容量要求取决于您的实施和用例所特有的各种因素。

要求 缓存间歇性 Internet 连接数据时所需的磁盘空间取决于以下因素：

- 上传的数据流数
- 每秒每个数据流的数据点数
- 每个数据点的大小
- 通信速度
- 预期网络中断
- 轮询并上传数据时所需的计算容量取决于以下因素：
  - 上传的数据流数
  - 每秒每个数据流的数据点数
- 将您的本地设备配置为访问以下 S3 存储桶：`iot-sitewise-gateway-<region>-748875242063`。
- 配置您的本地设备以确保可以访问以下端口：
  - 本地设备必须允许端口 443 上的网络入站流量。
  - 本地设备必须允许端口 443 和 8883 上的出站流量。

有关所需出站服务终端节点的完整列表，请参阅 [AWS IoT SiteWise Edge 网关所需的服务终端节点](#)。


- 以下端口保留供以下用户使用 AWS IoT SiteWise：  
80、443、3001、4569、4572、8000、8081、8082、8084、8085、8445、8085、8445、8086、9000、  
和 50010。使用预留端口通信可能导致连接终止。

 Note

AWS IoT Greengrass V2 直播管理器组件有自己的要求。有关更多信息，请参阅《AWS IoT Greengrass Version 2 开发人员指南》中的[配置](#)。

- Java 运行时环境 (JRE) 版本 11 或更高版本。Java 必须在设备的 PATH 环境变量中可用。要使用 Java 开发自定义组件，您必须安装 Java 开发工具包 (JDK)。我们建议你使用 [Amazon Corretto](#) 或 [OpenJDK](#)。

您必须具有以下权限才能使用 SiteWise Edge 网关：

 Note

如果您使用 AWS IoT SiteWise 控制台创建 SiteWise Edge 网关，则会为您添加这些权限。

- 您的 SiteWise Edge 网关的 IAM 角色必须允许您在 AWS IoT Greengrass V2 设备上使用 SiteWise 边缘网关来处理资产模型数据和资产数据。

该角色允许以下服务来代入角色：`credentials.iot.amazonaws.com`。

### 权限详细信息

该角色必须具有以下权限：

- `iotsitewise` — 允许主体在边缘检索资产模型数据和资产数据。
- `iot` — 允许您的 AWS IoT Greengrass V2 设备与之交互 AWS IoT。
- `logs` — 允许您的 AWS IoT Greengrass V2 设备向 Amazon Logs 发送 CloudWatch 日志。
- `s3` — 允许您的 AWS IoT Greengrass V2 设备从 Amazon S3 下载自定义组件工件。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:BatchPutAssetPropertyValue",
        "iotsitewise:List*",
        "iotsitewise:Describe*",
        "iotsitewise:Get*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:DescribeCertificate",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams",
        "s3:GetBucketLocation",
        "s3:GetObject",
        "iot:Connect",
        "iot:Publish",
        "iot:Subscribe",
        "iot:Receive",
        "iot:DescribeEndpoint"
      ]
    }
  ]
}
```

```

    ],
    "Resource": "*"
  }
]
}

```

## 创建 SiteWise Edge 网关

您可以使用 AWS IoT SiteWise 控制台创建 SiteWise Edge 网关。此过程详细介绍了如何创建将安装在自己的硬件上的自托管 SiteWise Edge 网关。有关创建在 Siemens In SiteWise Industrial Edge 上运行的 Edge 网关的信息，请参阅[在西门子工业 SiteWise 边缘上运行边缘](#)。

### 创建 SiteWise Edge 网关

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 Edge 网关。
3. 选择 创建网关。
4. 对于部署类型，请选择自托管网关。
5. 输入 E SiteWise Edge 网关的名称或使用生成的名称 AWS IoT SiteWise。
6. 在 Greengrass 设备操作系统下，选择要安装此 Edge 网关的设备的操作系统。 SiteWise

#### Note

数据处理包仅在 x86 平台上可用。


7. ( 可选 ) 要在边缘处理和组织数据，请在边缘功能下选择数据处理包。

#### Note

要向公司目录中的用户组授予访问此 SiteWise Edge 网关的权限，请参阅 [设置边缘功能](#)

8. ( 可选 ) 在高级配置下，执行以下操作：
  - 对于 Greengrass 核心设备，请在以下选项中选择一项：
    - 默认设置- AWS 自动使用默认设置在中创建 Greengrass 核心设备。 AWS IoT Greengrass V2

1. 输入 Greengrass 核心设备的名称或使用生成的名称。AWS IoT SiteWise
  - 高级设置-如果您要使用现有的 Greengrass 核心设备或手动创建核心设备，请选择此选项。
    1. 选择 Greengrass 核心设备或选择创建 Greengrass 核心设备在 AWS IoT Greengrass V2 控制台中创建。有关更多信息，请参阅《AWS IoT Greengrass Version 2 开发人员指南》中的[设置 AWS IoT Greengrass V2 核心设备](#)。
9. 选择 创建网关。
10. 在“生成 SiteWise Edge 网关安装程序”对话框中，选择“生成并下载”。AWS IoT SiteWise 会自动生成一个安装程序，您可以使用该安装程序来配置本地设备。


 Important

因此，请务必将安装程序文件保存在安全的位置。稍后您将使用该文件。

现在，您已经创建了 SiteWise Edge 网关，添加[数据源](#)，配置[发布者组件](#)，并让 SiteWise Edge 网关接收数据并将其发送到 AWS 云端。

## 在本地设备上安装 SiteWise Edge 网关软件

创建 SiteWise Edge 网关后，您需要在本地设备上安装 SiteWise Edge 网关软件。SiteWise 边缘网关软件可以安装在安装了 Linux 或 Windows 服务器操作系统的本地设备上。

 Important

确保您的本地设备已连接到互联网。

### Linux

以下过程使用 SSH 连接到您的本地设备。或者，您可以使用 USB 闪存驱动器或其他工具将安装程序文件传输到本地设备。如果您不想使用 SSH，请跳至下面的步骤 2：安装 SiteWise Edge 网关软件。

#### SSH 先决条件

在使用 SSH 连接到设备之前，请完成以下先决条件。

- 获取设备的 IP 地址。
- 获取用户名以连接到您的设备。
- 根据需要在您的本地计算机上安装 SSH 客户端。

您的本地计算机可能已默认安装了 SSH 客户端。您可以通过在命令行中键入 `ssh` 来进行验证。如果您的计算机无法识别该命令，可安装 SSH 客户端。

- Linux 和 macOS – 下载并安装 OpenSSH。有关更多信息，请参阅 <https://www.openssh.com>。

### 步骤 1：将安装程序复制到您的 SiteWise Edge 网关设备

以下说明说明了如何使用 SSH 客户端连接到本地设备。

1. 要连接到您的设备，请在计算机的终端窗口中运行以下命令，将 `###` 和 `IP` 替换为具有更高权限和 IP 地址的用户名。

```
ssh username@IP
```

2. 要将 AWS IoT SiteWise 生成的安装程序文件传输到您的 SiteWise Edge 网关设备，请运行以下命令。

#### Note

- `path-to-saved-installer` 替换为计算机上用于保存安装程序文件的路径和安装程序文件的名称。
- 将 `IP ##` 替换为本地设备的 IP 地址。
- `directory-to-receive-installer` 替换为本地设备上用于接收安装程序文件的路径。

```
scp path-to-saved-installer.sh user-name@IP-address:directory-to-receive-installer
```

### 步骤 2：安装 SiteWise Edge 网关软件

在以下步骤中，在 SiteWise Edge 网关设备的终端窗口中运行命令。

1. 向安装程序文件授予执行权限。

```
chmod +x path-to-installer.sh
```

2. 运行安装程序。

```
sudo ./path-to-installer.sh
```

## Windows server

### 先决条件

要安装 SiteWise Edge 网关软件，您必须具备以下先决条件：

- 安装了 Windows 服务器 2019 或更高版本
- 管理员权限
- PowerShell 已安装版本 5.1 或更高版本
- SiteWise Edge 网关安装程序已下载到将在其中进行配置的 Windows 服务器

### 步骤 1：以管理员 PowerShell 身份运行

1. 在要安装 SiteWise 边缘网关的 Windows 服务器上，以管理员身份登录。
2. 在 Windows 搜索栏 PowerShell 中输入。
3. 在搜索结果中，打开 Windows PowerShell 应用程序上的上下文（右键单击）菜单。选择以管理员身份运行。

### 步骤 2：安装 SiteWise Edge 网关软件

在 SiteWise Edge Gateway 设备的终端窗口中运行以下命令。

1. 解除封锁 SiteWise Edge 网关安装程序。

```
unblock-file path-to-installer.ps1
```

2. 运行安装程序。

```
./path-to-installer.ps1
```



**Note**

如果在系统上禁用脚本执行，请将脚本执行策略更改为 RemoteSigned。

```
Set-ExecutionPolicy RemoteSigned
```

## 启用边缘数据处理

您可以使用 AWS IoT SiteWise Edge 在本地收集、存储、组织和监控设备数据。您可以使用 SiteWise Edge 对工业数据进行建模，使用 M SiteWise onitor 创建仪表盘供操作人员在本地图像化数据。您可以在本地处理数据并将其发送到 AWS Cloud，也可以使用 AWS IoT SiteWise API 在本地进行处理。

借 AWS IoT SiteWise 助 Edge，您可以在本地处理原始数据，并选择仅向发送聚合数据，AWS Cloud 以优化带宽使用率和云存储成本。

**Note**

- AWS IoT SiteWise 在 Edge 网关上保留您的 SiteWise 边缘数据长达 30 天。数据的保留期取决于设备的可用磁盘空间。
- 如果您的 SiteWise Edge 网关已与 Edge 网关断开连接 30 天，则[数据处理包](#)将自动禁用。AWS Cloud

## 设置边缘功能

AWS IoT SiteWise 提供了以下数据包，您的 SiteWise Edge 网关可用于确定如何收集和处理您的数据。选择包以为 Edge 网关启用 SiteWise 边缘功能。

- 数据收集包使您的 SiteWise 边缘网关能够从多台 OPC-UA 服务器收集数据，然后将数据从边缘导出到。AWS Cloud 向 SiteWise Edge 网关添加数据源后，它将变为活动状态。
- 数据处理包使您的 SiteWise 边缘网关能够在边缘处理您的设备数据。例如，您可以使用资产模型来计算指标和转换。有关资产模型和资产的更多信息，请参阅[工业资产建模](#)。

**Note**

- 数据处理包仅在 x86 平台上可用。

- 数据处理包不支持网络代理。

## 配置边缘功能

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 Edge 网关。
3. 选择要为其激活 SiteWise 边缘功能的 Edge 网关。
4. 在 Edge 功能部分，选择编辑
5. 在 Edge 功能部分，选择启用数据处理包（会产生额外费用）。
6. （可选）在 Edge LDAP 连接部分，您可以授予公司目录中的用户组访问此 SiteWise Edge 网关的权限。用户组可以使用轻型目录访问协议 (LDAP) 凭据访问 SiteWise Edge 网关。然后，他们可以使用 AWS IoT SiteWise 应用程序、AWS IoT SiteWise API 操作或其他工具来管理 SiteWise Edge 网关。AWS OpsHub 有关更多信息，请参阅 [管理 SiteWise Edge 网关](#)。

### Note

您也可以使用 Linux 或 Windows 凭据访问边 SiteWise 缘网关。有关更多信息，请参阅 [使用 Linux 操作系统凭据访问您的 SiteWise Edge 网关](#)。

- a. 选择“已激活”。
  - b. 对于提供者名称，为您的 LDAP 提供者输入一个名称。
  - c. 在“主机名或 IP 地址”中，输入 LDAP 服务器的主机名或 IP 地址。
  - d. 对于端口，输入端口号。
  - e. 对于基本可分辨名称 (DN)，输入基本的可分辨名称 (DN)。  
  
支持以下属性类型：公用名 (CN)、LocalityName (L)、名称 (ST)、组织 stateOrProvince 名称 (O)、(OU)、国家名称 organizationalUnitName (C)、街道地址 (STREET)、域名组件 (DC) 和用户 ID (UID)。
  - f. 对于管理员组 DN，输入 DN。
  - g. 对于用户组 DN，输入 DN。
7. 选择保存。

现在，您已经在 Edge 网关上激活了 SiteWise 边缘功能，您需要为边缘配置资产模型。您的资产模型边缘配置指定计算资产属性的位置。您可以在边缘计算所有属性，或单独配置资产模型属性。资产模型属性包括[指标](#)、[变换](#)和[测量](#)。

有关资产属性的更多信息，请参阅 [the section called “定义数据属性”](#)。

创建资产模型后，您可以为边缘进行配置。有关您的边缘资产模型配置的更多信息，请参阅 [the section called “创建资产模型（控制台）”](#)。

#### Note

资产模型和仪表板每 10 分钟在 AWS Cloud 和您的 SiteWise Edge 网关之间自动同步一次。您也可以从本地 SiteWise Edge 网关应用程序手动同步。

## 在边缘处理数据

必须先为边缘配置资产模型，然后才能在 SiteWise 边缘处理边缘网关数据。您的资产模型边缘配置指定计算资产属性的位置。您可以选择在边缘计算所有属性并将结果发送到 AWS Cloud，也可以单独自定义每个资产属性的计算位置。有关更多信息，请参阅[启用边缘数据处理](#)。

资产属性包括指标、变换和度量：

- 指标是在指定的时间段内资产的汇总数据。您可以使用现有指标数据计算新的指标。AWS IoT SiteWise 始终将您的指标发送到 AWS 云端以进行长期存储。AWS IoT SiteWise 默认情况下，在 AWS 云端计算指标。您可以配置资产模型以计算边缘指标。AWS IoT SiteWise 将处理后的结果发送到 AWS 云端。
- 转换是将资产属性的数据点从一种形式映射到另一种形式的数学表达式。转换可以使用指标作为输入数据，并且必须在与其输入相同的位置进行计算和存储。如果将指标输入配置为在边缘计算，则 AWS IoT SiteWise 还会在边缘计算其关联的变换。
- 默认情况下，测量值格式为原始数据，您的设备会收集这些数据并发送到 AWS 云。您可以将资产模型配置为将数据存储于本地设备。

有关资产属性的更多信息，请参阅 [the section called “定义数据属性”](#)。

创建资产模型后，您可以为边缘进行配置。有关您的边缘资产模型配置的更多信息，请参阅 [the section called “创建资产模型（控制台）”](#)。

**Note**

资产模型和仪表板每 10 分钟在 AWS 云端和 SiteWise 边缘网关之间自动同步一次。您也可以从中手动同步[管理 SiteWise Edge 网关](#)。

您可以使用 AWS IoT SiteWise REST API 和 AWS Command Line Interface (AWS CLI) 在边 SiteWise 边缘网关查询边缘数据。在向 SiteWise Edge 网关查询边缘数据之前，必须满足以下先决条件：

- 必须为 REST API 设置您的凭证。有关设置凭证的更多信息，请参阅 [the section called “管理 SiteWise Edge 网关”](#)。
- SDK 终端节点必须指向您的 SiteWise Edge 网关的 IP 地址。您可以在您的 SDK 文档中了解更多信息。例如，请参阅 AWS SDK for Java 2.x Developer Guide (开发人员指南) 中的 [Specifying Custom Endpoints \(指定自定义端点\)](#)。
- 必须注册您的 SiteWise Edge 网关证书。您可以在软件开发工具包的文档中找到有关注册 SiteWise Edge 网关证书的更多信息。例如，请参阅 AWS SDK for Java 2.x 开发人员指南中的 [在 Node.js 中注册证书包](#)。

有关使用查询数据的更多信息 AWS IoT SiteWise，请参阅[从中查询数据 AWS IoT SiteWise](#)。

## 配置 AWS IoT SiteWise 发布者组件

创建 AWS IoT SiteWise Edge 网关并安装软件后，设置发布者组件，这样您的 SiteWise Edge 网关就可以将数据导出到 AWS 云端。有关更多信息，请参阅《AWS IoT Greengrass Version 2 开发者指南》中的 [AWS IoT SiteWise Publisher](#)。

### Console

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 Edge 网关。
3. 选择要为其配置发布商的 SiteWise Edge 网关。
4. 在“发布者配置”部分，选择“编辑”
5. 对于发布顺序，请选择以下选项之一：
  - 先发布最旧的数据-默认情况下，SiteWise Edge 网关会先将最旧的数据发布到云端。
  - 首先发布最新数据 — SiteWise Edge 网关首先将最新数据发布到云端。

6. (可选) 如果您不希望 SiteWise Edge 网关压缩您的数据, 请取消选择“上传数据时激活压缩”。
7. (可选) 如果您不想发布旧数据, 请选择“排除过期数据”, 然后执行以下操作:
  - 对于截止期, 输入一个值并选择一个单位。截止期必须介于 5 分钟到 7 天之间。例如, 如果截止期为三天, 则超过三天的数据不会发布到云端。
8. (可选) 要设置有关如何在本地设备上处理数据的自定义设置, 请选择本地存储设置并执行以下操作:
  - a. 在保留期中输入一个数字并选择一个单位。保留期必须介于 1 分钟到 30 天之间, 且大于或等于轮换期。例如, 如果保留期为 14 天, 则 SiteWise Edge 网关会在边缘存储超过 14 天后删除所有早于指定截止期限的数据。
  - b. 在轮换期中输入一个数字并选择一个单位。轮换期必须大于一分钟, 并且等于或小于保留期。例如, 假设轮换周期为两天, SiteWise Edge 网关会批量处理并将早于截止期限的数据保存到单个文件中。边 SiteWise 缘网关每两天将一批数据传输到以下本地目录一次: `/greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/exports`。
  - c. 在存储容量中, 输入一个大于或等于 1 的值。如果存储容量为 2 GB, 则当本地存储的数据超过 2 GB 时, SiteWise Edge 网关会开始删除数据。
9. 选择保存。

## AWS CLI

您可以使用 [UpdateGatewayCapabilityConfiguration](#) API 来配置发布者。将 `capabilityNamespace` 参数设置为 `iotsitewise:publisher:2`。

发布者提供了您可以自定义的以下配置参数:

### SiteWisePublisherConfiguration

#### publishingOrder

数据发布到云的顺序。此参数的值可以是以下值之一:

- `TIME_ORDER` (先发布最早的数据) — 默认情况下, 网关会先将最早的数据发布到云。
- `RECENT_DATA` (先发布最新数据) — 网关会先将最新数据发布到云。

#### dropPolicy

(可选) 控制将数据发布到云的策略。

## cutoffAge

早于截止期的数据不会发布到云。截止期必须介于 5 分钟到 7 天之间。

当您指定截止期时，可以使用 m、h 和 d。请注意，m 表示分钟，h 表示小时，d 表示天。

## exportPolicy

( 可选 ) 管理边缘数据存储的策略。本策略适用于早于截止期的数据。

## retentionPeriod

您的 SiteWise Edge 网关在存储到指定保留期后，会从本地存储中删除边缘上所有早于截止期的数据。保留期必须介于 1 分钟到 30 天之间，且大于或等于轮换期。

当您指定保留期时，可以使用 m、h 和 d。请注意，m 表示分钟，h 表示小时，d 表示天。

## rotationPeriod

将早于截止期的数据进行批处理和保存至单个文件的时间间隔。SiteWise Edge 网关在每个轮换周期结束时将一批数据传输到以下本地目录：`/greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/exports`。轮换期必须多于 1 分钟，且等于或小于保留期。

当您指定轮换期时，可以使用 m、h 和 d。请注意，m 表示分钟，h 表示小时，d 表示天。

## exportSizeLimitGB

允许存储在本地的数据最大以 GB 为单位。如果超过此配额，SiteWise Edge 网关会开始删除最早的数据，直到本地存储的数据大小等于或小于配额。该参数值必须大于或等于 1。

Example 发布者配置：

发布者命名空间：`iotsitewise:publisher:2`

```
{
  "SiteWisePublisherConfiguration": {
    "publishingOrder": "TIME_ORDER",
    "dropPolicy": {
      "cutoffAge": "7d",
      "exportPolicy": {
```

```
        "retentionPeriod": "7d",
        "rotationPeriod": "6h",
        "exportLocation": "/greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/
exports",
        "exportSizeLimitGB": 10
    }
}
}
```

## 配置数据源

设置 AWS IoT SiteWise Edge 网关后，您可以配置数据源，以便您的 SiteWise Edge 网关可以将来自本地工业设备的数据摄取到 AWS IoT SiteWise。每个源都代表一个本地服务器，例如 OPC-UA 服务器，您的 SiteWise 边缘网关可以连接和检索工业数据流。有关设置 SiteWise Edge 网关的更多信息，请参阅[配置 AWS IoT Greengrass V1 SiteWise Edge 网关](#)。

### Note

AWS IoT SiteWise 每次添加或编辑源时都会重新启动 SiteWise Edge 网关。您的 SiteWise Edge 网关在重启时不会提取数据。重启 SiteWise Edge 网关的时间取决于 E SiteWise dge 网关源上的标签数量。重启时间可以从几秒钟（对于标签较少的 SiteWise Edge 网关）到几分钟（对于具有许多标签的 SiteWise Edge 网关）不等。

创建源后，您可以将数据流与资产属性关联。有关如何创建和使用资产的更多信息，请参阅[工业资产建模](#)和[将工业数据流映射到资产属性](#)。

您可以查看 CloudWatch 指标以验证数据源是否已连接 AWS IoT SiteWise。有关更多信息，请参阅[AWS IoT Greengrass Version 2 网关指标](#)。

目前，AWS IoT SiteWise 支持以下数据源协议：

- [OPC-UA](#) — 一种用于工业自动化的通信协议 machine-to-machine ( M2M ) 。

### Note

SiteWise AWS IoT Greengrass V2 当前运行的边缘网关不支持 Modbus TCP 和以太网 IP 源。

## 主题

- [配置 OPC-UA 源](#)
- [配置数据来源身份验证](#)
- [为源服务器数据选择目标](#)

## 配置 OPC-UA 源

您可以使用 AWS IoT SiteWise 控制台或 SiteWise Edge 网关功能来定义并向边缘网关添加 OPC-UA 源，以表示本地 OPC-UA 服务器。SiteWise

## 主题

- [配置 OPC-UA 源 \(控制台\)](#)
- [配置 OPC-UA 源 \(CLI\)](#)
- [使您的 OPC-UA 源服务器能够信任 Edge 网关 SiteWise](#)
- [使用 OPC-UA 筛选数据摄取范围](#)
- [使用 OPC-UA 节点筛选条件](#)

## 配置 OPC-UA 源 (控制台)

使用控制台配置 OPC-UA 源 AWS IoT SiteWise

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 网关。
3. 选择边 SiteWise 缘网关以添加 OPC-UA 源。
4. 选择添加数据来源。
5. 输入源名称。
6. 输入数据源服务器的本地终端节点。端点可以是 IP 地址或主机名。您也可以向本地端点添加端口号。例如，您的本地终端节点可能如下所示：**opc.tcp://203.0.113.0:49320**

### Note

如果您的 SiteWise 边缘网关装有 Deployment type 西门子工业边缘设备 (全新)，并且您想从与边缘应用程序相同的西门子工业边缘设备上运行的 Edge OPC UA 服务器应用程序提取数据，请 AWS IoT SiteWise 输入。**opc.tcp://ie-opcua:48010**



7. (可选) 对于可供选择的节点 ID，请添加节点筛选器以限制向中提取哪些数据流。AWS Cloud默认情况下，SiteWise Edge 网关使用服务器的根节点来摄取所有数据流。您可以使用节点筛选器通过仅包含您建模的数据的路径来减少 SiteWise Edge 网关的启动时间和 CPU 使用率 AWS IoT SiteWise。默认情况下，SiteWise 边缘网关会上传除以开头的路径之外的所有 OPC-UA 路径。/Server/要定义 OPC-UA 节点筛选条件，可以使用节点路径以及 \* 和 \*\* 通配符。有关更多信息，请参阅 [使用 OPC-UA 节点筛选条件](#)。
8. 在“目标”中，选择源数据的目的地：
  - AWS IoT SiteWise 实时-选择此选项可将数据直接发送到 AWS IoT SiteWise 存储器。实时摄取和监控数据，并在边缘处理数据。
  - AWS IoT SiteWise 使用 Amazon S3 进行缓冲 — 以镶木地板格式将数据发送到 Amazon S3，然后导入 AWS IoT SiteWise 存储。选择此选项可批量摄取数据，并以经济实惠的方式存储历史数据。您可以配置您的首选 Amazon S3 存储桶位置，以及您希望将数据上传到 Amazon S3 的频率。您还可以选择在数据摄取后如何处理数据。AWS IoT SiteWise 您可以选择同时在 Amazon S3 SiteWise 和 Amazon S3 中提供数据，也可以选择自动将其从 Amazon S3 中删除。
    - Amazon S3 存储桶是一种暂存和缓冲机制，支持 parquet 格式的文件。
    - 如果您选中“将数据导入 AWS IoT SiteWise 存储”复选框，则会先将数据上传到 Amazon S3，然后上传到 AWS IoT SiteWise 存储中。
      - 如果您选中“从 Amazon S3 中删除数据”复选框，则在数据导入 SiteWise 存储后，数据将从 Amazon S3 中删除。
      - 如果您清除“从 Amazon S3 中删除数据”复选框，则数据将存储在 Amazon S3 和 SiteWise 存储中。
    - 如果清除“将数据导入 AWS IoT SiteWise 存储”复选框，则数据将仅存储在 Amazon S3 中。它不会导入到 SiteWise 存储中。

[管理数据存储](#)有关 AWS IoT SiteWise 提供的各种存储选项的详细信息，请访问。要了解有关定价选项的更多信息，请参阅[AWS IoT SiteWise 定价](#)。

- AWS IoT Greengrass 流管理器 — 使用 AWS IoT Greengrass 流管理器将数据发送到以下 AWS Cloud 目的地：传入的频道、Amazon Kinesis Data Streams 中的流 AWS IoT Analytics、中的资产属性或亚马逊简单存储服务 (Amazon S3) Simple Storage Service AWS IoT SiteWise 中的对象。有关更多信息，请参阅《AWS IoT Greengrass Version 2 开发者指南》中的 [“在 AWS IoT Greengrass Core 上管理数据流”](#)。

输入 AWS IoT Greengrass 直播的名称。

配置数据源时，用于选择的节点 ID 用于确定数据流的目的地。

- 如果使用 Amazon S3 将相同的数据同时发布到 AWS IoT SiteWise 实时和 AWS IoT SiteWise 缓冲区，则必须添加两个发布到两个目标的数据源。
- 要拆分数据，以便使用 Amazon S3 将其中一部分 AWS IoT SiteWise 实时发布到 B AWS IoT SiteWise *uffered*，另一部分发布到 *Buffered*，您必须筛选以下数据别名：

```
/Alias01/Data1  
/Alias02/Data1  
/Alias03/Data1  
/Alias03/Data2
```

例如，您可以使用 Amazon S3 添加指向 `/**/Data1` 节点筛选器的 AWS IoT SiteWise 实时数据源，以及另一个指向 `/**/Data2` AWS IoT SiteWise 缓冲的数据源

9. 在“高级配置”窗格中，您可以执行以下操作：

- a. 为源服务器和 SiteWise Edge 网关之间传输的连接和数据选择消息安全模式。此字段是 OPC-UA 安全策略和消息安全模式的组合。选择与您为 OPC-UA 服务器指定的相同的安全策略和消息安全模式。
- b. 如果您的来源需要身份验证，请从身份验证配置列表中选择一个 AWS Secrets Manager 密钥。SiteWise Edge 网关在连接到此数据源时使用此密钥中的身份验证凭据。您必须将密钥附加到 SiteWise Edge 网关的 AWS IoT Greengrass 组件，才能将其用于数据源身份验证。有关更多信息，请参阅 [the section called “配置数据源身份验证”](#)。

 Tip

您的数据服务器可能有一个名为 `允许匿名登录` 的选项。如果此选项为 `是`，则说明您的源不需要身份验证。

- c. (可选) 输入数据流前缀。SiteWise Edge 网关将此前缀添加到来自该源的所有数据流。使用数据流前缀，可以区分来自不同源的同名数据流。在您的账户中，每个数据流都应具有唯一的名称。
- d. (可选) 对于属性组，选择添加新组。
  - i. 为属性组输入名称。
  - ii. 对于属性：

1. 对于节点路径，添加 OPC-UA 节点过滤器以限制上传到哪些 OPC-UA 路径。AWS IoT SiteWise 其格式类似于用于选择的节点 ID。

iii. 对于组设置，请执行以下操作：

1. 在数据质量设置中，选择您希望 C AWS IoT SiteWise collector 采集的数据质量类型。

2. 对于扫描模式设置，请配置以下标准订阅属性：

- 对于扫描模式，请选择以下选项之一：有关扫描模式的更多信息，请参阅 [the section called “使用 OPC-UA 筛选数据摄取范围”](#)。
- 要发送每个数据点，请选择订阅并设置以下内容：
  - [数据更改触发器](#)-启动数据更改警报的条件。
  - [订阅队列大小](#)-OPC—UA 服务器上特定指标的队列深度，其中监控项目的通知已排队。
  - [订阅发布间隔](#)-创建订阅时指定的发布周期间隔（以毫秒为单位）。
  - [快照间隔](#)-用于确保 AWS IoT SiteWise Edge 摄取稳定的数据流的快照频率超时设置。
  - [扫描速率](#) — 您希望 SiteWise Edge 网关读取寄存器的速率。AWS IoT SiteWise 自动计算您的 SiteWise Edge 网关允许的最低扫描速率。
- 要按特定间隔发送数据点，请选择轮询并输入扫描速率。

3. 如果您选择 Subscibe 的扫描模式，请为信号源配置 Deadband 类型和相关设置。这控制了您的源发送给您的 AWS IoT SiteWise 数据以及它丢弃了哪些数据。有关死区设置的更多信息，请参阅 [the section called “使用 OPC-UA 筛选数据摄取范围”](#)。

10. 选择保存。

## 配置 OPC-UA 源 (CLI)

您可以使用为 SiteWise Edge 网关定义 OPC-UA 数据源。AWS CLI 为此，请创建一个 OPC-UA 功能配置 JSON 文件，然后使用 [update-gateway-capability-configuration](#) 命令更新 SiteWise Edge 网关配置。您必须在单个功能配置中定义所有 OPC-UA 源。

此功能具有以下命名空间。

- `iotsitewise:opcuacollector:2`

## 请求语法

```

{
  "sources": [
    {
      "name": "string",
      "endpoint": {
        "certificateTrust": {
          "type": "TrustAny" | "X509",
          "certificateBody": "string",
          "certificateChain": "string",
        },
        "endpointUri": "string",
        "securityPolicy": "NONE" | "BASIC128_RSA15" | "BASIC256" | "BASIC256_SHA256" |
"AES128_SHA256_RSAOAE" | "AES256_SHA256_RSAPSS",
        "messageSecurityMode": "NONE" | "SIGN" | "SIGN_AND_ENCRYPT",
        "identityProvider": {
          "type": "Anonymous" | "Username",
          "usernameSecretArn": "string"
        },
      },
      "nodeFilterRules": [
        {
          "action": "INCLUDE",
          "definition": {
            "type": "OpcUaRootPath",
            "rootPath": "string"
          }
        }
      ]
    },
    "measurementDataStreamPrefix": "string"
  },
  "destination": {
    "type": "StreamManager",
    "streamName": "string",
    "streamBufferSize": integer
  },
  "propertyGroups": [
    {
      "name": "string",
      "nodeFilterRuleDefinitions": [
        {
          "type": "OpcUaRootPath",
          "rootPath": "string"
        }
      ]
    }
  ]
}

```

```

    ],
    "deadband": {
      "type": "PERCENT" | "ABSOLUTE",
      "value": double,
      "eguMin": double,
      "eguMax": double,
      "timeoutMilliseconds": integer
    },
    "scanMode": {
      "type": "EXCEPTION" | "POLL",
      "rate": integer
    },
    "dataQuality": {
      "allowGoodQuality": true | false,
      "allowBadQuality": true | false,
      "allowUncertainQuality": true | false
    },
    "subscription": {
      "dataChangeTrigger": "STATUS" | "STATUS_VALUE" | "STATUS_VALUE_TIMESTAMP",
      "queueSize": integer,
      "publishingIntervalMilliseconds": integer,
      "snapshotFrequencyMilliseconds": integer
    }
  }
]
}
]
}

```

请求正文

来源

OPC-UA 源定义结构的列表，每个结构均包含以下信息：

name

源的唯一友好名称。

终端节点

一个包含以下信息的终端节点结构：

证书信托

一个包含以下信息的证书信任策略结构：

## 类型

源的证书信任模式。选择以下操作之一：

- **TrustAny**— 边 SiteWise 缘网关在连接到 OPC-UA 源时信任任何证书。
- **X509**— 边 SiteWise 缘网关在连接到 OPC-UA 源时信任 X.509 证书。如果选择此选项，则必须在 `certificateTrust` 中定义 `certificateBody`。也可以在 `certificateTrust` 中定义 `certificateChain`。

## 证书正文

( 可选 ) X.509 证书的正文。

如果您在 `certificateTrust` 中为 `type` 选择 X509，则此字段为必填字段。

## 证书链

( 可选 ) X.509 证书的信任链。

仅当您在 `certificateTrust` 中为 `type` 选择 X509 时使用此字段。

## EndpointURI

OPC-UA 源的本地终端节点。例如，您的本地终端节点可能类似于 `opc.tcp://203.0.113.0:49320`。

## 安全策略

要使用安全策略，可保护从 OPC-UA 源读取的消息。选择以下操作之一：

- **NONE**— 边 SiteWise 缘网关不保护来自 OPC-UA 来源的消息。建议您选择不同的安全策略。如果选择此选项，则还必须为 `messageSecurityMode` 选择 NONE。
- **BASIC256\_SHA256** — Basic256Sha256 安全策略。
- **AES128\_SHA256\_RSA0AEP** — Aes128\_Sha256\_Rsa0aep 安全策略。
- **AES256\_SHA256\_RSAPSS** — Aes256\_Sha256\_RsaPss 安全策略。
- **BASIC128\_RSA15** — ( 已弃用 ) OPC-UA 规范中已弃用 Basic128Rsa15 安全策略，因为它被视为不再安全。建议您选择不同的安全策略。有关更多信息，请参阅 [Basic128Rsa15](#)。
- **BASIC256** — ( 已弃用 ) OPC-UA 规范中已弃用 Basic256 安全策略，因为它被视为不再安全。建议您选择不同的安全策略。有关更多信息，请参阅 [Basic256](#)。

**⚠ Important**

如果您选择除 NONE 外的其他安全策略，则 `messageSecurityMode` 必须选择 SIGN 或 SIGN\_AND\_ENCRYPT。您还必须将源服务器配置为信任 SiteWise Edge 网关。有关更多信息，请参阅 [使您的 OPC-UA 源服务器能够信任 Edge 网关 SiteWise](#)。

## 消息 SecurityMode

用于保护与 OPC-UA 源的连接的消息安全模式。选择以下操作之一：

- NONE— 边 SiteWise 缘网关无法保护与 OPC-UA 源的连接。建议您选择不同的消息安全模式。如果选择此选项，则还必须为 `securityPolicy` 选择 NONE。
- SIGN— 在 SiteWise Edge 网关和 OPC-UA 源之间传输的数据已签名，但未加密。
- SIGN\_AND\_ENCRYPT — 对网关和 OPC-UA 源之间的传输中数据进行签名和加密。

**⚠ Important**

如果您选择除 NONE 外的消息安全模式，则必须选择除 NONE 外的 `securityPolicy`。您还必须将源服务器配置为信任 SiteWise Edge 网关。有关更多信息，请参阅 [使您的 OPC-UA 源服务器能够信任 Edge 网关 SiteWise](#)。

## 身份提供者

一个包含以下信息的身份提供程序结构：

### 类型

源所需的身份验证凭证的类型。选择以下操作之一：

- Anonymous — 源不需要身份验证即可连接。
- Username — 源需要用户名和密码才能连接。如果选择此选项，则必须在 `identityProvider` 中定义 `usernameSecretArn`。

### 用户名 SecretArn

( 可选 ) 密钥的 ARN。AWS Secrets Manager SiteWise Edge 网关在连接到此源时使用此密钥中的身份验证凭证。您必须将密钥附加到 SiteWise Edge 网关的 IoT SiteWise 连接器，才能将其用于源身份验证。有关更多信息，请参阅 [配置数据来源身份验证](#)。

如果您在 `identityProvider` 中为 `type` 选择 `Username`，则此字段为必填字段。

## 节点 FilterRules

定义要发送到云端的 OPC-UA 数据流路径的节点筛选规则结构列表。AWS 您可以使用节点筛选器通过仅包含您建模的数据的路径来减少 SiteWise Edge 网关的启动时间和 CPU 使用率 AWS IoT SiteWise。默认情况下，SiteWise 边缘网关会上传除以开头的路径之外的所有 OPC-UA 路径。/Server/要定义 OPC-UA 节点筛选条件，可以使用节点路径以及 \* 和 \*\* 通配符。有关更多信息，请参阅 [使用 OPC-UA 节点筛选条件](#)。

此列表中的每个结构都必须包含以下信息：

### action

此节点筛选器规则的操作。可以选择以下选项：

- INCLUDE— SiteWise Edge 网关仅包含符合此规则的数据流。

### 定义

一个包含以下信息的节点筛选器规则结构：

### 类型

此规则的节点筛选器路径的类型。可以选择以下选项：

- OpcUaRootPath— SiteWise Edge 网关根据 OPC-UA 路径层次结构的根目录评估此节点筛选路径。

### rootPath

根据 OPC-UA 路径层次结构的根进行评估的节点筛选器路径。此路径必须以 / 开头。

## 测量DataStream前缀

要在源中所有数据流的前面加上的字符串。SiteWise Edge 网关将此前缀添加到来自该源的所有数据流。使用数据流前缀，可以区分来自不同源的同名数据流。在您的账户中，每个数据流都应具有唯一的名称。

## 属性组

( 可选 ) 定义协议要求的 `deadband` 和 `scanMode` 的属性组列表。

### name

属性组的名称。这应该是一个唯一的标识符。



## 死带

deadband 结构包含以下信息：

### 类型

支持的死区类型。接受的值为：ABSOLUTE 和 PERCENT。

### 值

死区值。如果 type 是 ABSOLUTE，则此值为无单位的双精度。如果 type 是 PERCENT，则此值是 1 和 100 之间的双精度。

### egumin

( 可选 ) 使用 PERCENT 死区时的工程单位最小值。如果 OPC-UA 服务器未配置工程单元，则可以设置此项。

### eguMax

( 可选 ) 使用 PERCENT 死区时的工程单位最大值。如果 OPC-UA 服务器未配置工程单元，则可以设置此项。

### 超时毫秒

超时前的持续时间，以毫秒为单位。最少为 100。

## 扫描模式

scanMode 结构包含以下信息：

### 类型

支持的 scanMode 类型。接受的值为：POLL 和 EXCEPTION。

### 费率

扫描模式的采样间隔。

## 节点FilterRule定义

( 可选 ) 包含在属性组中的节点路径列表。属性组不能重叠。如果您不为此字段指定值，则该组包含根目录下的所有路径，且您无法创建其他属性组。nodeFilterRuleDefinitions 结构包含以下信息：

### 类型

OpcUaRootPath 是唯一支持的类型。这指定 rootPath 值是相对于 OPC-UA 浏览空间根目录的路径。

## rootPath

以逗号分隔的列表，用于指定要包含在属性组中的路径（相对于根路径）。

### 功能配置示例

以下示例根据存储在 JSON 文件中的有效负载定义了 OPC-UA SiteWise Edge 网关功能配置。

```
aws iotsitewise update-gateway-capability-configuration \  
--capability-namespace "iotsitewise:opcuacollector:2" \  
--capability-configuration file://opc-ua-configuration.json
```

### Example : OPC-UA 源配置

以下 `opc-ua-configuration.json` 文件定义基本的、不安全的 OPC-UA 源配置。

```
{  
  "sources": [  
    {  
      "name": "Wind Farm #1",  
      "endpoint": {  
        "certificateTrust": {  
          "type": "TrustAny"  
        },  
        "endpointUri": "opc.tcp://203.0.113.0:49320",  
        "securityPolicy": "NONE",  
        "messageSecurityMode": "NONE",  
        "identityProvider": {  
          "type": "Anonymous"  
        },  
        "nodeFilterRules": []  
      },  
      "measurementDataStreamPrefix": ""  
    }  
  ]  
}
```

### Example : 带有已定义属性组的 OPC-UA 源配置

以下 `opc-ua-configuration.json` 文件定义了一个基本的、不安全的 OPC-UA 源配置，其中包含已定义的属性组。

```
{
  "sources": [
    {
      "name": "source1",
      "endpoint": {
        "certificateTrust": {
          "type": "TrustAny"
        },
        "endpointUri": "opc.tcp://10.0.0.9:49320",
        "securityPolicy": "NONE",
        "messageSecurityMode": "NONE",
        "identityProvider": {
          "type": "Anonymous"
        },
        "nodeFilterRules": [
          {
            "action": "INCLUDE",
            "definition": {
              "type": "OpcUaRootPath",
              "rootPath": "/Utilities/Tank"
            }
          }
        ]
      },
      "measurementDataStreamPrefix": "propertyGroups",
      "propertyGroups": [
        {
          "name": "Deadband_Abs_5",
          "nodeFilterRuleDefinitions": [
            {
              "type": "OpcUaRootPath",
              "rootPath": "/Utilities/Tank/Temperature/TT-001"
            },
            {
              "type": "OpcUaRootPath",
              "rootPath": "/Utilities/Tank/Temperature/TT-002"
            }
          ],
          "deadband": {
            "type": "ABSOLUTE",
            "value": 5.0,
            "timeoutMilliseconds": 120000
          }
        }
      ]
    }
  ]
}
```

```

    },
    {
      "name": "Polling_10s",
      "nodeFilterRuleDefinitions": [
        {
          "type": "OpcUaRootPath",
          "rootPath": "/Utilities/Tank/Pressure/PT-001"
        }
      ],
      "scanMode": {
        "type": "POLL",
        "rate": 10000
      }
    },
    {
      "name": "Percent_Deadband_Timeout_90s",
      "nodeFilterRuleDefinitions": [
        {
          "type": "OpcUaRootPath",
          "rootPath": "/Utilities/Tank/Flow/FT-*"
        }
      ],
      "deadband": {
        "type": "PERCENT",
        "value": 5.0,
        "eguMin": -100,
        "eguMax": 100,
        "timeoutMilliseconds": 90000
      }
    }
  ]
}

```

### Example : 带属性的 OPC-UA 源配置

针对 `opc-ua-configuration.json` 的以下 JSON 示例定义具有以下属性的 OPC-UA 源配置：

- 信任任何证书。
- 使用 BASIC256 安全策略保护消息。
- 使用 SIGN\_AND\_ENCRYPT 模式保护连接。

- 使用存储在 Secrets Manager 密钥中的身份验证凭证。
- 筛选出数据流 ( 路径以 /WindFarm/2/WindTurbine/ 开头的数据流除外 ) 。
- 在每个数据流路径的开头添加 /Washington 以将此“Wind Farm #2”与另一个区域中的“Wind Farm #2”区分开来。

```
{
  "sources": [
    {
      "name": "Wind Farm #2",
      "endpoint": {
        "certificateTrust": {
          "type": "TrustAny"
        },
        "endpointUri": "opc.tcp://203.0.113.1:49320",
        "securityPolicy": "BASIC256",
        "messageSecurityMode": "SIGN_AND_ENCRYPT",
        "identityProvider": {
          "type": "Username",
          "usernameSecretArn":
            "arn:aws:secretsmanager:region:123456789012:secret:greenrass-windfarm2-auth-1ABCDE"
        },
        "nodeFilterRules": [
          {
            "action": "INCLUDE",
            "definition": {
              "type": "OpcUaRootPath",
              "rootPath": "/WindFarm/2/WindTurbine/"
            }
          }
        ]
      },
      "measurementDataStreamPrefix": "/Washington"
    }
  ]
}
```

Example : 使用证书信任的 OPC-UA 源配置

针对 `opc-ua-configuration.json` 的以下 JSON 示例定义具有以下属性的 OPC-UA 源配置 :

- 信任给定的 X.509 证书。

- 使用 BASIC256 安全策略保护消息。
- 使用 SIGN\_AND\_ENCRYPT 模式保护连接。

```
{
  "sources": [
    {
      "name": "Wind Farm #3",
      "endpoint": {
        "certificateTrust": {
          "type": "X509",
          "certificateBody": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCD6m7oRw0uX0jANBgkqhkiG9w
0BAQUFADCBiDELMAKGA1UEBhMVCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQHEwdTZ
WF0dGx1MQ8wDQYDVQKKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25zb2x1MRIw
EAYDVQQDEw1UZXR0Q21sYWVxHmAdBgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5
jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTEwNDI1MjA0NTIxWjCBiDELMAKGA1UEBh
MVCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQHEwdTZWF0dGx1MQ8wDQYDVQKKEwZBb
WF6b24xFDASBgNVBAsTC0lBTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVx
HmAdBgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wZ8wDQYJKoZIhvcNAQE
BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLyGVI
k60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
ITx0USQv7c7ugFFDzQGBzZswY6786m86gpEibb30hjZnzcVQAaRHhd1QWIMm2nr
AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4nUhVvxYUntneD9+h8Mg9q6q+auN
KyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJI1J00zbhNYS5f6Guo
EDmFJ10ZxBHjJnyp3780D8uTs7fLvjx79LjStbNYiytVbZPQUQ5Yaxu2jXnimvw
3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
          "certificateChain": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCD6m7oRw0uX0jANBgkqhkiG9w
0BAQUFADCBiDELMAKGA1UEBhMVCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQHEwdTZ
WF0dGx1MQ8wDQYDVQKKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25zb2x1MRIw
EAYDVQQDEw1UZXR0Q21sYWVxHmAdBgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5
jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTEwNDI1MjA0NTIxWjCBiDELMAKGA1UEBh
MVCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQHEwdTZWF0dGx1MQ8wDQYDVQKKEwZBb
WF6b24xFDASBgNVBAsTC0lBTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVx
HmAdBgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wZ8wDQYJKoZIhvcNAQE
BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLyGVI
k60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
ITx0USQv7c7ugFFDzQGBzZswY6786m86gpEibb30hjZnzcVQAaRHhd1QWIMm2nr
AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4nUhVvxYUntneD9+h8Mg9q6q+auN
KyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJI1J00zbhNYS5f6Guo
EDmFJ10ZxBHjJnyp3780D8uTs7fLvjx79LjStbNYiytVbZPQUQ5Yaxu2jXnimvw
```

```
3rrszlaEXAMPLE=
    -----END CERTIFICATE-----"
    },
    "endpointUri": "opc.tcp://203.0.113.2:49320",
    "securityPolicy": "BASIC256",
    "messageSecurityMode": "SIGN_AND_ENCRYPT",
    "identityProvider": {
        "type": "Anonymous"
    },
    "nodeFilterRules": []
},
"measurementDataStreamPrefix": ""

}
]
}
```

## 使您的 OPC-UA 源服务器能够信任 Edge 网关 SiteWise

如果在配置 OPC-UA 源时选择“无”以messageSecurityMode外的选项，则必须使源服务器能够信任 AWS IoT SiteWise Edge 网关。E SiteWise dge 网关会生成您的源服务器可能需要的证书。该过程因您的源服务器而异。有关更多信息，请参阅您的服务器文档。

以下过程概述了基本步骤。

### 使 OPC-UA 服务器能够信任 Edge 网关 SiteWise

1. 打开用于配置 OPC-UA 服务器的界面。
2. 输入 OPC-UA 服务器管理员的用户名和密码。
3. 在界面中找到可信客户端，然后选择AWS IoT SiteWise 网关客户端。
4. 选择 信任。

### 导出 OPC-UA 客户端证书

某些 OPC-UA 服务器需要访问 OPC-UA 客户端证书文件才能信任边缘网关。SiteWise 如果这适用于您的 OPC-UA 服务器，则可以使用以下步骤从 Edge 网关导出 OPC-UA 客户端证书。SiteWise 然后，可以在您的 OPC-UA 服务器上导入证书。

## 导出源的 OPC-UA 客户端证书文件

1. 运行以下命令来更改为包含证书文件的目录。# *sitewise-work* ### *aws.iot* #####  
*SiteWiseEdgeCollectorOpcua Greengrass* 工作文件夹并将#名称替换为数据源的名称。

```
#####Greengrass ##### /greengrass/v2/work/aws.iot#  
SiteWiseEdgeCollectorOpcua 在 Linux 和 C: /greengrass/v2/work/aws.iot ##  
SiteWiseEdgeCollector# Windows ### Opcua。
```

```
cd /sitewise-work/source-name/opcua-certificate-store
```

2. 该来源的 SiteWise Edge 网关的 OPC-UA 客户端证书位于文件中。aws-iot-opcua-client.pfx

运行以下命令来将证书导出到名为 aws-iot-opcua-client-certificate.pem 的 .pem 文件。

```
keytool -exportcert -v -alias aws-iot-opcua-client -keystore aws-iot-opcua-client.pfx -storepass amazon -storetype PKCS12 -rfc > aws-iot-opcua-client-certificate.pem
```

3. 将证书文件从 SiteWise Edge 网关传输到 OPC-UA 服务器。aws-iot-opcua-client-certificate.pem

为此，您可以使用常用软件（如 scp 程序）来通过 SSH 协议传输文件。有关更多信息，请参阅 Wikipedia 上的 [Secure copy](#)。

### Note

如果您的 SiteWise Edge 网关在亚马逊弹性计算云 (Amazon EC2) 上运行，并且您是首次连接到该网关，则必须配置连接的先决条件。有关更多信息，请参阅 Amazon EC2 用户指南中的[连接到您的 Linux 实例](#)。

4. 在 OPC-UA 服务器上导入证书文件以信任 SiteWise Edge 网关。aws-iot-opcua-client-certificate.pem 步骤因所使用的源服务器而异。请参阅服务器的文档。



## 使用 OPC-UA 筛选数据摄取范围

您可以使用扫描模式和死区范围来控制通过 OPC-UA 源摄取数据的方式。这些功能使您可以控制要采集的数据类型，以及服务器和 Edg SiteWise e 网关交换这些信息的方式和时间。

### 根据质量收集或筛选出数据

您可以配置数据质量设置以控制从 OPC-UA 源收集哪些数据。数据源在发送时将质量评级作为元数据包括在内。您可以选择以下一个或所有选项：

- Good
- Bad
- Uncertain

### 使用扫描模式控制数据收集频率

您可以配置 OPC-UA 扫描模式来控制从 OPC-UA 源收集数据的方式。您可以选择订阅或轮询模式。

- 订阅模式 — OPC-UA 源收集数据，按照您的扫描速率定义的频率发送到您的 SiteWise Edge 网关。服务器仅在值发生变化时才发送数据，因此这是您的 SiteWise Edge 网关接收数据的最大频率。
- 轮询模式 — 您的 SiteWise Edge 网关以您的扫描速率定义的设定频率轮询 OPC-UA 源。无论值是否发生变化，服务器都会发送数据，因此您的 SiteWise Edge 网关始终以此间隔接收数据。

#### Note

轮询模式选项会覆盖此源的死区设置。

### 使用死区范围筛选 OPC-UA 数据摄取

您可以对 OPC-UA 源属性组应用死区来筛选和丢弃某些数据，而不是将其发送到云端。AWS 死区指定了来自 OPC-UA 源的传入数据值的预期波动窗口。如果值在此窗口内，则您的 OPC-UA 服务器不会将其发送到云端。AWS 您可以使用死区过滤来减少正在处理和发送到 AWS 云端的数据量。要了解如何为 SiteWise Edge 网关设置 OPC-UA 源，请参阅 [the section called “配置数据源”](#)

#### Note

您的服务器会删除死区指定的窗口内的所有数据。您无法恢复此丢弃的数据。

## 死区类型

您可以为 OPC-UA 服务器属性组指定两种类型的死区。它们允许您选择将多少数据发送到 AWS 云端，以及丢弃多少数据。

- **百分比** — 使用测量值中预期波动的百分比来指定窗口。服务器根据此百分比计算出确切的窗口，并将超出窗口范围的数据发送到 AWS 云端。例如，在传感器上指定 2% 的死区值，范围从 -100 华氏度到 +100 华氏度，会告诉服务器在值变化 4 华氏度或更高时将数据发送到 AWS 云端。

### Note

如果您的源服务器未定义工程单位，则可以选择为此窗口指定最小和最大死区值。如果未提供工程单位范围，OPC-UA 服务器将默认为测量数据类型的完整范围。

- **绝对** — 使用精确单位指定窗口。例如，在传感器上将死区值指定为 2，告诉服务器在数据值变化至少 2 个单位时将数据发送到 AWS 云。在正常操作期间经常会出现波动的动态环境中，您可以使用绝对死区。

## 死区超时

您可以选择配置死区超时设置。在此超时之后，即使当前测量值处于预期的死区波动范围内，OPC-UA 服务器也将发送当前测量值。您可以使用超时设置来确保 AWS IoT SiteWise 始终采集稳定的数据流，即使值不超过定义的死区窗口也是如此。

## 使用 OPC-UA 节点筛选条件

在为 SiteWise Edge 网关定义 OPC-UA 数据源时，可以定义节点筛选器。节点过滤器允许您限制 SiteWise Edge 网关发送到云的数据流路径。您可以使用节点筛选器通过仅包含您建模的数据的路径来减少 SiteWise Edge 网关的启动时间和 CPU 使用率 AWS IoT SiteWise。默认情况下，SiteWise 边缘网关会上传除以开头的路径之外的所有 OPC-UA 路径。/Server/您可以使用在节点筛选条件中使用 \* 和 \*\* 通配符来包含多个数据流路径和一个筛选条件。要了解如何为 SiteWise Edge 网关设置 OPC-UA 源，请参阅 [配置数据源](#)

### Note

AWS IoT SiteWise 每次添加或编辑源时都会重新启动 SiteWise Edge 网关。您的 SiteWise Edge 网关在重启时不会提取数据。重启 SiteWise Edge 网关的时间取决于 E SiteWise dge 网

关源上的标签数量。重启时间可以从几秒钟（对于标签较少的 SiteWise Edge 网关）到几分钟（对于具有许多标签的 SiteWise Edge 网关）不等。

下表列出了可用于筛选 OPC-UA 数据源的通配符。

#### OPC-UA 节点筛选条件通配符

通配符	描述
*	匹配数据流路径中的单个级别。
**	匹配数据流路径中的多个级别。

#### Note

如果您为源配置了宽筛选器，然后将源更改为使用更严格的筛选器，则 AWS IoT SiteWise 会停止存储与新筛选器不匹配的数据。

#### Example 使用节点筛选条件的示例场景

考虑以下各项假定数据流：

- /WA/Factory 1/Line 1/PLC1
- /WA/Factory 1/Line 1/PLC2
- /WA/Factory 1/Line 2/Counter1
- /WA/Factory 1/Line 2/PLC1
- /OR/Factory 1/Line 1/PLC1
- /OR/Factory 1/Line 2/Counter2

使用前面的数据流，您可以定义节点筛选条件，限制 OPC-UA 源中要包含的数据。

- 要选择此示例中的所有节点，请使用 / 或 /\*\*/。您可以使用 \*\* 通配符包含多个目录或文件夹。
- 要选择所有 PLC 数据流，请使用 /\*/\*/\*/\*/PLC\* 或 /\*\*/PLC\*。
- 要选择此示例中的所有计数器，请使用 /\*\*/Counter\* 或 /\*/\*/\*/\*/Counter\*。

- 要选择 Line 2 中的所有计数器，请使用 `/**/Line 2/Counter*`。

## 配置数据来源身份验证

如果您的 OPC-UA 服务器需要身份验证凭据才能连接，则可以使用创建密钥并将其部署 AWS Secrets Manager 到您的 SiteWise Edge 网关。AWS Secrets Manager 对设备上的机密进行加密，以确保您的用户名和密码的安全，直到您需要使用它们为止。有关 AWS IoT Greengrass 密钥管理器组件的更多信息，请访问 [AWS IoT Greengrass Version 2 开发者指南中的密钥管理器](#)。

有关管理 Secrets Manager 密钥访问权限的信息，请访问：

- [谁有权访问你的 AWS Secrets Manager 秘密](#)。
- [确定账户内的请求是被允许还是被拒绝](#)。

### 步骤 1：创建源身份验证密钥

您可以使用 AWS Secrets Manager 为数据源创建身份验证密钥。在密钥中，定义 **username** 和 **password** 密钥/值对，其中包含数据来源的身份验证详细信息。

创建密钥（控制台）

1. 导航到 [AWS Secrets Manager 控制台](#)。
2. 选择 存储新密钥。
3. 在密钥类型下，选择其他密钥类型。
4. 在“键/值对”下，执行以下操作：
  1. 在第一个输入框中输入 **username**，在第二个输入框中输入用户名。
  2. 选择添加行。
  3. 在第一个输入框中输入 **password**，在第二个输入框中输入密码。
5. 对于加密密钥，选择 `aws/secretsManager`，然后选择下一步。
6. 在存储新密钥页面，输入密钥名称。
7. （可选）输入可帮助您识别此密钥的描述，然后选择下一步。
8. （可选）在存储新密钥页面，启用自动轮换。有关更多信息，请参阅 [AWS Secrets Manager 用户指南中的轮换密钥](#)。
9. 指定轮换计划。
10. 选择可以轮换此密钥的 Lambda 函数，然后选择下一步。

## 11. 检查您的密钥配置，然后选择存储。

要授权您的 SiteWise Edge 网关与之交互 AWS Secrets Manager，您的 SiteWise Edge 网关的 IAM 角色必须允许该 `secretsmanager:GetSecretValue` 操作。您可以使用 Greengrass 核心设备来搜索 IAM 策略。有关更新 IAM 策略的更多信息，请参阅 AWS Identity and Access Management 用户指南中的 [编辑 IAM 策略](#)。

### Example 策略

将 `secret-arn` 替换为您在上一步中创建的密钥的 Amazon 资源名称 (ARN)。有关如何获得 ARN 密钥的更多信息，请参阅 AWS Secrets Manager 用户指南中的 [从 AWS Secrets Manager 中检索密钥](#)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Effect": "Allow",
      "Resource": [
        "secret-arn"
      ]
    }
  ]
}
```

## 步骤 2：将密钥部署到您的 SiteWise Edge 网关设备

您可以使用 AWS IoT SiteWise 控制台将密钥部署到您的 SiteWise Edge 网关。

### 部署密钥 (控制台)

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 网关。
3. 从网关列表中，选择目标 SiteWise Edge 网关。
4. 在网关配置部分，选择 Greengrass 核心设备链接以打开与 Edge 网关关联 AWS IoT Greengrass 的核心。SiteWise
5. 在导航窗格中，选择部署。

6. 选择目标部署，然后选择修订。
7. 在指定目标页面，选择下一步。
8. 在选择组件页面的公共组件部分，关闭仅显示所选组件。
9. 搜索并选择 `aws.greengrass.SecretManager` 组件，然后选择“下一步”。
10. 从“选定组件”列表中选择 `aws.greengrass.SecretManager` 组件，然后选择配置组件。
11. 在要合并的配置字段，添加以下 JSON 对象。

#### Note

将 `secret-arn` 替换为您在上一步中创建密钥的 ARN。有关如何获得 ARN 密钥的更多信息，请参阅 AWS Secrets Manager 用户指南 中的 [从 AWS Secrets Manager 中检索密钥](#)。

```
{
  "cloudSecrets": [
    {
      "arn": "secret-arn"
    }
  ]
}
```

12. 选择确认。
13. 选择下一步。
14. 在配置高级设置页面上，选择下一步。
15. 查看您的部署配置，再选择部署。

### 步骤 3：添加身份验证配置

您可以使用 AWS IoT SiteWise 控制台向 SiteWise Edge 网关添加身份验证配置。

#### 添加身份验证配置（控制台）

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 从网关列表中，选择目标 SiteWise Edge 网关。
3. 从数据来源列表中选择目标数据来源，再选择编辑。

4. 在添加数据来源页面选择高级配置。
5. 对于身份验证配置，请选择您在上一步中部署的密钥。
6. 选择保存。

## 为源服务器数据选择目标

使用 Amazon S3 将 AWS IoT SiteWise 数据从边缘实时导出或批量导出。您也可以使用流将流发送到另一个组件。AWS IoT Greengrass

- AWS IoT SiteWise 实时-选择此选项可将数据直接发送到 AWS IoT SiteWise 存储器。实时摄取和监控数据，并在边缘处理数据。
- AWS IoT SiteWise 使用 Amazon S3 进行缓冲 — 以镶木地板格式将数据发送到 Amazon S3，然后导入 AWS IoT SiteWise 存储。选择此选项可批量摄取数据，并以经济实惠的方式存储历史数据。您可以配置您的首选 Amazon S3 存储桶位置，以及您希望将数据上传到 Amazon S3 的频率。您还可以选择在数据摄取后如何处理数据。AWS IoT SiteWise 您可以选择同时在 Amazon S3 SiteWise 和 Amazon S3 中提供数据，也可以选择自动将其从 Amazon S3 中删除。
  - Amazon S3 存储桶是一种暂存和缓冲机制，支持 parquet 格式的文件。
  - 如果您选中“将数据导入 AWS IoT SiteWise 存储”复选框，则会先将数据上传到 Amazon S3，然后上传到 AWS IoT SiteWise 存储中。
    - 如果您选中“从 Amazon S3 中删除数据”复选框，则在数据导入 SiteWise 存储后，数据将从 Amazon S3 中删除。
    - 如果您清除“从 Amazon S3 中删除数据”复选框，则数据将存储在 Amazon S3 和 SiteWise 存储中。
  - 如果清除“将数据导入 AWS IoT SiteWise 存储”复选框，则数据将仅存储在 Amazon S3 中。它不会导入到 SiteWise 存储中。

[管理数据存储](#)有关 AWS IoT SiteWise 提供的各种存储选项的详细信息，请访问。要了解有关定价选项的更多信息，请参阅[AWS IoT SiteWise 定价](#)。

- AWS IoT Greengrass 流管理器 — 使用 AWS IoT Greengrass 流管理器将数据发送到以下 AWS Cloud 目的地：传入的频道、Amazon Kinesis Data Streams 中的流 AWS IoT Analytics、中的资产属性或亚马逊简单存储服务 (Amazon S3) Simple Storage Service AWS IoT SiteWise 中的对象。有关更多信息，请参阅《AWS IoT Greengrass Version 2 开发者指南》中的“[在 AWS IoT Greengrass Core 上管理数据流](#)”。

以下示例显示所需的数据流消息结构。所有字段均为必填。

```
{
  "assetId": "string",
  "propertyAlias": "string",
  "propertyId": "string",
  "propertyValues": [
    {
      "quality": "string",
      "timestamp": {
        "offsetInNanos": number,
        "timeInSeconds": number
      },
      "value": {
        "booleanValue": boolean,
        "doubleValue": number,
        "integerValue": number,
        "stringValue": "string"
      }
    }
  ]
}
```

### Note

数据流消息的结构propertyAlias中必须包含 ( assetId和propertyId ) 或。

## assetId

( 可选 ) 要更新的资产的 ID。

## propertyAlias

( 可选 ) 标识属性的别名，例如 OPC-UA 服务器数据流路径。例如：

```
/company/windfarm/3/turbine/7/temperature
```

有关更多信息，请参阅《AWS IoT SiteWise 用户指南》中的将[工业数据流映射到资产属性](#)。

## propertyId

( 可选 ) 此条目的资产属性的 ID。



## propertyValues

( 必填 ) 要上传的属性值列表。您最多可以指定 10 个propertyValues数组元素。

### quality

( 可选 ) 资产属性值的质量。

### timestamp

( 必填 ) 资产属性值的时间戳。

### offsetInNanos

( 可选 ) 纳秒偏移量。timeInSeconds

### timeInSeconds

( 必需 ) Unix 纪元格式的时间戳日期，以秒为单位。分数纳秒数据由 offsetInNanos 提供。

### value

( 必填 ) 资产财产的价值。

#### Note

该value字段中只能存在以下值之一。

### booleanValue

( 可选 ) 布尔型 ( true或false ) 类型的资产属性数据。

### doubleValue

( 可选 ) 双精度型 ( 浮点数 ) 的资产属性数据。

### integerValue

( 可选 ) 整数 ( 整数 ) 类型的资产属性数据。

### stringValue

( 可选 ) 字符串类型的资产属性数据 ( 字符序列 ) 。

## 将合作伙伴数据源添加到 SiteWise Edge 网关

使用 AWS IoT SiteWise Edge 网关时，您可以将合作伙伴数据源连接到您的 SiteWise Edge 网关，并在您的 SiteWise Edge 网关和 AWS 云中接收来自合作伙伴的数据。这些合作伙伴数据来源是 AWS 与合作伙伴合作开发的 AWS IoT Greengrass 组件。添加合作伙伴数据源时，AWS IoT SiteWise 将创建此组件并将其部署到您的 SiteWise Edge 网关上。

要添加合作伙伴数据来源，请执行以下操作：

- [添加合作伙伴数据来源](#)
- 转到合作伙伴的 Web 门户并配置合作伙伴数据源，使其连接到 SiteWise Edge 网关。

### 主题

- [安全性](#)
- [添加合作伙伴数据来源](#)
- [在你的 SiteWise 边缘网关上设置 docker](#)
- [SiteWise 边缘网关合作伙伴数据源](#)

## 安全性

作为 AWS、客户和合作伙伴之间[责任共担模式](#)的一部分，以下内容描述了负责安全的不同方面：

### 客户责任

- 审查合作伙伴。
- 配置授予合作伙伴的网络访问权限。

### AWS 责任

- 将合作伙伴与客户 AWS 云资源隔离开来，合作伙伴需要的云资源除外。在本例中为 AWS IoT SiteWise 摄取。
- 将合作伙伴解决方案限制为合理使用 SiteWise 边缘网关计算机资源 ( CPU、内存、文件系统 )。

### 合作伙伴的责任

- 使用安全的默认值。
- 通过补丁和其他适当更新确保解决方案随着时间的推移而保持安全。

- 对客户数据保密。

## 添加合作伙伴数据来源

要将合作伙伴数据源连接到您的 SiteWise Edge 网关，请将其添加为数据源。当您将其添加为数据源时，AWS IoT SiteWise 会将私有 AWS IoT Greengrass 组件部署到您的 SiteWise Edge 网关。

### 先决条件

要添加合作伙伴数据来源，您必须执行以下操作：

- 在合作伙伴处创建账户。
- 绑定账户。

### 使用合作伙伴数据源创建 SiteWise Edge 网关

如果要创建新的 SiteWise Edge 网关，请完成中的步骤 [创建 SiteWise Edge 网关](#)。创建 SiteWise Edge 网关后，[将合作伙伴数据源添加到现有 SiteWise Edge 网关](#) 按照中的步骤添加合作伙伴数据源。

### 将合作伙伴数据源添加到现有 SiteWise Edge 网关

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 Gateways。
3. 选择要将合作伙伴数据源连接到的 SiteWise Edge 网关。
4. 对于数据来源，选择添加数据来源。
5. 对于来源类型，选择您要将 SiteWise Edge 网关连接到的合作伙伴。

#### Note

目前，EasyEdge 是唯一可用的合作伙伴数据源。首次添加 EasyEdge 数据源时，需要创建一个 [EasyEdge 帐户](#)。

6. 输入源名称。
7. 要授予合作伙伴访问数据来源的权限，请选择授权。
8. 要让 AWS IoT SiteWise 更新您的 AWS IoT SiteWise 发布者组件（如启用了数据处理包）以及处理器组件 AWS IoT SiteWise 处理器组件，请选择更新组件。
9. 选择保存。

## 在你的 SiteWise 边缘网关上设置 docker

要添加合作伙伴数据源，必须在本地设备上安装 [Docker Engine](#) 1.9.1 或更高版本。

### Note

版本 20.10 是经验证可与 SiteWise Edge 网关软件配合使用的最新版本。

## 验证是否已安装 Docker

要验证 Docker 是否已安装，请在连接到 Ed SiteWise ge 网关的终端上运行以下命令：

```
docker info
```

如果命令返回 `docker is not recognized` 结果，或者安装了旧版本的 Docker，请在[安装 Docker Engine](#) 后再继续。

## 设置 Docker

运行 Docker 容器组件的系统用户必须具有根或管理员权限，或者您必须将 Docker 配置为以非根用户或非管理员用户身份运行。

在 Linux 设备上，您必须向 `docker` 群组中添加一个 `ggc_user` 用户才能调用 Docker 命令，且无需 `sudo`。

要添加 `ggc_user` 或您用于运行 Docker 容器组件的非根用户至 `docker` 组中，请运行以下命令：

```
sudo usermod -aG docker ggc_user
```

有关更多信息，请参阅 [Docker 引擎的 Linux 安装后步骤](#)。

## SiteWise 边缘网关合作伙伴数据源

使用以下信息配置合作伙伴数据来源。

### EasyEdge

门户：

<https://studio.easyedge.io/>

EasyEdge 文档：

## [EasyEdge 对于 AWS](#)

[EasyEdge 要求](#)-有关 EasyEdge 要求的信息，包括配置防火墙所需的端点和端口。注意：您需要一个 EasyEdge 帐户才能访问此文档。

## 使用包

AWS IoT SiteWise 边缘网关使用不同的数据包来确定如何收集和处理您的数据。

目前，有以下包可供选择：

- 数据收集包 — 使用此包收集您的工业数据并将其发送到 AWS 云端目的地。默认情况下，此包会自动为您的 SiteWise Edge 网关启用。
- 数据处理包 — 使用此包启用 SiteWise 边缘网关与边缘配置的资产模型和资产的通信。您可以使用边缘配置来控制要在现场计算和处理的资产数据。然后，您可以将数据发送到 AWS IoT SiteWise 或其他 AWS 服务。有关数据处理包的更多信息，请参阅 [the section called “启用边缘数据处理”](#)。

## 升级包

### Important

将数据处理包版本从 2.0.x 前的（包括）版本升级到 2.1.x 版将导致本地存储的测量数据丢失。

SiteWise 边缘网关使用不同的数据包来确定如何收集和处理您的数据。您可以使用 AWS IoT SiteWise 控制台升级软件包。

升级包（控制台）

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 网关。
3. 在网关列表中，选择包含要升级的包的 SiteWise Edge 网关。
4. 在网关配置部分，选择可用的软件更新。
5. 在编辑软件版本页面的网关组件更新部分，执行以下操作：
  - 要更新 OPC-UA 收集器，请选择一个版本，然后选择“部署”。

- 要更新发布服务器，请选择一个版本，然后选择部署。
  - 要更新数据处理包，请选择一个版本，然后选择部署。
6. 部署完新版本后，选择“完成”。

如果您在升级包时遇到问题，请参阅 [无法将包部署到 SiteWise Edge 网关](#)。

## 管理 SiteWise Edge 网关

您可以使用 AWS IoT SiteWise 控制台和 API 操作来管理 AWS IoT SiteWise Edge 网关。您也可以使用 [AWS OpsHub or Window AWS IoT SiteWise s](#) 应用程序从本地设备管理 SiteWise Edge 网关的某些方面。

我们强烈建议您使用 [AWS OpsHub or AWS IoT SiteWise](#) 应用程序来监控本地设备上的磁盘使用情况。您还可以监控 `Gateway.AvailableDiskSpace` 和 `Gateway.UsedPercentageDiskSpace` Amazon CloudWatch 指标，并创建警报，以便在磁盘空间不足时收到通知。有关 Amazon CloudWatch 警报的更多信息，请参阅 [基于静态阈值创建 CloudWatch 警报](#)。

确保您的设备有足够的空间容纳后续数据。当您的本地设备空间即将用完时，该服务会自动删除时间戳最早的少量数据，以便为即将到来的数据腾出空间。

如要检查服务是否删除了您的数据，请执行以下操作：

1. 登录以 AWS OpsHub 获取 AWS IoT SiteWise 应用程序。
2. 选择设置。
3. 对于日志，指定时间范围，再选择下载。
4. 解压缩该日志文件。
5. 如果日志文件包含以下消息，则服务会删除您的数据：为了防止 SiteWise Edge 网关存储空间不足，已删除了 # 字节的数据。

## 使用 AWS IoT SiteWise 控制台管理您的 SiteWise Edge 网关

您可以使用 AWS IoT SiteWise 控制台配置、更新和监控您 AWS 账户中的所有 SiteWise Edge 网关。

您可以通过导航到 [AWS IoT SiteWise 控制台](#) 中的 SiteWise Edge 网关页面来查看您的 Edge 网关。要访问特定网关的 Edge 网关详细信息页面，请选择 Edge 网关的名称。

在 Edge 网关详细信息页面的概述选项卡中，您可以执行以下操作：

- 在“数据源”部分，更新数据源配置并配置其他数据源
- 选择 Open metrics 可在 CloudWatch 指标控制台中查看每个数据源摄取的数据点数 CloudWatch
- 在 Edge 功能部分，单击编辑将数据包添加到您的 SiteWise Edge 网关
- 在网关配置部分，查看 SiteWise Edge 网关的连接状态
- 在发布者配置部分，查看 SiteWise Edge 网关同步状态和 AWS IoT SiteWise 发布者组件的配置

在 Edge 网关详细信息页面的更新选项卡中，您可以查看部署到 Edge 网关的当前组件和包版本。当新版本可用时，您也可以在此处部署新版本。

## 使用 AWS OpsHub 管理 SiteWise 边缘网关 AWS IoT SiteWise

您可以使用 f AWS OpsHub or AWS IoT SiteWise 应用程序来管理和监控您的 SiteWise Edge 网关。此应用程序提供以下监控和管理选项：

- 在概述下，您可以执行以下操作：
  - 查看 SiteWise Edge 网关详细信息，帮助您深入了解 SiteWise Edge 网关设备数据、发现问题并提高 SiteWise Edge 网关的性能。
  - 查看 SiteWise 监控来自本地服务器和边缘设备的数据的 Monitor 门户。有关更多信息，请参阅 [AWS IoT SiteWise Monitor 应用程序指南](#) 中的 [什么是 AWS IoT SiteWise Monitor](#)。
- 在 Health 下，有一个控制面板，用于显示来自 SiteWise 边缘网关的数据。领域专家（例如流程工程师）可以使用仪表板来查看 SiteWise 边缘网关行为的概述。
- 在“资产”下，查看部署到本地设备的资产以及上次为资产属性收集或计算的价值。
- 在 VPC 设置下，执行下列操作：
  - 如果安装了数据处理包，请查看 SiteWise Edge 网关配置信息并将资源与 AWS 云同步。
  - 下载可用于使用其他工具访问 SiteWise Edge 网关的身份验证文件。
  - 下载可用于对 SiteWise Edge 网关进行故障排除的日志。
  - 查看部署到 SiteWise Edge 网关的 AWS IoT SiteWise 组件。

### Important

以下内容需要用 AWS OpsHub 于 AWS IoT SiteWise：

- 您的本地设备和 f AWS OpsHub o AWS IoT SiteWise r 应用程序必须连接到同一个网络。
- 必须启用数据处理包。

## 使用管理 SiteWise Edge 网关 AWS OpsHub

1. 下载并安装适用[AWS OpsHubAWS IoT SiteWise 于 Windows 的](#)应用程序。
2. 打开 应用程序。
3. 如果您没有为网关设置本地证书，请按照以下步骤[使用本地操作系统凭据访问您的 SiteWise Edge 网关](#)进行设置。
4. 您可以使用 Linux 或轻型目录访问协议 (LDAP) 凭据登录 SiteWise Edge 网关。要登录您的 SiteWise Edge 网关，请执行以下任一操作：

### Linux

1. 在“主机名或 IP 地址”中，输入本地设备的主机名或 IP 地址。
2. 在身份验证中，请选择 Linux。
3. 在用户名中，输入您的 Linux 操作系统的用户名。
4. 在密码中，输入您的 Linux 操作系统的密码。
5. 选择 Sign in ( 登录 )。

### LDAP

1. 在“主机名或 IP 地址”中，输入本地设备的主机名或 IP 地址。
2. 在身份验证中，请选择 LDAP。
3. 对于用户名，请输入您的 LDAP 用户名。
4. 对于 密码，请输入您的 LDAP 密码。
5. 选择 Sign in ( 登录 )。

## 使用本地操作系统凭据访问您的 SiteWise Edge 网关

除了轻量级目录访问协议 (LDAP) 之外，您还可以使用 Linux 或 Windows 凭据来访问您的 SiteWise Edge 网关。

### Important

要使用 Linux 凭据访问您的 SiteWise Edge 网关，必须激活 SiteWise Edge 网关的数据处理包。



## 使用 Linux 操作系统凭据访问您的 SiteWise Edge 网关

下列步骤假定您使用的是装有 Ubuntu 的设备。如果您使用的是其他 Linux 发行版，请查阅您设备的相关文档。

### 创建 Linux 用户群体

1. 要创建管理员组，请运行以下命令。

```
sudo groupadd --system SWE_ADMIN_GROUP
```

SWE\_ADMIN\_GROUP 群组中的用户可以允许 SiteWise Edge 网关的管理员访问权限。

2. 要创建用户组，请运行以下命令。

```
sudo groupadd --system SWE_USER_GROUP
```

SWE\_USER\_GROUP 群组中的用户可以允许 SiteWise Edge 网关的只读访问权限。

3. 要向管理员组添加用户，请运行以下命令。将###和##替换为要添加的用户名和密码。

```
sudo useradd -p $(openssl passwd -1 password) user-name
```

4. 要将用户添加到 SWE\_ADMIN\_GROUP 或 SWE\_USER\_GROUP 中，请将###替换为您在上一步中添加的用户名。

```
sudo usermod -a -G SWE_ADMIN_GROUP user-name
```

现在，您可以使用用户名和密码登录 for AWS IoT SiteWise 应用程序上的 SiteWise Edge 网关。AWS OpsHub

## 使用 Windows 凭据访问您的 SiteWise Edge 网关

下列步骤假定您使用的是装有 Windows 的设备。

### Important

安全是双方共同承担 AWS 的责任。创建包含至少 12 个字符以及大写、小写、数字和符号组合的强密码策略。此外，将 Windows 防火墙规则设置为允许端口 443 的传入流量，并阻止所有其他端口的传入流量。

## 创建 Windows Server 用户群体

1. 以管理员 PowerShell 身份运行。
  - a. 在要安装 SiteWise 边缘网关的 Windows 服务器上，以管理员身份登录。
  - b. 在 Windows 搜索栏 PowerShell 中输入。
  - c. 在搜索结果中，右键单击 Windows PowerShell 应用程序。选择以管理员身份运行。
2. 要创建管理员组，请运行以下命令。

```
net localgroup SWE_ADMIN_GROUP /add
```

您必须是 SWE\_ADMIN\_GROUP 群组中的用户才能允许 SiteWise Edge 网关的管理员访问权限。

3. 要创建用户组，请运行以下命令。

```
net localgroup SWE_USER_GROUP /add
```

您必须是该 SWE\_USER\_GROUP 组中的用户才能允许 SiteWise Edge 网关的只读访问权限。

4. 要添加用户，请运行以下命令。将###和##替换为要创建的用户名和密码。

```
net user user-name password /add
```

5. 要向管理员组添加用户，请运行以下命令。将###替换为要添加的用户名。

```
net localgroup SWE_ADMIN_GROUP user-name /add
```

现在，您可以使用用户名和密码登录 for AWS IoT SiteWise 应用程序上的 SiteWise Edge 网关。AWS OpsHub

## 管理 SiteWise Edge 网关证书

您可以在边缘 SiteWise 网关设备上 SiteWise 使用监控器和第三方应用程序，例如 Grafana。这些应用程序需要与服务建立 TLS 连接。SiteWise 边缘网关目前使用自签名证书。如果您使用浏览器打开应用程序（例如 Monito SiteWise r 门户），则可能会收到证书不可信的警告。

以下内容显示如何从 for AWS IoT SiteWise 应用程序下载可信证书。AWS OpsHub

1. 登录到应用程序。

2. 选择设置。
3. 对于身份验证，请选择下载证书。

以下假设您使用的是谷歌浏览器或 FireFox。如果您使用其他浏览器，请查阅您的浏览器相关文档。要将上一步中下载的证书添加到浏览器，请执行下列操作之一：

- 如果您使用 Google Chrome，请按照 Google Chrome 企业帮助文章中的[设置证书](#)进行操作。
- 如果您使用 Firefox，请按照 Oracle 文档中的[将证书加载至 Mozilla 或 Firefox 浏览器](#)进行操作。

## 更改 SiteWise Edge 网关组件包的版本

您可以使用 AWS IoT SiteWise 控制台更改 SiteWise Edge 网关上组件包的版本。

### 更改 SiteWise Edge 网关组件包的版本

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在左侧导航窗格中，选择 网关。
3. 选择要更改其包版本的 SiteWise Edge 网关。
4. 在网关配置下，选择查看软件版本。
5. 在编辑软件版本页面上，对于要更新其版本的包，选择要部署的版本，然后选择部署。
6. 选择完成。

## 在西门子工业 SiteWise 边缘上运行边缘

您可以通过在设备上运行 Edge 网关将数据从西门子工业 SiteWise 边缘设备摄取到您的设备上。为此，您需要创建一个 SiteWise 边缘网关资源，其部署目标为西门子工业边缘设备（全新），下载配置文件，然后通过西门子工业边缘管理 (IEM) 门户将其上传到您的西门子应用程序。有关在 Siemens In AWS IoT SiteWise Industrial Edge 上运行 Edge 的更多信息，包括如何设置所需的西门子资源，请参阅[什么是工业边缘？](#)在西门子文档中。

### Note

西门子不是 AWS IoT SiteWise Edge 的供应商或供应商。西门子工业边缘市场 Marketplace 是一个独立的市场。

## 主题

- [先决条件](#)
- [安全性](#)
- [创建配置文件](#)
- [故障排除](#)
- [联系我们](#)

## 先决条件

要在西门子工业 AWS IoT SiteWise 边缘上运行 Edge，您需要满足以下条件：

- [西门子数字交易平台账户](#)
- 西门子工业边缘中心 (iehub) 账户
- 西门子工业边缘管理 (IEM) 实例
- 西门子工业边缘设备 (IED) 或西门子工业边缘虚拟设备 (ieVD)
- 访问西门子工业边缘设备部署目标。要获得访问权限，请转到[AWS IoT SiteWise 控制台](#)并选择请求访问权限。

## 安全性

作为我们的客户和合作伙伴之间[AWS 责任共担模式](#)的一部分，以下内容描述了谁对安全的不同方面负责：

### 客户责任

- 审查合作伙伴。
- 配置授予合作伙伴的网络访问权限。
- 对运行 AWS IoT SiteWise Edge 的设备进行物理保护。

### AWS 责任

- 将合作伙伴与客户的 AWS 云资源隔离开来。

### 合作伙伴的责任

- 使用安全的默认值。
- 通过补丁和其他适当更新确保解决方案随着时间的推移而保持安全。
- 对客户数据保密。

- 审查合作伙伴市场上可用的其他应用程序。

在此功能的预览阶段，合作伙伴和通过合作伙伴市场安装的其他应用程序可以访问 AWS IoT SiteWise 缓存在合作伙伴设备上的客户数据。

## 创建配置文件

拥有合适的西门子帐户和 IEM 实例后，就可以创建部署类型为西门子工业 SiteWise 边缘设备的边缘网关。

### 创建配置文件

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 Edge 网关。
3. 选择 创建网关。
4. 对于部署类型，请选择西门子工业边缘设备-全新。
5. 输入 E SiteWise dge 网关的名称或使用生成的名称 AWS IoT SiteWise。
6. ( 可选 ) 在高级配置下，执行以下操作：
  - 输入 AWS IoT Core 事物的名称或使用生成的名称 AWS IoT SiteWise。
7. 选择 创建网关。
8. 在生成 SiteWise Edge 网关配置文件对话框中，选择生成并下载。AWS IoT SiteWise 自动生成用于配置 AWS IoT SiteWise Edge 应用程序的配置文件。

#### Important

确保将配置文件保存在安全的位置。稍后您将使用该文件。

现在，您已经创建了 SiteWise Edge 网关，请执行以下操作以完成 SiteWise Edge 网关的设置：

1. [添加数据源](#)
2. [配置发布者组件](#)

配置文件并配置好 SiteWise 边缘网关后，请从西门子工业边缘市场 ( Siemens In AWS IoT SiteWise dustrial Edge Marketplace ) 下载边缘应用程序，然后使用西门子工业边缘管理 (IEM) 门户进行安装。

然后，通过西门子工业边缘管理 (IEM) 门户访问您的西门子工业边缘设备，并将配置文件上传到要安装 SiteWise 边缘网关的设备上。

## 故障排除

要对西门子工业 SiteWise 边缘设备上的边缘网关进行故障排除，您可以通过西门子工业边缘管理 (IEM) 或西门子工业边缘设备 (IED) 门户访问应用程序的日志。有关更多信息，请参阅 Siemens 文档中的[下载日志](#)。

我看见了“SESSION\_TAKEN\_OVER”或“com.aws.greengrass.mqttclient”。

MqttClient: 无法通过 Spooler 发布消息，将重试。' 在日志中

如果您在的日志com.aws.greengrass.mqttclient.MqttClient: Failed to publish the message via Spooler and will retry.中看到包含的警告SESSION\_TAKEN\_OVER或包含错误的信息/greengrass/v2/logs/greengrass.log，则可能是您正在尝试对多台设备上的多个 SiteWise Edge 网关使用相同的配置文件。每个 SiteWise Edge 网关都需要一个唯一的配置文件才能连接到您的 AWS 账户。

我看到了 'com.aws.greengrass.deployment。 IotJobsHelper: 未找到部署作业。' 或“部署结果已报告。” 在日志中

如果您在的日志Deployment result already reported.中看到com.aws.greengrass.deployment.IotJobsHelper: No deployment job found.或/greengrass/v2/logs/greengrass.log，则可能正在尝试重复使用相同的配置文件。

有多种解决方案：

- 如果要重复使用配置文件，请执行以下操作：
  1. 导航到 [AWS IoT SiteWise 控制台](#)。
  2. 在导航窗格中，选择 网关。
  3. 选择要重复使用的 SiteWise Edge 网关。
  4. 选择“更新”选项卡。
  5. 选择其他发布者版本，然后选择“部署”。
- 按照中的[创建配置文件](#)步骤创建新的配置文件。

我在日志中看到“缺少 AWS\_REGION 的配置文件”。

如果您在西门子日志 Config file missing AWS\_REGION 中看到，则配置文件的 JSON 已损坏。您需要创建一个新的配置文件。按照中的 [创建配置文件](#) 步骤创建新的配置文件。

## 联系我们

- 如果您想请求访问该应用程序，请转到 [AWS IoT SiteWise 控制台](#) 并选择请求访问权限。
- 如果您需要帮助排除应用程序故障，请转到 [AWS IoT SiteWise 控制台](#)，导航到 SiteWise Edge 网关的详细信息页面，然后选择获取支持。

## 在 SiteWise Edge 网关上筛选资产

您可以使用边缘筛选来更有效地管理资产，方法是仅将一部分资产发送到特定的 SiteWise 边缘网关以用于数据处理。如果您的资产按树结构或父子结构排列，则可以设置附加到 SiteWise Edge 网关的 IAM 角色的 IAM 策略，该策略仅允许将树的根（父级）及其子项发送到特定的 SiteWise Edge 网关。

### Note

如果要现将资源排列为树形结构，则在创建结构后，请进入添加到结构中的每项现有资源并选择“编辑”，然后选择“保存”以确保 AWS IoT SiteWise 识别新结构。

## 设置边缘筛选

在 Edge 网关上设置 SiteWise 边缘筛选，方法是将以下 IAM 策略添加到 SiteWise Edge 网关的 IAM 角色，将 `< root-asset-id >` 替换为要发送到 SiteWise Edge 网关的根资产的 ID。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "iotsitewise:DescribeAsset",
        "iotsitewise:ListAssociatedAssets"
      ],
      "Resource": "arn:aws:iotsitewise:*:*:asset/*",
      "Condition": {
```

```
        "StringNotLike": {
            "iotsitewise:assetHierarchyPath": "<root-asset-id>*"
        }
    }
}
]
```

如果您的 SiteWise Edge 网关上当前有您想要移除的资产，请登录您的 SiteWise Edge 网关并运行以下命令，AWS IoT SiteWise 通过删除缓存来强制 SiteWise Edge 网关与之同步。

```
sudo rm /greengrass/v2/work/aws.iot.SiteWiseEdgeProcessor/sync-app/
sync_resource_bundles/edge.json
```

## 在边缘使用 AWS IoT SiteWise API

您可以使用可用的 AWS IoT SiteWise API 子集以及特定于边缘的 API 来与资源模型及其边缘资产进行交互。必须将资产模型配置为在边缘运行。有关更多信息，请参见 [在边缘处理数据](#)。

使用这些 API 来收集有关您的资产模型和资产的数据，监控已部署的门户和仪表板指标，并在边缘收集资产数据。这在您的网络中提供了一个中央主机，无需 Web API 调用即可与 AWS IoT SiteWise 进行交互。

### 主题

- [适用于 AWS IoT SiteWise 边缘设备的所有可用 API](#)
- [仅适用于边缘设备的 API，适用于 AWS IoT SiteWise 边缘设备](#)
- [教程：在 SiteWise Edge 网关上获取资产模型列表](#)

## 适用于 AWS IoT SiteWise 边缘设备的所有可用 API

使用边缘设备时，您可以使用各种 API 在设备上与 AWS IoT SiteWise 进行本地交互并完成任务。

### 可用的 AWS IoT SiteWise API

以下 AWS IoT SiteWise API 可在边缘设备上使用：

- [ListAssetModels](#)
- [DescribeAssetModel](#)



- [ListAssets](#)
- [DescribeAsset](#)
- [DescribeAssetProperty](#)
- [ListAssociatedAssets](#)
- [GetAssetPropertyAggregates](#)
- [GetAssetPropertyValue](#)
- [GetAssetPropertyValueHistory](#)
- [ListDashboards](#)
- [ListPortals](#)
- [ListProjectAssets](#)
- [ListProjects](#)
- [DescribeDashboard](#)
- [DescribePortal](#)
- [DescribeProject](#)

## 可用的仅限边缘的 API

以下 API 可在边缘设备进行本地使用：

- [身份验证](#) — 使用此 API 获取用于进行 API 调用的 Sigv4 临时凭证。

## 仅适用于边缘设备的 API，适用于 AWS IoT SiteWise 边缘设备

除了边缘上可用的 AWS IoT SiteWise API，还有边缘特定的 API。这些边缘特定的 API 如下所述。

### 身份验证

从 SiteWise Edge 网关获取证书。您需要添加本地用户或使用 LDAP 或 Linux 用户群体连接到您的系统。有关添加用户的更多信息，请参阅 [LDAP](#) 或 [Linux 用户群体](#)。

### 请求语法

```
POST /authenticate HTTP/1.1
Content-type: application/json
{
```

```
"username": "string",  
"password": "string",  
"authMechanism": "string"  
}
```

## URI 请求参数

该请求不使用任何 URI 参数。

## Request body

请求接受采用 JSON 格式的以下数据。

### username

用于验证请求调用的用户名。

类型：字符串

必需：是

### password

请求凭证的用户密码。

类型：字符串

必需：是

### authMechanism

在主机中验证此用户的身份验证方法。

类型：字符串

有效值：ldap、linux、winnt

必需：是

## 响应语法

```
HTTP/1.1 200  
Content-type: application/json  
{
```

```
"accessKeyId": "string",
"secretAccessKey": "string",
"sessionToken": "string",
"region": "edge"
}
```

## 响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

以下数据以 JSON 格式返回。

### accessKeyId

用于标识临时安全凭证的访问密钥 ID。

长度限制：最小长度为 16。长度上限为 128。

模式：`[\w]*`

### secretAccessKey

用于签署请求的秘密访问密钥。

类型：字符串

### sessionToken

用户必须传递给服务 API 才能使用临时凭证的令牌。

类型：字符串

### region

您针对 API 调用的目标区域。

类型：常量 — edge

## 错误

### IllegalArgumentException

由于提供的正文文件格式错误，请求被拒绝。错误消息描述了具体的错误。

HTTP 状态代码：400

## AccessDeniedException

根据当前的身份提供商，用户无有效凭证。错误消息描述了身份验证机制。

HTTP 状态代码：403

## TooManyRequestsException

该请求已达到身份验证尝试次数上限。错误消息包含等待进行新身份验证尝试的时间。

HTTP 状态代码：429

## 教程：在 SiteWise Edge 网关上获取资产模型列表

您可以使用可用的 AWS IoT SiteWise API 子集以及特定于边缘的 API 来与资源模型及其边缘资产进行交互。本教程将引导您获取 AWS IoT SiteWise Edge 网关的临时凭证以及获取 SiteWise Edge 网关上的资产模型列表。

### 先决条件

在本教程的步骤中，您可以使用各种工具。要使用这些工具，请确保您安装了相应的必备工具。

要完成本教程，您需要：

- 已部署且正在运行 [SiteWise 边缘网关要求](#)
- 通过 SiteWise 端口 443 访问同一网络中的 Edge 网关。
- 安装 [OpenSSL](#)
- (f AWS OpsHub o AWS IoT SiteWise r) f [AWS OpsHubor AWS IoT SiteWise 应用程序](#)
- (curl) [curl](#) 已安装
- (Python) [urllib3](#) 已安装
- (Python) [Python3](#) 已安装
- (Python) [Boto3](#) 已安装
- (Python) [BotoCore](#) 已安装

### 步骤 1：获取 SiteWise Edge 网关服务签名证书

要与 SiteWise Edge 网关上可用的 API 建立 TLS 连接，您需要一个可信证书。您可以使用 OpenSSL 或 AWS OpsHub 为生成此证书。AWS IoT SiteWise

## OpenSSL

### Note

您需要安装 [OpenSSL](#) 才能运行此命令。

打开终端并运行以下命令从 SiteWise Edge 网关获取签名证书。<sitewise\_gateway\_ip>替换为 SiteWise 边缘网关的 IP。

```
openssl s_client -connect <sitewise_gateway_ip>:443 </dev/null 2>/dev/null | openssl x509 -outform PEM > GatewayCert.pem
```

## AWS OpsHub for AWS IoT SiteWise

您可以使用 AWS OpsHub 用于 AWS IoT SiteWise。有关更多信息，请参见 [管理 SiteWise Edge 网关](#)。

本教程使用下载的 SiteWise Edge 网关证书的绝对路径。运行以下命令导出证书的完整路径，将 <absolute\_path\_to\_certificate> 替换为证书的路径：

```
export PATH_TO_CERTIFICATE='<absolute_path_to_certificate>'
```

## 步骤 2：获取您的 SiteWise Edge 网关主机名

### Note

您需要安装 [OpenSSL](#) 才能运行此命令。

要完成本教程，您需要提供 SiteWise Edge 网关的主机名。要获取 SiteWise Edge 网关的主机名，请运行以下命令，<sitewise\_gateway\_ip>替换为 SiteWise Edge 网关的 IP：

```
openssl s_client -connect <sitewise_gateway_ip>:443 </dev/null 2>/dev/null | grep -Po 'CN = \K.*' | head -1
```

运行以下命令导出主机名供以后使用，<your\_edge\_gateway\_hostname>替换为 SiteWise Edge 网关的主机名：

```
export GATEWAY_HOSTNAME='<your_edge_gateway_hostname>'
```

### 步骤 3：获取 SiteWise Edge 网关的临时证书

现在，您已获得 SiteWise Edge 网关的签名证书和主机名，您需要获取临时证书，以便可以在网关上运行 API。您可以通过 Edge 网关获取这些凭证 AWS OpsHub，AWS IoT SiteWise 也可以使用 API 直接从 SiteWise Edge 网关获取这些证书。

#### Important

凭证每 4 小时过期一次，因此您应该在使用 SiteWise Edge 网关上的 API 之前获取证书。请勿将凭证缓存超过 4 小时。

使用 fo AWS OpsHub r 获取临时证书 AWS IoT SiteWise

#### Note

您需要为 [AWS IoT SiteWise 应用程序安装 AWS OpsHub](#)。

要为 AWS IoT SiteWise 应用程序使用 AWS OpsHub 获取您的临时凭证，请执行以下操作：

1. 登录应用程序。
2. 选择设置。
3. 对于身份验证，选择复制凭证。
4. 展开适合您环境的选项，再选择复制。
5. 保存凭证以供稍后使用。

使用 SiteWise Edge 网关 API 获取临时证书

要使用 SiteWise Edge 网关 API 获取临时证书，可以使用 Python 脚本或 curl，首先需要 SiteWise Edge 网关的用户名和密码。边 SiteWise 缘网关使用 Sigv4 身份验证和授权。有关添加用户的更多信息，请参阅 [LDAP](#) 或 [Linux 用户群体](#)。在以下步骤中，将使用这些凭证在您的 SiteWise Edge 网关上获取使用 AWS IoT SiteWise API 所需的本地证书。

## Python

### Note

您需要安装 [urllib3](#) 和 [Python3](#)。

### 获取 Python 使用凭证

1. 创建一个名为 `get_credentials.py` 的文件，再将以下代码复制到其中。

```
'''
The following demonstrates how to get the credentials from the SiteWise Edge
gateway. You will need to add local users or connect your system to LDAP/AD
https://docs.aws.amazon.com/iot-sitewise/latest/userguide/manage-gateways-
ggv2.html#create-user-pool

Example usage:
    python3 get_credentials.py -e https://<gateway_hostname> -c
    <path_to_certificate> -u '<gateway_username>' -p '<gateway_password>' -m
    '<method>'
'''
import urllib3
import json
import urllib.parse
import sys
import os
import getopt

'''
This function retrieves the AWS IoT SiteWise Edge gateway credentials.
'''
def get_credentials(endpoint, certificatePath, user, password, method):
    http = urllib3.PoolManager(cert_reqs='CERT_REQUIRED', ca_certs=
    certificatePath)
    encoded_body = json.dumps({
        "username": user,
        "password": password,
        "authMechanism": method,
    })

    url = urllib.parse.urljoin(endpoint, "/authenticate")
```

```
response = http.request('POST', url,
                        headers={'Content-Type': 'application/json'},
                        body=encoded_body)

if response.status != 200:
    raise Exception(f'Failed to authenticate! Response status
{response.status}')

auth_data = json.loads(response.data.decode('utf-8'))

accessKeyId = auth_data["accessKeyId"]
secretAccessKey = auth_data["secretAccessKey"]
sessionToken = auth_data["sessionToken"]
region = "edge"

return accessKeyId, secretAccessKey, sessionToken, region

def print_help():
    print('Usage:')
    print(f'{os.path.basename(__file__)} -e <endpoint> -c <path/to/certificate>
-u <user> -p <password> -m <method> -a <alias>')
    print('')
    print('-e, --endpoint    edge gateway endpoint. Usually the Edge gateway
hostname.')
    print('-c, --cert_path path to downloaded gateway certificate')
    print('-u, --user        Edge user')
    print('-p, --password   Edge password')
    print('-m, --method    (Optional) Authentication method (linux, winnt,
ldap), default is linux')
    sys.exit()

def parse_args(argv):
    endpoint = ""
    certificatePath = None
    user = None
    password = None
    method = "linux"

    try:
        opts, args = getopt.getopt(argv, "he:c:u:p:m:",
["endpoint=", "cert_path=", "user=", "password=", "method="])
    except getopt.GetoptError:
        print_help()
```



```
for opt, arg in opts:
    if opt == '-h':
        print_help()
    elif opt in ("-e", "--endpoint"):
        endpoint = arg
    elif opt in ("-u", "--user"):
        user = arg
    elif opt in ("-p", "--password"):
        password = arg
    elif opt in ("-m", "--method"):
        method = arg.lower()
    elif opt in ("-c", "--cert_path"):
        certificatePath = arg

if method not in ['ldap', 'linux', 'winnt']:
    print("not valid method parameter, required are ldap, linux, winnt")
    print_help()

if (user == None or password == None):
    print("To authenticate against edge user, password have to be passed
together, and the region has to be set to 'edge'")
    print_help()

if(endpoint == ""):
    print("You must provide a valid and reachable gateway hostname")
    print_help()

return endpoint,certificatePath, user, password, method

def main(argv):
    # get the command line args
    endpoint, certificatePath, user, password, method = parse_args(argv)

    accessKeyId, secretAccessKey, sessionToken, region=get_credentials(endpoint,
certificatePath, user, password, method)

    print("Copy and paste the following credentials into the shell, they are
valid for 4 hours:")
    print(f"export AWS_ACCESS_KEY_ID={accessKeyId}")
    print(f"export AWS_SECRET_ACCESS_KEY={secretAccessKey}")
    print(f"export AWS_SESSION_TOKEN={sessionToken}")
    print(f"export AWS_REGION={region}")
```

```
print()
```

```
if __name__ == "__main__":  
    main(sys.argv[1:])
```

2. 使用您创建的凭证替换 `<gateway_username>` 和 `<gateway_password>` 并在终端运行 `get_credentials.py`。

```
python3 get_credentials.py -e https://$GATEWAY_HOSTNAME -c $PATH_TO_CERTIFICATE  
-u '<gateway_username>' -p '<gateway_password>' -m 'linux'
```

curl

#### Note

您需要安装 [curl](#)。

获取 curl 使用凭证

1. 使用您创建的凭证替换 `<gateway_username>` 和 `<gateway_password>` 并在终端运行命令。

```
curl --cacert $PATH_TO_CERTIFICATE --location \  
-X POST https://$GATEWAY_HOSTNAME:443/authenticate \  
--header 'Content-Type: application/json' \  
--data-raw '{  
    "username": "<gateway_username>",  
    "password": "<gateway_password>",  
    "authMechanism": "linux"  
}'
```

该响应应当与以下内容相似：

```
{  
    "username": "sweuser",  
    "accessKeyId": "<accessKeyId>",  
    "secretAccessKey": "<secretAccessKey>",
```

```
"sessionToken": "<sessionToken>",
"sessionExpiryTime": "2022-11-17T04:51:40.927095Z",
"authMechanism": "linux",
"role": "edge-user"
}
```

2. 从您的终端运行以下命令。

```
export AWS_ACCESS_KEY_ID=<accessKeyId>
export AWS_SECRET_ACCESS_KEY=<secretAccessKey>
export AWS_SESSION_TOKEN=<sessionToken>
export AWS_REGION=edge
```

## 第 4 步：获取 SiteWise Edge 网关上的资产模型列表

现在，您已拥有签名证书、SiteWise Edge 网关主机名和 SiteWise Edge 网关的临时证书，您可以使用 ListAssetModels API 获取 SiteWise Edge 网关上的资产模型列表。

### Python

#### Note

您需要安装 [Python3](#)、[Boto 3](#)。 [BotoCore](#)

### 使用 Python 获取资产模型列表

1. 创建一个名为 list\_asset\_model.py 的文件，再将以下代码复制到其中。

```
import json
import boto3
import botocore
import os

# create the client using the credentials
client = boto3.client("iotsitewise",
    endpoint_url= "https://" + os.getenv("GATEWAY_HOSTNAME"),
    region_name=os.getenv("AWS_REGION"),
    aws_access_key_id=os.getenv("AWS_ACCESS_KEY_ID"),
    aws_secret_access_key=os.getenv("AWS_SECRET_ACCESS_KEY"),
    aws_session_token=os.getenv("AWS_SESSION_TOKEN"),
```

```
verify=os.getenv("PATH_TO_CERTIFICATE"),
config=botocore.config.Config(inject_host_prefix=False))

# call the api using local credentials
response = client.list_asset_models()
print(response)
```

2. 从终端运行 `list_asset_model.py`。

```
python3 list_asset_model.py
```

curl

**Note**

您需要安装 [curl](#)。

使用 curl 获取资产模型列表

从终端运行以下命令。

```
curl \
  --request GET https://$GATEWAY_HOSTNAME:443/asset-models \
  --cacert $PATH_TO_CERTIFICATE \
  --aws-sigv4 "aws:amz:edge:iotsitewise" \
  --user "$AWS_ACCESS_KEY_ID:$AWS_SECRET_ACCESS_KEY" \
  -H "x-amz-security-token:$AWS_SESSION_TOKEN"
```

该响应应当与以下内容相似：

```
{
  "assetModelSummaries": [
    {
      "arn": "arn:aws:iotsitewise:{region}:{account-id}:asset-model/{asset-
model-id}",
      "creationDate": 1.669245291E9,
      "description": "This is a small example asset model",
      "id": "{asset-model-id}",
      "lastUpdateDate": 1.669249038E9,
      "name": "Some Metrics Model",
```

```
        "status": {
            "error": null,
            "state": "ACTIVE"
        }
    },
    .
    .
    .
],
"nextToken": null
}
```

## 备份和恢复 SiteWise Edge 网关

本主题介绍如何恢复 SiteWise Edge 网关和备份指标数据。如果您在同一台计算机上遇到 SiteWise Edge 网关损坏的问题，并且需要对问题进行故障排除，请阅读 AWS IoT SiteWise 文档 [SiteWise Edge 网关问题疑难解答](#)。

### Note

本主题中介绍的指南适用于安装在 2.1.0 或更高 AWS IoT Greengrass V2 版本的 SiteWise Edge 网关。

## 指标数据的每日备份

如果您想在新机器上传输或恢复数据，则创建备份非常重要。备份数据可以大大降低在传输或恢复过程中丢失操作数据的风险。

influxdb 文件夹路径如下所示：

Linux

```
/greengrass/v2/work/aws.iot.SiteWiseEdgeProcessor/influxdb
```

Windows

```
C:\greengrass\v2\work\aws.iot.SiteWiseEdgeProcessor\influxdb
```

我们建议您备份整个文件夹及其项下的所有内容。

我们建议您定期将指标数据从 1.0 SiteWise Edge 备份到外部硬盘驱动器或 AWS 云端。

## 恢复 SiteWise Edge 网关

使用以下步骤恢复 SiteWise Edge 网关：

1. 使用创建 SiteWise Edge 网关时下载的安装脚本在新计算机上恢复 SiteWise Edge 网关。阅读在[本地设备上安装 SiteWise Edge 网关软件](#)过程以设置 SiteWise Edge 网关。

如果您丢失或找不到安装脚本，请联系 [AWS 客户支持](#)。

2. 安装 E SiteWise dge 网关后，登录[AWS IoT Greengrass 控制台](#)。
3. 要重新部署组件，请导航到管理，再在AWS IoT Greengrass 设备下选择核心设备。
4. 在AWS IoT Greengrass 核心设备表中，选择与您的 SiteWise Edge 网关对应的核心设备。
5. 进入设备页面后，打开部署选项卡并选择您的部署 ID，这将会使用您选择的 ID 打开部署页面。

The screenshot shows the AWS IoT Greengrass console interface. On the left is a navigation menu with categories like Monitor, Connect, Test, and Manage. The main content area is titled 'OriginalGatewayGreengrassCoreDevice-nu7HuEvoH'. It has an 'Overview' section with details like Thing ID, Status (Healthy), Greengrass Core software version (2.9.3), Platform (linux/amd64), and Status reported (1 minute ago). Below this is a 'Deployments' tab, which is highlighted with a red box. Under 'Deployments (1)', there is a table with one entry:

Deployment ID	Name	Target	Status on this device	Status reported
5b5cbd52-607f-4c2c-bc8a-708298e4925a	-	OriginalGatewayGreengrassCoreDevice-nu7HuEvoH	Succeeded	4 days ago

6. 进入部署页面后，按右上角的操作按钮，再选择修订选项启动新的部署。配置部署。如果您想保持部署原样，请跳至查看并部署。
7. 等待部署状态变为 Completed。

### Note

SiteWise Edge 上的所有组件也需要几分钟才能完全设置和运行。

## 恢复 AWS IoT SiteWise 数据

使用以下过程在新计算机上恢复数据。

1. 将该 influxdb 文件夹复制到新机器上。
2. 通过在终端中运行以下命令来停止 SiteWise EdgeProcessor 组件：

Linux

```
sudo /greengrass/v2/bin/greengrass-cli component stop -n  
aws.iot.SiteWiseEdgeProcessor
```

Windows

```
C:\greengrass\v2\bin\greengrass-cli component stop -n  
aws.iot.SiteWiseEdgeProcesso
```

3. 找到您备份数据的路径，然后运行以下命令：

Linux

```
sudo yes | sudo cp -rf <influxdb_backup_path> /greengrass/v2/work/  
aws.iot.SiteWiseEdgeProcessor/influxdb
```

PowerShell

```
Copy-Item -Recurse -Force <influxdb_backup_path>\* C:\greengrass  
\v2\work\aws.iot.SiteWiseEdgeProcessor\
```

Windows

```
robocopy <influxdb_backup_path> C:\greengrass\v2\work  
\aws.iot.SiteWiseEdgeProcessor\ /E
```

4. 重启组 SiteWiseEdgeProcessor 件：

Linux

```
sudo /greengrass/v2/bin/greengrass-cli component restart -n  
aws.iot.SiteWiseEdgeProcessor
```

Windows

```
C:\greengrass\v2\bin\greengrass-cli component restart -n  
aws.iot.SiteWiseEdgeProcessor
```

## 验证备份和恢复是否成功

使用此过程验证您的备份数据和 E SiteWise dge 网关恢复。

### Note

此过程要求您已经安装 AWS OpsHub 了 AWS IoT SiteWise。有关更多信息，请参阅[使用 AWS OpsHub 管理 SiteWise 边缘网关 AWS IoT SiteWise](#)。

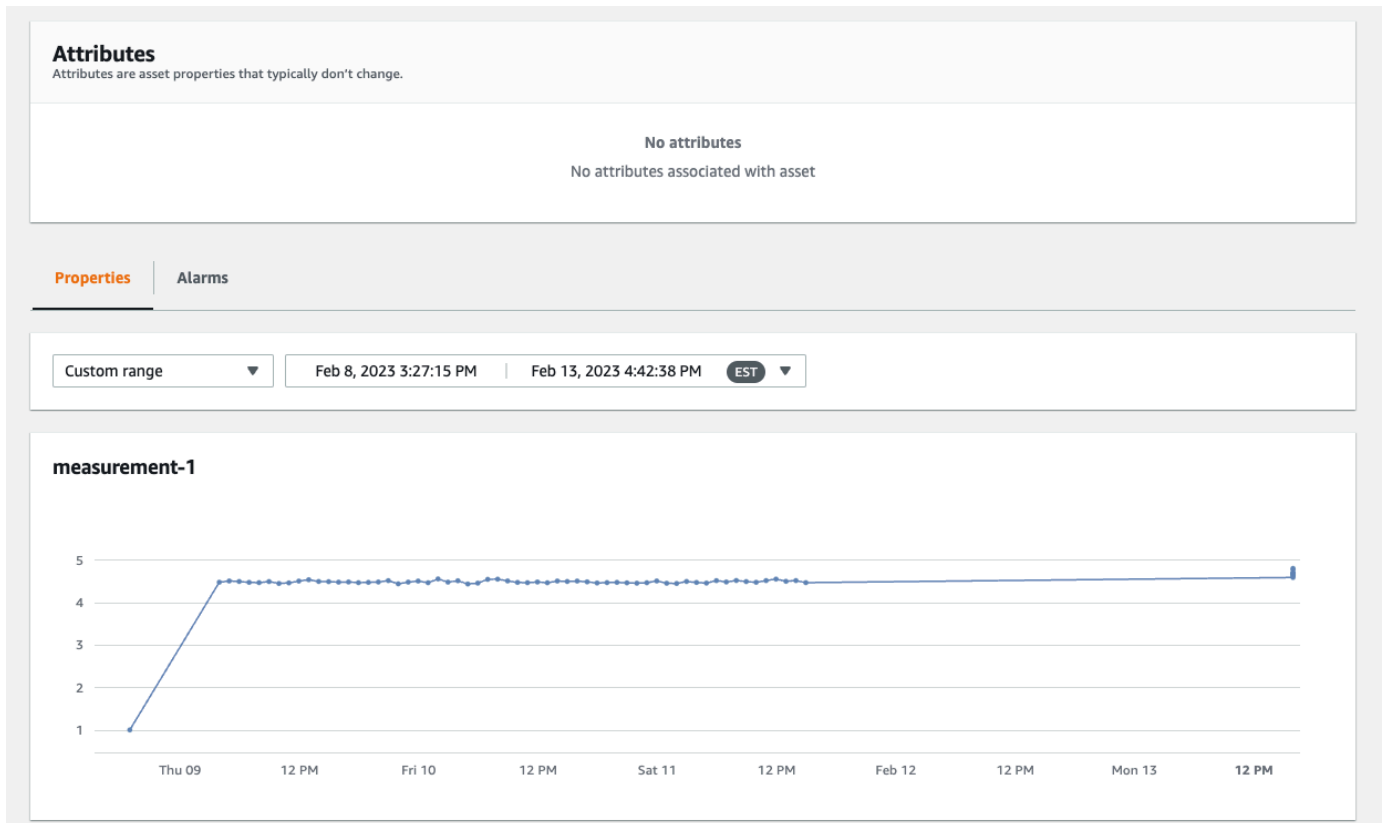
1. 开放 AWS OpsHub 给 AWS IoT SiteWise。
2. 在 SiteWise Edge 网关设置页面上，检查组件表中列出的每个组件的状态。确认状态颜色为绿色且读数显示为正在运行。

The screenshot displays the AWS OpsHub Gateway Settings page. At the top, there is a green banner indicating 'Connection successful'. Below this, the 'Gateway' section is active, showing 'Gateway configuration' with fields for Hostname or IP address (54.202.67.122), Data collection pack (2.2.0, Status: Enabled), and Data processing pack (2.1.29, Status: Enabled). The 'Authentication' section includes buttons for 'Download certificate' and 'Copy credentials'. The 'Logs' section has a 'Download' button. The 'Components' section features a table with the following data:

Name	Status
aws.lot.SiteWiseEdgeProcessor	RUNNING
aws.lot.SiteWiseEdgeCollectorOpcua	RUNNING
aws.lot.SiteWiseEdgePublisher	RUNNING

3. 在门户控制面板上验证您过去数据，检查过去数据和新数据是否都设置正确。过去数据和新数据之间会有停机时间。您应该可以看到一个未收集数据点的持续时间。





如果您在备份或恢复 SiteWise Edge 网关时遇到问题，请参阅以下疑难解答主题对 [E AWS IoT SiteWise dge 网关进行故障排除](#)。

## 设置 SiteWise 边缘网关 (AWS IoT Greengrass Version 1)

### Note

SiteWise 只有在 2021 年 7 月 29 日之前开始使用此功能时，在上 AWS IoT Greengrass V1 运行的 Edge 网关才可用。否则，您可以[设置在上运行的 SiteWise Edge 网关 AWS IoT Greengrass V2](#)。

您可以 AWS IoT SiteWise 使用 SiteWise Edge 网关向发送工业数据，从工业设备上传数据。E SiteWise dge 网关充当 AWS IoT SiteWise 与您的数据工业设备之间的中介。AWS IoT SiteWise 提供了可以部署在任何可以运行 SiteWise Edge 网关 AWS IoT Greengrass 的设备上的组 AWS IoT Greengrass 件。AWS IoT SiteWise 支持与 [OPC-UA 服务器协议](#) 链接。

如果您有在上运行的 AWS IoT SiteWise Edge 网关 AWS IoT Greengrass V1，则可以将您的 SiteWise Edge 网关升级到 AWS IoT Greengrass V2。有关更多信息，请参阅将 [SiteWise Edge 网关从升级到 AWS IoT Greengrass V1 的说明 AWS IoT Greengrass V2](#)。

## 主题

- [选择 AWS IoT Greengrass V1 SiteWise Edge 网关设备](#)
- [配置 AWS IoT Greengrass V1 SiteWise Edge 网关](#)
- [在 AWS IoT Greengrass V1 SiteWise Edge 网关上配置数据源](#)

## 选择 AWS IoT Greengrass V1 SiteWise Edge 网关设备

选择最适合您的工业运营的本地设备。您可以在任何可以运行的设备上配置 SiteWise Edge 网关 AWS IoT Greengrass。所有本地设备都必须满足以下要求：

- 支持 AWS IoT Greengrass 核心软件 v1.10.2 或更高版本。有关更多信息，请参阅 AWS IoT Greengrass Version 1 开发人员指南中的 [支持的平台和要求](#)。
- 至少有 4 GB 的内存。
- 至少有 10 GB 可用磁盘空间。
- 支持 Java 8 虚拟机 (JVM)。

如果您计划在边缘使用处理数据 AWS IoT SiteWise，则您的本地设备还必须满足以下要求：

- 配备 x86 64 位四核处理器。
- 至少有 16 GB 内存。
- 如果使用 Windows，则至少有 32 GB 的内存空间。
- 至少有 256 GB 可用磁盘空间。

缓存间歇性 Internet 连接数据时所需的磁盘空间取决于以下因素：

- 上传的数据流数
- 每秒每个数据流的数据点数
- 每个数据点的大小
- 通信速度
- 预期网络中断

轮询并上传数据时所需的计算容量取决于以下因素：

- 上传的数据流数
- 每秒每个数据流的数据点数

## 配置 AWS IoT Greengrass V1 SiteWise Edge 网关

边 AWS IoT SiteWise 缘网关充当您的工业设备和之间的中介 AWS IoT SiteWise。您可以在任何可以运行的设备上部署 SiteWise Edge 网关软件 AWS IoT Greengrass。有关更多信息，请参阅 [选择 AWS IoT Greengrass V1 SiteWise Edge 网关设备](#)。

您可以使用 E AWS IoT SiteWise dge 网关上的数据处理包在边 SiteWise 缘设备上本地处理数据。当您添加 SiteWise Edge 网关到其中时，即可执行此操作 AWS IoT SiteWise。有关在边缘处理数据的更多信息，请参阅 [the section called “启用边缘数据处理”](#)。

### Note

建议您与对您的本地和公司网络具有 IT 管理访问权限的人员完成以下步骤。这些步骤可能需要了解您的工业设备并有权配置防火墙设置的人员。

### 主题

- [设置 SiteWise Edge 网关环境](#)
- [创建 IAM policy 和角色](#)
- [配置群 AWS IoT Greengrass 组](#)
- [配置 AWS IoT SiteWise 连接器](#)
- [将 SiteWise Edge 网关添加到 AWS IoT SiteWise](#)

## 设置 SiteWise Edge 网关环境

在此过程中，您将安装 AWS IoT Greengrass 并配置您的 SiteWise Edge 网关以与一起使用 AWS IoT SiteWise。

**Note**

此节包括使用 `apt` 命令安装软件包的说明。适用于运行 Ubuntu 或类似工具的系统。如果您使用的不是类似的系统，请参阅供您分发的文档并使用推荐的软件包安装程序。

## 设置 E SiteWise dge 网关

1. 根据需要修改 SiteWise Edge 网关的 [BIOS](#) 设置，如下所示。
  - a. 如果适用，请确保 SiteWise Edge 网关在可能出现停电后自动重启。
  - b. 如果适用，请确保 SiteWise Edge 网关不会进入休眠或睡眠状态。
2. 确保 SiteWise Edge 网关连接到互联网。
3. (可选) 要在没有鼠标、键盘和显示器的情况下使用 SiteWise Edge 网关，请执行以下步骤在 SiteWise Edge 网关 ssh 上进行设置：
  - a. 如果您尚未安装 SSH 软件包，请运行以下命令。

```
sudo apt install ssh
```

- b. 运行以下命令。

```
service ssh status
```

- c. 在输出中搜索 `Active: active (running)` 以确认 SSH 服务器处于运行状态。
- d. 按 Q 退出。

运行以下命令使用 SSH 从另一台计算机连接到 SiteWise Edge 网关。将 `###` 替换为用户登录名，将 `IP` 替换为 SiteWise Edge 网关的 IP 地址。

```
ssh username@IP
```

您可以使用 `-p port-number` 参数连接到默认端口 22 以外的端口。

4. 下载并安装 AWS IoT Greengrass 核心软件 v1.10.2 或更高版本，然后为您 SiteWise 的 Edge 网关创建一个 AWS IoT Greengrass 群组。为此，请参阅 [AWS IoT Greengrass 开发人员指南](#) 中的 [AWS IoT Greengrass 入门](#)。

建议您运行 [AWS IoT Greengrass 设备设置脚本](#) 以快速开始。如果您想更仔细地查看 AWS IoT Greengrass 需求和流程，可以按照 [模块 1](#) 和 [模块 2](#) 中的步骤进行设置 AWS IoT Greengrass。

**⚠ Important**

查看支持 [AWS IoT SiteWise 的地区](#)。在为其选择区域时 AWS IoT Greengrass，请确保该区域也支持 AWS IoT SiteWise。否则，您将无法将 SiteWise Edge 网关连接到 AWS IoT SiteWise。

在继续下一步之前，您应该在 SiteWise Edge 网关上安装 AWS IoT Greengrass Core 软件。

5. 运行以下命令安装 Java 8：

```
sudo apt update
sudo apt install openjdk-8-jre
```

您将在本指南后面安装的 SiteWise Edge 网关软件使用 Java 8 运行时。

6. 运行以下命令验证是否已成功安装 Java。

```
java -version
```

7. C AWS IoT Greengrass ore 软件假设一个 java8 目录。运行以下命令将 Java 安装链接到该 java8 目录。

```
sudo ln -s /usr/bin/java /usr/bin/java8
```

8. 运行以下命令创建 /var/sitewise 数据目录并授予该目录的 ggc\_user 权限。AWS IoT SiteWise 将数据存储在目录中。您是在本过程 AWS IoT Greengrass 前面进行设置 ggc\_user 时创建的。

```
sudo mkdir /var/sitewise
sudo chown ggc_user /var/sitewise
sudo chmod 700 /var/sitewise
```

/var/sitewise 是 AWS IoT SiteWise 使用的默认目录。您可以自定义目录路径（例如，/var/sitewise 替换为 /var/custom/path/），但这样做需要在创建 SiteWise Edge 网关之后执行额外的步骤。有关更多信息，请参阅 [配置 AWS IoT SiteWise 连接器](#) 中的步骤 6。

9. 如果需要，请求您的 IT 管理员将以下终端节点和端口添加到本地网络的允许列表：

- 端口 443、8443 和 8883

**⚠ Important**

您可以将 AWS IoT Greengrass Core 配置为仅使用端口 443 进行所有网络通信。有关更多信息，请参阅 AWS IoT Greengrass 开发人员指南中的[通过端口 443 或网络代理连接](#)。

- 您的 SiteWise 边缘网关的 IP 地址（端口 443）。要获取 IP 地址，请运行 `ip address` 或 `ifconfig` 命令，并记下 `inet` 值（例如，`203.0.113.0`）。
- AWS IoT SiteWise 数据端点：`data.iotsitewise.region.amazonaws.com`（端口 443）。
- SiteWise Edge 网关使用的以下 AWS 终端节点。您可以在 `/greengrass-root/config/config.json` 文件中找到这些内容。将 `greengrass-root` 替换为 AWS IoT Greengrass 安装的根目录。
  - `ggHost`：`greengrass-ats.iot.region.amazonaws.com`（端口 443、8443 和 8883）。
  - `iotHost`：`prefix-ats.iot.region.amazonaws.com`（端口 443、8443 和 8883）。

有关更多信息，请参阅 [AWS IoT Greengrass 终端节点和限额](#)。

10. 如果 AWS IoT Greengrass Core 软件尚未运行，请运行以下命令来启动 AWS IoT Greengrass Core 软件。将 `greengrass-root #####`。AWS IoT Greengrass 默认的 `greengrass-root` 为 `/greengrass`。

```
cd /greengrass-root/ggc/core
sudo ./greengrassd start
```

您应该会看到以下消息：`Greengrass successfully started with PID: some-PID-number`

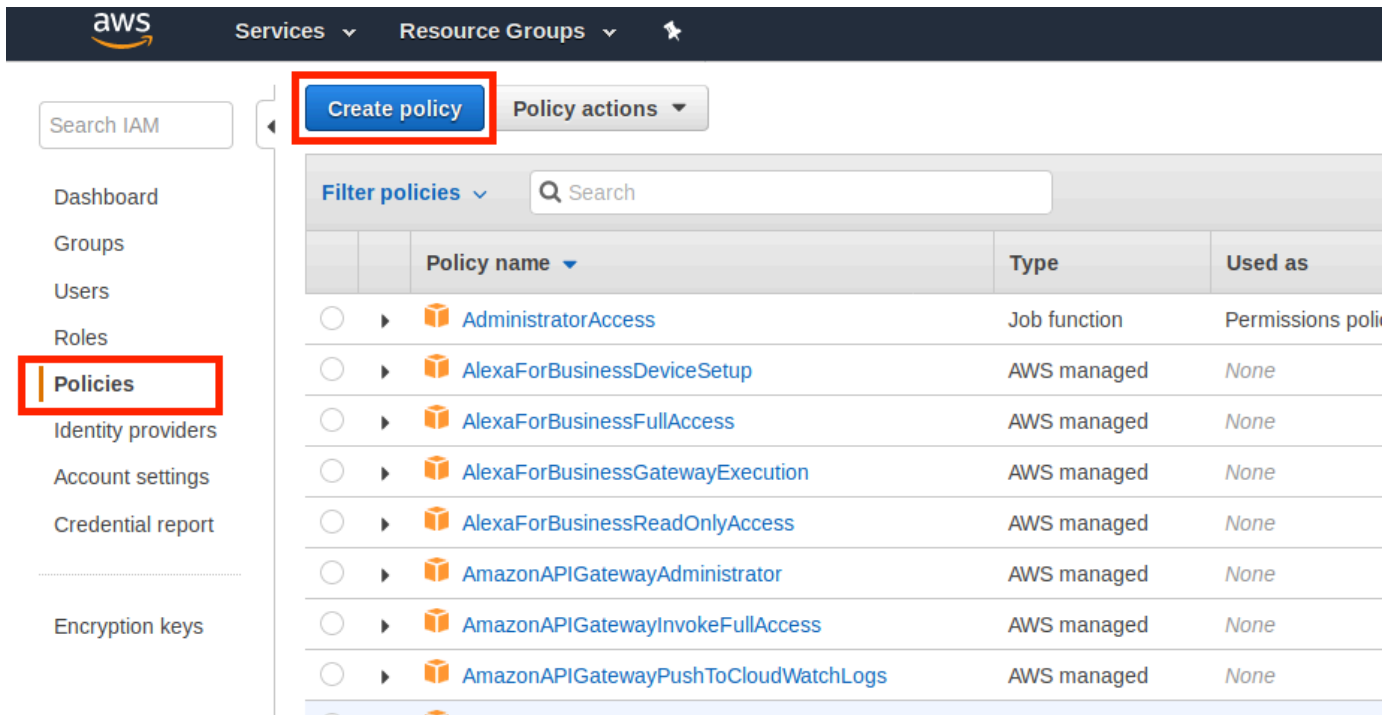
11. 将 AWS IoT Greengrass Core 软件配置为在 SiteWise Edge 网关开启时自动启动。请参阅 SiteWise Edge 网关操作系统的文档。

## 创建 IAM policy 和角色

您必须创建 AWS Identity and Access Management (IAM) 策略和角色才能允许 SiteWise Edge 网关 AWS IoT SiteWise 代表您进行访问。

### 要创建 IAM policy 和角色

1. 导航到 [IAM 控制台](#)。
2. 在导航窗格中，选择 **策略**，然后选择 **创建策略**。



3. 在 JSON 选项卡上，删除策略字段的当前内容，然后将以下策略粘贴到该字段中。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "*"
    }
  ]
}
```

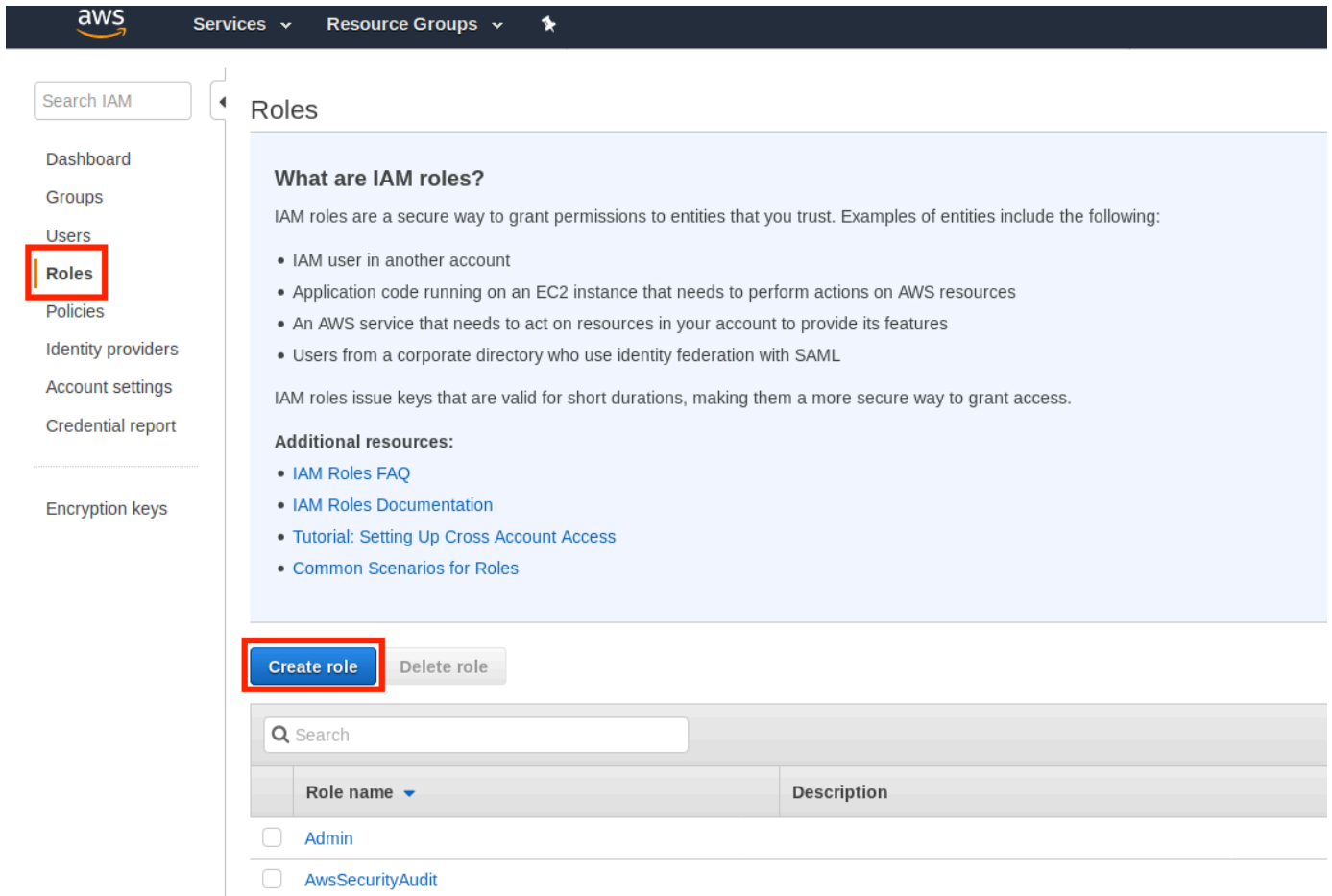
**Note**

为了提高安全性，您可以在Condition属性中指定 AWS IoT SiteWise 资产层次结构路径。以下示例是指定资产层次结构路径的信任策略。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "iotsitewise:assetHierarchyPath": [
            "/root node asset ID",
            "/root node asset ID/*"
          ]
        }
      }
    }
  ]
}
```

4. 选择查看策略。
5. 输入策略的名称和描述，然后选择 创建角色。
6. 在导航窗格中，选择 角色，然后选择 创建角色。





Search IAM

Dashboard

Groups

Users

**Roles**

Policies

Identity providers

Account settings

Credential report

Encryption keys

## Roles

### What are IAM roles?

IAM roles are a secure way to grant permissions to entities that you trust. Examples of entities include the following:

- IAM user in another account
- Application code running on an EC2 instance that needs to perform actions on AWS resources
- An AWS service that needs to act on resources in your account to provide its features
- Users from a corporate directory who use identity federation with SAML

IAM roles issue keys that are valid for short durations, making them a more secure way to grant access.

**Additional resources:**

- [IAM Roles FAQ](#)
- [IAM Roles Documentation](#)
- [Tutorial: Setting Up Cross Account Access](#)
- [Common Scenarios for Roles](#)

**Create role** Delete role

Search


Role name	Description
<input type="checkbox"/> Admin	
<input type="checkbox"/> AwsSecurityAudit	

7. 在选择受信任实体的类型下，选择 AWS 服务。在选择将使用此角色的服务下面，选择 Greengrass 作为将使用该角色的服务，然后选择 下一步：权限。


## Create role

1 2 3 4


## Select type of trusted entity




**AWS service**  
EC2, Lambda and others



**Another AWS account**  
Belonging to you or 3rd party



**Web identity**  
Cognito or any OpenID provider



**SAML 2.0 federation**  
Your corporate directory

Allows AWS services to perform actions on your behalf. [Learn more](#)

## Choose the service that will use this role

**EC2**

Allows EC2 instances to call AWS services on your behalf.

**Lambda**

Allows Lambda functions to call AWS services on your behalf.

API Gateway	CodeBuild	EC2 - Fleet	Inspector	Redshift
AWS Support	CodeDeploy	EKS	IoT	Rekognition
AppSync	Config	EMR	Kinesis	S3
Application Auto Scaling	Connect	ElasticCache	Lambda	SMS
Application Discovery Service	DMS	Elastic Beanstalk	Lex	SNS
Auto Scaling	Data Lifecycle Manager	Elastic Container Service	Machine Learning	SWF
Batch	Data Pipeline	Elastic Transcoder	Macie	SageMaker
CloudFormation	DeepLens	ElasticLoadBalancing	MediaConvert	Service Catalog
CloudHSM	Directory Service	Glue	OpsWorks	Step Functions
CloudTrail	DynamoDB	<b>Greengrass</b>	RAM	Storage Gateway
CloudWatch Events	EC2	GuardDuty	RDS	Trusted Advisor

## Select your use case

\* Required

Cancel

Next: Permissions

8. 搜索您创建的策略，选中复选框，然后选择下一步：标签。

## Create role

1 2 3 4

### ▼ Attach permissions policies

Choose one or more policies to attach to your new role.

Create policy ↻

Filter policies  Showing 1 result

	Policy name	Used as	Description
<input checked="" type="checkbox"/>	SiteWiseDemo	None	Policy for the SiteWise demo.

### ▶ Set permissions boundary

\* Required

Cancel

Previous

Next: Tags

9. (可选) 将标签添加到您的角色，然后选择 下一步：审核。

10. 输入角色的名称和描述，然后选择创建角色。

## Create role



## Review

Provide the required information below and review this role before you create it.

**Role name\***

Use alphanumeric and '+,=,@-\_' characters. Maximum 64 characters.

**Role description**

Allows Greengrass to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+,=,@-\_' characters.

**Trusted entities** AWS service: greengrass.amazonaws.com

**Policies** [SiteWiseDemo](#)

**Permissions boundary** Permissions boundary is not set

No tags were added.

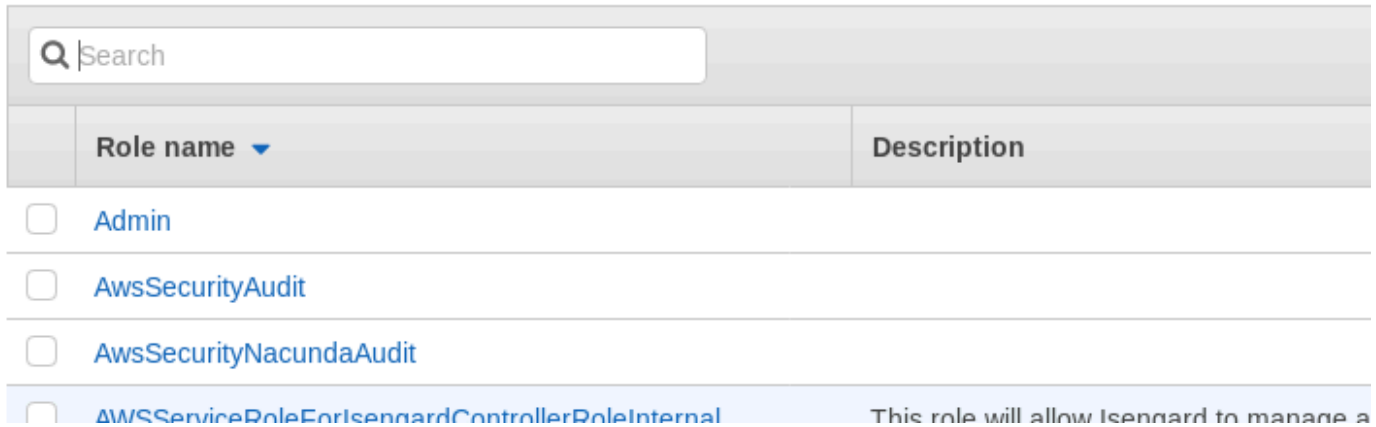
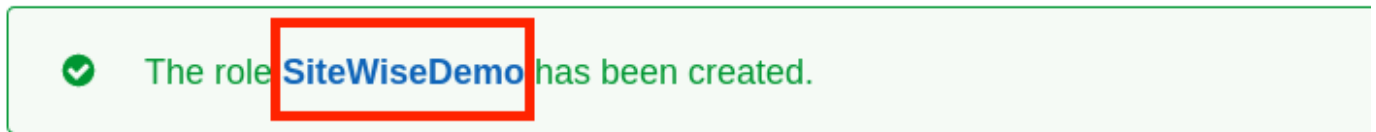
\* Required

[Cancel](#)

[Previous](#)

[Create role](#)

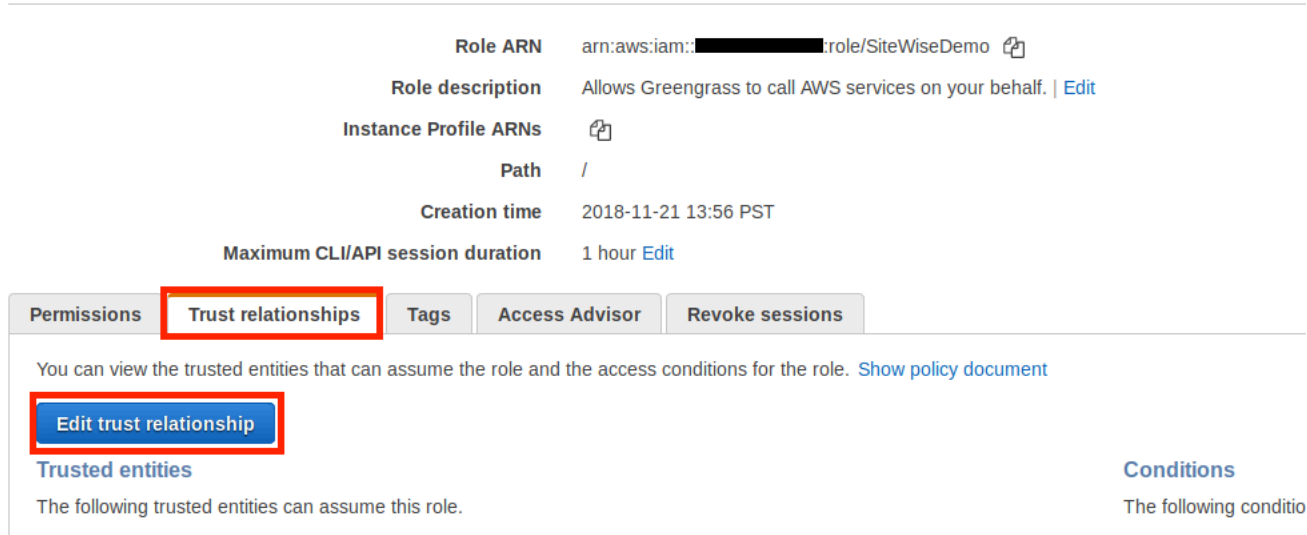
11. 在绿色横幅中，选择指向新角色的链接。您还可以使用搜索字段查找该角色。



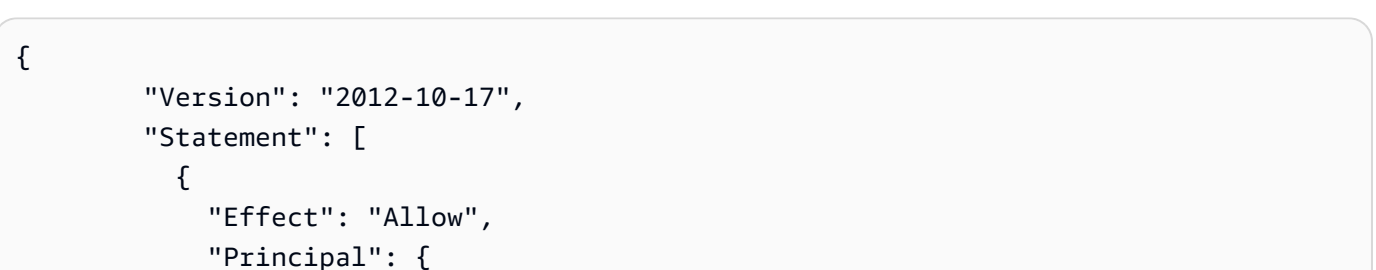
12. 选择 信任关系 选项卡，然后选择 编辑信任关系。

Roles > SiteWiseDemo

## Summary



13. 将策略字段的当前内容替换为以下内容，然后选择 更新信任策略。

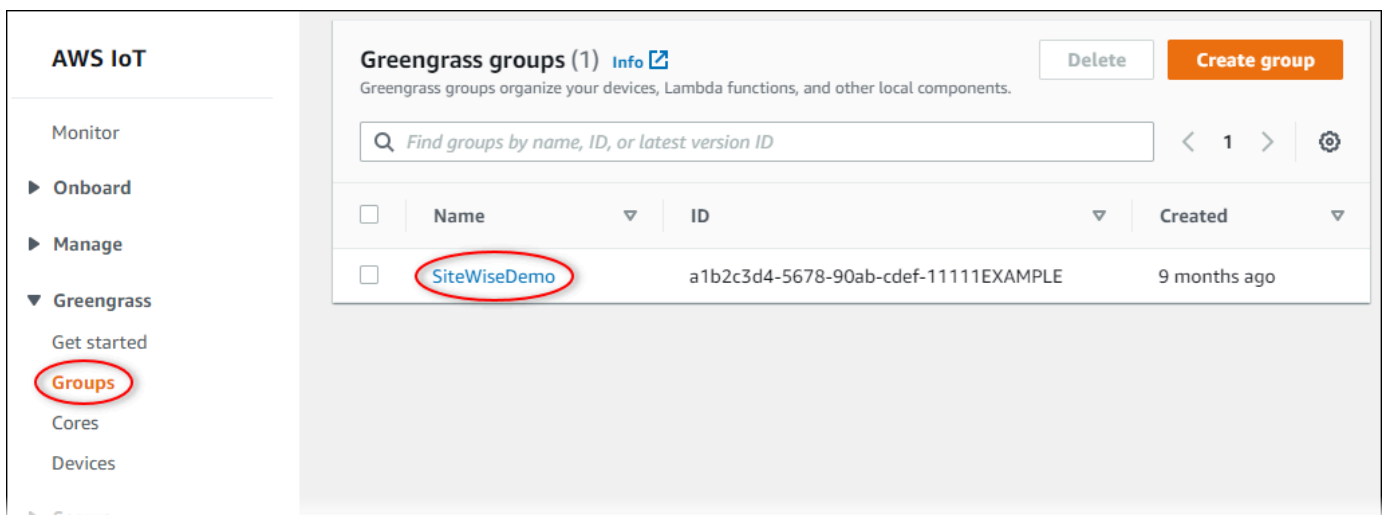


```
        "Service": "greengrass.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

## 配置群 AWS IoT Greengrass 组

将 IAM 角色附加到组并启用流管理器

1. 导航到 [AWS IoT Greengrass 控制台](#)。
2. 在左侧导航窗格中的 Greengrass 下，选择 组，然后选择您在 [设置 SiteWise Edge 网关环境](#) 中创建的组。



3. 在左侧导航窗格中，选择 设置。在 组角色 部分中，选择 添加角色。

GREENGRASS GROUP

## SiteWiseDemo

Not deployed Actions ▾

- Deployments
- Subscriptions
- Cores
- Devices
- Lambdas
- Resources
- Connectors
- Tags
- Settings**

**Group Role** Add Role

No role has been attached to the SiteWiseDemo Group

**Group ID**

1ff7b6c9-06d9-46f5-9f3e-88894dc19b37

**Certification authority (CA) and local connection configuration**

**Device certificate lifetime period**  
By changing this setting you control the period during which a Device can establish a communication with its Core. The next new period will be 7 days.

4. 选择在 [创建 IAM policy 和角色](#) 中创建的角色，然后选择 保存。

## Your Group's IAM Role

Adding an IAM Role to your Group establishes a trust relationship between your trusting account and the Core.

Select an IAM Role with a Greengrass Role Type

Search Role name

SiteWiseDemo

Cancel Back Save

5. 在 设置 页面上的 流管理器 部分中，选择 编辑。

流管理器是一项功能 AWS IoT Greengrass，它使您的 C AWS IoT Greengrass core 能够将数据流式传输到 AWS 云端。SiteWise Edge 网关要求启用流管理器。有关更多信息，请参阅 [《AWS IoT Greengrass Version 1 开发人员指南》](#) 中的 [管理 AWS IoT Greengrass 核心上的数据流](#)。

[Update default Lambda execution configuration](#)

## Stream manager

[Edit](#)

Stream manager enables the Core to ingest and process data streams and export them to cloud targets. [Learn more](#)

### Status

Disabled

## CloudWatch logs configuration

[Edit](#)

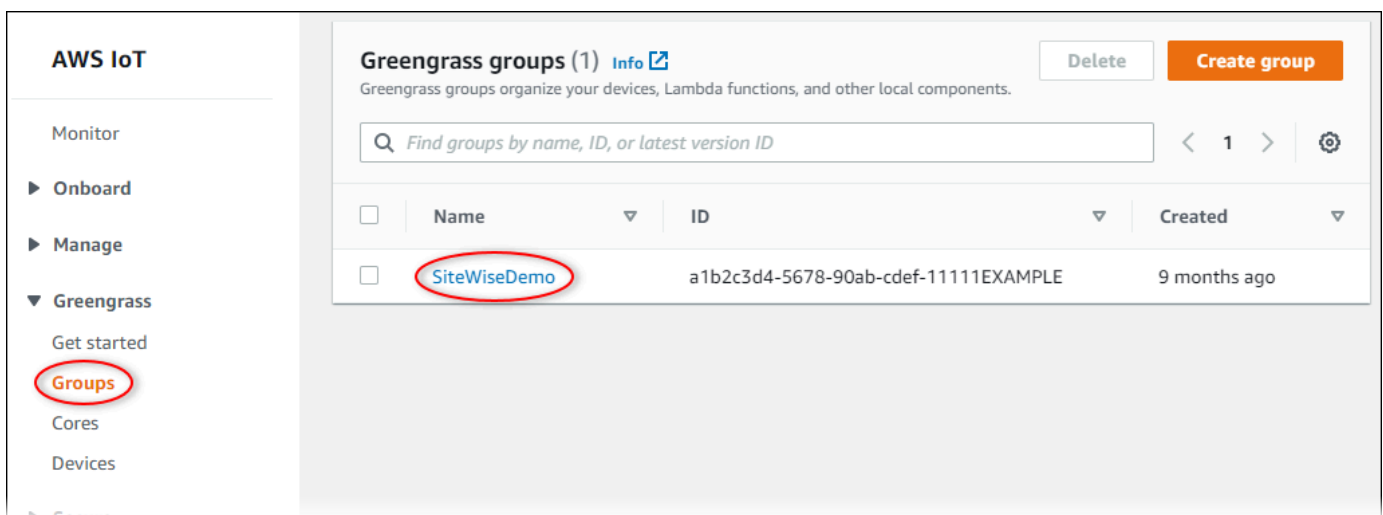
6. 选择 启用，然后选择 保存。
7. 在左上角，选择 服务 为下一过程做好准备。

## 配置 AWS IoT SiteWise 连接器

在此过程中，您将在 Greengrass 组上配置 AWS IoT SiteWise 连接器。组件是预先构建的模块，可加快常见边缘场景的开发生命周期。有关更多信息，请参阅AWS IoT Greengrass Version 1 开发人员指南中的 [AWS IoT Greengrass 连接器](#)。

### 配置 AWS IoT SiteWise 连接器

1. 导航到 [AWS IoT Greengrass 控制台](#)。
2. 在左侧导航窗格中的 Greengrass 下，选择 组，然后选择您在 [设置 SiteWise Edge 网关环境](#) 中创建的组。



3. 在左侧导航页面中，选择 连接器。在连接器页面上，选择 添加连接器。



GREENGRASS GROUP


# SiteWiseDemo

Not deployed Actions ▾

- Deployments
- Subscriptions
- Cores
- Devices
- Lambdas
- Resources
- Connectors**
- Tags
- Settings

## Connectors

Connectors are modules that provide built-in integration with services, protocols, or infrastructure. [Learn more](#)



**Accelerate your development**

Connectors make it easier to develop applications by providing built-in integration with services, protocols, or infrastructure. [Learn more](#)

**Add a connector**

4. SiteWise从列表中选择 IoT，然后选择下一步。

ADD A CONNECTOR TO YOUR GREENGRASS GROUP

## Select a connector

STEP 1/2

Select a connector to add to this group. Connectors that are already in the group are disabled in the list. [Learn more](#)

<input type="radio"/>	CloudWatch Metrics	Version: 2	<a href="#">Learn more</a>
<input type="radio"/>	Device Defender	Version: 2	<a href="#">Learn more</a>
<input type="radio"/>	Docker Application Deployment	Version: 1	<a href="#">Learn more</a>
<input checked="" type="radio"/>	IoT SiteWise	Version: 2	<a href="#">Learn more</a>
<input type="radio"/>	IoT Analytics	Version: 2	<a href="#">Learn more</a>
<input type="radio"/>	Kinesis Firehose	Version: 3	<a href="#">Learn more</a>
<input type="radio"/>	ML Feedback	Version: 1	<a href="#">Learn more</a>
<input type="radio"/>	ML Image Classification ARMv7	Version: 2	<a href="#">Learn more</a>
<input type="radio"/>	ML Image Classification Aarch64 JTX2	Version: 2	<a href="#">Learn more</a>
<input type="radio"/>	ML Image Classification x86_64	Version: 2	<a href="#">Learn more</a>

[Cancel](#) [Next](#)

- 如果您的服务器需要身份验证，则可以使用服务器的用户名和密码创建 AWS Secrets Manager 密钥。然后将每个密钥附加到 Greengrass 组并在用户名/密码密钥的 ARN 列表下选择。有关如何创建和配置密钥的更多信息，请参阅 [配置源身份验证](#)。也可以稍后将密钥添加到连接器。

## List of ARNs for OPC-UA username/password secrets (optional)

List of AWS Secret ARNs

2 secrets selected		Create ↗	Refresh	Clear	Close
Search					
<input checked="" type="checkbox"/>	greengrass-factory1-auth				
<input checked="" type="checkbox"/>	greengrass-factory2-auth				

- 如果您将 SiteWise Edge 网关设置为不同的路径 `/var/sitewise`，请输入该路径作为本地存储路径。
- (可选) 输入连接器的最大磁盘缓冲区大小。如果内 AWS IoT Greengrass 核失去与 AWS 云的连接，连接器会缓存数据，直到它能够成功连接。如果缓存大小超过最大磁盘缓冲区大小，则连接器会丢弃队列中最旧的数据。
- 选择 添加。
- 在右上角的 操作 菜单中，选择 部署。
- 选择 自动检测 启动部署。

如果部署失败，请再次选择 Deploy (部署)。如果部署仍然失败，请参阅 [AWS IoT Greengrass 部署故障排除](#)。

## 将 SiteWise Edge 网关添加到 AWS IoT SiteWise

在此过程中，您将 SiteWise 边缘网关的 Greengrass 组添加到。AWS IoT SiteWise 向注册 SiteWise Edge 网关后 AWS IoT SiteWise，该服务可以将您的数据源配置部署到您的 SiteWise Edge 网关。

### 将 SiteWise Edge 网关添加到 AWS IoT SiteWise

- 导航到 [AWS IoT SiteWise 控制台](#)。
- 选择 添加网关。
- 在添加 SiteWise 网关页面上，执行以下操作：

- a. 输入 E SiteWise dge 网关的名称。考虑在名称中包含 SiteWise Edge 网关的位置，这样您就可以轻松识别它。
- b. 在 Greengrass 组 ID 上，选择您此前创建的 Greengrass 组。

### Example

AWS IoT SiteWise > Gateways > Add SiteWise gateway

## Add SiteWise gateway

**Select a connected gateway**

SiteWise utilizes an on-premises gateway that collects data from local data servers and uploads the selected data. Once you or your IT Administrator have installed the software, registered it to AWS IoT Greengrass and connected it to your local network you can add it to the SiteWise service.  
[Learn more about this process and ordering hardware](#)

Gateway name  
Using the deployment location as a name makes identifying your gateway easier.

Alexandria

Greengrass group ID  
SiteWise gateway appliances must be connected to via AWS IoT Greengrass.

SiteWiseDemo

Cancel Add gateway

- c. (可选) 在边缘功能中，选择数据处理包。这使您的 SiteWise Edge 网关与为边缘配置的任何资产模型和资产之间可以进行通信。有关更多信息，请参阅 [the section called “启用边缘数据处理”](#)。

#### **⚠ Important**

如果将数据处理包添加到 SiteWise Edge 网关，则必须在 AWS IoT Greengrass 组中配置和部署 SiteWise Edge 连接器。按下列步骤操作。

- d. 选择 添加网关。
4. 如果您将数据处理包添加到 SiteWise Edge 网关，请在您的 AWS IoT Greengrass 群组上配置和部署 AWS IoT SiteWise 数据处理器连接器。按照中的 [the section called “配置 AWS IoT SiteWise 连接器”](#) 步骤配置 AWS IoT SiteWise 数据处理器连接器：

- a. 对于“在 AWS IoT Greengrass 控制台中选择连接器”，选择“AWS IoT SiteWise 数据处理器”。
- b. 在本地存储路径中，输入 SiteWise Edge 网关的路径。
- c. 选择 添加。
- d. 在右上角的操作菜单中，选择部署，再选择自动检测开始部署。

部署 SiteWise 边缘网关后，您可以为希望边 SiteWise 缘网关从中提取数据的每台工业设备添加一个来源。有关更多信息，请参阅 [配置数据源](#)。

您可以查看 Amazon CloudWatch 指标以验证您的 SiteWise Edge 网关是否已连接 AWS IoT SiteWise。有关更多信息，请参阅 [AWS IoT Greengrass Version 1 网关指标](#)。

## 在 AWS IoT Greengrass V1 SiteWise Edge 网关上配置数据源

设置 AWS IoT SiteWise Edge 网关后，您可以配置数据源，以便您的 SiteWise Edge 网关可以将来自本地工业设备的数据摄取到 AWS IoT SiteWise。每个源都代表一个本地服务器，例如 OPC-UA 服务器，您的 SiteWise 边缘网关可以连接和检索工业数据流。有关设置 SiteWise Edge 网关的更多信息，请参阅[配置 AWS IoT Greengrass V1 SiteWise Edge 网关](#)。

### Note

AWS IoT SiteWise 每次添加或编辑源时都会重新启动 SiteWise Edge 网关。您的 SiteWise Edge 网关在重启时不会提取数据。重启 SiteWise Edge 网关的时间取决于 E SiteWise dge 网关源上的标签数量。重启时间可以从几秒钟（对于标签较少的 SiteWise Edge 网关）到几分钟（对于具有许多标签的 SiteWise Edge 网关）不等。

创建源后，您可以将数据流与资产属性关联。有关如何创建和使用资产的更多信息，请参阅 [工业资产建模](#) 和 [将工业数据流映射到资产属性](#)。

您可以查看 CloudWatch 指标以验证数据源是否已连接 AWS IoT SiteWise。有关更多信息，请参阅 [AWS IoT Greengrass Version 1 网关指标](#)。

目前，AWS IoT SiteWise 支持以下数据源协议：

- [OPC-UA](#) — 一种用于工业自动化的通信协议 machine-to-machine ( M2M )。
- [Modbus TCP](#) — 一种用于与可编程逻辑控制器 (PLC) 接口的数据通信协议。

- [以太网/IP \(EIP\)](#) — 一种工业网络协议，可将通用工业协议 (CIP) 与标准以太网相适应。

### Note

SiteWise AWS IoT Greengrass V2 当前运行的边缘网关不支持 Modbus TCP 和以太网 IP 源。

## 主题

- [配置 Modbus TCP 源](#)
- [配置以太网/IP \(EIP\) 源](#)
- [配置源身份验证](#)
- [升级连接器](#)

## 配置 Modbus TCP 源

您可以使用 AWS IoT SiteWise 控制台或 AWS IoT SiteWise 边缘网关功能来定义并向边 SiteWise 边缘网关添加 Modbus TCP 源。此源代表本地 Modbus TCP 服务器。

### Note

- SiteWise AWS IoT Greengrass V2 当前运行的边缘网关不支持 Modbus TCP 源。
- 必须安装 AWS IoT SiteWise 连接器才能使用 Modbus TCP 源。

当您的 E SiteWise dge 网关收到数据类型时，您可以使用 Modbus TCP 源将源中的数据类型转换为不同的数据类型。源数据类型决定了您可以为目标数据选择的数据类型。还可以选择使用 Modbus TCP 源来选择交换字节。下表提供了有关兼容的源数据类型、目标数据类型和交换模式的更多信息。

有关交换模式的更多信息，请参阅有关 Modbus 消息编码的 [Modbus RTU 消息中如何编码真实 \(浮点\) 和 32 位数据](#) 的文章。

源数据类型	兼容的目标数据类型	兼容的交换模式	兼容的连接器版本
ASCII	String	noSwap	2

源数据类型	兼容的目标数据类型	兼容的交换模式	兼容的连接器版本
UTF8	String	noSwap	2
ISO8859	String	noSwap	2
Int16	整数、双精度、字符串	noSwap	1 和 2
Int32	整数、双精度、字符串	NoSwap、byteSwap、wordSwap、WordSwap	1 和 2
浮点型	双精度，字符串	NoSwap、byteSwap、wordSwap、WordSwap	1 和 2
布尔值	布尔值	noSwap	1 和 2
十六进制转储	String	noSwap	1 和 2

## 主题


- [配置 Modbus TCP 源 \(控制台\)](#)
- [配置 Modbus TCP 源 \(CLI\)](#)

## 配置 Modbus TCP 源 (控制台)

### 要配置 Modbus TCP 源


1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在左侧导航窗格中，选择 网关。
3. 在要为其创建源的 SiteWise Edge 网关上，选择管理，然后选择查看详细信息。
4. 选择右上角的 新建源。
5. 对于协议选项，请选择 Modbus TCP。
6. 对于 Modbus TCP 源配置，请输入源的名称。
7. 对于源 IP 地址，输入数据来源服务器的 IP 地址。

8. (可选) 输入源服务器的端口和设备 ID。
9. (可选) 在最小请求间持续时间中，输入向服务器发送后续请求间的时间间隔。您的 SiteWise Edge 网关会根据您的设备和您拥有的寄存器数量自动计算允许的最小间隔。
10. 对于属性组，输入名称。
11. 对于属性：
  - a. 在标签中，输入您的寄存器集的属性别名。例如，**TT-001**。
  - b. 在寄存器地址中，输入启动寄存器集的寄存器地址。
  - c. 对于源数据类型，选择要转换数据的 Modbus TCP 数据类型。默认为十六进制转储。

 Note

您选择的源数据类型决定了您可以选择的数据大小、目标数据类型和交换模式。有关更多信息，请参阅 [the section called “配置 Modbus TCP 源”](#)。

- d. 对于数据大小，输入从寄存器地址开始时要读取的寄存器数量。这取决于您为此源选择的源数据类型。
  - e. 在目标数据类型中 AWS IoT SiteWise，选择要将数据转换为的数据类型。默认值为字符串。目标类型必须与您为此源选择的源数据类型兼容。有关更多信息，请参阅 [the section called “配置 Modbus TCP 源”](#)。
  - f. 对于交换模式，选择要用于从寄存器集读取数据的数据交换模式。交换模式必须与您为此源选择的源数据类型兼容。有关更多信息，请参阅 [the section called “配置 Modbus TCP 源”](#)。
12. 对于扫描速率，更新您希望 SiteWise Edge 网关读取寄存器的速率。AWS IoT SiteWise 自动计算您的 SiteWise Edge 网关允许的最低扫描速率。
  13. (可选) 对于目标，选择源数据的发送位置。默认情况下，您的来源会将数据发送到 AWS IoT SiteWise。您可以使用 AWS IoT Greengrass 流将数据导出到本地目标或 AWS 云端。


 Note

如果要在边缘处理来自该源的数据，则必须选择 AWS IoT SiteWise 作为源数据的目的地 AWS IoT SiteWise。有关在边缘处理数据的更多信息，请参阅 [the section called “启用边缘数据处理”](#)。

如需将您的数据发送到其他目标：



- a. 在目标选项中，选择其他目标。
- b. 对于 Greengrass 直播名称，请输入直播的确切名称。AWS IoT Greengrass

 Note


您可以使用已经创建的流，也可以创建新的 AWS IoT Greengrass 流导出数据。如果要使用现有流，则必须输入该流的确切名称，否则将创建一个新流。有关使用 AWS IoT Greengrass 流的更多信息，请参阅 AWS IoT Greengrass 开发者指南中的[管理数据流](#)。

#### 14. 选择 添加源。

AWS IoT SiteWise 将 SiteWise Edge 网关配置部署到您的 AWS IoT Greengrass 核心。您无需手动启动部署。

#### 配置 Modbus TCP 源 (CLI)

您可以在 SiteWise 边缘网关功能中定义 Modbus TCP 数据源。您必须在单个功能配置中定义所有 Modbus TCP 源。

 Note

必须安装 AWS IoT SiteWise 连接器才能使用 Modbus TCP 源。

此功能具有以下版本。

版本	命名空间
1	iotsitewise:modbuscollector:1

#### Modbus TCP 功能配置参数

在功能配置中定义 Modbus TCP 源时，您必须在 capabilityConfiguration JSON 文档中指定以下信息：

## 来源

Modbus TCP 源定义结构的列表，每个结构均包含以下信息：

### name

源的唯一友好名称。

### measurementDataStream前缀

( 可选 ) 要在源中所有数据流的前面加上的字符串。SiteWise Edge 网关将此前缀添加到来自该源的所有数据流。使用数据流前缀，可以区分来自不同源的同名数据流。在您的账户中，每个数据流都应具有唯一的名称。

### 目的地

包含以下信息的目标结构：

### 类型

目标的类型。

### 直播名称

AWS IoT Greengrass 直播的名称。

### streamBufferSize

流缓冲的大小。

### 终端节点

一个包含以下信息的终端节点结构：

### IP 地址

Modbus TCP 源的 IP 地址。

### port (远程调试端口)

( 可选 ) Modbus TCP 源的端口。

### unitID

( 可选 ) UnitID。默认值为 1。

### minimumInterRequest持续时间

每次请求间的最短持续时间 ( 以毫秒为单位 ) 。

## 属性组

定义协议要求的标签定义的属性组列表。

`name`

属性组的名称。这应该是一个唯一的标识符。

`tagPathDefinitions`

测量值在源中的位置。例如，字节和字序、地址和转换类型。每个 `MeasurementPathDefinition` 的结构都由连接器定义。

`扫描模式`

定义源的扫描模式行为和可配置参数。

## 配置以太网/IP (EIP) 源

您可以使用 AWS IoT SiteWise 控制台或 SiteWise Edge 网关功能来定义以太网 IP 源并将其添加到您的 SiteWise Edge 网关。此源代表本地以太网 IP 服务器。

### Note

- SiteWise AWS IoT Greengrass V2 当前运行的边缘网关不支持以太网 IP 源。
- 必须安装 AWS IoT SiteWise 连接器才能使用以太网 IP 源。

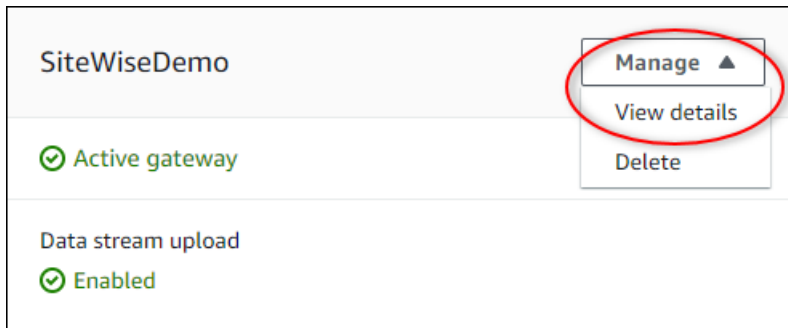
## 主题

- [配置以太网/IP 源 \( 控制器 \)](#)
- [配置以太网/IP 源 \(CLI\)](#)

## 配置以太网/IP 源 ( 控制器 )

### 配置以太网/IP 源

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在左侧导航窗格中，选择 **网关**。
3. 在要为其创建源的 SiteWise Edge 网关上，选择 **管理**，然后选择 **查看详细信息**。



4. 选择右上角的 新建源。
5. 对于协议选项，请选择以太网/IP (EIP)。
6. 对于 EtherNet/IP 源配置，请输入源的名称。
7. 对于源 IP 地址，输入数据来源服务器的 IP 地址。
8. (可选) 输入源服务器的端口。
9. 在最小请求间持续时间中，输入向服务器发送后续请求间的时间间隔。您的 SiteWise Edge 网关会根据您的设备和您拥有的寄存器数量自动计算允许的最小间隔。
10. 对于属性组，输入名称。
11. 对于属性：
  - a. 在标签中，输入您的寄存器集的属性别名。例如，**boiler.inlet.temperature.value**。
  - b. 在目标数据类型中 AWS IoT SiteWise，选择要将数据转换为的数据类型。默认值为字符串。
12. 对于扫描速率，更新您希望 SiteWise Edge 网关读取寄存器的速率。AWS IoT SiteWise 自动计算您的 SiteWise Edge 网关允许的最低扫描速率。
13. (可选) 对于目标，选择源数据的发送位置。默认情况下，您的来源会将数据发送到 AWS IoT SiteWise。您可以使用 AWS IoT Greengrass 流将数据导出到本地目标或 AWS 云端。

#### Note

如果要在边缘处理来自该源的数据，则必须选择 AWS IoT SiteWise 作为源数据的目的地 AWS IoT SiteWise。有关在边缘处理数据的更多信息，请参阅 [the section called “启用边缘数据处理”](#)。

如需将您的数据发送到其他目标：

- a. 在目标选项中，选择其他目标。

b. 对于 Greengrass 直播名称，请输入直播的确切名称。AWS IoT Greengrass

**Note**

您可以使用已经创建的流，也可以创建新的 AWS IoT Greengrass 流导出数据。如果要使用现有流，则必须输入该流的确切名称，否则将创建一个新流。有关使用 AWS IoT Greengrass 流的更多信息，请参阅 AWS IoT Greengrass 开发者指南中的[管理数据流](#)。

14. 选择 添加源。

AWS IoT SiteWise 将 SiteWise Edge 网关配置部署到您的 AWS IoT Greengrass 核心。您无需手动启动部署。

### 配置以太网/IP 源 (CLI)

您可以在 E SiteWise dge 网关功能中定义 EIP 数据源。您必须在单个功能配置中定义所有 EIP 源。

**Note**

必须安装 AWS IoT SiteWise 连接器才能使用以太网 IP 源。

此功能具有以下版本。

版本	命名空间
1	ioticsitewise:eipcollector:1

### EIP 功能配置参数

在功能配置中定义 EIP 源时，您必须在 capabilityConfiguration JSON 文档中指定以下信息：

#### 来源

EIP 源定义结构的列表，每个结构均包含以下信息：

#### name

源的唯一友好名称。最多可包含 256 个字符。

## destinationPathPrefix

( 可选 ) 要在源中所有数据流的前面加上的字符串。 SiteWise Edge 网关将此前缀添加到来自该源的所有数据流。使用数据流前缀，可以区分来自不同源的同名数据流。在您的账户中，每个数据流都应具有唯一的名称。

### 目的地

包含以下信息的目标结构：

#### 类型

目标的类型。

#### 直播名称

AWS IoT Greengrass 直播的名称。

#### streamBufferSize

流缓冲的大小。

### 终端节点

一个包含以下信息的终端节点结构：

#### IP 地址

EIP 源的 IP 地址。

#### port (远程调试端口)

( 可选 ) EIP 源的端口。接受的值为 1 到 65535 之间的数字。

### minimumInterRequest持续时间

( 可选 ) 每次请求间的最短持续时间 ( 以毫秒为单位 )。

### 属性组

定义协议要求的标签定义的属性组列表。每个来源可以有一个属性组。

#### name

属性组的名称。这应该是一个唯一标识符，最大长度为 256 个字符。

#### tagPathDefinitions

指定要从以太网/IP 设备收集的数据以及如何将其转换为输出的结构列表。

## 类型

tagPathDefinition 的类型。例如，EIPTagPath。

## path

tagPathDefinition 的路径。路径中的每个标签的最大长度为 40 个字符，并且能够以字母或下划线开头。标签不能包含连续或后置下划线。路径的前缀为任意值。

destinationPathPrefix

## dstDataType

输出标签数据的数据类型。接受的值为 : integer、double、string 和 boolean。

## 扫描模式

定义源的扫描模式行为和可配置参数。

## 类型

扫描模式行为的类型。接受的值为 POLL。

## 费率

连接器应从以太网/IP 源读取标签的速率（以毫秒为单位）。

## 配置源身份验证

如果 OPC-UA 服务器需要身份验证凭证才能连接，您可以为 AWS Secrets Manager 中的每个源定义密钥中的用户名和密码。然后，您将该密钥添加到您的 Greengrass 群组 and SiteWise 物联网连接器中，使该密钥可供您的 Edge 网关使用。SiteWise 有关更多信息，请参阅《AWS IoT Greengrass Version 1 开发人员指南》中的[将密钥部署到内 AWS IoT Greengrass 核](#)。

在 SiteWise Edge 网关获得密钥后，您可以在配置源时选择该密钥。然后，SiteWise Edge 网关在连接到源时使用来自密钥的身份验证凭据。有关更多信息，请参阅[配置数据源](#)。

## 主题

- [创建源身份验证密钥](#)
- [添加密钥至 Greengrass 组](#)
- [向 IoT SiteWise 连接器添加机密](#)

## 创建源身份验证密钥

在此过程中，您将为 Secrets Manager 中的源创建身份验证密钥。在密钥中，定义 **username** 和 **password** 密钥/值对，其中包含源的身份验证详细信息。

### 创建源身份验证密钥

1. 导航到 [Secrets Manager 控制台](#)。
2. 选择 存储新密钥。
3. 在 选择密钥类型 下，选择 其他密钥类型。
4. 为 OPC-UA 服务器的身份验证值输入 **username** 和 **password** 键值对，然后选择 下一步。

The screenshot shows the AWS Secrets Manager console interface. The 'Select secret type' section has four radio button options: 'Credentials for RDS database', 'Credentials for Redshift cluster', 'Credentials for DocumentDB database', and 'Other type of secrets (e.g. API key)'. The 'Other type of secrets' option is selected and highlighted with a red box. Below this, the 'Specify the key/value pairs to be stored in this secret' section is active, showing a table with two rows: 'username' and 'password'. Each row has an empty input field for the value and a 'Remove' button. A '+ Add row' link is visible below the table. At the bottom, the 'Select the encryption key' section shows a dropdown menu with 'DefaultEncryptionKey' selected and a refresh button. The 'Next' button is highlighted in orange at the bottom right.

5. 输入以 **greengrass-** 开头的密钥名称，例如 **greengrass-factory1-auth**。

### **⚠ Important**

您必须为默认 AWS IoT Greengrass 服务角色使用 **greengrass-** 前缀以访问您的密钥。如果要命名不带此前缀的密钥，则必须授予访问密钥的 AWS IoT Greengrass 自定义权



限。有关更多信息，请参阅《AWS IoT Greengrass Version 1 开发者指南》中的“[允许 AWS IoT Greengrass 获取机密值](#)”。

## Store a new secret

### Secret name and description Info

#### Secret name

Give the secret a name that enables you to find and manage it easily.

greengrass-factory1-auth

Secret name must contain only alphanumeric characters and the characters /\_+@-

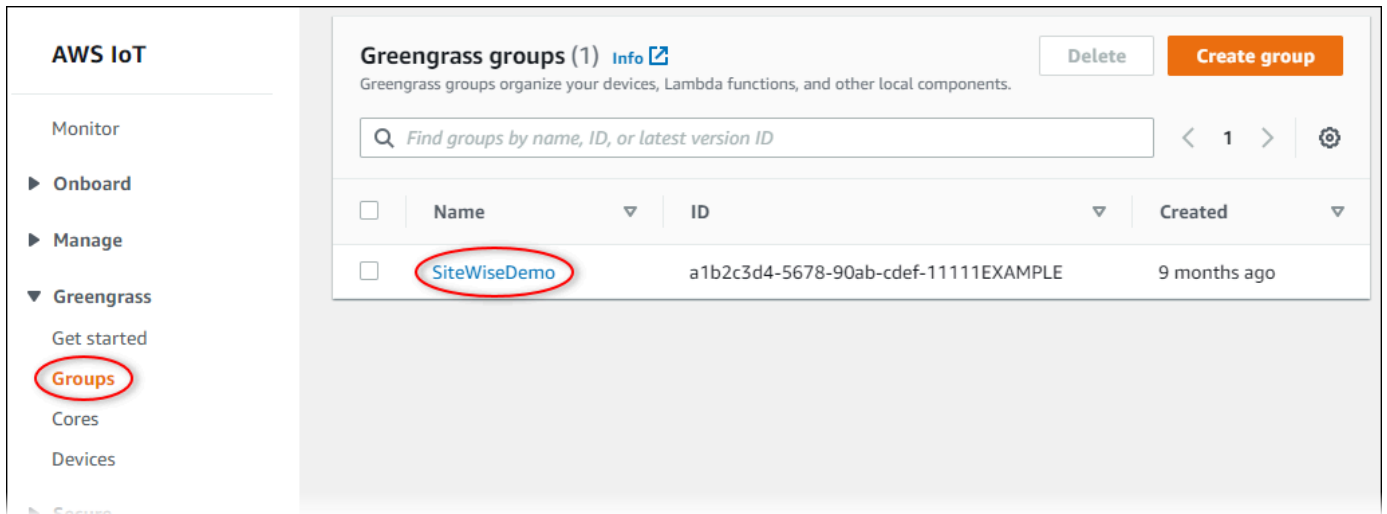
6. 输入描述，然后选择下一步。
7. （可选）在配置自动轮换页面上，为您的密钥配置自动轮换。如果您配置自动轮换，则必须在每次轮换密钥时重新部署 Greengrass 组。
8. 在配置自动旋转页面，选择下一步。
9. 查看您的新密钥并选择存储。

## 添加密钥至 Greengrass 组

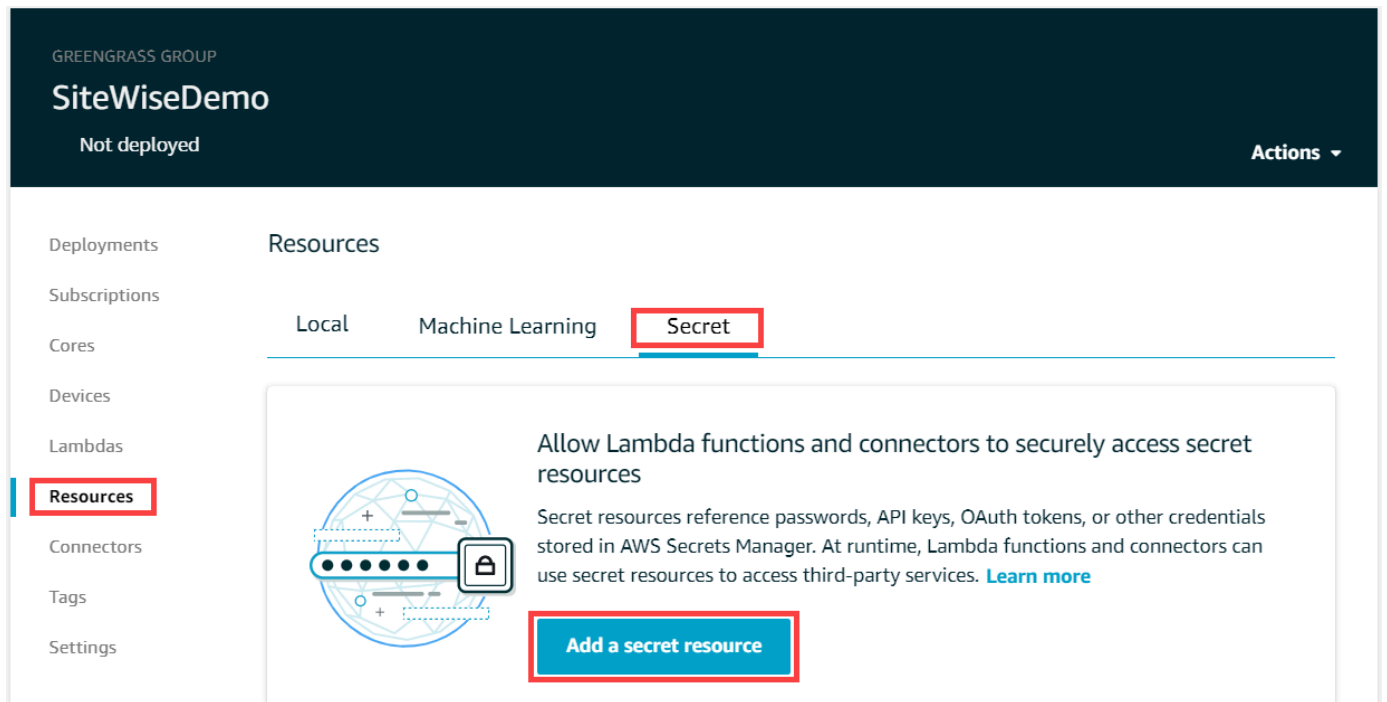
在此过程中，您将源身份验证密钥添加到您的 AWS IoT Greengrass 群组中，使其可供您的 IoT SiteWise 连接器使用。

## 添加一个密钥至您的 Greengrass 组

1. 导航到 [AWS IoT Greengrass 控制台](#)。
2. 在导航窗格的 Greengrass 下，选择组，然后选择您的组。



3. 在导航页面中，选择资源。
4. 在资源页面上，选择 密钥 选项卡，然后选择 添加密钥资源。



5. 选择 选择，从列表中选择您的密钥。
6. 选择下一步。
7. 在 密钥资源名称 中，输入密钥资源的名称，然后选择 保存。

ADD A RESOURCE TO YOUR GREENGRASS GROUP

## Name your secret resource

STEP 3/3

Your secret resource will be added to the group. Give it a unique name so you can easily identify it. [Learn more](#)

**Secret resource name**

The name can contain alphanumeric characters, colons, underscores, and dashes.

**Secret name**  
greengrass-factory1-auth

**Labels**  
AWSCURRENT

[Cancel](#) [Back](#) [Save](#)

### 向 IoT SiteWise 连接器添加机密

在此过程中，您将源身份验证密钥添加到 IoT SiteWise 连接器中，使其可供 AWS IoT SiteWise 和您的 SiteWise Edge 网关使用。

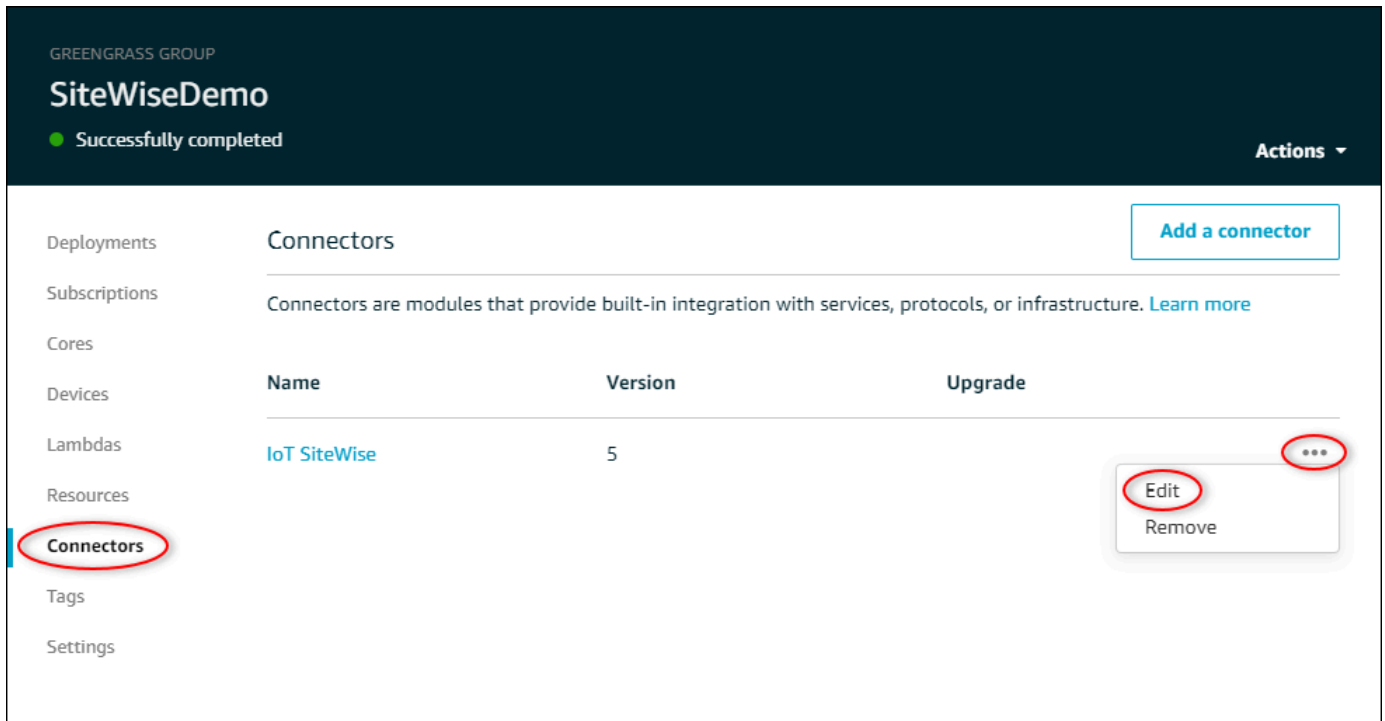
### 向 IoT SiteWise 连接器添加密钥

1. 导航到 [AWS IoT Greengrass 控制台](#)。
2. 在导航窗格的 Greengrass 下，选择组，然后选择您的组。

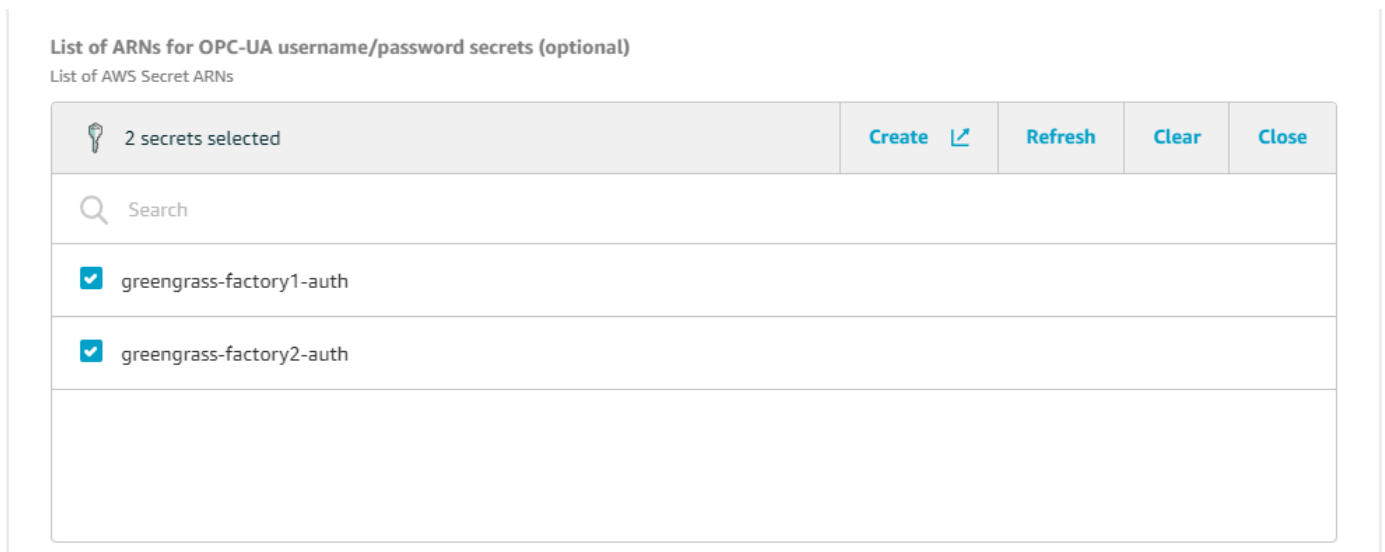
The screenshot shows the AWS IoT Greengrass console interface. On the left, the navigation pane is expanded to 'Greengrass', and the 'Groups' link is highlighted with a red circle. The main content area displays 'Greengrass groups (1)' with a search bar and a table of groups. The table has columns for 'Name', 'ID', and 'Created'. One group is listed with the name 'SiteWiseDemo', which is circled in red. The ID is 'a1b2c3d4-5678-90ab-cdef-11111EXAMPLE' and it was created '9 months ago'. There are 'Delete' and 'Create group' buttons at the top right of the table area.

<input type="checkbox"/>	Name	ID	Created
<input type="checkbox"/>	SiteWiseDemo	a1b2c3d4-5678-90ab-cdef-11111EXAMPLE	9 months ago

3. 在导航页面中，选择连接器。
4. 选择 IoT SiteWise 连接器的省略号图标以打开选项菜单，然后选择编辑。



5. 在 OPC-UA 用户名/密码密钥的 ARN 列表下，选择选择，然后选择要添加到此 Edge 网关的每个密钥。SiteWise 如果您需要创建密钥，请参阅[创建源身份验证密钥](#)。



如果您的密钥未显示，请选择 刷新。如果您的密钥仍未显示，请检查您是否已[将密钥添加到您的 Greengrass 组](#)。

6. 选择保存。
7. 在右上角的 操作 菜单中，选择 部署。

## 8. 选择 自动检测 启动部署。

如果部署失败，请再次选择 Deploy (部署)。如果部署仍然失败，请参阅 [AWS IoT Greengrass 部署故障排除](#)。

在部署组后，您可以配置使用新密钥的源。有关更多信息，请参阅 [配置数据源](#)。

## 升级连接器

### Important

[IoT SiteWise 连接器版本 6](#) 引入了新要求：[AWS IoT Greengrass 核心软件 v1.10.0](#) 和 [直播管理器](#)。在升级连接器之前，请检查您的 SiteWise Edge 网关是否满足这些要求，否则您将无法部署 SiteWise Edge 网关。

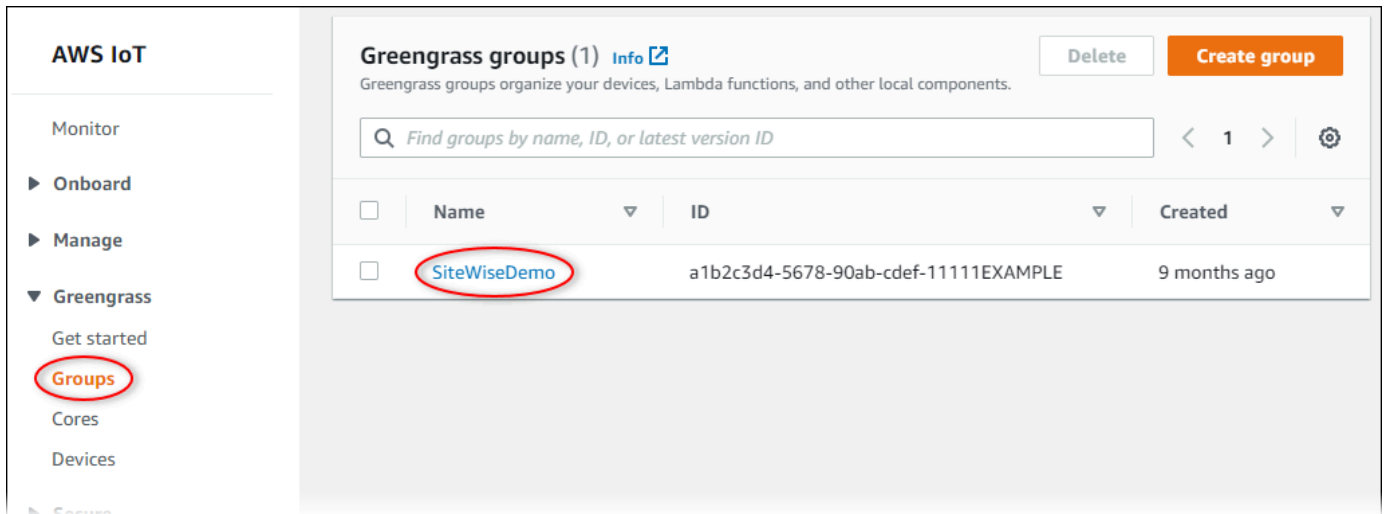
在新的 IoT 连接器版本发布后，您可以轻松升级 SiteWise Edge 网关的 SiteWise 连接器。

### Note

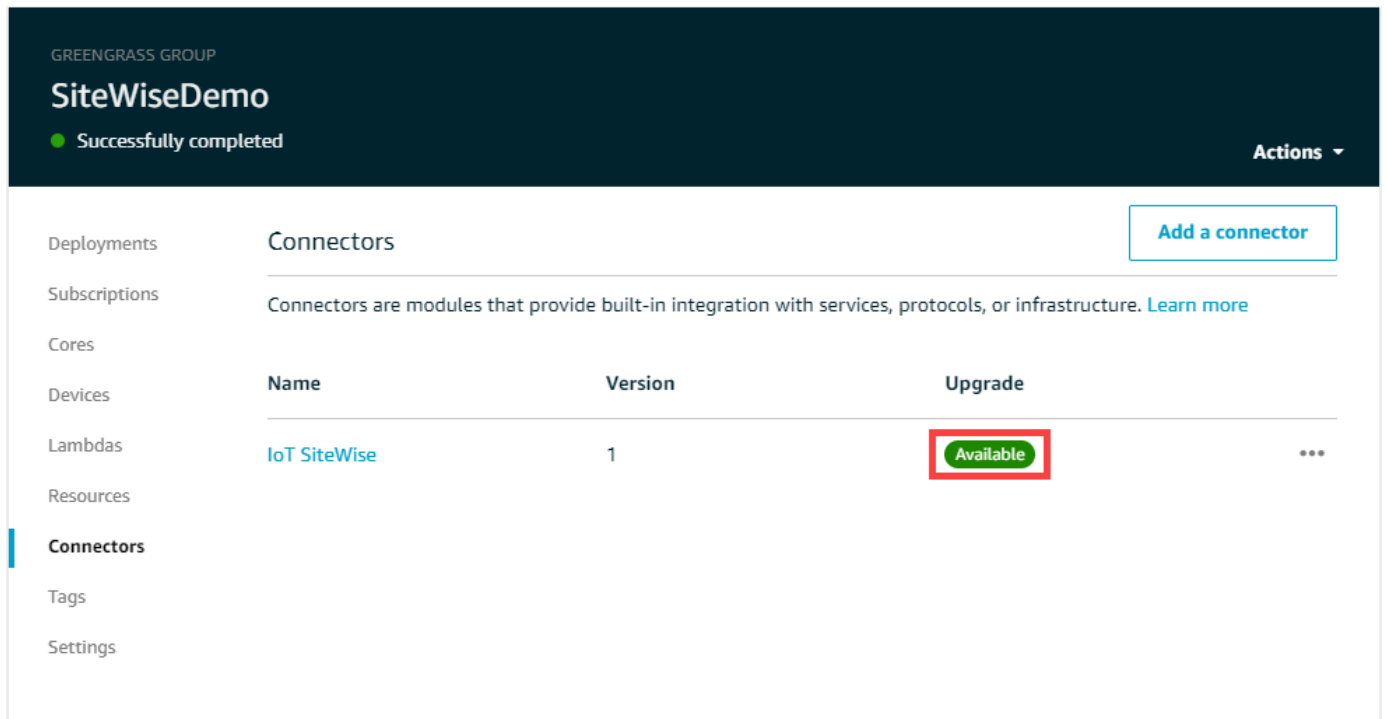
在此过程中，您将重新部署 Greengrass 群组并重启您的 Edge 网关。SiteWise 您的 SiteWise Edge 网关在重启时不会提取数据。重启 SiteWise Edge 网关的时间取决于 E SiteWise dge 网关源上的标签数量。重启时间可以从几秒钟（对于标签较少的 SiteWise Edge 网关）到几分钟（对于具有许多标签的 SiteWise Edge 网关）不等。

## 升级 IoT SiteWise 连接器

1. 导航到 [AWS IoT Greengrass 控制台](#)。
2. 在导航窗格中的 Greengrass 下，选择群组，然后选择您在设置 Edge 网关时创建的群组。  
SiteWise



3. 在导航窗格中，选择连接器。
4. 在“连接器”页面上，选择 IoT SiteWise 连接器旁边的“可用”。



如果没有看到 可用 元素，则连接器已经是最新版本。

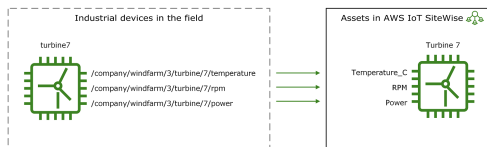
5. 在 升级连接器 页面上，输入连接器的参数，然后选择 升级。
6. 在右上角的 操作 菜单中，选择 部署。
7. 选择 自动检测 启动部署。

如果部署失败，请再次选择 Deploy (部署)。如果部署仍然失败，请参阅 [AWS IoT Greengrass 部署故障排除](#)。

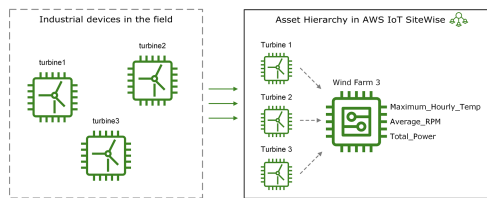
## 工业资产建模

您可以使用 AWS IoT SiteWise 资产创建工业运营的虚拟表示。资产代表将一个或多个数据流上传到的设备、设备或进程。AWS Cloud 例如，资产设备可以是风力涡轮机，它将空气温度、螺旋桨旋转速度和功率输出时间序列测量结果发送到 AWS IoT SiteWise 中的资产属性。

每个数据流对应一个唯一的属性别名。例如，别名 `/company/windfarm/3/turbine/7/temperature` 唯一标识来自风电场 #3 的涡轮机 #7 的温度数据流。您可以将 AWS IoT SiteWise 资产配置为使用数学表达式转换传入的测量数据，例如将温度数据从摄氏度转换为华氏度。



资产还可以表示设备的逻辑分组，如整个风电场。您可以将资产与其他资产相关联，来创建表示复杂工业运营的资产层次结构。资产可以访问其关联子资产中的数据。这样，您就可以使用 AWS IoT SiteWise 表达式来计算汇总指标，例如风力发电场的净功率输出。



您必须根据资产模型创建每项资产。资产模型是标准化资产格式的声明式结构。资产模型在相同类型的多个资产上强制执行一致的信息，以便您可以处理代表设备组的资产中的数据。在上图中，您对所有三个涡轮机使用相同的资产模型，因为所有涡轮机共享一组公有属性。

您也可以创建组件模型。组件模型是一种特殊类型的资产模型，您可以将其包含在资产模型或其他组件模型中。您可以使用组件模型来定义在多个资产模型之间共享的常见可重复使用的子组件，例如传感器、电机等。

定义资产模型后，您可以创建工业资产。要创建资产，请选择 ACTIVE 资产模型以从该模型创建资产。然后，您可以填充特定于资产的信息，例如数据流别名和属性。在上图中，您可以从一个资产模型创建三个涡轮机资产，然后为每个涡轮机关联像 `/company/windfarm/3/turbine/7/temperature` 一样的数据流别名。

您还可以更新和删除现有资源、资产模型和组件模型。更新资产模型时，基于该资产模型的每个资产都会反映您对基础模型所做的任何更改。更新组件模型时，这适用于基于引用该组件模型的每个资产模型的每项资产。



您的资产模型可能非常复杂，例如，在对具有许多子组件的复杂设备进行建模时。为了帮助保持此类资产模型的组织性和可维护性，您可以使用自定义复合模型对相关属性进行分组或重复使用共享组件。有关更多信息，请参阅 [自定义复合模型 \(组件\)](#)。

## 主题

- [资产和模型状态](#)
- [自定义复合模型 \(组件\)](#)
- [使用对象 ID](#)
- [创建资产模型和组件模型](#)
- [创建资产](#)
- [搜索资产](#)
- [将工业数据流映射到资产属性](#)
- [更新属性值](#)
- [关联资产和取消资产关联](#)
- [更新资产和模型](#)
- [删除资产和模型](#)
- [使用资产和模型进行批量操作](#)

## 资产和模型状态

创建、更新或删除资产、资产模型或组件模型时，更改需要一段时间才能传播。AWS IoT SiteWise 异步解析这些操作并更新每个资源的状态。每个资产、资产模型和组件模型都有一个状态字段，其中包含资源的状态和任何错误消息（如果适用）。状态可以是以下值之一：

- ACTIVE— 资源处于活动状态。这是您可以查询资产、资产模型和组件模型并与之交互的唯一状态。
- CREATING— 资源正在创建中。
- UPDATING— 资源正在更新中。
- DELETING— 资源正在被删除。
- PROPAGATING—（仅限资产模型和组件模型）更改将传播到所有依赖资源（从资产模型传播到资产，或从组件模型传播到资产模型）。
- FAILED— 资源在创建或更新操作期间未能验证，可能是由于表达式中存在循环引用。您可以删除处于 FAILED 状态的资源。

一些创建、更新和删除操作ACTIVE会将资产、资产模型或组件模型 AWS IoT SiteWise 置于操作解析之外的状态。要在执行其中一个操作后查询资源或与之交互，必须等到状态变为ACTIVE。否则，您的请求将失败。

## 主题

- [检查资产的状态](#)
- [检查资产模型或组件模型的状态](#)

## 检查资产的状态

您可以使用 AWS IoT SiteWise 控制台或 API 来检查资产的状态。

## 主题

- [检查资产的状态 \(控制台\)](#)
- [检查资产的状态 \(AWS CLI\)](#)

## 检查资产的状态 (控制台)

使用以下过程可以在 AWS IoT SiteWise 控制台中检查资产的状态。

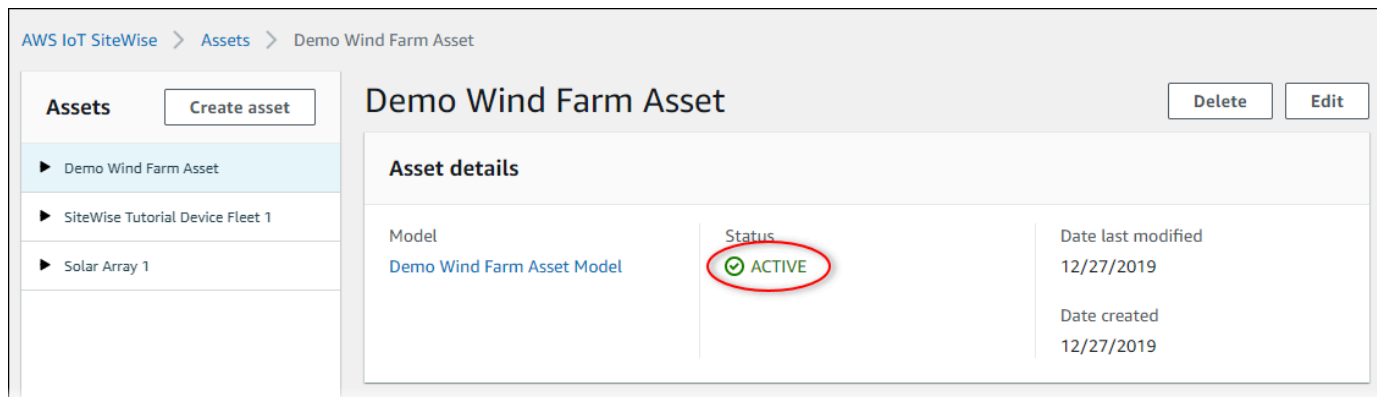
### 检查资产的状态 (控制台)

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 资产。
3. 选择要检查的资产。

#### Tip

您可以选择箭头图标来展开资产层次结构以查找资产。

4. 在 资源详细信息面板中查找 状态。



## 检查资产的状态 (AWS CLI)

您可以使用 AWS Command Line Interface (AWS CLI) 来检查资产的状态。

要检查资产的状态，请使用带 `assetId` 参数的 [DescribeAsset](#) 操作。

### 检查资产的状态 (AWS CLI)

- 运行以下命令以描述资产。将 `## ID` 替换为资产的 ID 或外部 ID。外部 ID 是用户定义的 ID。有关更多信息，请参阅《AWS IoT SiteWise 用户指南》中的 [使用外部 ID 引用对象](#)。

```
aws iotsitewise describe-asset --asset-id asset-id
```

此操作将返回包含资产详细信息的响应。响应包含一个具有以下结构的 `assetStatus` 对象：

```
{
  ...
  "assetStatus": {
    "state": "String",
    "error": {
      "code": "String",
      "message": "String"
    }
  }
}
```

资产的状态位于 JSON 对象的 `assetStatus.state` 中。

## 检查资产模型或组件模型的状态

您可以使用 AWS IoT SiteWise 控制台或 API 来检查资产模型或组件模型的状态。

### 主题

- [检查资产模型或组件模型的状态 \(控制台\)](#)
- [检查资产模型或组件模型的状态 \(AWS CLI\)](#)

### 检查资产模型或组件模型的状态 (控制台)

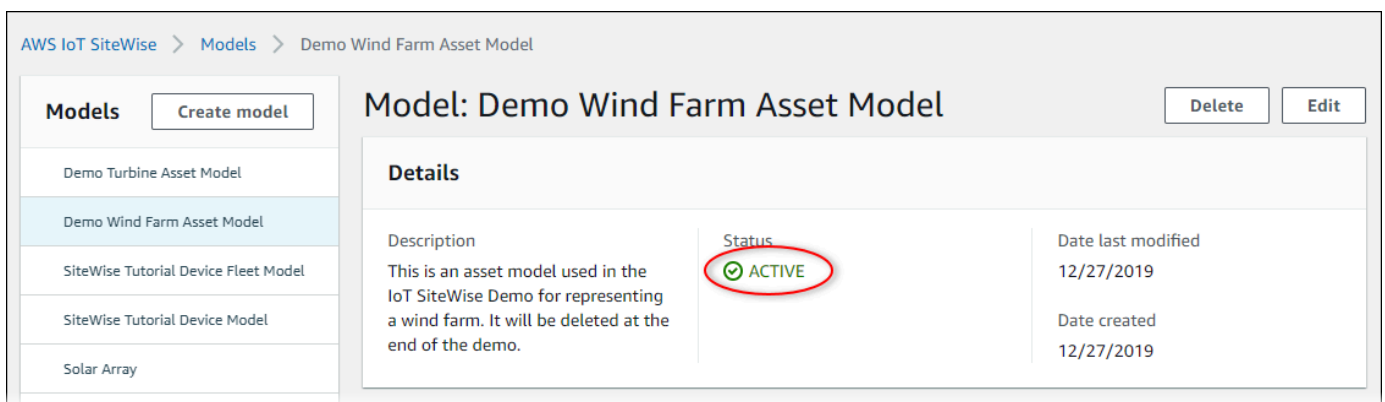
使用以下步骤在 AWS IoT SiteWise 控制台中检查资产模型或组件模型的状态。

#### Tip

资产模型和组件模型均列在导航窗格的模型下。所选资源模型或组件模型的“详细信息”面板会显示其类型。

### 检查资产模型或组件模型的状态 (控制台)

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 **模型**。
3. 选择要检查的型号。
4. 在详细信息面板中查找状态。



### 检查资产模型或组件模型的状态 (AWS CLI)

您可以使用 AWS CLI 来检查资产模型或组件模型的状态。

要检查资产模型或组件模型的状态，请使用带 `assetModelId` 参数的 [“DescribeAsset模型”](#) 操作。

### Tip

AWS CLI 将组件模型定义为一种资产模型。因此，您可以对两种类型的 [DescribeAsset模型](#) 使用相同的模型操作。响应中的 `assetModelType` 字段表示它是 `ASSET_MODEL` 还是 `COMPONENT_MODEL`。

## 检查资产模型或组件模型的状态 (AWS CLI)

- 运行以下命令来描述模型。将 `#### ID #####` 或组件模型的 ID 或外部 ID。外部 ID 是用户定义的 ID。有关更多信息，请参阅《AWS IoT SiteWise 用户指南》中的 [使用外部 ID 引用对象](#)。

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

该操作返回包含模型详细信息的响应。响应中包含具有以下结构的 `assetModelStatus` 对象。

```
{
  ...
  "assetModelStatus": {
    "state": "String",
    "error": {
      "code": "String",
      "message": "String"
    }
  }
}
```

模型的状态 `assetModelStatus.state` 在 JSON 对象中。

## 自定义复合模型（组件）

当你对特别复杂的工业资产（例如包含许多零件的复杂机器）进行建模时，保持资产模型的井然有序和可维护可能会成为一项挑战。

在这种情况下，您可以将自定义复合模型或组件（如果您使用的是控制台）添加到现有资产模型和组件模型中。它们通过对相关属性进行分组和重复使用子组件定义来帮助您保持井井有条。

有两种类型的自定义复合模型：

- 内联自定义复合模型定义了一组分组属性，这些属性适用于自定义复合模型所属的资产模型或组件模型。您可以使用它们对相关属性进行分组。它们由名称、描述和一组资产模型属性组成。它们不可重复使用。
- 基于组件模型的自定义复合模型引用要包含在资产模型或组件模型中的组件模型。您可以使用它们在模型中包含标准子组件。它们由名称、描述及其引用的组件模型的 ID 组成。它们没有自己的属性；引用的组件模型将其关联属性提供给任何创建的资产。

以下各节说明如何在设计中使用自定义复合模型。

主题

- [内联自定义复合模型](#)
- [Component-model-based 自定义复合模型](#)
- [使用路径引用自定义复合模型属性](#)

## 内联自定义复合模型

内联自定义复合模型提供了一种通过对相关属性进行分组来组织资产模型的方法。

例如，假设您要对机器人资产进行建模。该机器人包括伺服马达、电源和电池。这些组成部分中的每一个都有自己的属性，您要将其包含在模型中。您可以定义一个名为的资产模型`robot_model`，其属性如下所示。

- `robot_model`
  - `servo_status` ( 整数 )
  - `servo_position` ( 双 )
  - `powersupply_status` ( 整数 )
  - `powersupply_temperature` ( 双 )
  - `battery_status` ( 整数 )
  - `battery_charge` ( 双 )

但是，在某些情况下，可能有许多子组件，或者子组件本身可能具有许多属性。在这些情况下，可能有太多的属性，以至于在模型根目录的单个平面列表中引用和维护它们变得很麻烦，就像前面的示例一样。

要处理此类情况，您可以使用内联自定义复合模型对属性进行分组。内联自定义复合模型是定义其自身属性的自定义复合模型。例如，你可以像下面这样对机器人进行建模。

- robot\_model
  - servo
    - status ( 整数 )
    - position ( 双 )
  - powersupply
    - status ( 整数 )
    - temperature ( 双 )
  - battery
    - status ( 整数 )
    - charge ( 双 )

在前面的示例中servo，powersupply、和battery是在robot\_model资产模型中定义的内联自定义复合模型的名称。然后，这些复合模型中的每一个都定义了自己的属性。

#### Note

在这种情况下，每个自定义复合模型都定义了自己的属性，因此所有属性都是资产模型本身的一部分（在本例robot\_model中）。这些属性不会与任何其他资产模型或组件模型共享。例如，如果您创建了其他资产模型，其中也有一个名为的内联自定义复合模型servo，那么对servo内部进行更改robot\_model不会影响其他资产模型的servo定义。

如果要想实现这样的共享（例如，为伺服只有一个定义，所有资产模型都可以共享），则可以为其创建一个组件模型，然后创建参照该模型的基于组件模型的复合模型。有关详细信息，请参阅以下部分。

有关如何创建内联自定义复合模型的信息，请参见[创建自定义复合模型（组件）](#)。

## Component-model-based 自定义复合模型

可以在中创建元件模型 AWS IoT SiteWise 来定义标准的、可重复使用的子组件。创建组件模型后，可以在其他资产模型和组件模型中为其添加引用。为此，您可以将component-model-based 自定义复合模型添加到要引用该组件的任何模型中。您可以从多个模型中添加对元件的引用，也可以在同一个模型中多次添加引用。

通过这种方式，您可以避免在模型之间重复相同的定义。它还可以简化模型的维护，因为您对组件模型所做的任何更改都将反映在使用该模型的所有资产模型中。

例如，假设您的工业设备中有许多类型的设备都使用相同类型的伺服电机。其中一些在一台设备中就有许多伺服电机。您可以为每种设备类型创建资产模型，但不想servo每次都重复其定义。您只想对其建模一次，然后将其用于各种资产模型。如果您稍后更改的定义servo，则将在所有模型和资源中对其进行更新。

要以这种方式对前一个示例中的机器人进行建模，可以将伺服电动机、电源和电池定义为组件模型，如下所示。

- servo\_component\_model
  - status ( 整数 )
  - position ( 双 )
- powersupply\_component\_model
  - status ( 整数 )
  - temperature ( 双 )
- battery\_\_component\_model
  - status ( 整数 )
  - charge ( 双 )

然后，您可以定义引用这些组件的资产模型robot\_model，例如。多个资产模型可以引用同一个组件模型。您还可以在一个资产模型中多次引用同一个组件模型，例如，如果您的机器人中有多个伺服电机。



- robot\_model
  - servo1 ( 参考 : servo\_component\_model )
  - servo2 ( 参考 : servo\_component\_model )
  - servo3 ( 参考 : servo\_component\_model )
  - powersupply ( 参考 : powersupply\_component\_model )
  - battery ( 参考 : battery\_component\_model )

有关如何创建组件模型的信息，请参见[创建组件模型](#)。

有关如何在其他模型中引用您的组件模型的信息，请参见[创建自定义复合模型 \( 组件 \)](#)。

## 使用路径引用自定义复合模型属性

在资产模型、组件模型或自定义复合模型上创建属性时，可以从使用其值的其他属性（例如[变换](#)和[指标](#)）中引用该属性。

AWS IoT SiteWise 为您提供了不同的方式来引用您的房产。最简单的方法通常是使用其属性 ID。但是，如果您要引用的属性位于自定义复合模型上，则可能会发现改为按路径引用该属性会更有用。

路径是一个有序的路径段序列，它根据其在资产模型和复合模型中的嵌套复合模型中的位置来指定属性。

### 获取属性路径

你可以从属性的path字段中获取[AssetModel](#)属性的路径。

例如，假设您的资产模型robot\_model包含一个具有属性的自定义复合模型servoposition。如果您调用 [DescribeAssetModelCompositeModel](#)servo，则该position属性将列出一个如下所示的path字段：

```
"path": [  
  {  
    "id": "asset model ID",  
    "name": "robot_model"  
  },  
  {  
    "id": "composite model ID",  
    "name": "servo"  
  }  
]
```

```
    },  
    {  
      "id": "property ID",  
      "name": "position"  
    }  
  ]  
}
```

## 使用属性路径

在定义引用其他属性（例如转换或指标）的属性时，可以使用属性路径。

一个属性使用一个变量来引用另一个属性。有关使用变量的更多信息，请参阅[在公式表达式中使用变量](#)。

在定义变量来引用属性时，可以使用该属性的 ID 或其路径。

要定义使用被引用属性路径的变量，请指定其值的 `propertyPath` 字段。

例如，要定义具有通过路径引用属性的指标的资产模型，可以将这样的负载传递给[CreateAsset模型](#)：

```
{  
  ...  
  "assetModelProperties": [  
    {  
      ...  
      "type": {  
        "metric": {  
          ...  
          "variables": [  
            {  
              "name": "variable name",  
              "value": {  
                "propertyPath": [  
                  path segments  
                ]  
              }  
            },  
            ...  
          ]  
        }  
      },  
      ...  
    },  
    ...  
  ],  
}
```

```
    ...  
  ],  
  ...  
}
```

## 使用对象 ID

AWS IoT SiteWise 定义各种类型的永久对象，例如资产、资产模型、属性和层次结构。所有这些对象都有唯一的标识符，您可以使用这些标识符来检索、更新和删除它们。

AWS IoT SiteWise 为客户提供了不同的身份创建选项。AWS IoT SiteWise 默认情况下，在创建对象时会为您生成一个。您还可以为您的对象提供他们自己的 ID。

主题

- [使用对象 UUID](#)
- [使用外部 ID](#)

## 使用对象 UUID

中的每个永久对象都 AWS IoT SiteWise 有一个用于标识它的 [UUID](#)。例如，资产模型具有资产模型 ID，资产具有资产 ID，依此类推。此 ID 是在您创建对象时分配的，并且在对象的生命周期内保持不变。

创建新对象时，默认情况下会为您 AWS IoT SiteWise 生成一个唯一的 ID。您也可以在创建时以 UUID 格式提供自己的 ID。

### Note

UUID 在创建它的 AWS 区域内必须是全局唯一的，并且对于相同的对象类型。AWS IoT SiteWise 自动为您生成 ID 时，它始终是唯一的。如果您选择自己的身份证，请确保它是唯一的。

例如，如果您通过调用 Model 来创建新的资产 [CreateAsset模型](#)，则可以在请求的可选 `assetModelId` 字段中提供自己的 UUID。

相比之下，如果您在请求 `assetModelId` 中省略，则会为新的资产模型 AWS IoT SiteWise 生成一个 UUID。

## 使用外部 ID

要以 UUID 以外的某种格式定义您自己的 ID，可以分配一个外部 ID。例如，如果您重复使用系统中未使用的 ID，或者为了更便于人阅读 AWS，则可以这样做。外部 ID 的格式更加灵活。您可以在 AWS IoT SiteWise API 操作中使用它们来引用你的对象，否则你将使用 UUID。

与 UUID 一样，每个外部 ID 在其上下文中都必须是唯一的。例如，不能有两个具有相同外部 ID 的资产模型。此外，与 UUID 一样，一个对象在其生命周期中只能有一个外部 ID，这是无法更改的。

### 外部 ID 和 UUID 之间的区别

外部 ID 与 UUID 有以下区别：

- 每个对象都有一个 UUID，但外部 ID 是可选的。
- AWS IoT SiteWise 从不生成外部 ID。这些都是你自己提供的。
- 如果对象还没有 ID，则可以随时分配外部 ID。

### 外部 ID 的格式

有效的外部 ID 具有以下属性：

- 长度介于 2 到 128 个字符之间。
- 第一个和最后一个字符必须是字母数字 ( A-Z、a-z、0-9 )。
- 除第一个和最后一个以外的字符必须是字母数字，或者是以下字符之一：\_ - . :

例如，外部 ID 必须符合以下正则表达式：

```
[a-zA-Z0-9][a-zA-Z0-9_\-\.]*[a-zA-Z0-9]+
```

### 使用外部 ID 引用对象

在许多地方，你可以使用对象的 UUID 来引用对象，如果有的话，你可以改用它的外部 ID。为此，请将外部 ID 附加到字符串 `externalId:` 中。

例如，假设您有一个资产模型，其 UUID ( 资产模型 ID ) 为 `a1b2c3d4-5678-90ab-cdef-11111EXAMPLE`，并且还具外部 ID `myExternalId`。致电 [DescribeAsset 模特](#) 获取详细信息。您可以使用以下任一值作为的值 `assetModelId`：

- 使用资产模型 ID (UUID) 本身：`a1b2c3d4-5678-90ab-cdef-11111EXAMPLE`

- 使用外部 ID : externalId:myExternalId

```
aws iotsitewise describe-asset-model --asset-model-id a1b2c3d4-5678-90ab-
cdef-11111EXAMPLE
aws iotsitewise describe-asset-model --asset-model-id externalId:myExternalId
```

### Note

前externalId:缀本身不是外部 ID 的一部分。只有在为接受 UUID 或外部 ID 的 API 操作提供外部 ID 时，才需要提供前缀。例如，在查询或更新现有对象时提供前缀。在为对象定义外部 ID 时（例如创建资产模型时），请不要包含前缀。

您可以通过这种方式使用外部 ID 代替 UUID 来执行中的许多 API 操作 AWS IoT SiteWise，但不是全部。例如 [GetAssetPropertyValue](#)，必须使用 UUID；它不支持使用外部 ID。

要确定特定的 API 操作是否支持这种用法，请查阅 [API 参考](#)。

## 创建资产模型和组件模型

AWS IoT SiteWise 资产模型和组件模型可推动工业数据的标准化。资产模型或组件模型包含名称、描述、资产属性以及（可选）将属性组合在一起的自定义复合模型，或者引用子组件的组件模型。

- 您可以使用资产模型来创建资产。除了上面列出的功能外，资产模型还可以包含定义资产之间关系的层次结构定义。
- 组件模型表示资产模型或其他组件模型中的子组件。创建组件模型时，可以在资产模型和其他组件模型中为其添加引用。但是，您不能直接从组件模型创建资源。

创建资产模型或组件模型后，您可以为其创建自定义复合模型，以便将属性组合在一起或引用现有的组件模型。

有关如何创建资产模型和组件模型的详细信息，请参阅以下各节。

### 主题

- [创建资产模型](#)
- [创建组件模型](#)

- [定义数据属性](#)
- [创建自定义复合模型 \( 组件 \)](#)

## 创建资产模型

AWS IoT SiteWise 资产模型推动工业数据的标准化。资产模型包含名称、描述、资产属性和资产层次结构定义。例如，您可以使用温度、每分钟旋转次数 (RPM) 和功率属性来定义风力涡轮机模型。然后，可以使用净功率输出属性和风力涡轮机层次结构定义来定义风电场模型。

### Note

- 我们建议您从最低级别的节点开始对运营进行建模。例如，在创建风电场模型之前创建风力涡轮机模型。资产层次结构定义包含对现有资产模型的引用。通过这种方法，您可以在创建模型时定义资产层次结构。
- 资产模型不能包含其他资产模型。如果必须定义一个可以在另一个模型中作为子组件参考的模型，则应改为创建一个组件->模型。有关更多信息，请参阅 [创建组件模型](#)。

以下各节介绍如何使用 AWS IoT SiteWise 控制台或 API 创建资产模型。以及可用于创建模型的不同类型的资产属性和资产层次结构。

### 主题

- [创建资产模型 \( 控制台 \)](#)
- [创建资产模型 \(AWS CLI\)](#)
- [资产模型示例](#)
- [定义资产模型层次结构](#)

## 创建资产模型 ( 控制台 )


您可以使用 AWS IoT SiteWise 控制台创建资产模型。AWS IoT SiteWise 控制台提供各种功能，例如公式 auto 补全，可以帮助您定义有效的资产模型。

### 创建资产模型 ( 控制台 )

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 **模型**。

3. 选择创建模型。
4. 在创建模型页面上，执行以下操作：
  - a. 为资产模型输入名称，例如 **Wind Turbine** 或 **Wind Turbine Model**。此名称对于您账户在此区域的所有模型都必须是唯一的。
  - b. (可选) 为模型添加外部 ID。这是用户定义的 ID。有关更多信息，请参阅《AWS IoT SiteWise 用户指南》中的 [使用外部 ID 引用对象](#)。
  - c. (可选) 为模型添加测量值定义。测量值代表来自您的设备的数据流。有关更多信息，请参阅 [定义来自设备的数据流 \(测量\)](#)。
  - d. (可选) 为模型添加转换定义。转换是将数据从一种表单映射到另一种表单的公式。有关更多信息，请参阅 [转换数据 \(转换\)](#)。
  - e. (可选) 为模型添加指标定义。指标是按时间间隔汇总数据的公式。指标可以从关联资产的指标中输入数据，这样您就可以计算统计信息，从而提供对整个操作或部分操作的见解。有关更多信息，请参阅 [聚合来自属性和其他资产 \(指标\) 的数据](#)。
  - f. (可选) 为模型添加层次结构定义。层次结构是资产之间的关系。有关更多信息，请参阅 [定义资产模型层次结构](#)。
  - g. (可选) 为资产模型添加标签。有关更多信息，请参阅 [为资源添加 AWS IoT SiteWise 标签](#)。
  - h. 选择创建模型。


创建资产模型时，AWS IoT SiteWise 控制台会导航到新模型的页面。在此页面上，您可以查看模型的状态，最初为创建。此页面会自动更新，因此您可以等待模型的状态更新。

 Note

复杂模型的资产模型创建流程最多可能需要几分钟时间。在资产模型状态为活动后，您可以使用资产模型创建资产。有关更多信息，请参阅 [资产和模型状态](#)。

5. 或者，创建资产模型后，您可以将资产模型配置为边缘。有关 SiteWise Edge 的更多信息，请参阅 [启用边缘数据处理](#)。
  - a. 在模型页面上，选择针对边缘进行配置。
  - b. 在模型配置页面上，为您的模型选择边缘配置。这控制了 AWS IoT SiteWise 可以在何处计算和存储与此资产模型关联的属性。有关配置边缘模型的更多信息，请参阅 [the section called “设置边缘功能”](#)。

- c. 对于自定义边缘配置，请选择 AWS IoT SiteWise 要计算和存储每个资产模型属性的位置。

 Note

必须为同一位置配置关联的转换和指标。有关配置边缘模型的更多信息，请参阅 [the section called “设置边缘功能”](#)。

- d. 选择保存。在模型页面上，您的边缘配置现在应该为已配置。

## 创建资产模型 (AWS CLI)

您可以使用 AWS Command Line Interface (AWS CLI) 来创建资产模型。

使用 [CreateAsset模型](#) 操作创建具有属性和层次结构的资产模型。此操作需要具有以下结构的负载。

```
{
  "assetModelType": "ASSET_MODEL",
  "assetModelName": "String",
  "assetModelDescription": "String",
  "assetModelProperties": Array of AssetModelProperty,
  "assetModelHierarchies": Array of AssetModelHierarchyDefinition
}
```

## 创建资产模型 (AWS CLI)

1. 创建一个名为 `asset-model-payload.json` 的文件，然后将以下 JSON 对象复制到该文件中。

```
{
  "assetModelType": "ASSET_MODEL",
  "assetModelName": "",
  "assetModelDescription": "",
  "assetModelProperties": [

  ],
  "assetModelHierarchies": [

  ],
  "assetModelCompositeModels": [

  ]
}
```



```
}
```

2. 使用首选的 JSON 文本编辑器编辑 `asset-model-payload.json` 文件的以下内容：
  - a. 输入资产模型的名称 (`assetModelName`)，如 **Wind Turbine** 或 **Wind Turbine Model**。此名称在您账户中的所有资产模型和组件模型中必须是唯一的 AWS 区域。
  - b. (可选) 输入资产模型的外部 ID (`assetModelExternalId`)。这是用户定义的 ID。有关更多信息，请参阅《AWS IoT SiteWise 用户指南》中的 [使用外部 ID 引用对象](#)。
  - c. (可选) 输入资产模型的说明 (`assetModelDescription`)，或删除 `assetModelDescription` 键/值对。
  - d. (可选) 定义模型的资产属性 (`assetModelProperties`)。有关更多信息，请参阅 [定义数据属性](#)。
  - e. (可选) 定义模型的资产层次结构 (`assetModelHierarchies`)。有关更多信息，请参阅 [定义资产模型层次结构](#)。
  - f. (可选) 为模型定义警报。警报会监控其他特性，以便您可以识别何时需要注意设备或流程。每个警报定义都是一个复合模型 (`assetModelCompositeModels`)，用于标准化警报使用的属性集。有关更多信息，请参阅 [通过警报监控数据](#) 和 [在资产模型上定义警报](#)。
  - g. (可选) 为资产模型添加标签 (`tags`)。有关更多信息，请参阅 [为资源添加 AWS IoT SiteWise 标签](#)。
3. 运行以下命令，根据 JSON 文件中的定义创建资产模型。

```
aws iotsitewise create-asset-model --cli-input-json file://asset-model-payload.json
```

此操作将返回包含您在创建资产时引用的 `assetModelId` 的响应。响应中还包含模型的状态 (`assetModelStatus.state`)，最初为 `CREATING`。在传播更改之前，资产模型的状态为 `CREATING`。

#### Note

复杂模型的资产模型创建流程最多可能需要几分钟时间。要检查资产模型的当前状态，请使用 [DescribeAsset模型](#) 操作，方法是指定 `assetModelId`。在资产模型状态为 `ACTIVE` 后，您可以使用资产模型创建资产。有关更多信息，请参阅 [资产和模型状态](#)。

4. (可选) 为您的资产模型创建自定义复合模型。使用自定义复合模型，可以在模型内对属性进行分组，也可以通过引用组件模型来包含子组件。有关更多信息，请参阅 [创建自定义复合模型 \(组件\)](#)。

## 资产模型示例

本节包含资产模型定义示例，您可以使用这些示例使用 AWS CLI 和 AWS IoT SiteWise 软件开发工具包创建资产模型。这些资产模型代表风力涡轮机和风电场。风力涡轮机资产会摄取原始传感器数据并计算功率和平均风速等值。风电场资产计算诸如风电场中所有风力涡轮机的总功率之类的值。

### 主题

- [风力涡轮机资产模型](#)
- [风电场资产模型](#)

### 风力涡轮机资产模型

以下资产模型表示风电场中的涡轮机。风力涡轮机摄取传感器数据来计算功率和平均风速等值。

#### Note

此示例模型类似于 AWS IoT SiteWise 演示中的风力涡轮机模型。有关更多信息，请参阅 [使用演示 AWS IoT SiteWise 示](#)。

```
{
  "assetModelType": "ASSET_MODEL",
  "assetModelName": "Wind Turbine Asset Model",
  "assetModelDescription": "Represents a turbine in a wind farm.",
  "assetModelProperties": [
    {
      "name": "Location",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "Renton"
        }
      }
    },
    {
      "name": "Make",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "Amazon"
        }
      }
    }
  ]
}
```

```
    }
  }
},
{
  "name": "Model",
  "dataType": "INTEGER",
  "type": {
    "attribute": {
      "defaultValue": "500"
    }
  }
},
{
  "name": "Torque (KiloNewton Meter)",
  "dataType": "DOUBLE",
  "unit": "kNm",
  "type": {
    "measurement": {}
  }
},
{
  "name": "Wind Direction",
  "dataType": "DOUBLE",
  "unit": "Degrees",
  "type": {
    "measurement": {}
  }
},
{
  "name": "RotationsPerMinute",
  "dataType": "DOUBLE",
  "unit": "RPM",
  "type": {
    "measurement": {}
  }
},
{
  "name": "Wind Speed",
  "dataType": "DOUBLE",
  "unit": "m/s",
  "type": {
    "measurement": {}
  }
},
},
```

```
{
  "name": "RotationsPerSecond",
  "dataType": "DOUBLE",
  "unit": "RPS",
  "type": {
    "transform": {
      "expression": "rpm / 60",
      "variables": [
        {
          "name": "rpm",
          "value": {
            "propertyId": "RotationsPerMinute"
          }
        }
      ]
    }
  }
},
{
  "name": "Overdrive State",
  "dataType": "DOUBLE",
  "type": {
    "transform": {
      "expression": "gte(torque, 3)",
      "variables": [
        {
          "name": "torque",
          "value": {
            "propertyId": "Torque (KiloNewton Meter)"
          }
        }
      ]
    }
  }
},
{
  "name": "Average Power",
  "dataType": "DOUBLE",
  "unit": "Watts",
  "type": {
    "metric": {
      "expression": "avg(torque) * avg(rps) * 2 * 3.14",
      "variables": [
        {
```

```
        "name": "torque",
        "value": {
            "propertyId": "Torque (Newton Meter)"
        }
    },
    {
        "name": "rps",
        "value": {
            "propertyId": "RotationsPerSecond"
        }
    }
],
"window": {
    "tumbling": {
        "interval": "5m"
    }
}
}
},
{
    "name": "Average Wind Speed",
    "dataType": "DOUBLE",
    "unit": "m/s",
    "type": {
        "metric": {
            "expression": "avg(windspeed)",
            "variables": [
                {
                    "name": "windspeed",
                    "value": {
                        "propertyId": "Wind Speed"
                    }
                }
            ],
            "window": {
                "tumbling": {
                    "interval": "5m"
                }
            }
        }
    }
},
{
```

```

    "name": "Torque (Newton Meter)",
    "dataType": "DOUBLE",
    "unit": "Nm",
    "type": {
      "transform": {
        "expression": "knm * 1000",
        "variables": [
          {
            "name": "knm",
            "value": {
              "propertyId": "Torque (KiloNewton Meter)"
            }
          }
        ]
      }
    }
  },
  {
    "name": "Overdrive State Time",
    "dataType": "DOUBLE",
    "unit": "Seconds",
    "type": {
      "metric": {
        "expression": "statetime(overdrive_state)",
        "variables": [
          {
            "name": "overdrive_state",
            "value": {
              "propertyId": "Overdrive State"
            }
          }
        ]
      },
      "window": {
        "tumbling": {
          "interval": "5m"
        }
      }
    }
  }
],
"assetModelHierarchies": []
}

```

## 风电场资产模型

以下资产模型表示由多台风力涡轮机组成的风电场。该资产模型定义了风力涡轮机模型的[层次结构](#)。这允许风电场根据风电场中所有风力涡轮机的数据计算值（例如平均功率）。

### Note

此示例模型类似于 AWS IoT SiteWise 演示中的风力发电场模型。有关更多信息，请参阅 [使用演示 AWS IoT SiteWise 示](#)。

此资产模型依赖于[风力涡轮机资产模型](#)。将 `propertyId` 和 `childAssetModelId` 值替换为现有风力涡轮机资产模型中的对应值。

```
{
  "assetModelName": "Wind Farm Asset Model",
  "assetModelDescription": "Represents a wind farm.",
  "assetModelProperties": [
    {
      "name": "Code",
      "dataType": "INTEGER",
      "type": {
        "attribute": {
          "defaultValue": "300"
        }
      }
    },
    {
      "name": "Location",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "Renton"
        }
      }
    },
    {
      "name": "Reliability Manager",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "Mary Major"
        }
      }
    }
  ]
}
```

```

    }
  }
},
{
  "name": "Total Overdrive State Time",
  "dataType": "DOUBLE",
  "unit": "seconds",
  "type": {
    "metric": {
      "expression": "sum(overdrive_state_time)",
      "variables": [
        {
          "name": "overdrive_state_time",
          "value": {
            "propertyId": "ID of Overdrive State Time property in Wind Turbine
Asset Model",
            "hierarchyId": "Turbine Asset Model"
          }
        }
      ],
      "window": {
        "tumbling": {
          "interval": "5m"
        }
      }
    }
  }
},
{
  "name": "Total Average Power",
  "dataType": "DOUBLE",
  "unit": "Watts",
  "type": {
    "metric": {
      "expression": "sum(turbine_avg_power)",
      "variables": [
        {
          "name": "turbine_avg_power",
          "value": {
            "propertyId": "ID of Average Power property in Wind Turbine Asset
Model",
            "hierarchyId": "Turbine Asset Model"
          }
        }
      ]
    }
  }
}

```



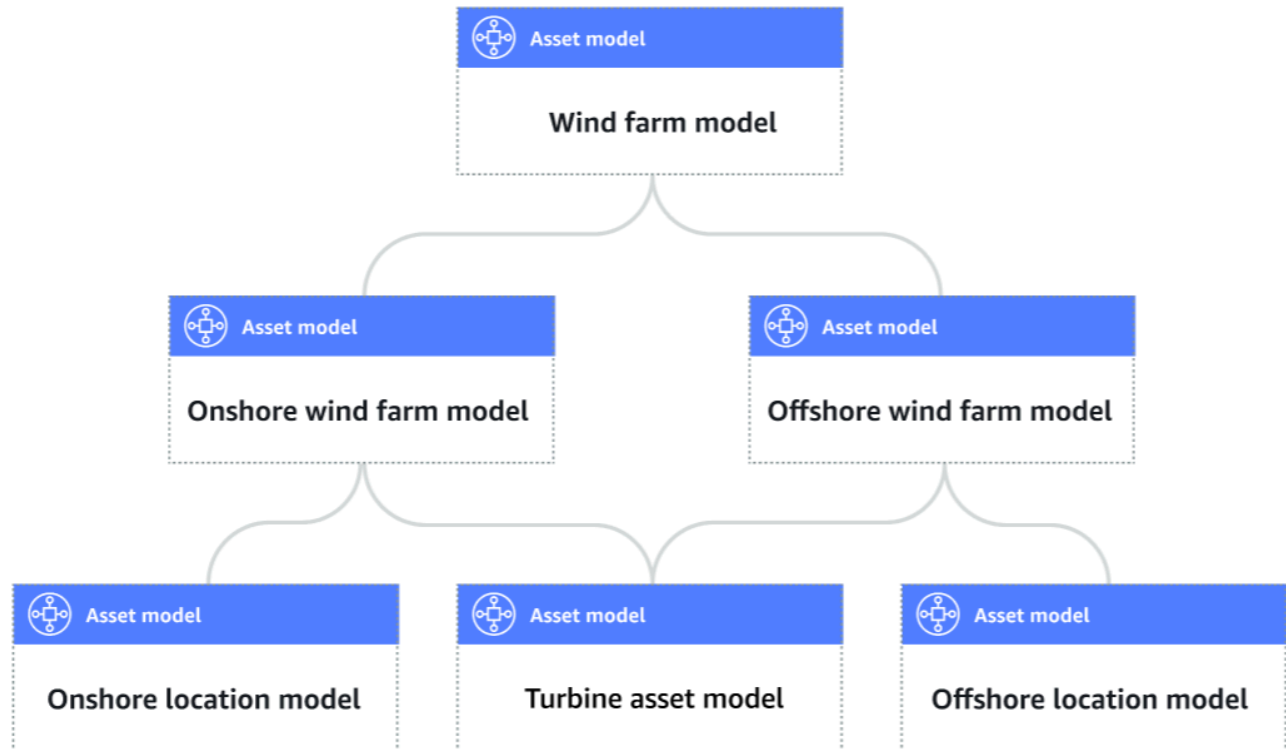
```
    ],
    "window": {
      "tumbling": {
        "interval": "5m"
      }
    }
  }
},
"assetModelHierarchies": [
  {
    "name": "Turbine Asset Model",
    "childAssetModelId": "ID of Wind Turbine Asset Model"
  }
]
}
```

## 定义资产模型层次结构

您可以定义资产模型层次结构，以便在工业运营中的资产模型之间创建逻辑关联。例如，您可以定义由陆上和海上风电场组成的风电场。陆上风电场包含涡轮机和陆上位置。海上风电场包含涡轮机和海上位置。



## Asset model hierarchy



当您通过层次将子资产模型与父资产模型关联时，父资产模型的指标可以从子资产模式的指标输入数据。您可以使用资产模型层次结构和指标来计算统计信息，从而提供对整个操作或部分操作的见解。有关更多信息，请参阅 [聚合来自属性和其他资产（指标）的数据](#)。

每个层次结构定义父资产模型和子资产模型之间的关系。在父资产模型中，您可以为同一个子资产模型定义多个层次结构。例如，如果您的风电场中有两种不同类型的风力涡轮机，其中所有风力涡轮机都由相同的资产模型表示，则可以为每种类型定义一个层次结构。然后，您可以在风电场模型中定义指标，以计算每种风力涡轮机的独立和组合统计数据。

一个父资产模型可以与多个子资产模型相关联。例如，如果您的陆上风电场和海上风电场由两种不同的资产模型表示，则可以将这些资产模型与相同的父风电场资产模型关联起来。

子资产模型也可以与多个父资产模型相关联。例如，如果您有两种不同类型的风电场，其中所有风力涡轮机都由相同的资产模型表示，则可以将风力涡轮机资产模型与不同的风电场资产模型相关联。

**Note**

定义资产模型层次结构时，子资产模型必须是 ACTIVE 或具有之前的 ACTIVE 版本。有关更多信息，请参阅 [资产和模型状态](#)。

定义分层资产模型并创建资产后，您可以关联资产以完成父子关系。有关更多信息，请参阅 [创建资产](#) 和 [关联资产和取消资产关联](#)。

**主题**

- [定义资产模型层次结构 \(控制台\)](#)
- [定义资产层次结构 \(\)AWS CLI](#)

**定义资产模型层次结构 (控制台)**

在 AWS IoT SiteWise 控制台中为资产模型定义层次结构时，需要指定以下参数：

- 层次结构名称 – 层次结构的名称，如 **Wind Turbines**。
- 层次模型 – 子资产模型。
- 层次结构外部 ID (可选) -这是用户定义的 ID。有关更多信息，请参阅《AWS IoT SiteWise 用户指南》中的 [使用外部 ID 引用对象](#)。

有关更多信息，请参阅 [创建资产模型 \(控制台\)](#)。

**定义资产层次结构 ()AWS CLI**

使用 AWS IoT SiteWise API 为资产模型定义层次结构时，需要指定以下参数：

- name – 层次结构的名称，如 **Wind Turbines**。
- childAssetModelId— 层次结构的子资产模型的 ID 或外部 ID。您可以使用 [ListAsset模型](#) 操作来查找现有资产模型的 ID。

**Example 层次结构定义示例**

以下示例演示一个表示风电场与风力涡轮机的关系的资产模型层次结构。此对象是 [AssetModel层次结构](#) 的示例。有关更多信息，请参阅 [创建资产模型 \(AWS CLI\)](#)。

```
{
  ...
  "assetModelHierarchies": [
    {
      "name": "Wind Turbines",
      "childAssetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
    },
  ],
]
}
```

## 创建组件模型

使用 AWS IoT SiteWise 组件模型定义子组件，您可以从资产模型或其他组件模型中引用这些子组件。这样，您就可以在多个其他模型中重复使用组件的定义，也可以在同一个模型中多次重复使用该组件的定义。

定义组件模型的过程与定义资产模型非常相似。与资产模型一样，组件模型也具有名称、描述和资产属性。但是，组件模型不能包含资产层次结构定义，因为组件模型本身不能用于直接创建资产。组件模型也无法定义警报。

例如，您可以使用电机温度、编码器温度和绝缘电阻属性为伺服电动机定义组件。然后，您可以为包含伺服电机的设备（例如数控机床）定义资产模型。

### Note

- 我们建议您从最低级别的节点开始对运营进行建模。例如，在创建 CNC 机床的资产模型之前，先创建伺服电机组件。资产模型包含对现有组件模型的引用。
- 您不能直接从组件模型创建资源。要创建使用您的组件的资产，必须为资产创建资产模型。然后，为其创建引用您的组件的自定义复合模型。有关创建资产模型的更多信息，请参阅[创建资产模型](#)。有关创建自定义复合模型的更多信息，请参阅[创建自定义复合模型 \(组件\)](#)。

以下各节介绍如何使用 AWS IoT SiteWise API 创建组件模型。

### 主题

- [创建组件模型 \(AWS CLI\)](#)
- [组件模型示例](#)

## 创建组件模型 (AWS CLI)

您可以使用 AWS Command Line Interface (AWS CLI) 来创建组件模型。

使用 “[CreateAsset模型](#)” 操作创建具有属性的组件模型。此操作需要具有以下结构的有效负载：

```
{
  "assetModelType": "COMPONENT_MODEL",
  "assetModelName": "String",
  "assetModelDescription": "String",
  "assetModelProperties": Array of AssetModelProperty,
}
```

## 创建组件模型 (AWS CLI)

1. 创建一个名为的文件 `component-model-payload.json`，然后将以下 JSON 对象复制到该文件中：

```
{
  "assetModelType": "COMPONENT_MODEL",
  "assetModelName": "",
  "assetModelDescription": "",
  "assetModelProperties": [

  ]
}
```

2. 使用首选的 JSON 文本编辑器编辑 `component-model-payload.json` 文件的以下内容：
  - a. 输入组件模型的名称 (`assetModelName`)，例如 **Servo Motor** 或 **Servo Motor Model**。此名称在您账户中的所有资产模型和组件模型中必须是唯一的 AWS 区域。
  - b. (可选) 输入组件模型的外部 ID (`assetModelExternalId`)。这是用户定义的 ID。有关更多信息，请参阅《AWS IoT SiteWise 用户指南》中的 [使用外部 ID 引用对象](#)。
  - c. (可选) 输入资产模型的说明 (`assetModelDescription`)，或删除 `assetModelDescription` 键/值对。
  - d. (可选) 为组件模型定义资产属性 (`assetModelProperties`)。有关更多信息，请参阅 [定义数据属性](#)。
  - e. (可选) 为资产模型添加标签 (`tags`)。有关更多信息，请参阅 [为资源添加 AWS IoT SiteWise 标签](#)。

3. 运行以下命令，根据 JSON 文件中的定义创建组件模型。

```
aws iotsitewise create-asset-model --cli-input-json file://component-model-payload.json
```

该操作会返回一个响应，其中 `assetModelId` 包含您在资产模型或其他组件模型中添加对组件模型的引用时所引用的响应。响应中还包含模型的状态 (`assetModelStatus.state`)，最初为 `CREATING`。在更改传播 `CREATING` 之前，组件模型的状态为。

#### Note

对于复杂的模型，元件模型的创建过程最多可能需要几分钟。要检查组件模型的当前状态，请使用 [DescribeAsset模型](#) 操作，方法是指定 `assetModelId`。组件模型状态变为后 `ACTIVE`，您可以在资产模型或其他组件模型中添加对组件模型的引用。有关更多信息，请参阅 [资产和模型状态](#)。

4. (可选) 为组件模型创建自定义复合模型。使用自定义复合模型，可以在模型内对属性进行分组，也可以通过引用另一个组件模型来包含子组件。有关更多信息，请参阅 [创建自定义复合模型 \(组件\)](#)。

## 组件模型示例

本节包含一个组件模型定义示例，您可以使用该示例使用 AWS CLI 和 AWS IoT SiteWise 软件开发工具包创建组件模型。此组件模型表示可以在其他设备（例如数控机床）中使用的伺服电动机。

### 主题

- [伺服电机组件模型](#)

### 伺服电机组件模型

以下组件模型表示可在数控机床等设备中使用的伺服电动机。伺服电机提供各种测量值，例如温度和电阻。这些测量值可用作从参考伺服电机组件模型的资产模型创建的资产的属性。

```
{
  "assetModelName": "ServoMotor",
  "assetModelType": "COMPONENT_MODEL",
  "assetModelProperties": [
    {
```

```
    "dataType": "DOUBLE",
    "name": "Servo Motor Temperature",
    "type": {
      "measurement": {}
    },
    "unit": "Celsius"
  },
  {
    "dataType": "DOUBLE",
    "name": "Spindle speed",
    "type": {
      "measurement": {}
    },
    "unit": "rpm"
  }
]
```

## 定义数据属性

资产属性是每个资产中包含资产数据的结构。资产属性可以是以下任一类型：

- 属性 – 通常为静态的资产特征，如设备制造商或地理区域。有关更多信息，请参阅 [定义静态数据 \(属性\)](#)。
- 测量值 – 资产的原始设备的传感器数据流，如时间戳转速值或以摄氏度为单位的时间戳温度值。测量值由数据流别名定义。有关更多信息，请参阅 [定义来自设备的数据流 \(测量\)](#)。
- 转换 – 资源转换后的时间序列值，如以华氏度为单位的时间戳温度值。转换由表达式和随该表达式一起使用的变量定义。有关更多信息，请参阅 [转换数据 \(转换\)](#)。
- 指标 – 在指定时间间隔内聚合的资产数据，如每小时平均温度。指示由时间间隔、表达式和随该表达式一起使用的变量定义。指标表达式可以输入关联资产的指标属性，这样您就可以计算出代表您的操作或操作子集的指标。有关更多信息，请参阅 [聚合来自属性和其他资产 \(指标\) 的数据](#)。

有关更多信息，请参阅 [创建资产模型](#)。

有关如何使用测量值、转换和指标计算整体设备效率 (OEE) 的示例，请参阅 [计算 OEE AWS IoT SiteWise](#)。

### 主题

- [定义静态数据 \(属性\)](#)

- [定义来自设备的数据流 \( 测量 \)](#)
- [转换数据 \( 转换 \)](#)
- [聚合来自属性和其他资产 \( 指标 \) 的数据](#)
- [使用公式表达式](#)

## 定义静态数据 ( 属性 )

资产属性表示通常为静态的信息，如设备制造商或地理位置。您从资产模型创建的每个资产都包含该模型的属性。

### 主题

- [定义属性 \( 控制台 \)](#)
- [定义属性 \(AWS CLI\)](#)

### 定义属性 ( 控制台 )

在 AWS IoT SiteWise 控制台中为资产模型定义属性时，需要指定以下参数：

- 名称 – 属性的名称。
- 默认值 – ( 可选 ) 此属性的默认值。从模型创建的资产具有此属性值。有关如何覆盖从模型创建的资产中的默认值的更多信息，请参阅[更新属性值](#)。
- 数据类型 – 属性的数据类型，如以下类型之一：
  - 字符串 – 最多 1024 字节的字符串。
  - 整数 – 有符号的 32 位整数，范围是 [-2,147,483,648, 2,147,483,647]。
  - 双精度 – 具有 [-10<sup>100</sup>, 10<sup>100</sup>] 范围和 IEEE 754 双精度的浮点数。
  - 布尔值 – true 或 false。
- 外部 ID – ( 可选 ) 这是用户定义的 ID。有关更多信息，请参阅《AWS IoT SiteWise 用户指南》中的[使用外部 ID 引用对象](#)。

有关更多信息，请参阅 [创建资产模型 \( 控制台 \)](#)。

### 定义属性 (AWS CLI)

使用 AWS IoT SiteWise API 为资产模型定义属性时，需要指定以下参数：

- name – 属性的名称。



- `defaultValue` – ( 可选 ) 此属性的默认值。从模型创建的资产具有此属性值。有关如何覆盖从模型创建的资产中的默认值的更多信息，请参阅[更新属性值](#)。
- `dataType` – 属性的数据类型，如以下类型之一：
  - `STRING` – 最多 1024 字节的字符串。
  - `INTEGER` – 有符号的 32 位整数，范围是 [-2,147,483,648, 2,147,483,647]。
  - `DOUBLE` – 具有 [-10<sup>100</sup>, 10<sup>100</sup>] 范围和 IEEE 754 双精度的浮点数。
  - `BOOLEAN` – `true` 或 `false`。
- `externalId` – ( 可选 ) 这是用户定义的 ID。有关更多信息，请参阅《AWS IoT SiteWise 用户指南》中的 [使用外部 ID 引用对象](#)。

### Example 属性定义示例

以下示例演示一个表示具有默认值的资产型号的属性。[此对象是包含属性的属性的示例](#)。AssetModel 您可以将此对象指定为 [CreateAsset 模型](#) 请求负载的一部分，以创建属性属性。有关更多信息，请参阅 [创建资产模型 \(AWS CLI\)](#)。

```
{
  ...
  "assetModelProperties": [
    {
      "name": "Model number",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "BLT123"
        }
      }
    }
  ],
  ...
}
```

### 定义来自设备的数据流 ( 测量 )

测量值表示设备的原始传感器数据流，如带有时间戳的温度值或带有时间戳的每分钟转数 (RPM) 值。

#### 主题

- [定义测量值 \( 控制台 \)](#)

- [定义测量 \(AWS CLI\)](#)

### 定义测量值 (控制台)

在 AWS IoT SiteWise 控制台中为资产模型定义测量值时，需要指定以下参数：

- 名称 – 属性的名称。
- 单位 – (可选) 属性的科学单位，例如毫米或摄氏度。
- 数据类型 – 属性的数据类型，如以下类型之一：
  - 字符串 – 最多 1024 字节的字符串。
  - 整数 – 有符号的 32 位整数，范围是 [-2,147,483,648, 2,147,483,647]。
  - 双精度 – 具有 [-10<sup>100</sup>, 10<sup>100</sup>] 范围和 IEEE 754 双精度的浮点数。
  - 布尔值 – true 或 false。
- 外部 ID – (可选) 这是用户定义的 ID。有关更多信息，请参阅《AWS IoT SiteWise 用户指南》中的 [使用外部 ID 引用对象](#)。

有关更多信息，请参阅 [创建资产模型 \(控制台\)](#)。

### 定义测量 (AWS CLI)

使用 AWS IoT SiteWise API 定义资产模型的衡量标准时，需要指定以下参数：

- name – 属性的名称。
- dataType – 属性的数据类型，如以下类型之一：
  - STRING – 最多 1024 字节的字符串。
  - INTEGER – 有符号的 32 位整数，范围是 [-2,147,483,648, 2,147,483,647]。
  - DOUBLE – 具有 [-10<sup>100</sup>, 10<sup>100</sup>] 范围和 IEEE 754 双精度的浮点数。
  - BOOLEAN – true 或 false。
- unit – (可选) 属性的科学单位，例如毫米或摄氏度。
- externalId – (可选) 这是用户定义的 ID。有关更多信息，请参阅《AWS IoT SiteWise 用户指南》中的 [使用外部 ID 引用对象](#)。

## Example 测量值定义示例

以下示例演示一个表示资产温度传感器读数的测量值。此对象是包含[测量值的AssetModel属性](#)的示例。您可以将此对象指定为[CreateAsset模型](#)请求负载的一部分，以创建测量属性。有关更多信息，请参阅[创建资产模型 \(AWS CLI\)](#)。

在定义资产模型时，[测量值](#)结构为空结构，因为您以后将每个资产配置为使用唯一的设备数据流。有关如何将资产的测量值属性连接到设备的传感器数据流的更多信息，请参阅[将工业数据流映射到资产属性](#)。

```
{
  ...
  "assetModelProperties": [
    {
      "name": "Temperature C",
      "dataType": "DOUBLE",
      "type": {
        "measurement": {}
      },
      "unit": "Celsius"
    }
  ],
  ...
}
```

## 转换数据 ( 转换 )

转换是将资产属性的数据点从一种形式映射到另一种形式的数学表达式。转换表达式由资产属性变量、文字、运算符和函数组成。变换后的数据点与输入数据点有关 one-to-one 系。AWS IoT SiteWise 每当任何输入属性接收到新的数据点时，都会计算一个新的转换数据点。

例如，如果您的资产具有以摄氏度为单位的名为 Temperature\_C 的温度测量流，您可以使用公式  $Temperature\_F = 9/5 * Temperature\_C + 32$  将每个数据点转换为华氏度。每次在 Temperature\_C 测量流中 AWS IoT SiteWise 接收到一个数据点时，相应的 Temperature\_F 值将在几秒钟内计算出来，并作为 Temperature\_F 属性提供。

如果您的转换包含多个变量，则较早到达的数据点会立即启动计算。举一个例子，其中零件制造商使用转换来监控产品质量。制造商根据零件类型使用不同的标准，使用以下测量值来表示过程：

- Part\_Number - 用于标识零件类型的字符串。
- Good\_Count - 一个整数，如果零件符合标准，则该整数将增加 1。

- `Bad_Count` - 一个整数，如果零件不符合标准，则该整数会增加 1。

制造商还创建了一个等于 `if(eq(Part_Number, "BLT123") and (Bad_Count / (Good_Count + Bad_Count) > 0.1), "Caution", "Normal")` 的转换 `Quality_Monitor`。

此转换可监控为特定零件类型生产的不良零件的百分比。如果部件号为 BLT123 且不良部件的百分比超过 10% (0.1)，则转换将返回 "Caution"。否则，转换将返回 "Normal"。

#### Note

- 如果 `Part_Number` 在其他测量之前收到新的数据点，则 `Quality_Monitor` 转换将使用新 `Part_Number` 值和最新的 `Good_Count` 和 `Bad_Count` 值。为避免错误，请在下次制造运行之前重置 `Good_Count` 和 `Bad_Count`。
- 如果您只想在所有变量都收到新的数据点后才计算表达式，请使用[指标](#)。

## 主题

- [定义转换 \(控制台\)](#)
- [定义变换 \(AWS CLI\)](#)

## 定义转换 (控制台)

在 AWS IoT SiteWise 控制台中为资产模型定义转换时，需要指定以下参数：

- 名称 - 属性的名称。
- 单位 - (可选) 属性的科学单位，例如毫米或摄氏度。
- 数据类型 - 转换的数据类型，可以是双精度或字符串。
- 外部 ID - (可选) 这是用户定义的 ID。有关更多信息，请参阅《AWS IoT SiteWise 用户指南》中的[使用外部 ID 引用对象](#)。
- 公式 - 转换表达式。转换表达式不能使用聚合函数或时间函数。要打开 auto complete 功能，请开始键入或按向下箭头键。有关更多信息，请参阅[使用公式表达式](#)。

**⚠ Important**

转换可以输入整型、双精度、布尔或字符串类型的属性。布尔值转换为 0 (假) 和 1 (真)。

转换必须输入一个或多个非特性属性。每次输入非特性属性接收新数据点时，AWS IoT SiteWise 都会计算一个新的转换数据点。新的属性值不会启动变换更新。资产属性数据 API 操作的请求速率同样适用于转换计算结果。

公式表达式只能输出双精度值或字符串值。嵌套表达式可以输出其他数据类型，例如字符串，但整个公式的计算结果必须为数字或字符串。您可以使用 [jp 函数](#) 将字符串转换为数字。布尔值必须为 1 (真) 或 0 (假)。有关更多信息，请参阅 [未定义值、无限值和溢出值](#)。

有关更多信息，请参阅 [创建资产模型 \(控制台\)](#)。

**定义变换 (AWS CLI)**

使用 AWS IoT SiteWise API 为资产模型定义转换时，需要指定以下参数：

- name – 属性的名称。
- unit – (可选) 属性的科学单位，例如毫米或摄氏度。
- dataType – 转换的数据类型，必须为 DOUBLE 或 STRING。
- externalId – (可选) 这是用户定义的 ID。有关更多信息，请参阅《AWS IoT SiteWise 用户指南》中的 [使用外部 ID 引用对象](#)。
- expression – 转换表达式。转换表达式不能使用聚合函数或时间函数。有关更多信息，请参阅 [使用公式表达式](#)。
- variables – 定义要在表达式中使用的资产的其他属性的变量列表。每个变量结构包含一个在表达式中使用的简单名称，以及一个指定哪个属性与该变量关联的 value 结构。value 结构包含以下信息：
  - propertyId – 要从中输入值的属性的 ID。您可以使用属性的名称而不是其 ID。

**⚠ Important**

转换可以输入整型、双精度、布尔或字符串类型的属性。布尔值转换为 0 (假) 和 1 (真)。

转换必须输入一个或多个非特性属性。每次输入非特性属性接收新数据点时，AWS IoT SiteWise 都会计算一个新的转换数据点。新的属性值不会启动变换更新。资产属性数据 API 操作的请求速率同样适用于转换计算结果。

公式表达式只能输出双精度值或字符串值。嵌套表达式可以输出其他数据类型，例如字符串，但整个公式的计算结果必须为数字或字符串。您可以使用 [jp 函数](#) 将字符串转换为数字。布尔值必须为 1 (真) 或 0 (假)。有关更多信息，请参阅 [未定义值、无限值和溢出值](#)。

## Example 转换定义

以下示例演示了将资产的温度测量数据从摄氏度转换为华氏度的转换属性。此对象是包含[变换的AssetModel属性](#)的示例。您可以将此对象指定为[CreateAsset模型](#)请求负载的一部分，以创建转换属性。有关更多信息，请参阅 [创建资产模型 \(AWS CLI\)](#)。

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "Temperature F",
      "dataType": "DOUBLE",
      "type": {
        "transform": {
          "expression": "9/5 * temp_c + 32",
          "variables": [
            {
              "name": "temp_c",
              "value": {
                "propertyId": "Temperature C"
              }
            }
          ]
        }
      },
      "unit": "Fahrenheit"
    }
  ],
  ...
}
```

## Example 包含三个变量的转换定义

以下示例演示了一个转换属性，如果超过 10% 的 BLT123 部件不符合标准，则该属性会返回警告消息 ("Caution")。否则，它将返回一条信息消息 ("Normal")。

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "Quality_Monitor",
      "dataType": "STRING",
      "type": {
        "transform": {
          "expression": "if(eq(Part_Number,\"BLT123\") and (Bad_Count / (Good_Count +
Bad_Count) > 0.1), \"Caution\", \"Normal\")",
          "variables": [
            {
              "name": "Part_Number",
              "value": {
                "propertyId": "Part Number"
              }
            },
            {
              "name": "Good_Count",
              "value": {
                "propertyId": "Good Count"
              }
            },
            {
              "name": "Bad_Count",
              "value": {
                "propertyId": "Bad Count"
              }
            }
          ]
        }
      }
    }
  ]
}
...
}
```

## 聚合来自属性和其他资产 ( 指标 ) 的数据

指标是使用聚合函数处理所有输入数据点并按指定时间间隔输出单个数据点的数学表达式。例如，指标可以从温度数据流计算每小时平均温度。

指标可以从关联资产的指标中输入数据，这样您就可以计算统计信息，从而提供对整个操作或部分操作的见解。例如，指标可以计算风电场中所有风力涡轮机的每小时平均温度。有关如何定义资产之间的关联的更多信息，请参阅[定义资产模型层次结构](#)。

指标也可以输入来自其他属性的数据，而无需在每个时间间隔内聚合数据。如果您在公式中指定了[属性](#)，则 AWS IoT SiteWise 在计算公式时会使用该属性的[最新](#)值。如果您在公式中指定指标，则 AWS IoT SiteWise 使用[最后](#)一个值作为计算公式的时间间隔。这意味着您可以定义  $OEE = Availability * Quality * Performance$  这样的指标，其中 Availability、Quality 和 Performance 都是同一资产模型上的其他指标。

AWS IoT SiteWise 还会为所有资产属性自动计算一组基本聚合指标。要降低计算成本，您可以使用这些聚合，而不是为基本计算定义自定义指标。有关更多信息，请参阅[查询资产属性聚合](#)。

### 主题

- [定义指标 \( 控制台 \)](#)
- [定义指标 \(AWS CLI\)](#)

### 定义指标 ( 控制台 )

在 AWS IoT SiteWise 控制台中为资产模型定义指标时，需要指定以下参数：

- 名称 – 属性的名称。
- 数据类型 - 转换的数据类型，可以是双精度或字符串。
- 外部 ID- ( 可选 ) 这是用户定义的 ID。有关更多信息，请参阅《AWS IoT SiteWise 用户指南》中的[使用外部 ID 引用对象](#)。
- 公式 – 指标表达式。指标表达式可以使用[聚合函数](#)从属性中输入层次结构中所有关联资产的数据。开始键入或按向下箭头键以打开自动完成功能。有关更多信息，请参阅[使用公式表达式](#)。

#### Important

指标只能输入整型、双精度、布尔或字符串类型的属性。布尔值转换为 0 ( 假 ) 和 1 ( 真 )。



如果您在指标表达式中定义任何指标输入变量，这些输入必须与输出指标具有同一个时间间隔。

公式表达式只能输出双精度值或字符串值。嵌套表达式可以输出其他数据类型，例如字符串，但整个公式的计算结果必须为数字或字符串。您可以使用 [jp 函数](#) 将字符串转换为数字。布尔值必须为 1 (真) 或 0 (假)。有关更多信息，请参阅 [未定义值、无限值和溢出值](#)。

- 时间间隔 – 时间间隔指标。AWS IoT SiteWise 支持以下滚动窗口时间间隔，其中每个间隔在前一个间隔结束时开始：
  - 1 分钟 – 1 分钟，在每分钟结束时计算 (凌晨 12:00:00、凌晨 12:01:00、凌晨 12:02:00，依此类推)。
  - 5 分钟 – 5 分钟，在每 5 分钟结束时计算，从整点开始 (中午 12:00:00、中午 12:05:00、中午 12:10:00，依此类推)。
  - 15 分钟 – 15 分钟，在每 15 分钟结束时计算，从整点开始 (凌晨 12:00:00、凌晨 12:15:00、凌晨 12:30:00，依此类推)。
  - 1 小时 – 1 小时 (60 分钟)，在每小时结束时计算，采用 UTC 格式 (凌晨 12:00:00、凌晨 01:00:00、凌晨 02:00:00，依此类推)。
  - 1 天 – 1 天 (24 小时)，在每天结束时计算，采用 UTC 格式 (星期一凌晨 12:00:00、星期二凌晨 12:00:00，依此类推)。
  - 1 周 – 1 周 (7 天)，在每个星期日结束时计算，采用 UTC 格式 (每个星期一凌晨 12:00:00)。
  - 自定义间隔 - 您可以输入一分钟到一周之间的任何时间间隔。
- 偏移日期 - 或者，聚合数据的参考日期。
- 偏移时间 - 或者，聚合数据的参考时间。偏移时间必须介于 00:00:00 和 23:59:59 之间。
- 偏移时区 - 或者，偏移的时区。如果未指定，默认偏移时区为通用协调时间 (UTC)。

### 支持的时区

- (UTC+ 00:00) 协调世界时间
- (UTC+ 01:00) 欧洲中部时间
- (UTC+ 02:00) 东欧
- (UTC+ 03:00) 东非时间
- (UTC+ 04:00) 近东时间
- (UTC+ 05:00) 巴基斯坦拉合尔时间
- (UTC+ 05:30) 印度标准时间

- (UTC+ 07:00) 越南标准时间
- (UTC+ 08:00) 中国台湾时间
- (UTC+ 09:00) 日本标准时间
- (UTC+ 09:30) 澳大利亚中部时间
- (UTC+ 10:00) 澳大利亚东部时间
- (UTC+ 11:00) 所罗门标准时间
- (UTC+12:00) 新西兰标准时间
- (UTC-11:00) 中途岛时间
- (UTC-10:00) 夏威夷标准时间
- (UTC-09:00) 阿拉斯加标准时间
- (UTC-08:00) 太平洋标准时间
- (UTC-07:00) 凤凰城标准时间
- (UTC-06:00) 中部标准时间
- (UTC-05:00) 东部标准时间
- (UTC-04:00) 波多黎各和美属维尔京群岛时间
- (UTC-03:00) 阿根廷标准时间
- (UTC-02:00) 南乔治亚时间
- (UTC-01:00) 中非时间

### Example 带偏移量的自定义时间间隔 (控制台)

以下示例向您展示如何定义 12 小时的时间间隔，偏移时间为 2021 年 2 月 20 日下午 6:30:30 (太平洋标准时间)。

#### 使用偏移量定义自定义间隔

1. 对于 时间间隔，选择自定义间隔。
2. 对于 时间间隔，执行以下操作之一：
  - 输入 **12**，然后选择 小时。
  - 输入 **720**，然后选择 分钟。
  - 输入 **43200**，然后选择 秒。

**⚠ Important**

无论单位如何，时间间隔都必须为整数。

3. 对于 偏移日期，选择 2021 年 2 月 20 日。
4. 对于 偏移时间，输入 **18:30:30**。
5. 对于 偏移时区，选择 (UTC-08:00) 太平洋标准时间。

如果您在 2021 年 7 月 1 日下午 06:30:30 ( 太平洋标准时间 ) 之前或之后创建指标，则第一个汇总结果将在 2021 年 7 月 1 日下午 06:30:30 ( 太平洋标准时间 ) 获得。第二个汇总结果是 2021 年 7 月 2 日上午 06:30:30 ( 太平洋标准时间 )，依此类推。

### 定义指标 (AWS CLI)

使用 AWS IoT SiteWise API 为资产模型定义指标时，需要指定以下参数：

- name – 属性的名称。
- dataType – 指标的数据类型，必须为 DOUBLE 或 STRING。
- externalId – ( 可选 ) 这是用户定义的 ID。有关更多信息，请参阅《AWS IoT SiteWise 用户指南》中的 [使用外部 ID 引用对象](#)。
- expression – 指标表达式。指标表达式可以使用 [聚合函数](#) 从属性中输入层次结构中所有关联资产的数据。有关更多信息，请参阅 [使用公式表达式](#)。
- window – 指标翻滚窗口的时间间隔和偏移量，其中每个间隔从前一个间隔结束时开始：
  - interval – 滚动窗口的时间间隔。间隔时间必须介于 1 分钟和 1 周之间。
  - offsets – 翻滚窗口的偏移。

有关更多信息，请参阅 AWS IoT SiteWise API 参考 [TumblingWindow](#) 中的。

### Example 带有偏移量 (AWS CLI) 的自定义时间间隔

以下示例向您展示如何定义 12 小时的时间间隔，偏移时间为 2021 年 2 月 20 日下午 06:30:30 ( 太平洋标准时间 )。

```
{
```

```

"window": {
  "tumbling": {
    "interval": "12h",
    "offset": " 2021-07-23T18:30:30-08"
  }
}
}

```

如果您在 2021 年 7 月 1 日下午 06:30:30 (太平洋标准时间) 之前或之后创建指标，则第一个汇总结果将在 2021 年 7 月 1 日下午 06:30:30 (太平洋标准时间) 获得。第二个汇总结果是 2021 年 7 月 2 日上午 06:30:30 (太平洋标准时间)，依此类推。

- **variables** – 定义要在表达式中使用的资产或子资产的其他属性的变量列表。每个变量结构包含一个在表达式中使用的简单名称，以及一个指定哪个属性与该变量关联的 **value** 结构。value 结构包含以下信息：
  - **propertyId** – 从中提取值的属性的 ID。如果属性是在当前模型中定义的 (而不是在层次结构的模型中定义的)，您可以使用属性的名称而不是其 ID。
  - **hierarchyId** – (可选) 从中查询属性的子资产的层次结构的 ID。您可以使用层次结构定义的名称而不是其 ID。如果省略此值，则会在当前模型中 AWS IoT SiteWise 找到该属性。

#### Important

指标只能输入整型、双精度、布尔或字符串类型的属性。布尔值转换为 0 (假) 和 1 (真)。

如果您在指标表达式中定义任何指标输入变量，这些输入必须与输出指标具有同一个时间间隔。

公式表达式只能输出双精度值或字符串值。嵌套表达式可以输出其他数据类型，例如字符串，但整个公式的计算结果必须为数字或字符串。您可以使用 [jp 函数](#) 将字符串转换为数字。布尔值必须为 1 (真) 或 0 (假)。有关更多信息，请参阅 [未定义值、无限值和溢出值](#)。

- **unit** – (可选) 属性的科学单位，例如毫米或摄氏度。

### Example 指标定义示例

以下示例演示一个指标属性，该属性可聚合资产的温度测量值数据，以计算每小时最高温度 (以华氏度为单位)。此对象是包含 [指标的 AssetModel 属性](#) 的示例。您可以将此对象指定为 [CreateAsset 模型](#) 请求负载的一部分，以创建指标属性。有关更多信息，请参阅 [创建资产模型 \(AWS CLI\)](#)。

```

{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "Max temperature",
      "dataType": "DOUBLE",
      "type": {
        "metric": {
          "expression": "max(temp_f)",
          "variables": [
            {
              "name": "temp_f",
              "value": {
                "propertyId": "Temperature F"
              }
            }
          ],
          "window": {
            "tumbling": {
              "interval": "1h"
            }
          }
        }
      },
      "unit": "Fahrenheit"
    }
  ],
  ...
}

```

### Example 输入关联资产数据的指标定义示例

以下示例演示了一个指标属性，该属性汇总了多台风力涡轮机的平均功率数据，以计算风电场的总平均功率。此对象是包含指标的[AssetModel属性](#)的示例。您可以将此对象指定为[CreateAsset模型](#)请求负载的一部分，以创建指标属性。

```

{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "Total Average Power",

```

```
    "dataType": "DOUBLE",
    "type": {
      "metric": {
        "expression": "avg(power)",
        "variables": [
          {
            "name": "power",
            "value": {
              "propertyId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
              "hierarchyId": "Turbine Asset Model"
            }
          }
        ],
        "window": {
          "tumbling": {
            "interval": "5m"
          }
        }
      },
      "unit": "kWh"
    },
    ...
  ]
}
```

## 使用公式表达式

通过使用公式表达式，您可以定义数学函数以转换和聚合原始工业数据，从而深入了解您的运营情况。公式表达式组合文字、运算符、函数和变量来处理数据。有关如何定义使用公式表达式的资产属性的更多信息，请参阅[转换数据（转换）](#)和[聚合来自属性和其他资产（指标）的数据](#)。转换和指标是公式属性。

### 主题

- [在公式表达式中使用变量](#)
- [在公式表达式中使用文字](#)
- [在公式表达式中使用运算符](#)
- [在公式表达式中使用常量](#)
- [在公式表达式中使用函数](#)
- [公式表达式教程](#)

## 在公式表达式中使用变量

变量表示公式表达式中的 AWS IoT SiteWise 资产属性。使用变量在表达式中输入来自其他资产属性的值，以便您可以处理来自常量属性 ([属性](#))、原始数据流 ([测量](#)) 和其他公式属性的数据。

变量可以表示来自同一资产模型或关联子资产模型的资产属性。只有指标公式才能输入子资产模型中的变量。

您可以在控制台和 API 中使用不同的名称来识别变量。

- AWS IoT SiteWise 控制台-在表达式中使用资产属性名称作为变量。
- AWS IoT SiteWise API (AWS CLI, AWS SDK)-使用[ExpressionVariable](#)结构定义变量，该结构需要变量名和对资产属性的引用。变量名称可包含大小写字母、数字和下划线。然后，使用变量名在表达式中引用资产属性。

变量名称区分大小写。

有关更多信息，请参阅[定义转换](#)和[定义指标](#)。

### 使用变量引用属性

变量的值定义了它所引用的属性。AWS IoT SiteWise 提供了不同的方法来执行此操作。

- 按属性 ID：您可以指定属性的唯一 ID (UUID) 来对其进行识别。
- 按名称：如果该属性属于同一资产模型，则可以在属性 ID 字段中指定其名称。
- 按路径：变量值可以通过其路径来引用属性。有关更多信息，请参阅[使用路径引用自定义复合模型属性](#)。

#### Note

AWS IoT SiteWise 控制台不支持变量。它们由 AWS IoT SiteWise API (包括 AWS Command Line Interface AWS CLI) 和 AWS 软件开发工具包使用。

您在响应中收到的变量 AWS IoT SiteWise 包含有关该值的完整信息，包括 ID 和路径。

但是，当您将变量传递到时 AWS IoT SiteWise (例如，在“创建”或“更新”调用中)，您只需要指定其中一个即可。例如，如果您指定路径，则无需提供 ID。

## 在公式表达式中使用文字

可以在公式表达式中定义数字和字符串文字。

- ### 数字

使用数字和科学记数法来定义整数和双精度。您可以使用 [E 记数法](#) 来表示带有科学记数法的数字。

示例：1、2.0、.9、-23.1、7.89e3、3.4E-5

- ### 字符串

使用 ' ( 引号 ) 和 " ( 双引号 ) 字符来定义字符串。起点和结尾的报价类型必须匹配。要对与您用来声明字符串的引号相匹配的引号进行转义，请将该引号字符包含两次。这是 AWS IoT SiteWise 字符串中唯一的转义字符。

示例：'active'、"inactive"、'{"temp": 52}'、"{""temp"": ""high""}"。

## 在公式表达式中使用运算符

您可以在公式表达式中使用以下常见的运算符。

运算符	描述
+	<p>如果两个操作数都是数字，则此运算符将左右操作数相加。</p> <p>如果任一操作数是字符串，则此运算符将左右操作数连接为字符串。例如，表达式的 <code>1 + 2 + " is three"</code> 计算结果为 <code>"3 is three"</code>。连接字符串最多可以包含 1024 个字符。如果字符串超过 1024 个字符，则 AWS IoT SiteWise 不会为该计算输出数据点。</p>
-	<p>从左侧操作数中减去右侧操作数。</p> <p>只能将此运算符与数字操作数一起使用。</p>
/	<p>左侧操作数除以右侧操作数。</p>



运算符	描述
	只能将此运算符与数字操作数一起使用。
*	将左右操作数相乘。 只能将此运算符与数字操作数一起使用。
^	将左操作数作为右操作数的幂（指数）。 只能将此运算符与数字操作数一起使用。
%	返回左侧操作数除以右侧操作数得到的余数。结果具有与左侧操作数相同的符号。此行为不同于 modulo 操作。 只能将此运算符与数字操作数一起使用。
$x < y$	如果 $x$ 小于 $y$ ，则返回 1，否则，返回 0。
$x > y$	如果 $x$ 大于 $y$ ，则返回 1，否则，返回 0。
$x \leq y$	如果 $x$ 小于或等于 $y$ ，则返回 1，否则，返回 0。
$x \geq y$	如果 $x$ 大于或等于 $y$ ，则返回 1，否则，返回 0。
$x == y$	如果 $x$ 等于 $y$ ，则返回 1，否则，返回 0。
$x != y$	如果 $x$ 不等于 $y$ ，则返回 1，否则，返回 0。

运算符	描述
!x	<p>如果 x 等于 0 (假)，则返回 1，否则，返回 0。</p> <p>如果出现以下情况，则 x 的计算结果为假：</p> <ul style="list-style-type: none"><li>• x 是一个数字操作数，其计算结果为 0。</li><li>• x 计算结果为空字符串。</li><li>• x 计算结果为空数组。</li><li>• x 计算结果为 None。</li></ul>
x and y	<p>如果 x 的计算结果为 0 (假)，则返回 0。否则，返回 y 的计算结果。</p> <p>如果出现以下情况，则 x 或 y 的计算结果为假：</p> <ul style="list-style-type: none"><li>• x 或 y 是一个数字操作数，其计算结果为 0。</li><li>• x 或 y 的计算结果为空字符串。</li><li>• x 或 y 的计算结果为空数组。</li><li>• x 或 y 的计算结果为 None。</li></ul>
x or y	<p>如果 x 等于 1 (真)，则返回 1。否则，返回 y 的计算结果。</p> <p>如果出现以下情况，则 x 或 y 的计算结果为假：</p> <ul style="list-style-type: none"><li>• x 或 y 是一个数字操作数，其计算结果为 0。</li><li>• x 或 y 的计算结果为空字符串。</li><li>• x 或 y 的计算结果为空数组。</li><li>• x 或 y 的计算结果为 None。</li></ul>

运算符	描述
<code>not x</code>	<p>如果 <code>x</code> 等于 <code>0</code> (假), 则返回 <code>1</code>, 否则, 返回 <code>0</code>。</p> <p>如果出现以下情况, 则 <code>x</code> 的计算结果为假:</p> <ul style="list-style-type: none"><li>• <code>x</code> 是一个数字操作数, 其计算结果为 <code>0</code>。</li><li>• <code>x</code> 计算结果为空字符串。</li><li>• <code>x</code> 计算结果为空数组。</li><li>• <code>x</code> 计算结果为 <code>None</code>。</li></ul>
<code>[]</code> <code>s[index]</code>	<p>返回字符串 <code>s</code> 索引 <code>index</code> 上的字符。这等同于 Python 中的索引语法。</p> <p>Example 示例</p> <ul style="list-style-type: none"><li>• <code>"Hello!"[1]</code> 返回 <code>e</code>。</li><li>• <code>"Hello!"[-2]</code> 返回 <code>o</code>。</li></ul>

运算符	描述
<p data-bbox="115 306 152 342">[]</p> <p data-bbox="115 386 436 422"><code>s[start:end:step]</code></p>	<p data-bbox="829 226 1495 306">返回字符串 <code>s</code> 的片段。这等同于 Python 中的切片语法。此运算符具有以下参数：</p> <ul data-bbox="829 352 1507 743" style="list-style-type: none"> <li>• <code>start</code> – 或者，切片的包含起始索引。默认值为 <code>0</code>。</li> <li>• <code>end</code> – 或者，切片的独占结束索引。默认为字符串的长度。</li> <li>• <code>step</code> – 或者，切片中每个步骤要递增的数字。例如，您可以指定 <code>2</code> 返回一个包含所有其他字符的切片，或者指定 <code>-1</code> 反转切片。默认值为 <code>1</code>。</li> </ul> <p data-bbox="829 821 1435 903">您可以省略 <code>step</code> 参数以使用其默认值。例如，<code>s[1:4:1]</code> 等同于 <code>s[1:4]</code>。</p> <p data-bbox="829 947 1500 1029">参数必须是整数或 <a href="#">无</a> 常量。如果指定 <code>none</code>，则 AWS IoT SiteWise 使用该参数的默认值。</p> <p data-bbox="829 1073 1027 1108">Example 示例</p> <ul data-bbox="829 1155 1398 1478" style="list-style-type: none"> <li>• <code>"Hello!"[1:4]</code> 返回 <code>"ell"</code>。</li> <li>• <code>"Hello!"[:2]</code> 返回 <code>"He"</code>。</li> <li>• <code>"Hello!"[3:]</code> 返回 <code>"lo!"</code>。</li> <li>• <code>"Hello!"[:-4]</code> 返回 <code>"He"</code>。</li> <li>• <code>"Hello!"[::2]</code> 返回 <code>"Hlo"</code>。</li> <li>• <code>"Hello!"[::-1]</code> 返回 <code>"!olleH"</code>。</li> </ul>

## 在公式表达式中使用常量

您可以在表达式中使用以下常见的数学常数。所有常量均不区分大小写。

**Note**

如果定义的变量与常量同名，则该变量会覆盖该常量。

常量	描述
pi	数字 pi ( $\pi$ ) : 3.141592653589793
e	数字 e : 2.718281828459045
true	等同于数字 1。在中 AWS IoT SiteWise，布尔值转换为其等效数字。
false	等同于数字 0。在中 AWS IoT SiteWise，布尔值转换为其等效数字。
none	等同于无值。您可以使用此常量将任何内容输出为 <a href="#">条件表达式</a> 的结果。

### 在公式表达式中使用函数

您可以使用以下函数对公式表达式中的数据进行操作。

转换和指标支持不同的功能。下表显示了哪些类型的函数与每种类型的公式属性兼容。

**Note**

一个公式表达式中最多可以包含 10 个函数。

函数类型	转换	指标
<a href="#">在公式表达式中使用常用函数</a>	 是	 是

函数类型	转换	指标
<a href="#">在公式表达式中使用比较函数</a>	 是	 是
<a href="#">在公式表达式中使用条件函数</a>	 是	 是
<a href="#">在公式表达式中使用字符串函数</a>	 是	 是
<a href="#">在公式表达式中使用聚合函数</a>	 否	 是
<a href="#">在公式表达式中使用时间函数</a>	 是	 是
<a href="#">在公式表达式中使用日期和时间函数</a>	 是	 是

## 函数语法

您可以使用以下语法来创建函数：

## 常规语法

在常规语法中，函数名称后面是带有零个或多个参数的圆括号。

*function\_name(argument1, argument2, argument3, ...)*。例如，使用常规语法的函数可能看起来像 `log(x)` 和 `contains(s, substring)`。

## 统一函数调用语法 (UFCS)

UFCS 允许您在面向对象编程中使用方法调用的语法来调用函数。在 UFCS 中，第一个参数后面跟着点 (`.`)，然后是函数名和括号内的其余参数（如果有）。

*argument1.function\_name(argument2, argument3, ...)*。例如，带有 UFCS 的函数可能看起来像 `x.log()` 和 `s.contains(substring)`。

您也可以使用 UFCS 来链接后续函数。AWS IoT SiteWise 使用当前函数的计算结果作为下一个函数的第一个参数。

例如，您可以使用 `message.jp('$ .status').lower().contains('fail')`，而不使用 `contains(lower(jp(message, '$ .status')), 'fail')`。

有关更多信息，请参阅 [D Programming Language](#) 页面。

### Note

您可以将 UFCS 用于所有 AWS IoT SiteWise 功能。  
AWS IoT SiteWise 函数不区分大小写。例如，可以互换使用 `lower(s)` 和 `Lower(s)`。

## 在公式表达式中使用常用函数

在[转换](#)和[指标](#)中，您可以使用以下函数来计算转换和指标中的常见数学函数。

函数	描述
<code>abs(x)</code>	返回 $x$ 的绝对值。
<code>acos(x)</code>	返回 $x$ 的反余弦值。
<code>asin(x)</code>	返回 $x$ 的正弦值。

函数	描述
<code>atan(x)</code>	返回 $x$ 的反正切值。
<code>cbrt(x)</code>	返回 $x$ 的立方根。
<code>ceil(x)</code>	返回大于 $x$ 的最接近整数。
<code>cos(x)</code>	返回 $x$ 的余弦值。
<code>cosh(x)</code>	返回 $x$ 的双曲余弦值。
<code>cot(x)</code>	返回 $x$ 的余切值。
<code>exp(x)</code>	返回 $e$ 的 $x$ 次幂。
<code>expm1(x)</code>	返回 $\exp(x) - 1$ 。对于较小的 $\exp(x) - 1$ 值，使用此函数可以更准确地计算 $x$ 。
<code>floor(x)</code>	返回小于 $x$ 的最接近整数。
<code>log(x)</code>	返回 $x$ 的 $\log_e$ (以 $e$ 为底)。
<code>log10(x)</code>	返回 $x$ 的 $\log_{10}$ (以 $10$ 为底)。
<code>log1p(x)</code>	返回 $\log(1 + x)$ 。对于较小的 $\log(1 + x)$ 值，使用此函数可以更准确地计算 $x$ 。
<code>log2(x)</code>	返回 $x$ 的 $\log_2$ (以 $2$ 为底)。
<code>pow(x, y)</code>	返回 $x$ 的 $y$ 次幂。这等同于 $x ^ y$ 。
<code>signum(x)</code>	返回 $x$ 的符号 ( $-1$ 表示负输入, $0$ 表示零输入, $+1$ 表示正输入 )。
<code>sin(x)</code>	返回 $x$ 的正弦值。
<code>sinh(x)</code>	返回 $x$ 的双曲正弦值。
<code>sqrt(x)</code>	返回 $x$ 的平方根。



函数	描述
<code>tan(x)</code>	返回 $x$ 的正切值。
<code>tanh(x)</code>	返回 $x$ 的双曲正切值。

### 在公式表达式中使用比较函数

在[变换](#)和[指标](#)中，您可以使用以下比较函数来比较两个值并输出1（真）或0（假）。AWS IoT SiteWise 按字典顺序比较字符串。

函数	描述
<code>gt(x, y)</code>	<p>如果 <math>x</math> 大于 <math>y</math>，则返回 1，否则，返回 0 (<math>x &gt; y</math>)。</p> <p>如果 <math>x</math> 和 <math>y</math> 的类型不兼容，例如数字和字符串，则此函数不返回值。</p>
<code>gte(x, y)</code>	<p>如果 <math>x</math> 大于或等于 <math>y</math>，则返回 1，否则，返回 0 (<math>x \geq y</math>)。</p> <p>AWS IoT SiteWise 如果参数在相对容差范围内，则认为它们是相等的<math>1E-9</math>。其行为类似于 Python 中的 <a href="#">isclose</a> 函数。</p> <p>如果 <math>x</math> 和 <math>y</math> 的类型不兼容，例如数字和字符串，则此函数不返回值。</p>
<code>eq(x, y)</code>	<p>如果 <math>x</math> 等于 <math>y</math>，则返回 1，否则，返回 0 (<math>x == y</math>)。</p> <p>AWS IoT SiteWise 如果参数在相对容差范围内，则认为它们是相等的<math>1E-9</math>。其行为类似于 Python 中的 <a href="#">isclose</a> 函数。</p> <p>如果 <math>x</math> 和 <math>y</math> 的类型不兼容，例如数字和字符串，则此函数不返回值。</p>

函数	描述
<code>lt(x, y)</code>	<p>如果 <math>x</math> 小于 <math>y</math>，则返回 1，否则，返回 0 (<math>x &lt; y</math>)。</p> <p>如果 <math>x</math> 和 <math>y</math> 的类型不兼容，例如数字和字符串，则此函数不返回值。</p>
<code>lte(x, y)</code>	<p>如果 <math>x</math> 小于或等于 <math>y</math>，则返回 1，否则，返回 0 (<math>x \leq y</math>)。</p> <p>AWS IoT SiteWise 如果参数在相对容差范围内，则认为它们是相等的 <math>1E-9</math>。其行为类似于 Python 中的 <a href="#">isclose</a> 函数。</p> <p>如果 <math>x</math> 和 <math>y</math> 的类型不兼容，例如数字和字符串，则此函数不返回值。</p>
<code>isnan(x)</code>	<p>如果 <math>x</math> 等于 NaN，则返回 1，否则，返回 0。</p> <p>如果 <math>x</math> 是字符串，则此函数不返回值。</p>

### 在公式表达式中使用条件函数

在[变换](#)和[指标](#)中，您可以使用以下函数来检查条件并返回不同的结果，无论该条件的计算结果为 true 还是 false。

函数	描述
<code>if(condition, result_if_true, result_if_false)</code>	<p>计算 <code>condition</code>，如果条件计算结果为真，则返回 <code>result_if_true</code>，如果条件计算结果为 false，则返回 <code>result_if_false</code>。</p> <p><code>condition</code> 必须是数字。此函数将 0 和一个空字符串视为 false，其他所有内容（包括 NaN）都视为 true。布尔值转换为 0（假）和 1（真）。</p>

函数	描述
	<p>您可以从此函数返回 <a href="#">无常量</a>，以丢弃特定条件的输出。这意味着您可以筛选出不符合条件的数据点。有关更多信息，请参阅 <a href="#">筛选数据点</a>。</p> <p>Example 示例</p> <ul style="list-style-type: none"> <li>• <code>if(0, x, y)</code> 返回变量 <code>y</code>。</li> <li>• <code>if(5, x, y)</code> 返回变量 <code>x</code>。</li> <li>• 如果变量 <code>temp</code> 大于 <code>300</code>，则 <code>if(gt(temp, 300), x, y)</code> 返回变量 <code>x</code>。</li> <li>• 如果 <code>if(gt(temp, 300), temp, none)</code> 大于或等于 <code>300</code>，则返回变量 <code>temp</code>，如果 <code>temp</code> 小于 <code>300</code>，则返回 <code>none</code> (无值)。</li> </ul> <p>我们建议您将 UFCS 用于嵌套条件函数，其中一个或多个参数是条件函数。您可以使用 <code>if(condition, result_if_true)</code> 计算条件和 <code>elif(condition, result_if_true, result_if_false)</code> 计算其他条件。</p> <p>例如，您可以使用 <code>if(condition1, result1_if_true).elif(condition2, result2_if_true, result2_if_false)</code>，而不使用 <code>if(condition1, result1_if_true, if(condition2, result2_if_true, result2_if_false))</code>。</p> <p>您还可以链接其他中间条件函数。例如，您可以使用 <code>if(condition1, result1_if_true).elif(condition2, result2_if_true).elif(condition3, result3_if_true, result3_i</code></p>

函数	描述
	<p>f_false) 而不是嵌套多个 if 语句，例如 <code>if(condition1, result1_if_true, if(condition2, result2_if_true, if(condition3, result3_if_true result3_if_false)))</code> 。</p> <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p><b>⚠ Important</b></p> <p>您必须将 <code>elif(condition, result_if_true, result_if_false)</code> 与 UFCS 一起使用。</p> </div>

### 在公式表达式中使用字符串函数

在[转换](#)和[指标](#)中，您可以使用以下函数对字符串进行操作。有关更多信息，请参阅[在公式中使用字符串](#)。

#### ⚠ Important

公式表达式只能输出双精度值或字符串值。嵌套表达式可以输出其他数据类型，例如字符串，但整个公式的计算结果必须为数字或字符串。您可以使用[jp 函数](#)将字符串转换为数字。布尔值必须为 1 (真) 或 0 (假)。有关更多信息，请参阅[未定义值、无限值和溢出值](#)。

函数	描述
<code>len(s)</code>	返回字符串 <code>s</code> 的长度。
<code>find(s, substring)</code>	返回字符串 <code>substring</code> 中字符串 <code>s</code> 的索引。
<code>contains(s, substring)</code>	如果字符串 <code>s</code> 包含字符串 <code>substring</code> ，则返回 1，否则，返回 0。
<code>upper(s)</code>	以大写形式返回字符串 <code>s</code> 。

函数	描述
lower(s)	以小写形式返回字符串 s。

函数	描述
jp(s, json_path)	<p>s使用<a href="#">JsonPath</a>表达式计算字符串json_path并返回结果。</p> <p>请使用该函数执行以下操作：</p> <ul style="list-style-type: none"><li>• 从序列化的JSON结构中提取值、数组或对象。</li><li>• 将字符串转换为数字。例如，公式jp('111', '\$')以数字形式返回111。</li></ul> <p>要从JSON结构中提取字符串值并将其作为数字返回，必须使用多个嵌套jp函数。外部jp函数从JSON结构中提取字符串，内部jp函数将字符串转换为数字。</p> <p>该字符串json_path必须包含一个字符串文字。这意味着json_path不能是计算结果为字符串的表达式。</p> <p>Example 示例</p> <ul style="list-style-type: none"><li>• jp('{ "status": "active", "value": 15 }', '\$.value') 返回15。</li><li>• jp('{ "measurement": { "reading": 25, "confidence": 0.95 } }', '\$.measurement.reading') 返回25。</li><li>• jp('[2,8,23]', '\$[2]') 返回23。</li><li>• jp('{ "values": [3,6,7] }', '\$.values[1]') 返回6。</li><li>• jp('111', '\$') 返回111。</li><li>• jp(jp('{ "measurement": { "reading": 25, "confidence": "0.95</li></ul>

函数	描述
<pre>join(s0, s1, s2, s3, ...)</pre>	<p>返回带有分隔符的串联字符串。此函数使用第一个输入字符串作为分隔符，并将剩余的输入字符串连接在一起。其行为类似于 Java 中的<a href="#">连接 (CharSequence 分隔符, CharSequence... 元素)</a> 函数。</p> <p>Example 示例</p> <ul style="list-style-type: none"> <li>• <code>join("-", "aa", "bb", "cc")</code> 返回 <code>aa-bb-cc</code></li> </ul>
<pre>format(expression: "format") 或 format("format", expression)</pre>	<p>返回指定格式的字符串。此函数将 <code>expression</code> 求值，然后以指定格式返回该值。其行为类似于 Java 中的<a href="#">format(String format, Object... args)</a> 函数。有关支持的格式的更多信息，请参阅 Java 平台标准版 7 API 规范中的<a href="#">类格式化程序</a>下的转换。</p> <p>Example 示例</p> <ul style="list-style-type: none"> <li>• <code>format(100+1: "d")</code> 返回字符串，<code>101</code>。</li> <li>• <code>format("The result is %d", 100+1)</code> 返回字符串，<code>The result is 101</code>。</li> </ul>

函数	描述
f'expression'	<p>返回一个连接的字符串。使用此格式化函数，您可以使用简单的表达式来连接和格式化字符串。这些函数可能包含嵌套表达式。可以使用 {} (大括号) 对表达式进行插值。这与 Python 中的 <a href="#">格式化字符串字面值</a> 类似。</p> <p>Example 示例</p> <ul style="list-style-type: none"> <li>f'abc{1+2: "f"}d' 返回 abc3.000000d 。要计算此示例表达式，请执行以下操作： <ol style="list-style-type: none"> <li>format(1+2: "f") 返回浮点数 3.000000。</li> <li>join(' ', "abc", 1+2, 'd') 返回字符串，abc3.000000d 。</li> </ol> <p>您也可以使用以下方式编写表达式：join(' ', "abc", format(1+2: "f"), 'd') 。</p> </li> </ul>

## 在公式表达式中使用聚合函数

仅在 [指标](#) 中，您可以使用以下函数以聚合每个时间间隔的输入值，并计算单个输出值。聚合函数可以从关联资产聚合数据。

聚合函数参数可以是 [变量](#)、[数字文字](#)、[时间函数](#)、嵌套表达式或聚合函数。公式 max(latest(x), latest(y), latest(z)) 使用聚合函数作为参数，并返回 x、y、和 z 属性的最大当前值。

可以在聚合函数中使用嵌套表达式。当您使用嵌套表达式时，以下规则将适用：

- 每个参数都只能有一个变量。

### Example

例如，支持 avg(x\*(x-1)) 和 sum(x/2 )/avg(y^2 )。

例如，不支持 min(x/y)。

- 每个参数可以有多个嵌套表达式。



## Example

例如，支持  $\text{sum}(\text{avg}(x^2)/2)$ 。

- 不同的参数可以有不同的变量。

## Example

例如，支持  $\text{sum}(x/2, y*2)$ 。

### Note

- 如果您的表达式包含测量值，则 AWS IoT SiteWise 使用当前时间间隔内测量值的最后一个值来计算聚合。
- 如果您的表达式包含属性，则 AWS IoT SiteWise 使用属性的最新值来计算聚合。

函数	描述
$\text{avg}(x_0, \dots, x_n)$	<p>返回给定变量在当前时间间隔内的平均值。</p> <p>仅当给定变量在当前时间间隔内至少有一个数据点时，此函数才会输出数据点。</p>
$\text{sum}(x_0, \dots, x_n)$	<p>返回给定变量在当前时间间隔内的值总和。</p> <p>仅当给定变量在当前时间间隔内至少有一个数据点时，此函数才会输出数据点。</p>
$\text{min}(x_0, \dots, x_n)$	<p>返回给定变量在当前时间间隔内的最小值。</p> <p>仅当给定变量在当前时间间隔内至少有一个数据点时，此函数才会输出数据点。</p>
$\text{max}(x_0, \dots, x_n)$	<p>返回给定变量在当前时间间隔内的最大值。</p> <p>仅当给定变量在当前时间间隔内至少有一个数据点时，此函数才会输出数据点。</p>

函数	描述
<code>count(x<sub>0</sub>, ..., x<sub>n</sub>)</code>	<p>返回给定变量在当前时间间隔内的数据点总数。有关如何计算满足条件的数据点数量的更多信息，请参阅<a href="#">对符合条件的数据点进行计数</a>。</p> <p>此函数会计算每个时间间隔的数据点。</p>
<code>stdev(x<sub>0</sub>, ..., x<sub>n</sub>)</code>	<p>返回给定变量在当前时间间隔内的标准差。</p> <p>仅当给定变量在当前时间间隔内至少有一个数据点时，此函数才会输出数据点。</p>

### 在公式表达式中使用时间函数

使用时间函数根据数据点的时间戳返回值。

### 在指标中使用时间函数

仅在[指标](#)中，您可以使用以下函数，这些函数根据数据点的时间戳返回值。

时态函数参数必须是来自本地资产模型或嵌套表达式的属性。这意味着您不能在时态函数中使用子资源模型中的属性。

可以在时态函数中使用嵌套表达式。当您使用嵌套表达式时，以下规则将适用：

- 每个参数都只能有一个变量。

例如，支持 `latest( t*9/5 + 32 )`。

- 参数不能是聚合函数。

例如，不支持 `first( sum(x) )`。

函数	描述
<code>first(x)</code>	返回给定变量在当前时间间隔内具有最早时间戳的值。

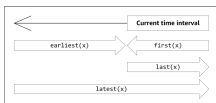
函数	描述
last(x)	返回给定变量在当前时间间隔内具有最晚时间戳的值。
earliest(x)	<p>返回给定变量在当前时间间隔开始之前的最后一个值。</p> <p>如果输入属性在其历史记录中具有至少一个数据点，则此函数将计算每个时间间隔的数据点。有关详细信息，请参阅 <a href="#">time-range-defintion</a>。</p>
latest(x)	<p>返回给定变量的最后一个值，其中包含当前时间间隔结束前的最新时间戳。</p> <p>如果输入属性在其历史记录中具有至少一个数据点，则此函数将计算每个时间间隔的数据点。有关详细信息，请参阅 <a href="#">time-range-defintion</a>。</p>
statetime(x)	<p>返回给定变量在当前时间间隔内为正值的的时间（以秒为单位）。您可以使用<a href="#">比较函数</a>创建转换属性，以供 statetime 函数使用。</p> <p>例如，如果 Idle 属性为 0 或 1，您可以使用以下表达式计算每个时间间隔的闲置时间：IdleTime = statetime(Idle)。有关更多信息，请参阅 <a href="#">statetime 使用情景示例</a>。</p> <p>此函数不支持将指标属性作为输入变量。</p> <p>如果输入属性在其历史记录中具有至少一个数据点，则此函数将计算每个时间间隔的数据点。</p>

函数	描述
<code>TimeWeightedAvg(x, [interpolation])</code>	<p>返回输入数据的平均值，使用点之间的时间间隔进行加权。</p> <p>有关计算和间隔的详细信息，请参阅<a href="#">时间加权函数参数</a>。</p> <p>可选参数 <code>interpolation</code> 必须是字符串常量：</p> <ul style="list-style-type: none"><li>• <code>locf</code> – 这是默认值。该计算使用上次观测到的向前结转计算算法计算数据点之间的间隔。在这种方法中，数据点是作为最后一个观测值计算的，直到下一个输入数据点的时间戳。</li></ul> <p>将一个良好的数据点之后的值外推为其值，直到下一个数据点的时间戳。</p> <ul style="list-style-type: none"><li>• <code>linear</code> – 计算使用线性插值计算算法计算数据点之间的间隔。</li></ul> <p>两个良好数据点之间的值是通过这些数据点的值之间的线性插值推断出来的。</p> <p>良好数据点和不良数据点之间的值或最后一个良好数据点之后的值，将被推断为一个良好数据点。</p>

函数	描述
<p><code>TimeWeightedStDev(x, [algo])</code></p>	<p>返回输入数据的标准差，使用点之间的时间间隔进行加权。</p> <p>有关计算和间隔的详细信息，请参阅<a href="#">时间加权函数参数</a>。</p> <p>该计算使用上次观测到的向前结转计算算法计算数据点之间的间隔。在这种方法中，数据点是作为最后一个观测值计算的，直到下一个输入数据点的时间戳。权重计算为数据点或窗口边界之间的时间间隔（以秒为单位）。</p> <p>可选参数 <code>algo</code> 必须是字符串常量：</p> <ul style="list-style-type: none"> <li>• <code>f</code> – 这是默认值。它返回带有频率权重的无偏加权样本方差，其中以秒 <code>TimeWeight</code> 为单位计算。这种算法通常是在标准差下假设的，被称为加权样本的贝塞尔标准差校正。</li> <li>• <code>p</code> – 返回有偏差的加权样本方差，也称为总体方差。</li> </ul> <p>以下公式用于计算，其中：</p> <ul style="list-style-type: none"> <li>• <math>S_p</math> = 总体标准差</li> <li>• <math>S_f</math> = 频率标准差</li> <li>• <math>X_i</math> = 传入数据</li> <li>• <math>\omega_i</math> = 等于以秒为单位的时间间隔的权重</li> <li>• <math>\mu^*</math> = 传入数据的加权平均值</li> </ul> <p>总体标准差方程：</p> $S_p^2 = \frac{\sum_{i=1}^N \omega_i (x_i - \mu^*)^2}{\sum_{i=1}^N \omega_i}$

函数	描述
	频率标准差方程： $S_f^2 = \frac{\sum_{i=1}^N \omega_i (x_i - \mu^*)^2}{\sum_{i=1}^N \omega_i - 1}$

下图显示了如何相对于当前时间间隔 AWS IoT SiteWise 计算时间函数 `firstlastearliestlatest`、`、` 和 `、`。



### Note

- 的时间 `first(x)` 范围 `last(x)` 是 ( 当前窗口开始, 当前窗口结束 ]。
- 的时间范围 `latest(x)` 是 ( 时间开始, 当前窗口结束 )。
- 的时间范围 `earliest(x)` 是 ( 时间开始, 上一个窗口结束 )。

## 时间加权函数参数

为聚合窗口计算的时间加权函数会考虑以下因素：

- 窗口内的数据点
- 数据点之间的时间间隔
- 窗口前的最后一个数据点
- 窗口之后的第一个数据点 ( 对于某些算法 )

术语：

- 不良数据点 - 任何质量不佳或非数字值的数据点。在窗口结果计算中不考虑这一点。
- 不良间隔 - 不良数据点之后的间隔。第一个已知数据点之前的间隔也被视为不良间隔。
- 良好数据点 - 任何具有良好质量和数值的数据点。

**Note**

- AWS IoT SiteWise 仅在计算变换和指标时才消耗GOOD质量数据。它会忽略 UNCERTAIN 和 BAD 数据点。
- 第一个已知数据点之前的间隔被视为不良间隔。请参阅[the section called “公式表达式教程”](#)了解更多信息。

最后一个已知数据点之后的间隔无限期地持续下去，影响所有后续窗口。当新的数据点到达时，该函数会重新计算间隔。

按照上述规则，将计算聚合窗口结果并仅限于窗口边界。默认情况下，只有当整个窗口都是良好间隔时，该函数才会发送窗口结果。

如果窗口良好间隔小于窗口长度，则该函数不会发送窗口。

当影响窗口结果的数据点发生变化时，即使数据点在窗口之外，该函数也会重新计算窗口。

如果输入属性的历史记录中至少有一个数据点，并且计算已启动，则该函数计算每个时间间隔的时间加权聚合函数。

**Example statetime 使用情景示例**

考虑这样一个使用情景示例，您有一个具有以下属性的资产：

- Idle – 一个为 0 或 1 的测量值。当该值为 1 时，机器处于空闲状态。
- Idle Time – 一个指标，该指标使用 `statetime(Idle)` 公式计算机器在每分钟内处于空闲状态的时间（以秒为单位）。

Idle 属性具有以下数据点。

Timestamp	2:00:00 PM	2:00:30 PM	2:01:15 PM	2:02:45 PM	2:04:00 PM
Idle	0	1	1	0	0

AWS IoT SiteWise 根据的值每分钟计算一次 Idle Time 属性 Idle。此计算完成后，Idle Time 属性将具有以下数据点。

Timestamp	2:00:00 PM	2:01:00 PM	2:02:00 PM	2:03:00 PM	2:04:00 PM
Idle Time	不适用	30	60	45	0

AWS IoT SiteWise 在每分钟结束 Idle Time 时执行以下计算。

- At 2:00 PM (for 1:59 PM to 2:00 PM)
  - 在 2:00 PM 之前没有 Idle 数据，因此计算不会得出任何数据点。
- At 2:01 PM (for 2:00 PM to 2:01 PM)
  - 在 2:00:00 PM，机器处于活动状态 (Idle 为 0)。
  - 在 2:00:30 PM，机器处于空闲状态 (Idle 为 1)。
  - Idle 在 2:01:00 PM 的时间间隔结束之前没有再次变化，因此 Idle Time 是 30 秒。
- At 2:02 PM (for 2:01 PM to 2:02 PM)
  - 在 2:01:00 PM，机器处于空闲状态 (基于 2:00:30 PM 的上一个数据点)。
  - 在 2:01:15 PM，机器仍处于空闲状态。
  - Idle 在 2:02:00 PM 的时间间隔结束之前没有再次变化，因此 Idle Time 是 60 秒。
- At 2:03 PM (for 2:02 PM to 2:03 PM)
  - 在 2:02:00 PM，机器处于空闲状态 (基于 2:01:15 PM 的上一个数据点)。
  - 在 2:02:45 PM，机器处于活动状态。
  - Idle 在 2:03:00 PM 的时间间隔结束之前没有再次变化，因此 Idle Time 是 45 秒。
- At 2:04 PM (for 2:03 PM to 2:04 PM)
  - 在 2:03:00 PM，机器处于活动状态 (基于 2:02:45 PM 的上一个数据点)。
  - Idle 在 2:04:00 PM 的时间间隔结束之前没有再次变化，因此 Idle Time 是 0 秒。

### Example 示例 TimeWeightedAvg 和 TimeWeightedStDev 场景

下表提供了这些一分钟窗口指标的示例输入和输出：Avg(x)，TimeWeightedAvg(x)，TimeWeightedAvg(x, "linear")，stDev(x)，timeWeightedStDev(x)，timeWeightedStDev(x, 'p')。

一分钟聚合窗口的示例输入：



**Note**

这些数据点都有GOOD质量。

03:00:00	4.0
03:01:00	2.0
03:01:10	8.0
03:01:50	20.0
03:02:00	14.0
03:02:05	10.0
03:02:10	3.0
03:02:30	20.0
03:03:30	0.0

汇总结果输出：

**Note**

无 – 未生成此窗口的结果。

时间	Avg(x)	TimeWeightedAvg(x)	TimeWeightedAvg(X, "linear")	stDev(X)	timeWeightedStDev(x)	timeWeightedStDev(x, 'p')
3:00:00	4	无	无	0	无	无
3:01:00	2	4	3	0	0	0

时间	Avg(x)	TimeWeightedAvg(x)	TimeWeightedAvg(X, "linear")	stDev(X)	timeWeightedStDev(x)	timeWeightedStDev(x, 'p')
3:02:00	14	9	13	6	5.4306100 41581775	5.3851648 07134504
3:03:00	11	13	12.875	8.5440037 4531753	7.7240544 37220943	7.6594168 62050705
3:04:00	0	10	2.5	0	10.084389 681792215	10
3:05:00	无	0	0	无	0	0

### 在转换中使用时间函数

仅在[转换](#)中，您可以使用 `pretrigger()` 函数在启动当前转换计算的属性更新之前，检索变量的 GOOD 质量值。

举一个制造商 AWS IoT SiteWise 用来监控机器状态的示例。制造商使用以下测量值和转换来表示过程：

- 测量值，`current_state`，可以是 0 或 1。
  - 如果本机处于清洁状态，则 `current_state` 等于 1。
  - 如果计算机处于制造状态，则 `current_state` 等于 0。
- 转换，`cleaning_state_duration`，等于 `if(pretrigger(current_state) == 1, timestamp(current_state) - timestamp(pretrigger(current_state)), none)`。此转换以 Unix 纪元格式返回机器处于清洁状态的时间（以秒为单位）。有关更多信息，请参阅[在公式表达式中使用条件函数 时间戳 \( \)](#) 函数。

如果机器处于清洁状态的时间超过预期，制造商可能会对机器进行调查。

您也可以在多变量转换中使用 `pretrigger()` 函数。例如，您有两个名为 `x` 和 `y` 的测量值和一个等于 `x + y + pretrigger(y)` 的转换 `z`。下表显示了 `x`、`y` 和 `z` 从上午 9:00 到 9:15 的值。

### Note

- 此示例假设测量值按时间顺序到达。例如，上午 09:00 x 的值在上午 09:05 x 的值之前到达。
- 如果上午 9:05 的数据点在上午 9:00 的数据点之前到达，则 z 不会在上午 9:05 进行计算。
- 如果上午 9:05 x 的值在上午 09:00 x 的值之前到达，并且 y 的值按时间顺序到达，则 z 在上午 9:05 等于  $22 = 20 + 1 + 1$ 。

	09:00 AM	09:05 AM	09:10 AM	09:15 AM
x	10	20		30
y	1	2	3	
$z = x + y + \text{pretrigger}(y)$	y 在上午 09:00 之前没有收到任何数据点。因此，z 不是在上午 09:00 计算的。	$23 = 20 + 2 + 1$ pretrigger(y) 等于 1。	$25 = 20 + 3 + 2$ x 没有收到新的数据点。 pretrigger(y) 等于 2。	$36 = 30 + 3 + 3$ y 没有收到新的数据点。因此，在上午 09:15 pretrigger(y) 等于 3。

### 在公式表达式中使用日期和时间函数

在[转换](#)和[指标](#)中，您可以通过以下方式使用日期和时间函数：

- 以 UTC 或本地时区检索数据点的当前时间戳。
- 使用参数（例如 year、month 和 day\_of\_month）构造时间戳。
- 使用 unix\_time 参数提取一个时间段，例如年或月。

函数	描述
now()	采用 Unix 纪元时间格式返回当前日期和时间（以秒为单位）。

函数	描述
timestamp()	<ul style="list-style-type: none"><li>在转换中，该函数采用 Unix 纪元时间格式的时间戳（以秒为单位）。</li></ul> <p>仅在转换中，您可以执行以下操作之一：</p> <ul style="list-style-type: none"><li>提供一个变量作为函数的参数。timestamp( <i>variable-name</i> ) 函数以 Unix 纪元格式返回指定变量的最新 GOOD 质量值的时间戳（以秒为单位）。</li></ul> <p>例如，如果您的资产有一个名为的转换属性 Temperature_F ，该属性使用 <math>9/5 * \text{Temperature\_C}</math> 公式将每个温度数据点从摄氏度转换为华氏度，则可以使用 timestamp(Temperature_F) 函数获取 Temperature_F 属性最新 GOOD 质量值的时间戳。</p> <ul style="list-style-type: none"><li>使用 pretrigger() 函数作为函数的参数。timestamp(pretrigger( <i>variable-name</i> )) 函数返回以 Unix 纪元格式启动当前转换计算的属性更新之前，指定变量的 GOOD 质量值的时间戳（以秒为单位）。有关更多信息，请参阅 <a href="#">在转换中使用时间函数</a>。</li></ul> <ul style="list-style-type: none"><li>在指标中，该函数以 Unix 纪元格式返回在当前窗口末尾检索到的时间戳（以秒为单位）。</li></ul>

函数	描述
<code>mktime(time_zone, year, month, day_of_month, hour, minute, second)</code>	<p>返回 Unix 纪元时间格式的输入时间（以秒为单位）。</p> <p>使用此函数存在以下要求：</p> <ul style="list-style-type: none"><li>• 时区参数必须是带引号的字符串 ('UTC')。如果未指定，默认时区为 UTC。</li></ul> <p>时区参数可以是第一个或最后一个参数。</p> <ul style="list-style-type: none"><li>• 年、月、月、日、时、分和秒参数必须按顺序排列。</li><li>• 需要年、月和日参数。</li></ul> <p>使用此函数存在以下限制：</p> <ul style="list-style-type: none"><li>• year - 有效值在 1970 到 2250 之间。</li><li>• month - 有效值在 1 到 12 之间。</li><li>• day-of-month - 有效值在 1 到 31 之间。</li><li>• hour - 有效值在 0 到 23 之间。</li><li>• minute - 有效值在 0 到 59 之间。</li><li>• second - 有效值在 0 到 60 之间。它可以是浮点数。</li></ul> <p>示例：</p> <ul style="list-style-type: none"><li>• <code>mktime(2020, 2, 29)</code></li><li>• <code>mktime('UTC+3', 2021, 12, 31, 22)</code></li><li>• <code>mktime(2022, 10, 13, 2, 55, 13.68, 'PST')</code></li></ul>

函数	描述
<code>localtime(unix_time, time_zone)</code>	<p>从 Unix 时间返回指定时区中的某年、一个月中的某天、一周中的某天、一年中的某天、小时、分钟或秒。</p> <p>使用此函数存在以下要求：</p> <ul style="list-style-type: none"> <li>• 时区参数必须是带引号的字符串 ('UTC')。如果未指定，默认时区为 UTC。</li> <li>• 采用 Unix 纪元时间格式是以秒为单位的时间（采用 Unix 纪元时间格式）。有效范围在 1-31556889864403199 之间。它可以是浮点数。</li> </ul> <p>响应示例：<code>2007-12-03T10:15:30+01:00[Europe/Paris]</code></p> <p><code>localtime(unix_time, time_zone)</code> 不是一个独立的函数。 <code>year()</code>、<code>mon()</code>、<code>mday</code>、<code>wday()</code>、<code>yday()</code>、<code>hour()</code> 和 <code>sec()</code> 函数采用 <code>localtime(unix_time, time_zone)</code> 作为参数。</p> <p>示例：</p> <ul style="list-style-type: none"> <li>• <code>year(localtime('GMT', 1605898608.8113723))</code></li> <li>• <code>now().localtime().year()</code></li> <li>• <code>timestamp().localtime('PST').year()</code></li> <li>• <code>localtime(1605289736, 'Europe/London').year()</code></li> </ul>
<code>year(localtime(unix_time, time_zone))</code>	<p>返回 <code>localtime(unix_time, time_zone)</code> 的年份。</p>

函数	描述
<code>mon(localtime(unix_time, time_zone))</code>	返回 <code>localtime(unix_time, time_zone)</code> 的月份。
<code>mday(localtime(unix_time, time_zone))</code>	返回 <code>localtime(unix_time, time_zone)</code> 一个月中的某天。
<code>wday(localtime(unix_time, time_zone))</code>	返回 <code>localtime(unix_time, time_zone)</code> 一周中的某天。
<code>yday(localtime(unix_time, time_zone))</code>	返回 <code>localtime(unix_time, time_zone)</code> 一年中的某天。
<code>hour(localtime(unix_time, time_zone))</code>	返回 <code>localtime(unix_time, time_zone)</code> 的某小时。
<code>minute(localtime(unix_time, time_zone))</code>	返回 <code>localtime(unix_time, time_zone)</code> 的某分钟。
<code>sec(localtime(unix_time, time_zone))</code>	返回 <code>localtime(unix_time, time_zone)</code> 的某秒。

## 支持的时区格式

您可以通过以下方式指定时区参数：

- 时区偏移量 - 指定 'Z' 表示 UTC 或偏移量 ('+2' 或 '-5')。
- ID 偏移量 - 结合时区缩写和偏移量。例如，'GMT+2' 和 'UTC-01:00'。时区缩写必须仅包含三个字母。
- 基于区域的 ID - 例如 'Etc/GMT+12' 和 'Pacific/Pago\_Pago'。

## 支持的时区缩写

日期和时间函数支持以下三个字母的时区缩写：

- 美国东部标准时间 (EST) - -05:00
- 美国夏威夷时间 (HST) - -10:00

- 北美山区时间 (MST) - -07:00
- ACT - 澳大利亚/达尔文
- AET - 澳大利亚/悉尼
- AGT- 美洲/阿根廷/布宜诺斯艾利斯
- ART - 非洲/开罗
- AST - 美洲/安克雷奇
- BET - 美洲/圣保罗
- BST - 亚洲/达卡
- CAT - 非洲/哈拉雷
- CET - 欧洲/巴黎
- CNT - 美洲/圣约翰斯
- CST - 美洲/芝加哥
- CTT - 亚洲/上海
- EAT - 非洲/亚的斯亚贝巴
- IET - 美洲/印第安纳州/印第安纳波利斯
- IST - 亚洲/加尔各答
- JST - 亚洲/东京
- MIT - 太平洋/阿皮亚
- NET - 亚洲/埃里温
- NST - 太平洋/奥克兰
- PLT - 亚洲/卡拉奇
- PRT - 美洲/波多黎各
- America/Los\_Angeles
- SST - 太平洋/瓜达尔卡纳尔岛
- VST - 亚洲/胡志明市

支持的基于区域的 ID

日期和时间函数支持以下基于区域的 ID，按其与 UTC+ 00:00 的关系进行组织：



- Etc/GMT+12 (UTC-12:00)
- 太平洋/帕果帕果 (UTC-11:00)
- 太平洋/萨摩亚 (UTC-11:00)
- 太平洋/纽埃岛 (UTC-11:00)
- 美国/萨摩亚 (UTC-11:00)
- Etc/GMT+11 (UTC-11:00)
- 太平洋/中途岛 (UTC-11:00)
- 太平洋/檀香山 (UTC-10:00)
- 太平洋/拉罗汤加岛 (UTC-10:00)
- 太平洋/塔希提岛 (UTC-10:00)
- 太平洋/约翰斯顿岛 (UTC-10:00)
- 美国/夏威夷 (UTC-10:00)
- SystemV/HST10 (UTC-10:00)
- Etc/GMT+10 (UTC-10:00)
- 太平洋/马克萨斯群岛 (UTC-09:30)
- Etc/GMT+9 (UTC-09:00)
- 太平洋/甘比尔 (UTC-09:00)
- 美国/阿特卡 (UTC-09:00)
- SystemV/YST9 (UTC-09:00)
- 美洲/埃达克岛 (UTC-09:00)
- 美国/阿留申 (UTC-09:00)
- Etc/GMT+8 (UTC-08:00)
- 美国/阿拉斯加 (UTC-08:00)
- 美洲/朱诺 (UTC-08:00)
- 美洲/梅特拉卡特拉 (UTC-08:00)
- 美洲/亚库塔特 (UTC-08:00)
- 太平洋/皮特凯恩岛 (UTC-08:00)
- 美洲/锡特卡 (UTC-08:00)

- 美洲/安克雷奇 (UTC-08:00)
- SystemV/PST8 (UTC-08:00)
- 美洲/诺姆 (UTC-08:00)
- SystemV/YST9YDT (UTC-08:00)
- 加拿大/育空地区 (UTC-07:00)
- 美国/太平洋-新 (UTC-07:00)
- Etc/GMT+7 (UTC-07:00)
- 美国/亚利桑那州 (UTC-07:00)
- 美国/道森克里克 (UTC-07:00)
- 加拿大/太平洋地区 (UTC-07:00)
- PST8PDT (UTC-07:00)
- SystemV/MST7 (UTC-07:00)
- 美洲/道森 (UTC-07:00)
- 墨西哥/ BajaNorte (世界标准时间 07:00)
- 美洲/蒂华纳 (UTC-07:00)
- 美洲/克雷斯頓 (UTC-07:00)
- 美洲/埃莫西约 (UTC-07:00)
- 美洲/圣伊莎贝尔 (UTC-07:00)
- 美洲/温哥华 (UTC-07:00)
- 美洲/恩塞纳达 (UTC-07:00)
- 美洲/凤凰城 (UTC-07:00)
- 美洲/怀特霍斯 (UTC-07:00)
- 美洲/纳尔逊堡 (UTC-07:00)
- SystemV/PST8PDT (UTC-07:00)
- 美洲/洛杉矶 (UTC-07:00)
- 美国/太平洋地区 (UTC-07:00)
- 美洲/萨尔瓦多 (UTC-06:00)
- 美洲/危地马拉 (UTC-06:00)
- 美洲/伯利兹 (UTC-06:00)

- 美洲/马那瓜 (UTC-06:00)
- 美洲/特古西加尔巴 (UTC-06:00)
- Etc/GMT+6 (UTC-06:00)
- 太平洋/复活节岛 (UTC-06:00)
- 墨西哥/ BajaSur (世界标准时间 06:00)
- 美洲/里贾纳 (UTC-06:00)
- 美洲/丹佛 (UTC-06:00)
- 太平洋/加拉帕戈斯群岛 (UTC-06:00)
- 美洲/耶洛奈夫 (UTC-06:00)
- 美洲/斯威夫特卡伦特 (UTC-06:00)
- 美洲/伊努维克 (UTC-06:00)
- 美洲/马萨特兰 (UTC-06:00)
- 美洲/博伊西 (UTC-06:00)
- 美洲/哥斯达黎加 (UTC-06:00)
- MST7MDT (UTC-06:00)
- SystemV/CST6 (UTC-06:00)
- 美洲/奇瓦瓦州 (UTC-06:00)
- 美洲/奥吉纳加 (UTC-06:00)
- 智利/ EasterIsland (世界标准时间 06:00)
- 美国/山区 (UTC-06:00)
- 美洲/埃德蒙顿 (UTC-06:00)
- 加拿大/山区 (UTC-06:00)
- 美洲/剑桥湾 (UTC-06:00)
- 纳瓦霍 (UTC-06:00)
- SystemV/MST7MDT (UTC-06:00)
- 加拿大/萨斯喀彻温省 (UTC-06:00)
- 美洲/希普罗克 (UTC-06:00)
- 美洲/巴拿马 (UTC-05:00)
- 美洲/芝加哥 (UTC-05:00)

- 美洲/埃鲁内佩 (UTC-05:00)
- Etc/GMT+5 (UTC-05:00)
- 墨西哥/整体 (UTC-05:00)
- 美洲/阿克里港 (UTC-05:00)
- 美洲/瓜亚基尔 (UTC-05:00)
- 美洲/兰京海口 (UTC-05:00)
- 美国/中部 (UTC-05:00)
- 美洲/雷尼河 (UTC-05:00)
- 美洲/印第安纳州/诺克斯 (UTC-05:00)
- 美洲/北达科他州/Beulah (UTC-05:00)
- 美洲/蒙特雷 (UTC-05:00)
- 美洲/牙买加 (UTC-05:00)
- 美洲阿蒂科肯 (UTC-05:00)
- 美洲/科勒尔港 (UTC-05:00)
- 美洲/北达科他州/中部 (UTC-05:00)
- 美洲/开曼群岛 (UTC-05:00)
- 美洲/印第安纳州/特尔城 (UTC-05:00)
- 美洲/墨西哥城 (UTC-05:00)
- 美洲/马塔莫罗斯 (UTC-05:00)
- CST6CDT (UTC-05:00)
- 美洲/诺克斯县 (UTC-05:00)
- 美洲/波哥大 (UTC-05:00)
- 美洲/梅诺米尼 (UTC-05:00)
- 美洲/雷索卢特 (UTC-05:00)
- SystemV/EST5 (UTC-05:00)
- 加拿大/中部 (UTC-05:00)
- 巴西/阿克里州 (UTC-05:00)
- 美洲/坎昆 (UTC-05:00)
- 美洲/利马 (UTC-05:00)

- 美洲/Bahia\_Banderas (UTC-05:00)
- 美国/印第安纳-斯塔克 (UTC-05:00)
- 美洲/里奥布朗库 (UTC-05:00)
- SystemV/CST6CDT (UTC-05:00)
- 牙买加 (UTC-05:00)
- 美洲/梅里达 (UTC-05:00)
- 美洲/北达科他州/新塞勒姆 (UTC-05:00)
- 美洲/温尼伯 (UTC-05:00)
- 美洲/库亚巴 (UTC-04:00)
- 美洲/马里戈特 (UTC-04:00)
- 美洲/印第安纳州/彼得斯堡 (UTC-04:00)
- 智利/大陆 (UTC-04:00)
- 美洲/大特克岛 (UTC-04:00)
- 古巴 (UTC-04:00)
- Etc/GMT+4 (UTC-04:00)
- 美洲/马瑙斯 (UTC-04:00)
- 美洲/韦恩堡 (UTC-04:00)
- 美洲/圣托马斯 (UTC-04:00)
- 美洲/安圭拉 (UTC-04:00)
- 美洲/哈瓦那 (UTC-04:00)
- 美国/密歇根州 (UTC-04:00)
- 美洲/巴巴多斯 (UTC-04:00)
- 美洲/路易斯维尔 (UTC-04:00)
- 美洲/库拉索 (UTC-04:00)
- 美洲/圭亚那 (UTC-04:00)
- 美洲/马提尼克岛 (UTC-04:00)
- 美洲/波多黎各 (UTC-04:00)
- 美洲/西班牙港 (UTC-04:00)
- SystemV/AST4 (UTC-04:00)

- 美洲/印第安纳州/Vevay (UTC-04:00)
- 美洲/印第安纳州/万森纳 (UTC-04:00)
- 美洲/克拉伦代克 (UTC-04:00)
- 美洲/安提瓜 (UTC-04:00)
- 美洲/印第安纳波利斯 (UTC-04:00)
- 美洲/伊卡卢伊特 (UTC-04:00)
- 美洲/圣文森特 (UTC-04:00)
- 美洲/肯塔基州/路易斯维尔 (UTC-04:00)
- 美洲/多米尼加 (UTC-04:00)
- 美洲/亚松森 (UTC-04:00)
- EST5EDT (UTC-04:00)
- 美洲/拿骚 (UTC-04:00)
- 美洲/肯塔基州/蒙蒂塞洛 (UTC-04:00)
- 巴西/西部 (UTC-04:00)
- 美洲/阿鲁巴 (UTC-04:00)
- 美洲/印第安纳州/印第安纳波利斯 (UTC-04:00)
- 美洲/圣地亚哥 (UTC-04:00)
- 美洲/拉巴斯 (UTC-04:00)
- 美洲/桑德贝 (UTC-04:00)
- 美洲/印第安纳州/马伦戈 (UTC-04:00)
- 美洲/巴朗克萨伯隆 (UTC-04:00)
- 美洲/圣多明哥 (UTC-04:00)
- 美国/东部 (UTC-04:00)
- 加拿大/东部 (UTC-04:00)
- 美洲/太子港 (UTC-04:00)
- 美洲/圣巴泰勒米 (UTC-04:00)
- 美洲/尼皮贡 (UTC-04:00)
- 美国/印第安纳州东部 (UTC-04:00)
- 美洲/圣露西亚 (UTC-04:00)

- 美洲/蒙特塞拉特 (UTC-04:00)
- 美洲/Lower\_Princes (UTC-04:00)
- 美洲/底特律 (UTC-04:00)
- 美洲/托尔托拉 (UTC-04:00)
- 美洲/波多韦 (UTC-04:00)
- 美洲/大坎普 (UTC-04:00)
- 美洲/弗吉尼亚 (UTC-04:00)
- 美洲/潘纳通 (UTC-04:00)
- 美洲/蒙特利尔 (UTC-04:00)
- 美洲/印第安纳州/威纳马克 (UTC-04:00)
- 美洲/博阿维斯塔 (UTC-04:00)
- 美洲/格林纳达 (UTC-04:00)
- 美洲/纽约 (UTC-04:00)
- 美洲/圣基茨 (UTC-04:00)
- 美洲/加拉加斯 (UTC-04:00)
- 美洲/瓜德罗普岛 (UTC-04:00)
- 美洲/多伦多 (UTC-04:00)
- SystemV/EST5EDT (UTC-04:00)
- 美洲/阿根廷/卡塔马卡 (UTC-03:00)
- 加拿大/大西洋地区 (UTC-03:00)
- 美洲/阿根廷/科尔多瓦 (UTC-03:00)
- 美洲/阿拉瓜伊纳 (UTC-03:00)
- 美洲/阿根廷/萨尔塔 (UTC-03:00)
- Etc/GMT+3 (UTC-03:00)
- 美洲/蒙得维的亚 (UTC-03:00)
- 巴西/东部 (UTC-03:00)
- 美洲/阿根廷/门多萨 (UTC-03:00)
- 美洲/阿根廷/里约热内卢加列戈斯 (UTC-03:00)
- 美洲/卡塔马卡 (UTC-03:00)

- 美洲/科尔多瓦 (UTC-03:00)
- 美洲/圣保罗 (UTC-03:00)
- 美洲/阿根廷/胡胡伊 (UTC-03:00)
- 美洲/卡宴 (UTC-03:00)
- 美洲/累西腓市 (UTC-03:00)
- 美洲/布宜诺斯艾利斯 (UTC-03:00)
- 美洲/帕拉马里博 (UTC-03:00)
- 美洲/蒙克顿 (UTC-03:00)
- 美洲/门多萨 (UTC-03:00)
- 美洲/圣塔伦 (UTC-03:00)
- 大西洋/百慕大 (UTC-03:00)
- 美洲/马塞约 (UTC-03:00)
- 大西洋/斯坦利 ( UTC-03:00 )
- 美洲/哈利法克斯 (UTC-03:00)
- 南极洲/罗瑟拉 (UTC-03:00)
- 美洲/阿根廷/圣路易斯 (UTC-03:00)
- 美洲/阿根廷/乌斯怀亚 (UTC-03:00)
- 南极洲/帕尔默 (UTC-03:00)
- 美洲/蓬塔阿雷纳斯 (UTC-03:00)
- 美洲/格莱斯贝 (UTC-03:00)
- 美洲/福塔雷萨 (UTC-03:00)
- 美洲/图勒岛 (UTC-03:00)
- 美洲/阿根廷/拉里奥哈 (UTC-03:00)
- 美洲/贝伦 (UTC-03:00)
- 美洲/胡胡伊 (UTC-03:00)
- 美洲/巴伊亚 (UTC-03:00)
- 美洲/古斯湾 (UTC-03:00)
- 美洲/阿根廷/圣胡安 (UTC-03:00)
- 美国/阿根廷/ (世界标准时间 03:00ComodRivadavia )



- 美洲/阿根廷/图库曼 (UTC-03:00)
- 美洲/罗萨里奥 (UTC-03:00)
- SystemV/AST4ADT (UTC-03:00)
- 美洲/阿根廷/布宜诺斯艾利斯 (UTC-03:00)
- 美洲/圣约翰斯 (UTC-02:30)
- 加拿大/纽芬兰 (UTC-02:30)
- 美洲/密克隆群岛 (UTC-02:00)
- Etc/GMT+2 (UTC-02:00)
- 美洲/戈特霍布 (UTC-02:00)
- 美洲/诺罗尼亚 (UTC-02:00)
- 巴西/ DeNoronha (世界标准时间 02:00)
- 大西洋/南乔治亚岛 (UTC-02:00)
- Etc/GMT+1 (UTC-01:00)
- 大西洋/佛得角 (UTC-01:00)
- 太平洋/基里巴斯圣诞岛 (UTC+14:00)
- Etc/GMT-14 (UTC+14:00)
- 太平洋/法考福环礁 (UTC+13:00)
- 太平洋/恩德伯里 (UTC+13:00)
- 太平洋/阿皮亚 (UTC+13:00)
- 太平洋/汤加塔布 (UTC+13:00)
- Etc/GMT-13 (UTC+13:00)
- NZ-CHAT (UTC+12:45)
- 太平洋/查塔姆群岛 (UTC+12:45)
- 太平洋/夸贾林 (UTC+12:00)
- 南极洲/ ( 世界标准时间+ 12 McMurdo :00 )
- 太平洋/瓦利斯 (UTC+12:00)
- 太平洋/斐济 (UTC+12:00)
- 太平洋/富纳富提 (UTC+12:00)
- 太平洋/瑙鲁 (UTC+12:00)

- 夸贾林岛 (UTC+ 12:00)
- 新西兰 (UTC+12:00)
- 太平洋/威克岛 (UTC+12:00)
- 南极洲/南极 (UTC+ 12:00)
- 太平洋/塔拉瓦 (UTC+12:00)
- 太平洋/奥克兰 (UTC+12:00)
- 亚洲/堪察加半岛 (UTC+12:00)
- Etc/GMT-12 (UTC+12:00)
- 亚洲/阿纳德尔 (UTC+12:00)
- 太平洋/马朱罗 (UTC+12:00)
- 太平洋/波纳佩岛 (UTC+11:00)
- 太平洋/布干维尔 (UTC+11:00)
- 南极洲/麦格理 (UTC+11:00)
- 太平洋/波纳佩岛 (UTC+11:00)
- 太平洋/埃法特岛 (UTC+11:00)
- 太平洋/诺福克 (UTC+11:00)
- 亚洲/马加丹 (UTC+11:00)
- 太平洋/科斯雷 (UTC+11:00)
- 亚洲/萨哈林 (UTC+11:00)
- 太平洋/努美阿 (UTC+11:00)
- Etc/GMT-11 (UTC+11:00)
- 亚洲/Srednekolymsk (UTC+11:00)
- 太平洋/瓜达尔卡纳尔岛 (UTC+11:00)
- 澳大利亚/豪勋爵 (UTC+ 10:30)
- 澳大利亚/LHI (UTC+10:30)
- 澳大利亚/霍巴特 (UTC+10:00)
- 太平洋/雅浦岛 (UTC+10:00)
- 澳大利亚/塔斯马尼亚州 (UTC+10:00)
- 太平洋/莫尔兹比港 (UTC+10:00)

- 澳大利亚/澳大利亚首都直辖区 (UTC+10:00)
- 澳大利亚/维多利亚州 (UTC+10:00)
- 太平洋/楚克岛 (UTC+10:00)
- 澳大利亚/昆士兰州 (UTC+10:00)
- 澳大利亚/堪培拉 (UTC+10:00)
- 澳大利亚/居里 (UTC+10:00)
- 太平洋/关岛 (UTC+10:00)
- 太平洋/特鲁克岛 (UTC+10:00)
- 澳大利亚/新南威尔士州 (UTC+10:00)
- 亚洲/符拉迪沃斯托克 (UTC+10:00)
- 太平洋/塞班岛 (UTC+10:00)
- 南极洲/杜蒙德维尔 (UTC+10:00)
- 澳大利亚/悉尼 (世界标准时间+ 10:00)
- 澳大利亚/布里斯班 (UTC+10:00)
- Etc/GMT-10 (UTC+10:00)
- 亚洲/乌斯季挪拉 (UTC+10:00)
- 澳大利亚/墨尔本 (UTC+10:00)
- 澳大利亚/利达曼 (UTC+10:00)
- 澳大利亚/北部 (UTC+09:30)
- 澳大利亚/Yancowinna (UTC+ 09:30)
- 澳大利亚/阿德莱德 (UTC+09:30)
- 澳大利亚/布罗肯希尔 (UTC+ 09:30)
- 澳大利亚/南部 (UTC+09:30)
- 澳大利亚/达尔文 (UTC+09:30)
- Etc/GMT-9 (UTC+09:00)
- 太平洋/帕劳 (UTC+09:00)
- 亚洲/赤塔 (UTC+09:00)
- 亚洲/帝力 (UTC+09:00)
- 亚洲/查亚普拉 (UTC+09:00)

- 亚洲/雅库茨克 (UTC+09:00)
- 亚洲/平壤 (UTC+09:00)
- 韩国 (UTC+09:00)
- 亚洲/首尔 (UTC+09:00)
- 亚洲/汉德加 (UTC+ 09:00)
- 日本 (UTC+09:00)
- 亚洲/东京 (UTC+09:00)
- 澳大利亚/尤克拉 (UTC+ 08:45)
- 亚洲/古晋 (UTC+08:00)
- 亚洲/重庆 (UTC+08:00)
- Etc/GMT-8 (UTC+08:00)
- 澳大利亚/珀斯 (UTC+08:00)
- 亚洲/澳门 (UTC+08:00)
- 亚洲/澳门 (UTC+08:00)
- 亚洲/乔巴山 (UTC+08:00)
- 亚洲/上海 (UTC+08:00)
- 南极洲/凯西 (UTC+ 08:00)
- 亚洲/乌兰巴托 (UTC+ 08:00)
- 亚洲/重庆 (UTC+08:00)
- 亚洲/乌兰巴托 (UTC+08:00)
- 亚洲/台北 (UTC+08:00)
- 亚洲/马尼拉 (UTC+08:00)
- 中国 (UTC+08:00)
- 亚洲/乌戎潘当 (UTC+ 08:00)
- 亚洲/哈尔滨 (UTC+08:00)
- 新加坡 (UTC+08:00)
- 亚洲/文莱 (UTC+08:00)
- 澳大利亚/西部 (UTC+08:00)

- 亚洲/香港 (UTC+ 08:00)
- 亚洲/望加锡 (UTC+08:00)
- 中国香港 (UTC+08:00)
- 亚洲/吉隆坡 (UTC+ 08:00)
- 亚洲/伊尔库茨克 (UTC+08:00)
- 亚洲/新加坡 (UTC+08:00)
- 亚洲/坤甸 (UTC+ 07:00)
- Etc/GMT-7 (UTC+07:00)
- 亚洲/金边 (UTC+07:00)
- 亚洲/新西伯利亚 (UTC+07:00)
- 南极洲/戴维斯 (UTC+07:00)
- 亚洲/托木斯克 (UTC+ 07:00)
- 亚洲/雅加达 (UTC+07:00)
- 亚洲/巴尔瑙尔 (UTC+07:00)
- 印度洋/圣诞岛 (UTC+07:00)
- 亚洲/胡志明市 (UTC+07:00)
- 亚洲/科布多 (UTC+ 07:00)
- 亚洲/曼谷 (UTC+07:00)
- 亚洲/万象 (UTC+07:00)
- 亚洲/新库兹涅茨克 (UTC+07:00)
- 亚洲/克拉斯诺亚尔斯克 (UTC+ 07:00)
- 亚洲/西贡 (UTC+07:00)
- 亚洲/仰光 (UTC+06:30)
- 亚洲/仰光 (UTC+06:30)
- 印度洋/科科斯群岛 (UTC+06:30)
- 亚洲/喀什 (UTC+06:00)
- Etc/GMT-6 (UTC+06:00)
- 亚洲/阿拉木图 (UTC+06:00)

- 亚洲/达卡 (UTC+06:00)
- 亚洲/鄂木斯克 (UTC+06:00)
- 亚洲/达卡 (UTC+06:00)
- 印度洋/查戈斯群岛 (UTC+06:00)
- 亚洲/Qyzylorda (UTC+06:00)
- 亚洲/比什凯克 (UTC+06:00)
- 南极洲/东方站 ( UTC+ 06:00 )
- 亚洲/乌鲁木齐 (UTC+06:00)
- 亚洲/廷布 (UTC+06:00)
- 亚洲/廷布 (UTC+06:00)
- 亚洲/加德满都 (UTC+05:45)
- 亚洲/加德满都 (UTC+05:45)
- 亚洲/加尔各答 (UTC+05:30)
- 亚洲/科伦坡 (UTC+05:30)
- 亚洲/加尔各答 (UTC+05:30)
- 亚洲/阿克套 (UTC+ 05:00)
- Etc/GMT-5 (UTC+05:00)
- 亚洲/撒马尔罕 (UTC+05:00)
- 亚洲/卡拉奇 (UTC+05:00)
- 亚洲/叶卡捷琳堡 (UTC+05:00)
- 亚洲/杜尚别 (UTC+05:00)
- 印度洋/马尔代夫 (UTC+05:00)
- 亚洲/Oral (UTC+ 05:00)
- 亚洲/塔什干 (UTC+05:00)
- 南极洲/莫森 (UTC+05:00)
- 亚洲/阿克托比 (UTC+ 05:00)
- 亚洲/阿什哈巴德 (UTC+05:00)
- 亚洲/阿什哈巴德 (UTC+ 05:00)

- 亚洲/阿特劳 (UTC+05:00)
- 印度洋/凯尔盖朗岛 (UTC+ 05:00)
- 伊朗 (UTC+04:30)
- 亚洲/德黑兰 (UTC+04:30)
- 亚洲/喀布尔 (UTC+04:30)
- 亚洲/埃里温 (UTC+04:00)
- Etc/GMT-4 (UTC+04:00)
- Etc/GMT-4 (UTC+04:00)
- 亚洲/迪拜 (UTC+04:00)
- 印度洋/留尼汪岛 (UTC+04:00)
- 欧洲/萨拉托夫 (UTC+ 04:00)
- 欧洲/萨马拉 (UTC+04:00)
- 印度洋/马埃岛 (UTC+04:00)
- 亚洲/巴库 (UTC+04:00)
- 亚洲/马斯喀特 (UTC+04:00)
- 欧洲/伏尔加格勒 (UTC+ 04:00)
- 欧洲/阿斯特拉罕 (UTC+04:00)
- 亚洲/第比利斯 (UTC+04:00)
- 欧洲/乌里扬诺夫斯克 (UTC+ 04:00)
- 亚洲/亚丁 (UTC+03:00)
- 非洲/内罗毕 (UTC+03:00)
- 欧洲/伊斯坦布尔 (UTC+03:00)
- Etc/GMT-3 (UTC+03:00)
- 欧洲/扎波罗热 (UTC+03:00)
- 以色列 (UTC+03:00)
- 印度洋/科摩罗 (UTC+03:00)
- 南极洲/昭和 (UTC+03:00)
- 非洲/摩加迪沙 (UTC+03:00)
- 欧洲/布加勒斯特 (UTC+03:00)

- 非洲/阿斯马拉 (UTC+03:00)
- 欧洲/玛丽港 (UTC+ 03:00)
- 亚洲/伊斯坦布尔 (UTC+03:00)
- 欧洲/蒂拉斯波尔 (UTC+ 03:00)
- 欧洲/莫斯科 (UTC+03:00)
- 欧洲/基希讷乌 (UTC+03:00)
- 欧洲/赫尔辛基 (UTC+03:00)
- 亚洲/贝鲁特 (UTC+03:00)
- 亚洲/特拉维夫 (UTC+ 03:00)
- 非洲/吉布提 (UTC+03:00)
- 欧洲/辛菲罗波尔 (UTC+03:00)
- 欧洲/索非亚 (UTC+03:00)
- 亚洲/加沙 (UTC+03:00)
- 非洲/阿斯马拉 (UTC+03:00)
- 欧洲/里加 (UTC+03:00)
- 亚洲/巴格达 (UTC+03:00)
- 亚洲/大马士革 (UTC+03:00)
- 非洲/达累斯萨拉姆 (UTC+ 03:00)
- 非洲/亚的斯亚贝巴 (UTC+ 03:00)
- 欧洲/乌日哥罗德 (UTC+ 03:00)
- 亚洲/耶路撒冷 (UTC+03:00)
- 亚洲/利雅得 (UTC+03:00)
- 亚洲/科威特 (UTC+03:00)
- 欧洲/基洛夫 (UTC+03:00)
- 非洲/坎帕拉 (UTC+03:00)
- 欧洲/明斯克 (UTC+03:00)
- 亚洲/卡塔尔 (UTC+03:00)
- 欧洲/基辅 (UTC+03:00)
- 亚洲/巴林 (UTC+03:00)



- 欧洲/维尔纽斯 (UTC+03:00)
- 印度洋/塔那那利佛 (UTC+03:00)
- 印度洋/马约特岛 (UTC+03:00)
- 欧洲/塔林 (UTC+03:00)
- 土耳其 (UTC+03:00)
- 非洲/朱巴 (UTC+03:00)
- 亚洲/尼科西亚 (UTC+03:00)
- 亚洲/法马古斯塔 (UTC+03:00)
- W-SU (UTC+03:00)
- EET (UTC+03:00)
- 亚洲/希伯伦 (UTC+03:00)
- 亚洲/安曼 (UTC+03:00)
- 欧洲/尼科西亚 (UTC+03:00)
- 欧洲/雅典 (UTC+03:00)
- 非洲/开罗 (UTC+02:00)
- 非洲/姆巴巴内 (UTC+02:00)
- 欧洲/布鲁塞尔 (UTC+02:00)
- 欧洲/华沙 (UTC+02:00)
- CET (UTC+02:00)
- 欧洲/卢森堡 (UTC+02:00)
- Etc/GMT-2 (UTC+02:00)
- 利比亚 (UTC+02:00)
- 非洲/基加利 (UTC+02:00)
- 非洲/的黎波里 (UTC+ 02:00)
- 欧洲/加里宁格勒 (UTC+ 02:00)
- 非洲/温得和克 (UTC+02:00)
- 欧洲/马耳他 (UTC+02:00)
- 欧洲/布辛根 (UTC+02:00)
-

- 欧洲/斯科普里 (UTC+ 02:00)
- 欧洲/萨拉热窝 (UTC+ 02:00)
- 欧洲/罗马 (UTC+02:00)
- 欧洲/苏黎世 (UTC+02:00)
- 欧洲/直布罗陀 (UTC+02:00)
- 非洲/卢本巴希 (UTC+02:00)
- 欧洲/瓦杜兹 (UTC+02:00)
- 欧洲/卢布尔雅那 (UTC+ 02:00)
- 欧洲/柏林 (UTC+02:00)
- 欧洲/斯德哥尔摩 (UTC+02:00)
- 欧洲/布达佩斯 (UTC+02:00)
- 欧洲/萨格勒布 (UTC+02:00)
- 欧洲地区 ( 巴黎 ) (UTC+02:00)
- 非洲/休达 (UTC+02:00)
- 欧洲/布拉格 (UTC+02:00)
- 南极洲/巨魔 (UTC+ 02:00)
- 非洲/哈博罗内 (UTC+02:00)
- 欧洲/哥本哈根 (UTC+02:00)
- 欧洲/维也纳 (UTC+02:00)
- 欧洲/地拉那 (UTC+02:00)
- MET (UTC+02:00)
- 欧洲/阿姆斯特丹 (UTC+02:00)
- 非洲/马普托 (UTC+02:00)
- 欧洲/圣马力诺 (UTC+ 02:00)
- 波兰 (UTC+02:00)
- 欧洲/安道尔 (UTC+02:00)
- 欧洲/奥斯陆 (UTC+02:00)
- 欧洲/波德戈里察 (UTC+02:00)
- 非洲/布琼布拉 (UTC+02:00)

- 大西洋/扬马延岛 (UTC+02:00)
- 非洲/马塞卢 (UTC+02:00)
- 欧洲/马德里 (UTC+02:00)
- 非洲/布兰太尔 (UTC+02:00)
- 非洲/卢萨卡 (UTC+02:00)
- 非洲/哈拉雷 (UTC+02:00)
- 非洲/喀土穆 (UTC+02:00)
- 非洲/约翰内斯堡 (UTC+02:00)
- 欧洲/贝尔格莱德 (UTC+ 02:00)
- 欧洲/布拉迪斯拉发 (UTC+02:00)
- 北极洲/朗伊尔城 (UTC+02:00)
- 埃及 (UTC+02:00)
- 欧洲/梵蒂冈 (UTC+02:00 )
- 欧洲/摩纳哥 (UTC+02:00)
- 欧洲/伦敦 (UTC+01:00)
- Etc/GMT-1 (UTC+01:00)
- 欧洲/泽西岛 (UTC+01:00)
- 欧洲/根西岛 (UTC+01:00)
- 欧洲/马恩岛 (UTC+ 01:00)
- 非洲/突尼斯 (UTC+01:00)
- 非洲/马拉博 (UTC+01:00)
- GB-Eire (UTC+01:00)
- 非洲/拉各斯 (UTC+01:00)
- 非洲/阿尔及尔 (UTC+01:00)
- GB (UTC+01:00)
- 葡萄牙 (UTC+01:00)
- 非洲/圣多美 (UTC+ 01:00)
- 非洲/恩贾梅纳 (UTC+01:00)
- 大西洋/法罗群岛 (UTC+ 01:00)

- 爱尔兰 (UTC+ 01:00)
- 大西洋/法罗群岛 (UTC+01:00)
- 欧洲/都柏林 (UTC+01:00)
- 非洲/利伯维尔 (UTC+01:00)
- 非洲/埃丰阿莱耶 (UTC+ 01:00)
- 非洲/埃丰阿莱耶 (UTC+ 01:00)
- 非洲/杜阿拉 (UTC+01:00)
- 非洲/布拉柴维尔 (UTC+01:00)
- 非洲/新港 (UTC+ 01:00)
- 大西洋/马德拉 (UTC+01:00)
- 欧洲/里斯本 (UTC+01:00)
- 大西洋/加那利 (UTC+01:00)
- 非洲/卡萨布兰卡 (UTC+01:00)
- 欧洲/贝尔法斯特 (UTC+01:00)
- 非洲/罗安达 (UTC+ 01:00)
- 非洲/金沙萨 (UTC+01:00)
- 非洲/班吉 (UTC+01:00)
- WET (UTC+01:00)
- 非洲/尼亚美 ( UTC+ 01:00 )
- GMT (UTC+00:00)
- Etc/GMT-0 (UTC+00:00)
- 大西洋/圣海伦娜 (UTC+ 00:00)
- Etc/GMT+0 (UTC+00:00)
- 非洲/班珠尔 (UTC+00:00)
- Etc/GMT (UTC+00:00)
- 非洲/弗里敦 (UTC+00:00)
- 非洲/巴马科 (UTC+00:00)
- 非洲/科纳克里 (UTC+00:00)
- 通用 (UTC+00:00)

- 非洲/努瓦克肖特 (UTC+00:00)
- UTC (UTC+00:00)
- Etc/通用 (UTC+00:00)
- 大西洋/亚速尔群岛 (UTC+00:00)
- 非洲/阿比让 (UTC+00:00)
- 非洲/阿克拉 (UTC+00:00)
- Etc/UCT (UTC+00:00)
- GMT0 (UTC+00:00)
- 祖鲁 (UTC+ 00:00) 祖鲁 (UTC+ 00:00)
- 非洲/瓦加杜古 (UTC+00:00)
- 大西洋/雷克雅未克 (UTC+00:00)
- Etc/祖鲁 (UTC+00:00)
- 冰岛 (UTC+00:00)
- 非洲/洛美 (UTC+00:00)
- 格林威治 (UTC+00:00)
- Etc/GMT0 (UTC+00:00)
- 美洲/丹马港 (UTC+ 00:00)
- 非洲/达喀尔 (UTC+00:00)
- 非洲/比绍 (UTC+00:00)
- Etc/格林威治 (UTC+00:00)
- 非洲/廷巴克图 (UTC+00:00)
- UCT (UTC+00:00)
- 非洲/蒙罗维亚 (UTC+00:00)
- Etc/UTC (UTC+00:00)

## 公式表达式教程

您可以按照这些教程在 AWS IoT SiteWise中使用公式表达式。

## 主题

- [在公式中使用字符串](#)
- [筛选数据点](#)
- [对符合条件的数据点进行计数](#)
- [公式中的最新数据](#)
- [公式中的数据质量](#)
- [未定义值、无限值和溢出值](#)

## 在公式中使用字符串

您可以对公式表达式中的字符串进行操作。您也可以从引用属性和测量属性的变量中输入字符串。

### Important

公式表达式只能输出双精度值或字符串值。嵌套表达式可以输出其他数据类型，例如字符串，但整个公式的计算结果必须为数字或字符串。您可以使用 [jp 函数](#) 将字符串转换为数字。布尔值必须为 1 (真) 或 0 (假)。有关更多信息，请参阅 [未定义值、无限值和溢出值](#)。

AWS IoT SiteWise 提供了可用于对字符串进行操作的以下公式表达式功能：

- [字符串文字](#)
- [索引运算符](#) (s[index])
- [切片运算符](#) (s[start:end:step])
- [比较函数](#)，您可以使用它按[字典顺序](#)比较字符串
- [字符串函数](#)，包括可以解析序列化的 JSON 对象并将字符串转换为数字的 jp 函数

## 筛选数据点

您可以使用 [if 函数](#) 筛选出不符合条件的数据点。if 函数计算条件并返回 true 和 false 结果的不同值。您可以使用 [无常量](#) if 函数的一个案例的输出，以丢弃该案例的数据点。

### 筛选出与条件匹配的数据点

- 创建一个转换，该转换使用 if 函数定义一个条件，该条件检查条件是否满足，并以 result\_if\_true 或 result\_if\_false 值的形式返回 none。

## Example 示例：过滤掉水未沸腾的数据点

考虑您在其中有一个测量值的场景 `temp_c`，它提供水在机器中的温度（以摄氏度为单位）。您可以定义以下转换来过滤掉水未沸腾的数据点：

- 转换：`boiling_temps = if(gte(temp_c, 100), temp_c, none)` – 如果温度大于或等于 100 摄氏度，则返回温度，否则不返回任何数据点。

## 对符合条件的数据点进行计数

您可以使用[比较函数](#)和[求和](#)来计算条件为真的数据点的数量。

## 计算与条件匹配的数据点

1. 创建一个使用比较函数在另一个属性上定义筛选条件的转换。
2. 创建一个汇总满足该条件的数据点的指标。

## Example 示例：计算水沸腾的数据点数量

考虑您在其中有一个测量值的场景 `temp_c`，它提供水在机器中的温度（以摄氏度为单位）。您可以定义以下转换和指标属性来计算水沸腾的数据点数量：

- 转换：`is_boiling = gte(temp_c, 100)` – 如果温度大于或等于 100 摄氏度，则计算结果为返回 1，否则计算结果为返回 0。
- 指标：`boiling_count = sum(is_boiling)` – 返回水沸腾的数据点数量。

## 公式中的最新数据

AWS IoT SiteWise 支持延迟摄取最长 7 天的数据。当 AWS IoT SiteWise 收到延迟数据时，它会重新计算过去窗口中输入延迟数据的任何指标的现有值。这些重新计算会产生数据处理费用。

### Note

在 AWS IoT SiteWise 计算输入后期数据的属性时，它使用每个属性的当前公式表达式。

在 AWS IoT SiteWise 重新计算某个指标的过去窗口后，它将替换该窗口的先前值。如果您为该指标启用了通知，则 AWS IoT SiteWise 还会发出属性值通知。这意味着，您可以对您以前收到了通知的相同

属性和时间戳收到新属性值更新通知。如果应用程序或数据湖使用属性值通知，则必须使用新值更新上一个值，以便其数据准确无误。

## 公式中的数据质量

在中 AWS IoT SiteWise，每个数据点都有一个质量代码，可以是以下代码之一：

- GOOD – 数据不受任何问题的影响。
- BAD – 数据受传感器故障等问题的影响。
- UNCERTAIN – 数据受传感器不准确等问题的影响。

AWS IoT SiteWise 在计算变换和指标时仅消耗GOOD质量数据。AWS IoT SiteWise 仅输出成功计算所需的GOOD质量数据。如果计算不成功，则 AWS IoT SiteWise 不会为该计算输出数据点。如果计算生成未定义值、无限值或溢出值，则会发生这种情况。

有关如何查询数据和按数据质量筛选的更多信息，请参阅[从中查询数据 AWS IoT SiteWise](#)。

## 未定义值、无限值和溢出值

某些公式表达式（例如  $x / 0$ 、 $\sqrt{-1}$ 、或  $\log(0)$ ）会计算在实数系统中未定义的值、无限值或超出所 AWS IoT SiteWise 支持的范围。当资产属性的表达式计算未定义、无限或溢出值时，AWS IoT SiteWise 不会为该计算输出数据点。

AWS IoT SiteWise 如果将非数值作为公式表达式的结果进行计算，也不会输出数据点。这意味着，如果您定义了一个计算字符串、数组或 [无常量](#) 的公式，则 AWS IoT SiteWise 不会为该计算输出数据点。

## Example 示例

以下每个公式表达式生成的值都 AWS IoT SiteWise 不能表示为数字。AWS IoT SiteWise 在计算这些公式表达式时不输出数据点。

- $x / 0$  未定义。
- $\log(0)$  未定义。
- $\sqrt{-1}$  在实数系统中未定义。
- "hello" + " world" 是一个字符串。
- `jp({'values':[3,6,7]}, '$.values')` 是一个数组。
- 当 temp 小于 300 时，`if(gte(temp, 300), temp, none)` 等于 none。



## 创建自定义复合模型（组件）

自定义复合模型或组件（如果您使用的是控制台）可以为您的资产模型和组件模型提供另一个层次的组织。您可以使用它们通过对属性进行分组或引用其他模型来构造模型。有关使用自定义复合模型的更多信息，请参阅[自定义复合模型（组件）](#)。

您可以在现有资产模型或组件模型中创建自定义复合模型。有两种类型的自定义复合模型。要对模型中的相关属性进行分组，可以创建内联自定义复合模型。要在资产模型或组件模型中引用组件模型，可以创建基于组件模型的自定义复合模型。

以下各节介绍如何使用 AWS IoT SiteWise API 创建自定义复合模型。

### 主题

- [创建内联组件（控制台）](#)
- [创建内联自定义复合模型 \(AWS CLI\)](#)
- [创建组 component-model-based 组件（控制台）](#)
- [创建 component-model-based 自定义复合模型 \(AWS CLI\)](#)

## 创建内联组件（控制台）

您可以使用 AWS IoT SiteWise 控制台创建定义其自身属性的内联组件。

### Note

由于这是一个内联组件，因此这些属性仅适用于当前资产模型，不会在其他任何地方共享。如果您需要生成可重复使用的模型（例如，在多个资产模型之间共享，或者在一个资产模型中包含多个实例），则应改为基于组件模型创建一个组件。有关详细信息，请参阅以下部分。

## 创建组件（控制台）

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 **模型**。
3. 选择要向其添加组件的资产模型。
4. 在“属性”选项卡上，选择“组件”。
5. 选择创建组件。
6. 在“创建组件”页面上，执行以下操作：

- a. 输入组件的名称，例如**ServoMotor**或**ServoMotor Model**。该名称在该地区您账户的所有组成部分中必须是唯一的。
- b. (可选) 为模型添加属性定义。属性表示很少更改的信息。有关更多信息，请参阅 [定义静态数据 \(属性\)](#)。
- c. (可选) 为模型添加测量值定义。测量值代表来自您的设备的数据流。有关更多信息，请参阅 [定义来自设备的数据流 \(测量\)](#)。
- d. (可选) 为模型添加转换定义。转换是将数据从一种表单映射到另一种表单的公式。有关更多信息，请参阅 [转换数据 \(转换\)](#)。
- e. (可选) 为模型添加指标定义。指标是按时间间隔汇总数据的公式。指标可以从关联资产的指标中输入数据，这样您就可以计算统计信息，从而提供对整个操作或部分操作的见解。有关更多信息，请参阅 [聚合来自属性和其他资产 \(指标\) 的数据](#)。
- f. 选择创建组件。

## 创建内联自定义复合模型 (AWS CLI)

您可以使用 AWS Command Line Interface (AWS CLI) 来创建定义其自身属性的内联自定义复合模型。

使用 [CreateAssetModelComposite模型](#) 操作创建具有属性的内联模型。此操作需要具有以下结构的负载。

### Note

由于这是一个内联复合模型，因此这些属性仅适用于当前的资产模型，不会在其他任何地方共享。它之所以成为“内联”，是因为它没有为该 `composedAssetModelId` 字段提供值。如果您需要生成可重复使用的模型（例如，在多个资产模型之间共享，或者在一个资产模型中包含多个实例），则应改为创建一个基于组件模型的复合模型。有关详细信息，请参阅以下部分。

```
{
  "assetModelCompositeModelName": "CNCLathe_ServoMotorA",
  "assetModelCompositeModelType": "CUSTOM",
  "assetModelCompositeModelProperties": [
    {
      "dataType": "DOUBLE",
```

```
    "name": "Servo Motor Temperature",
    "type": {
      "measurement": {}
    },
    "unit": "Celsius"
  },
  {
    "dataType": "DOUBLE",
    "name": "Spindle speed",
    "type": {
      "measurement": {}
    },
    "unit": "rpm"
  }
]
```

## 创建组 component-model-based 件 (控制台)

您可以使用 AWS IoT SiteWise 控制台基于组件模型创建组件。

### 创建组 component-model-based 件 (控制台)

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 **模型**。
3. 选择要向其添加组件的资产模型。
4. 在“属性”选项卡上，选择“组件”。
5. 选择创建组件。
6. 在“创建组件”页面上，执行以下操作：
  - a. 选择要作为组件基础的组件模型。
  - b. 输入组件的名称，例如 **ServoMotor** 或 **ServoMotor Model**。该名称在该地区您账户的所有组成部分中必须是唯一的。
  - c. 选择创建组件。

## 创建 component-model-based 自定义复合模型 (AWS CLI)

您可以使用在 AWS CLI 资产模型中创建 component-model-based 自定义复合模型。component-model-based 自定义复合模型是对已在其他地方定义的组件模型的引用。

使用 [CreateAssetModelComposite模型](#) 操作创建 component-model-based 自定义复合模型。此操作需要具有以下结构的负载。

#### Note

在此示例中，的值 `composedAssetModelId` 是现有组件模型的资产模型 ID 或外部 ID。有关更多信息，请参阅《AWS IoT SiteWise 用户指南》中的 [使用外部 ID 引用对象](#)。有关如何创建组件模型的示例，请参见 [创建组件模型 \(AWS CLI\)](#)。

```
{
  "assetModelCompositeModelName": "CNCLathe_ServoMotorA",
  "assetModelCompositeModelType": "CUSTOM",
  "composedAssetModelId": component model ID
}
```

由于它只是一个参考，因此 component-model-based 自定义复合模型除了名称之外没有自己的属性。

如果要将同一组件的多个实例添加到资产模型中（例如，具有多个伺服电动机的数控机床），则可以添加多个 component-model-based 自定义复合模型，每个模型都有自己的名称，但所有模型的引用都相同 `composedAssetModelId`。

您可以将组件嵌套在其他组件中。为此，您可以向其中一个组件模型添加 component-model-based 复合模型，如本示例所示。

## 创建资产

您可以从资产模型中创建资产。您必须具有资产模型，然后才能创建资产。如果您尚未创建资产模型，请参阅 [创建资产模型](#)。

#### Note

您只能从 ACTIVE 模型创建资产。如果模型的状态不是 ACTIVE，您可能需要等待几分钟（最长），然后才能从该模型创建资产。有关更多信息，请参阅 [资产和模型状态](#)。

### 主题

- [创建资产 \(控制台\)](#)

- [创建资产 \(AWS CLI\)](#)
- [配置新资产](#)

## 创建资产 ( 控制台 )

您可以使用 AWS IoT SiteWise 控制台创建资产。

### 创建资产 ( 控制台 )

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 资产。
3. 选择 创建资产。
4. 在 创建资产 页上，执行以下操作：
  - a. 对于 模型，选择要从中创建资产的资产模型。

#### Note

如果您的模型状态不为 活动，则必须等待其变为活动状态，如果状态为 失败，则必须解决问题。

- b. 为资产输入 名称。
- c. ( 可选 ) 为资产添加标签。有关更多信息，请参阅 [为资源添加 AWS IoT SiteWise 标签](#)。
- d. 选择 创建资产。

创建资产时，AWS IoT SiteWise 控制台会导航到新资产的页面。在此页面上，您可以查看资产的状态，最初为 正在创建。此页面会自动更新，因此您可以等待资产的状态更新。

#### Note

资产创建流程最多可能需要一分钟。在状态为活动后，您可以对资产执行更新操作。有关更多信息，请参阅 [资产和模型状态](#)。

创建资产后，请参阅[配置新资产](#)。

## 创建资产 (AWS CLI)

您可以使用 AWS Command Line Interface (AWS CLI) 根据资产模型创建资产。

您必须具有 `assetModelId` 才能创建资产。如果您创建了资产模型，但不知道该 [ListAsset模型assetModelId](#)，请使用 [模型 API](#) 查看所有资产模型。

要根据资产模型创建资产，请使用带有以下参数的 [CreateAssetAPI](#)：

- `assetName` – 新资产的名称。为你的资产命名以帮助你识别它。
- `assetModelId` – 资产的 ID。这是 UUID 格式的实际 ID，或者 `externalId:myExternalId` 如果有，则为。有关更多信息，请参阅《AWS IoT SiteWise 用户指南》中的 [使用外部 ID 引用对象](#)。

### 创建资产 (AWS CLI)

- 运行以下命令以创建资产。将 `####` 替换为资产名称，将资产模型 `ID #####` 的 ID 或外部 ID。

```
aws iotsitewise create-asset \  
  --asset-name asset-name \  
  --asset-model-id asset-model-id
```

此操作将按以下格式返回包含新资产的详细信息和状态的响应。

```
{  
  "assetId": "String",  
  "assetArn": "String",  
  "assetStatus": {  
    "state": "String",  
    "error": {  
      "code": "String",  
      "message": "String"  
    }  
  }  
}
```

资产的 `state` 为 `CREATING`，直至资产创建完毕。

### Note

资产创建流程最多可能需要一分钟。要检查资产的状态，请使用以您的资产 ID 作为 `assetId` 参数的 [DescribeAsset](#) 操作。资产的 `state` 为 `ACTIVE` 后，您可以对资产执行更新操作。有关更多信息，请参阅 [资产和模型状态](#)。

创建资产后，请参阅 [配置新资产](#)。

## 配置新资产

使用以下任何可选操作完成资产配置：

- [将工业数据流映射到资产属性](#)，如果您的资产具有测量值属性。
- [更新属性值](#)，如果您的资产具有唯一属性值。
- [关联资产和取消资产关联](#)，如果您的资产是父资产。

## 搜索资产

使用 AWS IoT SiteWise 控制台 搜索功能根据元数据和实时属性值筛选器查找资产。

### 先决条件

AWS IoT SiteWise 需要权限才能与之集成，AWS IoT TwinMaker 以便更好地组织和建模工业数据。如果您已授予权限 AWS IoT SiteWise，请使用 [ExecuteQuery](#) API。如果您尚未授予权限 AWS IoT SiteWise，并且需要入门方面的帮助，请参阅 [集成 AWS IoT SiteWise 和 AWS IoT TwinMaker](#)。

## 高级搜索开启 AWS IoT SiteWise 控制台

### 元数据搜索

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择“资产”下的“高级搜索”。
3. 在“高级搜索”下，选择“元数据”搜索选项。
4. 填写参数。尽可能多地填写字段，以便高效搜索。

- a. 资产名称-输入资产的完整名称或部分名称进行广泛搜索。
  - b. 属性名称-输入完整的属性名称或部分名称以进行广泛搜索。
  - c. 操作员-从以下选项选择一个运算符：
    - =
    - <
    - >
    - <=
    - >=
  - d. 属性值-将此值与属性的最新值进行比较。
  - e. 属性值类型-属性的数据类型。请从以下内容中选择：
    - 双
    - 整数
    - String
    - 布尔值
5. 选择搜索。
  6. 从“搜索结果”表中，从“名称”列中选择资产。这会将您带到该资产的详细资产页面。



**Assets**

Assets represent industrial devices and processes that send data streams to SiteWise. Models are structures that enforce a specific model of properties and hierarchies for all instances of each asset. You must create every asset from a model.

**Advanced search**

Use advanced search to find assets based on specific metadata. In addition, you can enter SQL queries directly in the query builder.

**Metadata search** | Query builder

Asset name: Level-2 | Property name: power\_max | Operator: > | Property value: 20 | Property value type: Double

Search results (2)

Name	Asset id	Description
<a href="#">Level-2-asset-1</a>	d0e9019b-9c38-4316-b574-38317aa38143	
<a href="#">Level-2-asset-2</a>	b9c0d2fc-1527-42ce-8ba2-d1a4e8ff43de	Example description

## 部分搜索

无需为资产搜索提供所有参数。以下是使用元数据搜索选项进行部分搜索的一些示例：

- 按名称查找资产：
  - 仅在“资产名称”字段中输入一个值。
  - “属性名称”和“属性值”字段为空。
- 查找包含具有特定名称的属性的资产：
  - 仅在“属性名称”字段中输入值。
  - 资产名称和属性值字段为空。
- 根据资产的最新价值查找资产：
  - 在“属性名称”和“属性值”字段中输入值。
  - 选择运算符和属性值类型。

## 查询生成器搜索

1. 导航到 AWS IoT SiteWise 控制台。
2. 在导航窗格中，选择“资产”下的“高级搜索”。

3. 在“高级搜索”下，选择“查询生成器”选项。
4. 在查询生成器窗格中，编写您的 SQL 查询以检索asset\_name、asset\_id和asset\_description。
5. 选择搜索。
6. 从“搜索结果”表中，从“名称”列中选择资产。这会将您带到该资产的详细资产页面。

**Assets**

Assets represent Industrial devices and processes that send data streams to SiteWise. Models are structures that enforce a specific model of properties and hierarchies for all instances of each asset. You must create every asset from a model.

**Advanced search**

Use advanced search to find assets based on specific metadata. In addition, you can enter SQL queries directly in the query builder.

Metadata search | **Query builder**

Query builder

```
SELECT a.asset_id, a.asset_name, a.asset_description
FROM asset a, asset_property p, latest_value_time_series ts
WHERE a.asset_name LIKE '%asset-2%' AND a.property_name = 'temperature_f' AND ts.double_value > 50.0
```

Clear Search

**Search results (2)**

Name	Asset id	Description
<a href="#">Level-2a-asset-2</a>	4fed596d-e903-4338-86db-34ca9301233a	Generator #3
<a href="#">Level-2b-asset-2</a>	b4ac2b24-4fce-4a72-9fea-ef6d0f741e8d	Generator #2

### Note

- SQL 查询中的SELECT子句必须包含asset\_name和asset\_id字段，以确保搜索结果表中的资产有效。
- 查询生成器仅在结果表中显示名称、资产 ID 和描述。向SELECT子句添加更多字段不会向结果表中添加更多列

## 将工业数据流映射到资产属性

可以在资产属性上定义属性别名。这有助于您在摄取或检索资产数据时识别资产属性。如果资产具有测量值属性，您可以定义属性别名来将数据流映射到这些测量值属性。

此过程要求您知道属性别名。

- 如果您使用 [Edge SiteWise 网关中的 OPC-UA 数据源从 OPC-UA 服务器提取数据](#)，则您的属性别名是“对象”节点下变量的路径，开头为。/

### Example

如果变量的路径是 `company/windfarm/3/turbine/7/temperature`，则您的属性别名是 `/company/windfarm/3/turbine/7/temperature`。

有关 OPC-UA 信息架构的更多信息，请参阅 OPC UA 在线参考中的 [信息模型和地址间距映射](#)。

### 注意

- 如果为 OPC-UA 源配置数据流前缀，则必须在来自该源的所有数据流的属性别名中包含该前缀。

### Example

如果 `/RentonWA` 是前缀，则前一个别名是 `/RentonWA/company/windfarm/3/turbine/7/temperature`。

- 属性别名最多包含 1,000 字节。OPC-UA 变量路径最多包含 4,096 字节。目前，AWS IoT SiteWise 不支持从具有长路径的 OPC-UA 变量中提取数据。

- 如果您使用 [SiteWise 边缘网关中的 Modbus TCP 数据源从 Modbus 服务器提取数据](#)，则您的属性别名为：

`Modbus register set tag name`

使用此值将数据从该寄存器集发送到资产属性。

- 如果您从其他来源（例如使用 [AWS IoT 规则](#) 或 [API](#)）提取数据，则必须定义自己的属性别名。您可以定义适用于您的设备配置的属性别名命名系统。例如，如果从 AWS IoT 事物中提取数据，则可以在属性别名中包含事物名称以唯一地标识数据流。有关此示例的更多信息，请参阅 [从 AWS IoT 事物中提取数据教程](#)。

在一个地区和 AWS 账户中，房产别名必须是唯一的。AWS IoT SiteWise 如果将属性别名设置为已存在于其他资产属性上的别名，则返回错误。

如果您有多个具有相同数据流路径的 OPC-UA 源，请为每个源的路径添加前缀以形成唯一的别名。有关更多信息，请参阅 [配置数据源](#)。

### Note

本节介绍如何为测量属性设置属性别名。有关如何为外部警报状态属性设置属性别名的更多信息，请参阅 [映射外部警报状态流](#)。

## 主题

- [设置属性别名 \(控制台\)](#)
- [设置属性别名 \(AWS CLI\)](#)

## 设置属性别名 (控制台)

您可以使用 AWS IoT SiteWise 控制台为资产属性设置别名。

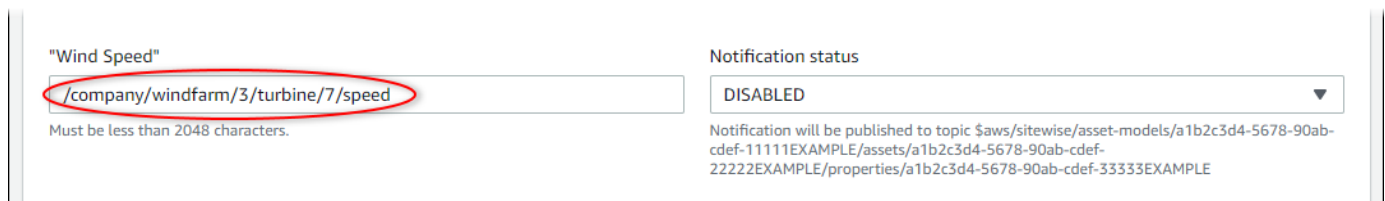
### 设置属性别名 (控制台)

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 资产。
3. 选择要为其设置属性别名的资产。

### Tip

您可以选择箭头图标来展开资产层次结构以查找资产。

4. 选择编辑。
5. 查找要为其设置别名的属性，然后输入属性别名。



The screenshot shows a form with two main sections. The first section is titled '"Wind Speed"' and contains a text input field with the value `/company/windfarm/3/turbine/7/speed`. Below the input field is a note: "Must be less than 2048 characters." The second section is titled "Notification status" and contains a dropdown menu with the value "DISABLED". Below the dropdown menu is a note: "Notification will be published to topic \$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE".

## 6. 选择保存。

### 设置属性别名 (AWS CLI)

使用 AWS Command Line Interface (AWS CLI) 为资产属性设置别名。

您必须知道资产的 `assetId` 和属性的 `propertyId` 才能完成此过程。您也可以使用外部 ID。如果您创建了资产但不知道该资产 `assetId`，请使用 [ListAssets](#) API 列出特定模型的所有资产。使用该 [DescribeAsset](#) 操作查看资产的属性，包括财产 ID。

使用 [UpdateAsset属性](#) 操作将数据流映射到资产的属性。指定以下参数：

- `assetId`— 资产的 ID 或外部 ID。有关更多信息，请参阅《AWS IoT SiteWise 用户指南》中的 [使用外部 ID 引用对象](#)。
- `propertyId`— 资产属性的 ID 或外部 ID。
- `propertyAlias` – 数据流到属性别名的路径。
- `propertyNotificationState` – 属性值通知状态：ENABLED 或 DISABLED。在更新属性别名时指定属性的现有通知状态。您可以通过“[DescribeAsset属性](#)”操作检索现有的通知状态。

如果省略此参数，则新通知状态为 DISABLED。有关属性通知的更多信息，请参阅[与其他 AWS 服务互动](#)。

### 设置属性别名 (AWS CLI)

1. 运行以下命令以检索属性的当前通知状态。将 `asset-id` 和 `property-id` 替换为资产属性的 ID。

```
aws iotsitewise describe-asset-property \  
  --asset-id asset-id \  
  --property-id property-id
```

此操作将按以下格式返回包含资产属性的详细信息的响应。属性通知状态位于 JSON 对象的 `assetProperty.notification.state` 中。

```
{  
  "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",  
  "assetName": "Wind Turbine 7",  
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
```

```

"assetProperty": {
  "id": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
  "name": "Wind Speed",
  "notification": {
    "topic": "$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/
assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE",
    "state": "ENABLED"
  },
  "dataType": "DOUBLE",
  "unit": "m/s",
  "type": {
    "measurement": {}
  }
}
}

```

2. 运行以下命令以设置资产属性的别名。将 *property-alias* 替换为属性别名，将 *notification-state* 替换为通知状态，或者省略 `--property-notification-state` 以禁用通知。您可以选择使用新单位更新资产的 `##`，以及 `--property-unit`。

```

aws iotsitewise update-asset-property \
  --asset-id asset-id \
  --property-id property-id \
  --property-alias property-alias \
  --property-notification-state notification-state \
  --property-unit unit

```

3. 要验证是否已设置别名，请运行以下命令来检索属性的详细信息。将 *asset-id* 和 *property-id* 替换为资产属性的 ID。

```

aws iotsitewise describe-asset-property \
  --asset-id asset-id \
  --property-id property-id

```

此操作将按以下格式返回包含资产属性的详细信息的响应。属性别名 `assetProperty.alias` 在 JSON 对象中，在本示例中设置为 `myAlias`。

```

{
  "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetName": "Wind Turbine 7",
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",

```

```
"assetProperty": {
  "alias": "myAlias",
  "id": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
  "name": "Wind Speed",
  "notification": {
    "topic": "$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/
assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE",
    "state": "ENABLED"
  },
  "dataType": "DOUBLE",
  "unit": "m/s",
  "type": {
    "measurement": {}
  }
}
```

## 更新属性值

资产继承其资产模型的属性，包括属性的默认值。某些情况下，您需要保留资产模型的默认属性，如资产制造商特征。其他情况下，您需要更新继承的属性，如资产的纬度和经度。

### Updating an attribute value (console)

您可以使用 AWS IoT SiteWise 控制台更新属性资产属性的值。

#### 更新属性值 (控制台)

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 资产。
3. 选择要更新其属性的资产。

#### Tip

您可以选择箭头图标来展开资产层次结构以查找资产。

4. 选择编辑。
5. 查找要更新的属性，然后输入其新值。

## 6. 选择保存。

## Updating an attribute value (AWS CLI)

您可以使用 AWS Command Line Interface (AWS CLI) 来更新属性值。

您必须知道资产的 `assetId` 和属性的 `propertyId` 才能完成此过程。您也可以使用外部 ID。如果您创建了资产但不知道该资产 `assetId`，请使用 [ListAssets](#) API 列出特定模型的所有资产。使用该 [DescribeAsset](#) 操作查看资产的属性，包括财产 ID。

使用 “[BatchPutAssetProperty值](#)” 操作为您的资产分配属性值。您可以使用此操作一次设置多个属性。此操作的负载包含条目列表，每个条目都包含资产 ID、属性 ID 和属性值。

## 更新属性的值 (AWS CLI)

1. 创建一个名为 `batch-put-payload.json` 的文件，并将以下 JSON 对象复制到该文件中。此示例负载演示如何设置风力涡轮机的纬度和经度。更新 ID、值和时间戳以修改用例的负载。

```
{
  "entries": [
    {
      "entryId": "windfarm3-turbine7-latitude",
      "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
      "propertyValues": [
        {
          "value": {
            "doubleValue": 47.6204
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          }
        }
      ]
    }
  ]
}
```



```

    },
    {
      "entryId": "windfarm3-turbine7-longitude",
      "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE",
      "propertyValues": [
        {
          "value": {
            "doubleValue": 122.3491
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          }
        }
      ]
    }
  ]
}

```

- 负载中的每个条目都包含一个 `entryId`，您可以将其定义为任何唯一的字符串。如果任何请求条目失败，则每个错误都将包含相应请求的 `entryId`，以便您知道要重试的请求。
- 要设置属性值，可以在每个属性属性的列表中包含一个 `timestamp-quality-value (TQV)` 结构。`propertyValues` 此结构必须包含新的 `value` 和当前的 `timestamp`。
  - `value` – 包含下列字段之一的结构，具体取决于要设置的属性的类型：
    - `booleanValue`
    - `doubleValue`
    - `integerValue`
    - `stringValue`
  - `timestamp`— 一种包含当前 Unix 纪元时间（以秒为单位）的结构，`timeInSeconds`。AWS IoT SiteWise 拒绝任何时间戳在过去存在超过 7 天或将来超过 5 分钟的数据点。

有关如何为 [BatchPutAssetPropertyValue](#) 准备负载的更多信息，请参阅[使用 API 摄取数据 AWS IoT SiteWise](#)。

2. 运行以下命令将属性值发送到 AWS IoT SiteWise：

```
aws iotsitewise batch-put-asset-property-value -\-cli-input-json file://batch-put-payload.json
```

# 关联资产和取消资产关联

如果资产的模型定义了任何子资产模型层次结构，则可以将子资产关联到您的资产。父资产可以访问和聚合关联资产中的数据。有关分层资产模型的更多信息，请参阅[定义资产模型层次结构](#)。

主题

- [关联资产和取消资产关联 \(控制台\)](#)
- [关联和取消关联资产 \(AWS CLI\)](#)

## 关联资产和取消资产关联 (控制台)

您可以使用 AWS IoT SiteWise 控制台关联和取消关联资产。

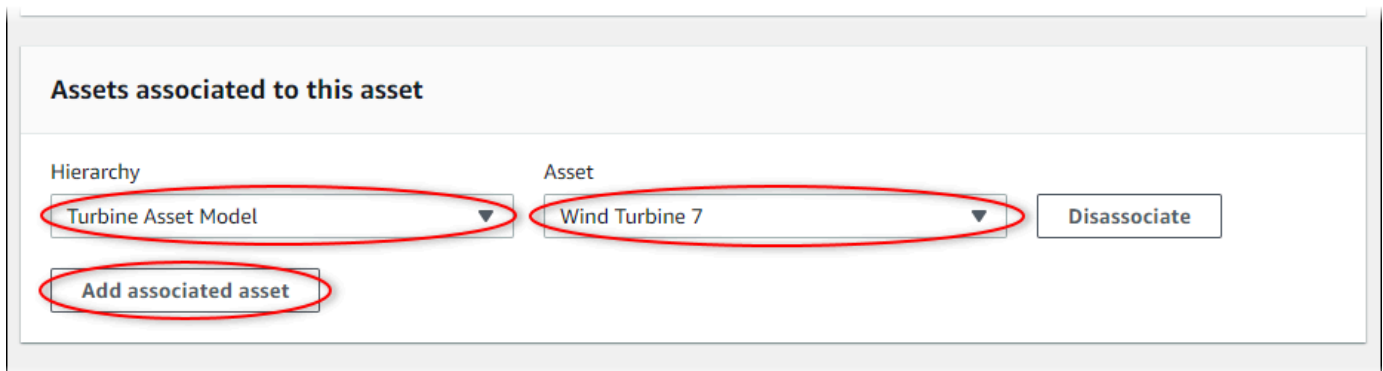
关联资产 (控制台)

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 资产。
3. 选择要为其关联子资产的父资产。

**i** Tip

您可以选择箭头图标来展开资产层次结构以查找资产。

4. 选择编辑。
5. 在 与此资产关联的资产 中，选择 添加关联的资产。



6. 对于 层次结构，选择定义父资产与子资产之间关系的层次结构。
7. 对于 资产，选择要关联的子资产。
8. 选择保存。

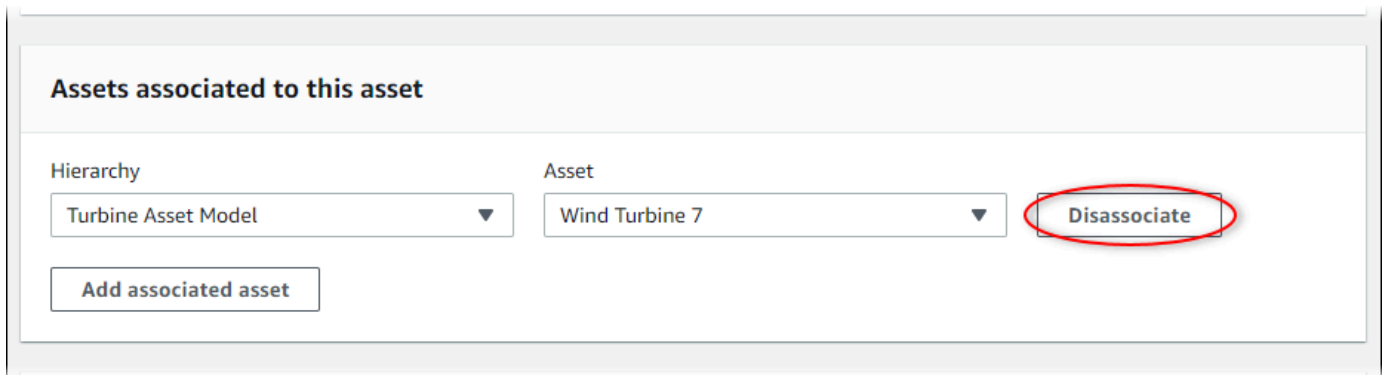
## 取消资产关联 ( 控制台 )

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 资产。
3. 选择要取消其子资产关联的父资产。

### Tip

您可以选择箭头图标来展开资产层次结构以查找资产。

4. 选择编辑。
5. 在 与此资产关联的资产 中，为资产选择 取消关联。



6. 选择保存。

## 关联和取消关联资产 ()AWS CLI

您可以使用 AWS Command Line Interface (AWS CLI) 来关联和取消关联资产。

对于此过程，您必须知道父资产模型中定义与子资产模型关系的层次结构的 ID (hierarchyId)。使用 [DescribeAsset](#) 操作在响应中查找层次结构 ID。

### 查找层次结构 ID

- 运行以下命令以描述父资产。将 **### ID** 替换为父资产的 ID 或外部 ID。

```
aws iotsitewise describe-asset --asset-id parent-asset-id
```

此操作将返回包含资产详细信息的响应。响应包含一个具有以下结构的 `assetHierarchies` 列表：

```
{
  ...
  "assetHierarchies": [
    {
      "id": "String",
      "name": "String"
    }
  ],
  ...
}
```

层次结构 ID 是资产层次结构列表中的层次结构的 id 值。

有了层次结构 ID 后，您可以将资产与该层次结构关联或取消关联。

要将子资产与父资产相关联，请使用 [AssociateAssets](#) 操作。要解除子资产与父资产的关联，请使用 [DisassociateAssets](#) 操作。请指定以下参数（对于这两个操作是相同的）：

- `assetId`— 父资产的 ID 或外部 ID。
- `hierarchyId`— 父资产中的层次结构 ID 或外部 ID。
- `childAssetId`— 子资产的 ID 或外部 ID。

### 关联资产 (AWS CLI)

- 运行以下命令以将子资产与父资产关联。 **#### ID### ID #### ID ##### ID#**

```
aws iotsitewise associate-assets \
  --asset-id parent-asset-id \
  --hierarchy-id hierarchy-id \
  --child-asset-id child-asset-id
```

### 解除资产的关联 ()AWS CLI

- 运行以下命令以取消子资产与父资产的关联。 **#### ID### ID #### ID ##### ID#**

```
aws iotsitewise disassociate-assets \
  --asset-id parent-asset-id \
  --hierarchy-id hierarchy-id \
```

```
--child-asset-id child-asset-id
```

## 更新资产和模型

您可以在中更新您的资产、资产模型和组件模型，AWS IoT SiteWise 以修改其名称和定义。这些更新操作是异步的，需要一段时间才能传 AWS IoT SiteWise播。在进行其他更改之前，请检查资产或模型的状态。您必须等待更改传播之后才能继续使用更新的资产或模型。

### 主题

- [更新资产](#)
- [更新资产模型和组件模型](#)
- [更新自定义复合模型 \( 组件 \)](#)

## 更新资产

您可以使用 AWS IoT SiteWise 控制台或 API 来更新资产的名称。

更新资产时，资产的状态为UPDATING 直至更改传播为止。有关更多信息，请参阅 [资产和模型状态](#)。

### 主题

- [更新资产 \( 控制台 \)](#)
- [更新资产 \(AWS CLI\)](#)

## 更新资产 ( 控制台 )

您可以使用 AWS IoT SiteWise 控制台更新资产详细信息。

### 更新资产 ( 控制台 )

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 资产。
3. 选择要更新的资产。

#### Tip

您可以选择箭头图标来展开资产层次结构以查找资产。

4. 选择编辑。
5. 更新资产 的名称。
6. ( 可选 ) 在此页面上，更新资产的其他信息。有关更多信息，请参阅下列内容：
  - [将工业数据流映射到资产属性](#)
  - [更新属性值](#)
  - [与其他 AWS 服务互动](#)
7. 选择保存。

## 更新资产 (AWS CLI)

您可以使用 AWS Command Line Interface (AWS CLI) 来更新资产 的名称。

使用 [UpdateAsset](#) 操作来更新资产。指定以下参数：

- `assetId` – 资产的 ID。这是 UUID 格式的实际 ID，或者 `externalId:myExternalId` 如果有，则为。有关更多信息，请参阅《AWS IoT SiteWise 用户指南》中的 [使用外部 ID 引用对象](#)。
- `assetName` – 资产的新名称。

### 更新资产 的名称 (AWS CLI)

- 运行以下命令以更新资产 的名称。将 **## ID #####** 的 ID 或外部 ID。使用 **#####** 更新资产名称。

```
aws iotsitewise update-asset \  
  --asset-id asset-id \  
  --asset-name asset-name
```

## 更新资产模型和组件模型

您可以使用 AWS IoT SiteWise 控制台或 API 来更新资产模型或组件模型。

您无法更改现有属性的类型或数据类型，也不能更改现有指标的窗口。您也无法将模型的类型从资产模型更改为组件模型，或者反之亦然。

### Important

- 如果您从资产模型或组件模型中移除某个属性，则 AWS IoT SiteWise 会删除该属性之前的所有数据。对于组件模型，这会影响使用该组件模型的所有资产模型，因此请特别注意了解您的更改可能应用的范围。
- 如果您从资产模型中移除层次结构定义，则 AWS IoT SiteWise 取消关联该层次结构中的所有资产。

更新资产模型时，基于该模型的每个资产都会反映您对基础模型所做的任何更改。在更改传播之前，每个资产都具有 UPDATING 状态。您必须等待这些资产恢复到 ACTIVE 状态，然后才能与它们进行交互。在此期间，更新的资产模型的状态将为 PROPAGATING。

更新组件模型时，包含该组件模型的每个资产模型都会反映所做的更改。在组件模型的变化传播之前，每个受影响的资产模型都具有 UPDATING 状态，然后是更新其关联资产时的状态，如前一段所述。PROPAGATING 必须等到这些资产模型恢复到 ACTIVE 状态后才能与它们进行交互。在此期间，更新的组件模型的状态将为 PROPAGATING。

有关更多信息，请参阅 [资产和模型状态](#)。

#### 主题

- [更新资产或组件模型 \(控制台\)](#)
- [更新资产或组件模型 \(AWS CLI\)](#)

## 更新资产或组件模型 (控制台)

您可以使用 AWS IoT SiteWise 控制台更新资产模型或组件模型。

### 更新资产模型或组件模型 (控制台)

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 **模型**。
3. 选择要更新的资产模型或组件模型。
4. 选择 **编辑**。
5. 在 **编辑模型** 页面上，执行以下任何操作：
  - 在 **模型详细信息** 中，更改模型的名称。

- 更改任何 属性定义。您无法更改现有属性的 数据类型。有关更多信息，请参阅 [定义静态数据 \(属性\)](#)。
- 更改任何 测量值定义。您无法更改现有测量值的 数据类型。有关更多信息，请参阅 [定义来自设备的数据流 \(测量\)](#)。
- 更改任何 转换定义。有关更多信息，请参阅 [转换数据 \(转换\)](#)。
- 更改任何 指标定义。您无法更改现有指标的 时间间隔。有关更多信息，请参阅 [聚合来自属性和其他资产 \(指标\) 的数据](#)。
- (仅限资产模型) 更改任何层次结构定义。您无法更改现有层次结构的 层次结构模型。有关更多信息，请参阅 [定义资产模型层次结构](#)。

6. 选择 Save (保存)。

## 更新资产或组件模型 (AWS CLI)

您可以使用 AWS Command Line Interface (AWS CLI) 更新资产模型或组件模型。

使用 [UpdateAsset模型](#) API 更新资产模型或组件模型的名称、描述和属性。仅对于资产模型，您可以更新层次结构。指定以下参数：

- `assetModelId` – 资产的 ID。这是 UUID 格式的实际 ID，或者 `externalId:myExternalId` 如果有，则为。有关更多信息，请参阅《AWS IoT SiteWise 用户指南》中的 [使用外部 ID 引用对象](#)。

在有效载荷中指定更新的模型。要了解资产模型或组件模型的预期格式，请参阅 [创建资产模型](#)。

### Warning

[UpdateAsset模型](#) API 会使用您在有效载荷中提供的模型覆盖现有模型。为避免删除模型的属性或层次结构，您必须在更新的模型负载中包含它们的 ID 和定义。要了解如何查询模型的现有结构，请参阅 [DescribeAsset模型](#) 操作。

### Note

以下过程只能更新类型的复合模型 AWS/ALARM。如果要更新 CUSTOM 复合模型，请改用 [UpdateAssetModelComposite模型](#)。有关更多信息，请参阅 [更新自定义复合模型 \(组件\)](#)。



## 更新资产模型或组件模型 (AWS CLI)

1. 运行以下命令以检索现有模型定义。将 `asset-model-id` 替换为要更新的资产模型或组件模型的 ID 或外部 ID。

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

该操作返回包含模型详细信息的响应。该响应具有以下结构。

```
{
  "assetModelId": "String",
  "assetModelArn": "String",
  "assetModelName": "String",
  "assetModelDescription": "String",
  "assetModelProperties": Array of AssetModelProperty,
  "assetModelHierarchies": Array of AssetModelHierarchyDefinition,
  "assetModelCompositeModels": Array of AssetModelCompositeModel,
  "assetModelCompositeModelSummaries": Array of AssetModelCompositeModelSummary,
  "assetModelCreationDate": "String",
  "assetModelLastUpdateDate": "String",
  "assetModelStatus": {
    "state": "String",
    "error": {
      "code": "String",
      "message": "String"
    }
  },
  "assetModelType": "String"
}
```

有关更多信息，请参阅[DescribeAsset模型](#)操作。

2. 创建一个名为 `update-asset-model.json` 的文件，并将上一命令的响应复制到该文件中。
3. 从 `update-asset-model.json` 中的 JSON 对象中删除以下键/值对：
  - `assetModelId`
  - `assetModelArn`
  - `assetModelCompositeModelSummaries`
  - `assetModelCreationDate`
  - `assetModelLastUpdateDate`

- `assetModelStatus`
- `assetModelType`

[UpdateAsset模型](#)操作需要具有以下结构的有效负载：

```
{
  "assetModelName": "String",
  "assetModelDescription": "String",
  "assetModelProperties": Array of AssetModelProperty,
  "assetModelHierarchies": Array of AssetModelHierarchyDefinition,
  "assetModelCompositeModels": Array of AssetModelCompositeModel
}
```

4. 在 `update-asset-model.json` 中，执行以下任何操作：
  - 更改资产模型的名称 (`assetModelName`)。
  - 更改、添加或删除资产模型的描述 (`assetModelDescription`)。
  - 更改、添加或删除资产模型的任何属性 (`assetModelProperties`)。您无法更改现有属性的 `dataType` 或现有指标的 `window`。有关更多信息，请参阅 [定义数据属性](#)。
  - 更改、添加或删除资产模型的任何层次结构 (`assetModelHierarchies`)。您无法更改现有层次结构的 `childAssetModelId`。有关更多信息，请参阅 [定义资产模型层次结构](#)。
  - 更改、添加或删除资产模型中任何类型为 AWS/ALARM (`assetModelCompositeModels`) 的复合模型。警报会监控其他特性，以便您可以识别何时需要注意设备或流程。每个警报定义都是一个复合模型，用于标准化警报使用的属性集。有关更多信息，请参阅 [通过警报监控数据](#) 和 [在资产模型上定义警报](#)。
5. 运行以下命令，使用存储在 `update-asset-model.json` 中的定义更新资产模型。将 `#### ID #####` 的 ID：

```
aws iotsitewise update-asset-model \
  --asset-model-id asset-model-id \
  --cli-input-json file://model-payload.json
```

## 更新自定义复合模型（组件）

您可以使用 AWS IoT SiteWise API 更新自定义复合模型，也可以使用 AWS IoT SiteWise 控制台更新组件。

## 主题

- [更新组件 \(控制台\)](#)
- [更新自定义复合模型 \(AWS CLI\)](#)

## 更新组件 (控制台)

您可以使用 AWS IoT SiteWise 控制台更新组件。

### 更新组件 (控制台)

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 **模型**。
3. 选择组件所在的资产模型。
4. 在“属性”选项卡上，选择“组件”。
5. 选择要更新的组件。
6. 选择编辑。
7. 在“编辑组件”页面上，执行以下任一操作：
  - 在 **模型详细信息** 中，更改模型的 **名称**。
  - 更改任何 **属性定义**。您无法更改现有属性的 **数据类型**。有关更多信息，请参阅 [定义静态数据 \(属性\)](#)。
  - 更改任何 **测量值定义**。您无法更改现有测量值的 **数据类型**。有关更多信息，请参阅 [定义来自设备的数据流 \(测量\)](#)。
  - 更改任何 **转换定义**。有关更多信息，请参阅 [转换数据 \(转换\)](#)。
  - 更改任何 **指标定义**。您无法更改现有指标的 **时间间隔**。有关更多信息，请参阅 [聚合来自属性和其他资产 \(指标\) 的数据](#)。
8. 选择 **Save (保存)**。

## 更新自定义复合模型 (AWS CLI)

您可以使用 AWS Command Line Interface (AWS CLI) 更新自定义复合模型。

要更新名称或描述，请使用 [UpdateAssetModelComposite模型](#) 操作。您也可以更新属性（仅限于内联自定义复合模型）。您无法更新 component-model-based 自定义复合模型的属性，因为其引用的组件模型提供了其关联的属性。

### ⚠ Important

如果您从自定义复合模型中移除某个属性，则 AWS IoT SiteWise 会删除该属性的所有先前数据。您无法更改现有属性的类型或数据类型。

要将现有的复合模型属性替换为具有相同属性的新属性name，请执行以下操作：

1. 提交移除全部现有属性的UpdateAssetModelCompositeModel请求。
2. 提交第二份包含新房产的UpdateAssetModelCompositeModel申请。新的资产属性将与前一个资产属性name相同，AWS IoT SiteWise 并将生成一个新的唯一资产id。

## 更新自定义复合模型 (AWS CLI)

1. 要检索现有的复合模型定义，请运行以下命令。将 `composite-model-id` 替换为要更新的自定义复合模型的 ID 或外部 ID，将 `asset-model-id` 替换为与自定义复合模型关联的资产模型。有关更多信息，请参阅《AWS IoT SiteWise 用户指南》。

```
aws iotsitewise describe-asset-model-composite-model \
--asset-model-composite-model-id composite-model-id \
--asset-model-id asset-model-id
```

有关更多信息，请参阅[DescribeAssetModelCompositeModel](#)操作。

2. 创建一个名为 `update-custom-composite-model.json` 的文件，然后将上一个命令的响应复制到该文件中。
3. 从中的 JSON 对象中移除所有键值对，但以下字段 `update-custom-composite-model.json` 除外：
  - `assetModelCompositeModelName`
  - `assetModelCompositeModelDescription` ( 如果存在 )
  - `assetModelCompositeModelProperties` ( 如果存在 )
4. 在 `update-custom-composite-model.json` 中，执行以下任何操作：
  - 更改的值 `assetModelCompositeModelName`。
  - 添加或移 `assetModelCompositeModelDescription` 除或更改其值。
  - 仅适用于内联自定义复合模型：在中更改、添加或移除资产模型的任何属性 `assetModelCompositeModelProperties`。

有关此文件所需格式的更多信息，请参阅[UpdateAssetModelComposite模型](#)的请求语法。

5. 运行以下命令，使用存储在中的定义更新自定义复合模型update-custom-composite-model.json。用##### ID 替换 composite-model-id，用其所在的##### ID ## asset-model-id。

```
aws iotsitewise update-asset-model-composite-model \  
--asset-model-composite-model-id composite-model-id \  
--asset-model-id asset-model-id \  
--cli-input-json file://update-custom-composite-model.json
```

## 删除资产和模型

当你使用完资源和模型 AWS IoT SiteWise 后，你可以将其删除。删除操作是异步的，需要一段时间才能传 AWS IoT SiteWise播。

### 主题

- [删除资产](#)
- [删除资产模型](#)

## 删除资产

您可以使用 AWS IoT SiteWise 控制台或 API 删除资产。

您必须先取消资产与其子资产和父资产的关联，然后才能删除该资产。有关更多信息，请参阅 [关联资产和取消资产关联](#)。如果使用 AWS Command Line Interface (AWS CLI)，则可以使用“[ListAssociated资产](#)”操作列出资产的子项。

删除资产时，资产的状态为 DELETING，直至更改传播为止。有关更多信息，请参阅 [资产和模型状态](#)。删除资产后，您将无法再查询该资产。如果查询，API 将返回 HTTP 404 响应。

### Important

AWS IoT SiteWise 删除已删除资产的所有属性数据。

### 主题

- [删除资产 \(控制台\)](#)
- [删除资产 \(AWS CLI\)](#)

## 删除资产 (控制台)

您可以使用 AWS IoT SiteWise 控制台删除资产。

### 删除资产 (控制台)

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 资产。
3. 选择要删除的资产。

#### Tip

您可以选择箭头图标来展开资产层次结构以查找资产。

4. 如果该资产具有任何 关联的资产，请删除每个资产。您可以选择资产的名称以导航到其页面，从中可以将其删除。
5. 在资产页面上，选择 删除。
6. 在“删除资源”对话框中，执行以下操作：
  - a. 输入 **Delete** 以确认删除。
  - b. 选择删除。

## 删除资产 (AWS CLI)

您可以使用 AWS Command Line Interface (AWS CLI) 删除资产。

使用 [DeleteAsset](#) 操作删除资产。指定以下参数：

- `assetId` – 资产的 ID。这是 UUID 格式的实际 ID，或者 `externalId:myExternalId` 如果有，则为。有关更多信息，请参阅《AWS IoT SiteWise 用户指南》中的 [使用外部 ID 引用对象](#)。

### 要删除资产 (AWS CLI)

1. 运行以下命令以列出资产的层次结构。将 **## ID #####** 的 ID 或外部 ID：

```
aws iotsitewise describe-asset --asset-id asset-id
```

此操作将返回包含资产详细信息的响应。响应包含一个具有以下结构的assetHierarchies列表：

```
{
  ...
  "assetHierarchies": [
    {
      "id": "String",
      "name": "String"
    }
  ],
  ...
}
```

有关更多信息，请参阅[DescribeAsset](#)操作。

- 对于每个层次结构，运行以下命令以列出与该层次结构关联的资产子项。将## *ID* 替换为资产的 ID 或外部 ID，将层次结构 *ID #####* 的 ID 或外部 ID。

```
aws iotsitewise list-associated-assets \
  --asset-id asset-id \
  --hierarchy-id hierarchy-id
```

有关更多信息，请参阅“[ListAssociated资产](#)”操作。

- 运行以下命令以删除每个关联的资产，然后删除该资产。将## *ID #####* 的 ID 或外部 ID。

```
aws iotsitewise delete-asset --asset-id asset-id
```

## 删除资产模型

您可以使用 AWS IoT SiteWise 控制台或 API 删除资产模型。

在删除资产模型之前，必须先删除从该资产模型中创建的所有资产。

删除资产模型时，它的状态为 DELETING，直至更改传播为止。有关更多信息，请参阅[资产和模型状态](#)。删除资产模型后，您将无法再查询该资产模型。如果查询，API 将返回 HTTP 404 响应。

## 主题

- [删除资产模型 \(控制台\)](#)
- [删除资产模型 \(AWS CLI\)](#)

### 删除资产模型 (控制台)

您可以使用 AWS IoT SiteWise 控制台删除资产模型。

#### 删除资产模型 (控制台)

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 **模型**。
3. 选择要删除的资产模型。
4. 如果模型具有任何资产，请删除每个资产。选择资产的名称以导航到其页面，您可以从中将其删除。有关更多信息，请参阅 [删除资产 \(控制台\)](#)。
5. 在模型页面上，选择 **删除**。
6. 在删除模型对话框中，执行以下操作：
  - a. 输入 **Delete** 以确认删除。
  - b. 选择 **删除**。

### 删除资产模型 (AWS CLI)

您可以使用 AWS Command Line Interface (AWS CLI) 删除资产模型。

使用 [DeleteAsset模型](#) 操作删除资产模型。指定以下参数：

- `assetModelId` – 资产的 ID。这是 UUID 格式的实际 ID，或者 `externalId:myExternalId` 如果有，则为。有关更多信息，请参阅《AWS IoT SiteWise 用户指南》中的 [使用外部 ID 引用对象](#)。

#### 要删除资产模型 (AWS CLI)

1. 运行以下命令以列出从模型创建的所有资产。将 `#### ID #####` 的 ID 或外部 ID。

```
aws iotsitewise list-assets --asset-model-id asset-model-id
```

有关更多信息，请参阅 [ListAssets](#) 操作。



2. 如果上一命令返回模型中的任何资产，请删除每个资产。有关更多信息，请参阅 [删除资产 \(AWS CLI\)](#)。
3. 运行以下命令以删除资产模型。将 `#### ID #####` 的 ID 或外部 ID。

```
aws iotsitewise delete-asset-model --asset-model-id asset-model-id
```

## 使用资产和模型进行批量操作

要处理大量资产或资产模型，请使用批量操作将资源批量导入和导出到其他位置。例如，您可以创建一个数据文件来定义 Amazon S3 存储桶中的资产或资产模型，然后使用批量导入在中创建或更新它们 AWS IoT SiteWise。或者，如果您有大量资产或资产模型 AWS IoT SiteWise，则可以将其导出到 Amazon S3。

### Note

您可以 AWS IoT SiteWise 通过调用 AWS IoT TwinMaker API 中的操作在中执行批量操作。无需设置 AWS IoT TwinMaker 或创建 AWS IoT TwinMaker 工作区即可执行此操作。您只需要一个 Amazon S3 存储桶，您可以在其中放置您的 AWS IoT SiteWise 内容。

### 主题

- [关键概念和术语](#)
- [支持的功能](#)
- [批量操作的先决条件](#)
- [运行批量导入任务](#)
- [运行批量导出任务](#)
- [作业进度跟踪和错误处理](#)
- [导入元数据示例](#)
- [导出元数据示例](#)
- [AWS IoT SiteWise 元数据传输任务架构](#)

## 关键概念和术语

AWS IoT SiteWise 批量导入和导出功能依赖于以下概念和术语：

- 导入：将资产或资产模型从 Amazon S3 存储桶中的文件移动到 AWS IoT SiteWise。
- 导出：将资产或资产模型从 Amazon S3 存储桶移 AWS IoT SiteWise 至 Amazon S3 存储桶的操作。
- 来源：您要从中移动内容的起始位置。

例如，Amazon S3 存储桶既是导入源，又 AWS IoT SiteWise 是导出源。

- 目的地：您要将内容移动到的目标位置。

例如，Amazon S3 存储桶是导出目的地，AWS IoT SiteWise 也是导入目的地。

- AWS IoT SiteWise 架构：此架构用于从中导入和导出元数据 AWS IoT SiteWise。
- 顶级资源：可以单独创建或更新的 AWS IoT SiteWise 资源，例如资产或资产模型。
- 子资源：顶级 AWS IoT SiteWise 资源中的嵌套资源。示例包括属性、层次结构和复合模型。
- 元数据：成功导入或导出资源所需的关键信息。元数据的示例包括资产和资产模型的定义。
- 元数据 TransferJob：运行时创建的对象 CreateMetadataTransferJob。

## 支持的功能

本主题说明了在运行批量操作时可以做什么。批量操作支持以下功能：

- 顶级资源创建：当您导入未定义 ID 或其 ID 与现有资源或资产模型不匹配的资产或资产模型时，它将被创建为新资源。
- 顶级资源替换：当您导入的资产或资产模型的 ID 与已存在的资产或资产模型相匹配时，它将替换现有资源。
- 子资源的创建、替换或删除：当您的导入替换顶级资源（例如资产或资产模型）时，新定义将替换所有子资源，例如属性、层次结构或复合模型。

例如，如果您在批量导入期间更新了资产模型，并且更新后的版本定义了原始版本中不存在的属性，则会创建一个新属性。如果它定义了一个已经存在的属性，则现有属性将被更新。如果更新的资产模型省略了原始资产模型中存在的属性，则该属性将被删除。

- 不删除顶级资源：批量操作不会删除资产或资产模型。批量操作只能创建或更新它们。

## 批量操作的先决条件

本节介绍批量操作的先决条件，包括在本地计算机之间 AWS 服务 交换资源的 AWS Identity and Access Management (IAM) 权限。在开始批量操作之前，请完成以下先决条件：

- 创建 Amazon S3 存储桶来存储资源。有关使用 Amazon S3 的更多信息，请参阅[什么是亚马逊 S3？](#)

## IAM 权限

要执行批量操作，您必须创建一个 AWS Identity and Access Management (IAM) 策略，该策略具有允许 Amazon S3 和您的本地计算机之间交换 AWS 资源的权限。AWS IoT SiteWise 有关创建 IAM 策略的更多信息，请参阅[创建 IAM 策略](#)。

要执行批量操作，您需要以下策略。

### AWS IoT SiteWise 政策

此策略允许访问批量操作所需 AWS IoT SiteWise 的 API 操作：

```
{
  "Sid": "SiteWiseApiAccess",
  "Effect": "Allow",
  "Action": [
    "iotsitewise:CreateAsset",
    "iotsitewise:CreateAssetModel",
    "iotsitewise:UpdateAsset",
    "iotsitewise:UpdateAssetModel",
    "iotsitewise:UpdateAssetProperty",
    "iotsitewise:ListAssets",
    "iotsitewise:ListAssetModels",
    "iotsitewise:ListAssetProperties",
    "iotsitewise:ListAssetModelProperties",
    "iotsitewise:ListAssociatedAssets",
    "iotsitewise:DescribeAsset",
    "iotsitewise:DescribeAssetModel",
    "iotsitewise:DescribeAssetProperty",
    "iotsitewise:AssociateAssets",
    "iotsitewise:DisassociateAssets",
    "iotsitewise:AssociateTimeSeriesToAssetProperty",
    "iotsitewise:DisassociateTimeSeriesFromAssetProperty",
    "iotsitewise:BatchPutAssetPropertyValue",
    "iotsitewise:BatchGetAssetPropertyValue",
    "iotsitewise:TagResource",
    "iotsitewise:UntagResource",
    "iotsitewise:ListTagsForResource",
    "iotsitewise:CreateAssetModelCompositeModel",
    "iotsitewise:UpdateAssetModelCompositeModel",
    "iotsitewise:DescribeAssetModelCompositeModel",
```

```

    "iotsitewise:DeleteAssetModelCompositeModel",
    "iotsitewise:ListAssetModelCompositeModels",
    "iotsitewise:ListCompositionRelationships",
    "iotsitewise:DescribeAssetCompositeModel"
  ],
  "Resource": "*"
}

```

## AWS IoT TwinMaker 政策

此策略允许访问您用来处理批量操作的 AWS IoT TwinMaker API 操作：

```

{
  "Sid": "MetadataTransferJobApiAccess",
  "Effect": "Allow",
  "Action": [
    "iottwinmaker:CreateMetadataTransferJob",
    "iottwinmaker:CancelMetadataTransferJob",
    "iottwinmaker:GetMetadataTransferJob",
    "iottwinmaker:ListMetadataTransferJobs"
  ],
  "Resource": "*"
}

```

## 亚马逊 S3 政策

此策略允许访问 Amazon S3 存储桶，以便传输用于批量操作的元数据。

For a specific Amazon S3 bucket

如果您使用一个特定的存储桶来处理批量操作元数据，则此策略提供对该存储桶的访问权限：

```

{
  "Effect": "Allow",
  "Action": [
    "s3:PutObject",
    "s3:GetObject",
    "s3:GetBucketLocation",
    "s3:ListBucket",
    "s3:AbortMultipartUpload",
    "s3:ListBucketMultipartUploads",
    "s3:ListMultipartUploadParts"
  ],

```

```
"Resource": [
  "arn:aws:s3:::bucket name",
  "arn:aws:s3:::bucket name/*"
]
```

To allow any Amazon S3 bucket

如果您要使用许多不同的存储桶来处理批量操作元数据，则此策略提供对任何存储桶的访问权限：

```
{
  "Effect": "Allow",
  "Action": [
    "s3:PutObject",
    "s3:GetObject",
    "s3:GetBucketLocation",
    "s3:ListBucket",
    "s3:AbortMultipartUpload",
    "s3:ListBucketMultipartUploads",
    "s3:ListMultipartUploadParts"
  ],
  "Resource": "*"
}
```

有关对导入和导出操作进行故障排除的信息，请参见[批量导入和导出疑难解答](#)。

## 运行批量导入任务

批量导入是将元数据移动到 AWS IoT SiteWise 工作区的操作。例如，批量导入可以将元数据从本地文件或 Amazon S3 存储桶中的文件移动到 AWS IoT SiteWise 工作空间。

### 步骤 1：准备要导入的文件

下载本 AWS IoT SiteWise 机格式文件以导入资源和资产模型。有关更多信息，请参阅[AWS IoT SiteWise 元数据传输任务架构](#)。

### 第 2 步：将准备好的文件上传到 Amazon S3

将文件上传到亚马逊 S3。有关详细信息，[请参阅《亚马逊简单存储服务用户指南》中的将文件上传到 Amazon S3](#)。

## 导入元数据 ( 控制台 )

您可以使用批量导入元数据。AWS IoT SiteWise 控制台 按照[步骤 1：准备要导入的文件](#)和[第 2 步：将准备好的文件上传到 Amazon S3](#)准备一个可以导入的文件。

将数据从 Amazon S3 导入到 AWS IoT SiteWise 控制台

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 从导航窗格中选择批量操作新建。
3. 选择“新建导入”以开始导入过程。
4. 在导入元数据页面上：
  - 选择“浏览亚马逊 S3”以查看亚马逊 S3 存储桶和文件。
  - 导航到包含准备好的导入文件的 Amazon S3 存储桶。
  - 选择要导入的文件。
  - 查看所选文件，然后选择“导入”。
5. 的“SiteWise 元数据批量操作”页面在“作业进度”表中 AWS IoT SiteWise 控制台 显示新创建的导入任务。

## 导入元数据 (AWS CLI)

要执行导入操作，请按以下步骤操作：

将数据从 Amazon S3 导入到 AWS CLI

1. 按照以下步骤创建指定要导入的资源的元数据文件[AWS IoT SiteWise 元数据传输任务架构](#)。将此文件存储在您的 Amazon S3 存储桶中。

有关要导入的元数据文件的示例，请参阅[导入元数据示例](#)。

2. 现在使用请求正文创建一个 JSON 文件。请求正文指定传输任务的来源和目的地。此文件与上一步中的文件是分开的。请务必将您的 Amazon S3 存储桶指定为源和目标。

以下示例显示了请求正文：

```
{
  "metadataTransferJobId": "your-transfer-job-Id",
  "sources": [{
    "type": "s3",
```

```

    "s3Configuration": {
      "location": "arn:aws:s3:::your-S3-bucket-name/
your_import_metadata.json"
    }
  }],
  "destination": {
    "type": "iotsitewise"
  }
}

```

3. CreateMetadataTransferJob通过运行以下 AWS CLI 命令来调用。在此示例中，上一步中的请求正文文件名为createMetadataTransferJobExport.json。

```

aws iottwinmaker create-metadata-transfer-job --region us-east-1 \
--cli-input-json file://createMetadataTransferJobImport.json

```

这将创建一个元数据传输任务，并开始传输所选资源的过程。

## 运行批量导出任务

批量导出是将元数据从 AWS IoT SiteWise 工作空间移动到 Amazon S3 存储桶的操作。

将 AWS IoT SiteWise 内容批量导出到 Amazon S3 时，您可以指定筛选条件来限制要导出的特定资产模型和资产。

必须在 JSON 请求的 iotSiteWiseConfiguration “来源” 部分中指定过滤器。

### Note

您可以在请求中包含多个筛选条件。批量操作将导出与任何筛选条件匹配的资产模型和资产。如果您不提供任何筛选条件，则批量操作会导出您的所有资产模型和资产。

### Example 带过滤器的请求正文

```

{
  "metadataTransferJobId": "your-transfer-job-id",
  "sources": [
    {

```

```
    "type": "iotsitewise",
    "iotSiteWiseConfiguration": {
      "filters": [
        {
          "filterByAssetModel": {
            "assetModelId": "asset model ID"
          }
        },
        {
          "filterByAssetModel": {
            "assetModelId": "asset model ID",
            "includeAssets": true
          }
        },
        {
          "filterByAssetModel": {
            "assetModelId": "asset model ID",
            "includeOffspring": true
          }
        }
      ]
    },
    "destination": {
      "type": "s3",
      "s3Configuration": {
        "location": "arn:aws:s3:::your-S3-bucket-location"
      }
    }
  }
}
```

## 导出元数据 (控制台)

以下过程说明了控制台导出操作：

在中创建导出任务 AWS IoT SiteWise 控制台

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 从导航窗格中选择批量操作新建。
3. 选择“新建导出”以开始导出过程。
4. 在导出元数据页面上：



- 输入导出任务的名称。这是您的 Amazon S3 存储桶中导出的文件所使用的名称。
  - 选择要导出的资源，这将为任务设置筛选条件：
    - 导出所有资产和资产模型。对资产和资产模型使用筛选器。
    - 导出资产。筛选您的资产。
      - 选择要用于导出筛选器的资源。
      - ( 可选 ) 添加后代或关联的资产模型。
    - 导出资产模型。根据您的资产模型进行筛选。
      - 选择要用于导出筛选器的资产模型。
      - ( 可选 ) 添加后代或关联资产，或两者兼而有之。
    - 选择下一步。
  - 导航到亚马逊 S3 存储桶：
    - 选择“浏览亚马逊 S3”以查看亚马逊 S3 存储桶和文件。
    - 导航到必须存放文件的 Amazon S3 存储桶。
    - 选择下一步。
  - 查看导出任务并选择导出。
5. 的“SiteWise 元数据批量操作”页面在“作业进度”表中 AWS IoT SiteWise 控制台 显示新创建的导入任务。

有关导出元数据时使用筛选器的不同方法，请参阅[导出元数据示例](#)。

## 导出元数据 (AWS CLI)

以下过程解释了 AWS CLI 导出操作：

将数据从导出 AWS IoT SiteWise 到 Amazon S3

1. 使用您的请求正文创建 JSON 文件。请求正文指定传输任务的来源和目的地。以下示例显示了一个请求正文示例：

```
{
  "metadataTransferJobId": "your-transfer-job-Id",
  "sources": [{
    "type": "iotsitewise"
  }],
}
```

```
"destination": {
  "type": "s3",
  "s3Configuration": {
    "location": "arn:aws:s3:::your-S3-bucket-location"
  }
}
```

请务必将您的 Amazon S3 存储桶指定为元数据传输任务的目的地。

#### Note

此示例将导出您的所有资产模型和资产。要将导出限制为特定的资产模型或资产，您可以在请求正文中加入筛选器。有关应用导出筛选器的更多信息，请参阅[导出元数据示例](#)。

2. 保存您的请求正文文件，以便在下一步中使用。在本示例中，文件命名为 `createMetadataTransferJobExport.json`。
3. `CreateMetadataTransferJob` 通过运行以下 AWS CLI 命令来调用：

```
aws iottwinmaker create-metadata-transfer-job --region us-east-1 \
  --cli-input-json file://createMetadataTransferJobExport.json
```

`createMetadataTransferJobExport.json` 使用您自己的传输文件名替换输入的 JSON 文件。

## 作业进度跟踪和错误处理

批量处理作业需要时间来处理。每个任务都按照 AWS IoT SiteWise 收到请求的顺序进行处理。它 `one-at-a-time` 针对每个账户进行处理。任务完成后，队列中的下一个任务会自动开始处理。AWS IoT SiteWise 异步解析任务，并在任务进行时更新每个任务的状态。每个作业都有一个状态字段，其中包含资源状态和错误消息（如果适用）。

状态可以是以下值之一：

- `VALIDATING`— 验证作业，包括提交的文件格式及其内容。
- `PENDING`— 作业在队列中。您可以从 AWS IoT SiteWise 控制台取消处于此状态的作业，但所有其他状态都将持续到最后。

- **RUNNING**— 处理作业。它正在创建和更新导入文件所定义的资源，或者根据所选的导出任务筛选器导出资源。如果取消，则不会删除此任务导入的任何资源。请参阅[查看任务进度和详细信息 \(控制台\)](#) 了解更多信息。
- **CANCELLING**— 任务正在被主动取消。
- **ERROR**— 一个或多个资源处理失败。有关更多信息，请查看详细的工作报告。请参阅[检查错误详情 \(控制台\)](#) 了解更多信息。
- **COMPLETED**— Job 已完成，没有错误。
- **CANCELLED**— 作业已取消且未排队。如果您取消了某项RUNNING任务，则不会从中删除取消时该任务已导入的资源 AWS IoT SiteWise。

## 主题

- [作业进度跟踪](#)
- [检查错误](#)

## 作业进度跟踪

查看任务进度和详细信息 (控制台)

[导出元数据 \(控制台\)](#) 要启动批量作业，请参阅[导入元数据 \(控制台\)](#) 或。

AWS IoT SiteWise 控制台中的 Job 进度概览：

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 从导航窗格中选择批量操作新建。
3. AWS IoT SiteWise 控制台中的作业进度表显示了批量操作任务的列表。
4. Job type 列描述它是导出任务还是导入任务。导入日期列显示作业的开始日期。
5. 状态列显示作业的状态。您可以选择作业以查看有关该作业的详细信息。
6. 所选作业成功后会显示成功，如果作业失败，则显示失败列表。每种资源类型还会显示错误描述。

AWS IoT SiteWise 控制台中的 Job 详情概述：

AWS IoT SiteWise 控制台中的作业进度表显示了批量操作任务的列表。

1. 选择一份工作以查看更多详情。
2. 对于导入任务，Data source ARN表示导入文件的 Amazon S3 位置。

- 对于导出任务，Data destination ARN表示文件在导出后的 Amazon S3 位置。
- Status和Status reason，提供有关当前工作的更多详细信息。有关更多信息，请参阅[作业进度跟踪和错误处理](#)。
- Queued position表示作业在流程队列中的位置。一次只能处理一个作业。排队位置为 1 表示接下来将处理该作业。
- 作业详细信息页面还会显示作业进度计数。
  - 作业进度计数类型有：
    - Total resources— 表示转移过程中的资产总数。
    - Succeeded— 表示在此过程中成功转移的资产数量。
    - Failed— 表示在此过程中失败的资产数量。
    - Skipped— 表示在此过程中跳过的资产数量。
- 作业状态为PENDING或VALIDATING，则显示所有作业进度计为-。这表示正在评估作业进度计数。
- 作业状态RUNNING显示计Total resources数，即已提交待处理的作业。详细计数 ( SucceededFailed、和Skipped ) 适用于已处理的资源。在任务的状态为COMPLETED或之前，详细Total resources计数的总和小于计数ERROR。
- 如果作业的状态为COMPLETED或ERROR，则Total resources计数等于详细计数 ( SucceededFailed、和Skipped ) 的总和。
- 如果作业的状态为ERROR，请查看 Job failures 表，了解有关特定错误和失败的详细信息。有关更多信息，请参阅[检查错误详情 \( 控制台 \)](#)。

### 查看工作进度和详细信息 (AWS CLI)

开始批量操作后，您可以使用以下 API 操作检查或更新其状态：

- 要检索有关特定任务的信息，请使用 [GetMetadataTransferJob](#) API 操作。

使用 **GetMetadataTransferJob** API 检索信息：

- 创建并运行传输作业。调用 GetMetadataTransferJob API。

Example AWS CLI 命令：

```
aws iottwinmaker get-metadata-transfer-job \
```

```
--metadata-transfer-job-id your_metadata_transfer_job_id \  
--region your_region
```

2. GetMetadataTransferJobAPI 返回一个带有以下参数的MetadataTransferJobProgress对象：

- succeededCount — 表示在此过程中成功转移的资产数量。
- failedCount — 表示在此过程中失败的资产数量。
- skippedCount — 表示在此过程中跳过的资产数量。
- TotalCount — 表示转移过程中的资产总数。

这些参数表示任务进度状态。如果状态为RUNNING，它们会帮助跟踪仍有待处理的资源数量。

如果您遇到架构验证错误，或者 failedCount 大于或等于 1，则作业进度状态将变为。ERROR任务的完整错误报告已放入您的 Amazon S3 存储桶中。有关更多信息，请参阅[检查错误](#)。

- 要列出当前作业，请使用 [ListMetadataTransferJobsAPI](#) 操作。

使用 JSON 文件根据返回的任务的当前状态对其进行筛选。请参阅以下步骤：

1. 要指定要使用的过滤器，请创建一个 AWS CLI 输入 JSON 文件。要使用：

```
{  
  "sourceType": "s3",  
  "destinationType": "iottwinmaker",  
  "filters": [{  
    "state": "COMPLETED"  
  }]  
}
```

有关有效state值的列表，请参阅 AWS IoT TwinMaker API 参考指南中的[ListMetadataTransferJobs筛选器](#)。

2. 在以下 AWS CLI 示例命令中使用 JSON 文件作为参数：

```
aws iottwinmaker list-metadata-transfer-job --region your_region \  
--cli-input-json file://ListMetadataTransferJobsExample.json
```

- 要取消任务，请使用 [CancelMetadataTransferJobAPI](#) 操作。此 API 取消了特定的元数据传输任务，而不会影响任何已导出或导入的资源：

```
aws iottwinmaker cancel-metadata-transfer-job \  
  --region your_region \  
  --metadata-transfer-job-id job-to-cancel-id
```

## 检查错误

### 检查错误详情 (控制台)

AWS IoT SiteWise 控制台中的错误详情：

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 有关批量操作作业的列表，AWS IoT SiteWise 控制台 请参阅中的作业进度表。
3. 选择作业以查看作业详细信息。
4. 如果作业的状态为COMPLETED或ERROR，则Total resources计数等于详细计数 ( SucceededFailed、和Skipped ) 的总和。
5. 如果作业的状态为ERROR，请查看 Job failures 表，了解有关特定错误和失败的详细信息。
6. Job 失败表显示了作业报告中的内容。该Resource type字段表示错误或失败的位置，如下所示：
  - 例如，Resource type字段Bulk operations template中的验证错误表示导入模板和元数据架构文件格式不匹配。请参阅[AWS IoT SiteWise 元数据传输任务架构](#)了解更多信息。
  - 该Resource type字段Asset中的失败表示该资产不是因为与其他资产发生冲突而创建的。有关 AWS IoT SiteWise 资源[错误和冲突的信息](#)，请参阅[常见错误](#)。

### 检查错误详情 (AWS CLI)

要处理和诊断传输任务期间产生的错误，请参阅以下有关使用 GetMetadataTransferJob API 操作的过程：

1. 创建并运行转接任务后，请致电 [GetMetadataTransferJob](#)：

```
aws iottwinmaker get-metadata-transfer-job \  
  --metadata-transfer-job-id your_metadata_transfer_job_id \  
  --region us-east-1
```

2. 看到作业状态变为后COMPLETED，就可以开始验证作业的结果了。

3. 当您调用 `GetMetadataTransferJob` 时，它会返回一个名为的对象 [MetadataTransferJobProgress](#)。

该 `MetadataTransferJobProgress` 对象包含以下参数：

- `failedCount`：表示在转移过程中失败的资产数量。
  - `skippedCount`：表示在转移过程中跳过的资产数量。
  - `succeededCount`：表示在转移过程中成功的资产数量。
  - `TotalCount`：表示转移过程中涉及的资产总数。
4. 此外，API 调用还会返回一个包含预签名 URL 的元素 `reportUrl`。如果您的调动任务有任何问题需要进一步调查，请访问此网址。

## 导入元数据示例

本节介绍如何创建元数据文件以通过单个批量导入操作导入资产模型和资产。

### 批量导入示例

您可以通过一次批量导入操作导入许多资产模型和资产。以下示例说明如何创建元数据文件来执行此操作。

在此示例场景中，您有各种工作场所，在工作单元中包含工业机器人。

该示例定义了两种资产模型：

- `RobotModel1`：此资产模型代表您在工作场所拥有的一种特定类型的机器人。机器人具有测量属性 `Temperature`。
- `WorkCell`：此资产模型代表您的一个工作场所中的一组机器人。资产模型定义了一个层次结构 `robotHierarchyOEM1`，以表示工作单元包含机器人的关系。

该示例还定义了一些资产：

- `WorkCell1`：波士顿基地内的工作单元
- `RobotArm123456`：该工作单元中的一个机器人
- `RobotArm987654`：那个工作单元里的另一个机器人

以下 JSON 元数据文件定义了这些资产模型和资产。使用此元数据进行批量导入会在其中创建资产模型和资产 AWS IoT SiteWise，包括它们的层次关系。

#### 用于导入的元数据文件

```
{
  "assetModels": [
    {
      "assetModelExternalId": "Robot.OEM1.3536",
      "assetModelName": "RobotModel1",
      "assetModelProperties": [
        {
          "dataType": "DOUBLE",
          "externalId": "Temperature",
          "name": "Temperature",
          "type": {
            "measurement": {
              "processingConfig": {
                "forwardingConfig": {
                  "state": "ENABLED"
                }
              }
            }
          },
          "unit": "fahrenheit"
        }
      ]
    },
    {
      "assetModelExternalId": "ISA95.WorkCell",
      "assetModelName": "WorkCell",
      "assetModelProperties": [],
      "assetModelHierarchies": [
        {
          "externalId": "workCellHierarchyWithOEM1Robot",
          "name": "robotHierarchyOEM1",
          "childAssetModelExternalId": "Robot.OEM1.3536"
        }
      ]
    }
  ],
  "assets": [
    {
      "assetExternalId": "Robot.OEM1.3536.123456",
```



```

        "assetName": "RobotArm123456",
        "assetModelExternalId": "Robot.OEM1.3536"
    },
    {
        "assetExternalId": "Robot.OEM1.3536.987654",
        "assetName": "RobotArm987654",
        "assetModelExternalId": "Robot.OEM1.3536"
    },
    {
        "assetExternalId": "BostonSite.Area1.Line1.WorkCell1",
        "assetName": "WorkCell1",
        "assetModelExternalId": "ISA95.WorkCell",
        "assetHierarchies": [
            {
                "externalId": "workCellHierarchyWithOEM1Robot",
                "childAssetExternalId": "Robot.OEM1.3536.123456"
            },
            {
                "externalId": "workCellHierarchyWithOEM1Robot",
                "childAssetExternalId": "Robot.OEM1.3536.987654"
            }
        ]
    }
]
}
}

```

## 模型和资产的初始加载示例

在此示例场景中，您的公司中有多个包含工业机器人的工作场所。

该示例定义了多个资产模型：

- **Sample\_Enterprise**— 此资产模型代表了这些网站所属的公司。资产模型定义了一个层次结构Enterprise to Site，以表示站点与企业的关系。
- **Sample\_Site**— 此资产模型代表公司内部的制造基地。资产模型定义了一个层次结构Site to Line，以表示线条与网站的关系。
- **Sample\_Welding Line**— 此资产模型代表工作场所内的装配线。资产模型定义了一个层次结构Line to Robot，以表示机器人与生产线的关系。
- **Sample\_Welding Robot**— 此资产模型代表您的工作场所中的一种特定类型的机器人。

该示例还根据资产模型定义资产。

- Sample\_AnyCompany Motor— 此资产是根据Sample\_Enterprise资产模型创建的。
- Sample\_Chicago— 此资产是根据Sample\_Site资产模型创建的。
- Sample\_Welding Line 1— 此资产是根据Sample\_Welding Line资产模型创建的。
- Sample\_Welding Robot 1— 此资产是根据Sample\_Welding Robot资产模型创建的。
- Sample\_Welding Robot 2— 此资产是根据Sample\_Welding Robot资产模型创建的。

以下 JSON 元数据文件定义了这些资产模型和资产。使用此元数据进行批量导入会在其中创建资产模型和资产 AWS IoT SiteWise，包括它们的层次关系。

用于载入资产和模型的 JSON 文件以供导入

```
{
  "assetModels": [
    {
      "assetModelExternalId": "External_Id_Welding_Robot",
      "assetModelName": "Sample_Welding Robot",
      "assetModelProperties": [
        {
          "dataType": "STRING",
          "externalId": "External_Id_Welding_Robot_Serial_Number",
          "name": "Serial Number",
          "type": {
            "attribute": {
              "defaultValue": "-"
            }
          },
          "unit": "-"
        },
        {
          "dataType": "DOUBLE",
          "externalId": "External_Id_Welding_Robot_Cycle_Count",
          "name": "CycleCount",
          "type": {
            "measurement": {}
          },
          "unit": "EA"
        },
        {
          "dataType": "DOUBLE",
          "externalId": "External_Id_Welding_Robot_Joint_1_Current",
```

```

        "name": "Joint 1 Current",
        "type": {
            "measurement": {}
        },
        "unit": "Amps"
    },
    {
        "dataType": "DOUBLE",
        "externalId": "External_Id_Welding_Robot_Joint_1_Max_Current",
        "name": "Max Joint 1 Current",
        "type": {
            "metric": {
                "expression": "max(joint1current)",
                "variables": [
                    {
                        "name": "joint1current",
                        "value": {
                            "propertyExternalId":
"External_Id_Welding_Robot_Joint_1_Current"
                        }
                    }
                ],
                "window": {
                    "tumbling": {
                        "interval": "5m"
                    }
                }
            }
        },
        "unit": "Amps"
    }
]
},
{
    "assetModelExternalId": "External_Id_Welding_Line",
    "assetModelName": "Sample_Welding_Line",
    "assetModelProperties": [
        {
            "dataType": "DOUBLE",
            "externalId": "External_Id_Welding_Line_Availability",
            "name": "Availability",
            "type": {
                "measurement": {}
            }
        },
    ],

```

```

        "unit": "%"
      }
    ],
    "assetModelHierarchies": [
      {
        "externalId": "External_Id_Welding_Line_T0_Robot",
        "name": "Line to Robot",
        "childAssetModelExternalId": "External_Id_Welding_Robot"
      }
    ]
  },
  {
    "assetModelExternalId": "External_Id_Site",
    "assetModelName": "Sample_Site",
    "assetModelProperties": [
      {
        "dataType": "STRING",
        "externalId": "External_Id_Site_Street_Address",
        "name": "Street Address",
        "type": {
          "attribute": {
            "defaultValue": "-"
          }
        },
        "unit": "-"
      }
    ],
    "assetModelHierarchies": [
      {
        "externalId": "External_Id_Site_T0_Line",
        "name": "Site to Line",
        "childAssetModelExternalId": "External_Id_Welding_Line"
      }
    ]
  },
  {
    "assetModelExternalId": "External_Id_Enterprise",
    "assetModelName": "Sample_Enterprise",
    "assetModelProperties": [
      {
        "dataType": "STRING",
        "name": "Company Name",
        "externalId": "External_Id_Enterprise_Company_Name",
        "type": {

```

```

        "attribute": {
            "defaultValue": "-"
        }
    },
    "unit": "-"
}
],
"assetModelHierarchies": [
    {
        "externalId": "External_Id_Enterprise_T0_Site",
        "name": "Enterprise to Site",
        "childAssetModelExternalId": "External_Id_Site"
    }
]
}
],
"assets": [
    {
        "assetExternalId": "External_Id_Welding_Robot_1",
        "assetName": "Sample_Welding Robot 1",
        "assetModelExternalId": "External_Id_Welding_Robot",
        "assetProperties": [
            {
                "externalId": "External_Id_Welding_Robot_Serial_Number",
                "attributeValue": "S1000"
            },
            {
                "externalId": "External_Id_Welding_Robot_Cycle_Count",
                "alias": "AnyCompany/Chicago/Welding Line/S1000/Count"
            },
            {
                "externalId": "External_Id_Welding_Robot_Joint_1_Current",
                "alias": "AnyCompany/Chicago/Welding Line/S1000/1/Current"
            }
        ]
    },
    {
        "assetExternalId": "External_Id_Welding_Robot_2",
        "assetName": "Sample_Welding Robot 2",
        "assetModelExternalId": "External_Id_Welding_Robot",
        "assetProperties": [
            {
                "externalId": "External_Id_Welding_Robot_Serial_Number",
                "attributeValue": "S2000"
            }
        ]
    }
]
}
]
}

```

```

    },
    {
      "externalId": "External_Id_Welding_Robot_Cycle_Count",
      "alias": "AnyCompany/Chicago/Welding Line/S2000/Count"
    },
    {
      "externalId": "External_Id_Welding_Robot_Joint_1_Current",
      "alias": "AnyCompany/Chicago/Welding Line/S2000/1/Current"
    }
  ]
},
{
  "assetExternalId": "External_Id_Welding_Line_1",
  "assetName": "Sample_Welding Line 1",
  "assetModelExternalId": "External_Id_Welding_Line",
  "assetProperties": [
    {
      "externalId": "External_Id_Welding_Line_Availability",
      "alias": "AnyCompany/Chicago/Welding Line/Availability"
    }
  ],
  "assetHierarchies": [
    {
      "externalId": "External_Id_Welding_Line_T0_Robot",
      "childAssetExternalId": "External_Id_Welding_Robot_1"
    },
    {
      "externalId": "External_Id_Welding_Line_T0_Robot",
      "childAssetExternalId": "External_Id_Welding_Robot_2"
    }
  ]
},
{
  "assetExternalId": "External_Id_Site_Chicago",
  "assetName": "Sample_Chicago",
  "assetModelExternalId": "External_Id_Site",
  "assetHierarchies": [
    {
      "externalId": "External_Id_Site_T0_Line",
      "childAssetExternalId": "External_Id_Welding_Line_1"
    }
  ]
},
{

```

```

"assetExternalId": "External_Id_Enterprise_AnyCompany",
"assetName": "Sample_AnyEnterprise Motor",
"assetModelExternalId": "External_Id_Enterprise",
"assetHierarchies": [
  {
    "externalId": "External_Id_Enterprise_T0_Site",
    "childAssetExternalId": "External_Id_Site_Chicago"
  }
]
}
]
}
}

```

以下屏幕截图显示的是运行前面的代码示例 AWS IoT SiteWise 控制台 之后显示的模型。

The screenshot shows the 'Models' page in the AWS IoT SiteWise console. It displays a table with 4 models, all with an 'ACTIVE' status. The table columns are Name, Status, Model type, Date created, and Date modified.

Name	Status	Model type	Date created	Date modified
<a href="#">Sample_Enterprise</a>	ACTIVE	Asset model	November 10, 2023 at 11:22:13 (UT...)	November 10, 202...
<a href="#">Sample_Site</a>	ACTIVE	Asset model	November 10, 2023 at 11:21:57 (UT...)	November 10, 202...
<a href="#">Sample_Welding Line</a>	ACTIVE	Asset model	November 10, 2023 at 11:21:40 (UT...)	November 10, 202...
<a href="#">Sample_Welding Robot</a>	ACTIVE	Asset model	November 10, 2023 at 11:21:24 (UT...)	November 10, 202...

以下屏幕截图显示了运行上一个代码示例 AWS IoT SiteWise 控制台 之后显示在中的模型、资产和层次结构。

The screenshot shows the 'Assets' page in the AWS IoT SiteWise console. It displays a tree view of assets under the 'Sample\_AnyEnterprise Motor' model. The assets are all 'ACTIVE' and were created on November 10, 2023.

Name	Description	Status	Date created	Date modified
Sample_AnyEnterprise Motor		ACTIVE	November 10, 2023 at 11:23:06 (UTC-5:00)	November 10, 2023 at 11:23:06 (UTC-...
Sample_Chicago		ACTIVE	November 10, 2023 at 11:22:57 (UTC-5:00)	November 10, 2023 at 11:22:57 (UTC-...
Sample_Welding Line 1		ACTIVE	November 10, 2023 at 11:22:48 (UTC-5:00)	November 10, 2023 at 11:22:48 (UTC-...
Sample_Welding Robot 1		ACTIVE	November 10, 2023 at 11:22:39 (UTC-5:00)	November 10, 2023 at 11:22:39 (UTC-...
Sample_Welding Robot 2		ACTIVE	November 10, 2023 at 11:22:30 (UTC-5:00)	November 10, 2023 at 11:22:30 (UTC-...

## 新增额外资产的示例

此示例定义了要导入到您账户中现有资产模型的其他资产：

- Sample\_Welding Line 2— 此资产是根据Sample\_Welding Line资产模型创建的。
- Sample\_Welding Robot 3— 此资产是根据Sample\_Welding Robot资产模型创建的。
- Sample\_Welding Robot 4— 此资产是根据Sample\_Welding Robot资产模型创建的。

要为本示例创建初始资产，请参阅[模型和资产的初始加载示例](#)。

以下 JSON 元数据文件定义了这些资产模型和资产。使用此元数据进行批量导入会在其中创建资产模型和资产 AWS IoT SiteWise，包括它们的层次关系。

用于加载其他资产的 JSON 文件

```
{
  "assets": [
    {
      "assetExternalId": "External_Id_Welding_Robot_3",
      "assetName": "Sample_Welding Robot 3",
      "assetModelExternalId": "External_Id_Welding_Robot",
      "assetProperties": [
        {
          "externalId": "External_Id_Welding_Robot_Serial_Number",
          "attributeValue": "S3000"
        },
        {
          "externalId": "External_Id_Welding_Robot_Cycle_Count",
          "alias": "AnyCompany/Chicago/Welding Line/S3000/Count"
        },
        {
          "externalId": "External_Id_Welding_Robot_Joint_1_Current",
          "alias": "AnyCompany/Chicago/Welding Line/S3000/1/Current"
        }
      ]
    },
    {
      "assetExternalId": "External_Id_Welding_Robot_4",
      "assetName": "Sample_Welding Robot 4",
      "assetModelExternalId": "External_Id_Welding_Robot",
      "assetProperties": [
```



```

        {
            "externalId": "External_Id_Welding_Robot_Serial_Number",
            "attributeValue": "S4000"
        },
        {
            "externalId": "External_Id_Welding_Robot_Cycle_Count",
            "alias": "AnyCompany/Chicago/Welding Line/S4000/Count"
        },
        {
            "externalId": "External_Id_Welding_Robot_Joint_1_Current",
            "alias": "AnyCompany/Chicago/Welding Line/S4000/1/Current"
        }
    ]
},
{
    "assetExternalId": "External_Id_Welding_Line_1",
    "assetName": "Sample_Welding Line 1",
    "assetModelExternalId": "External_Id_Welding_Line",
    "assetHierarchies": [
        {
            "externalId": "External_Id_Welding_Line_T0_Robot",
            "childAssetExternalId": "External_Id_Welding_Robot_1"
        },
        {
            "externalId": "External_Id_Welding_Line_T0_Robot",
            "childAssetExternalId": "External_Id_Welding_Robot_2"
        },
        {
            "externalId": "External_Id_Welding_Line_T0_Robot",
            "childAssetExternalId": "External_Id_Welding_Robot_3"
        }
    ]
},
{
    "assetExternalId": "External_Id_Welding_Line_2",
    "assetName": "Sample_Welding Line 2",
    "assetModelExternalId": "External_Id_Welding_Line",
    "assetHierarchies": [
        {
            "externalId": "External_Id_Welding_Line_T0_Robot",
            "childAssetExternalId": "External_Id_Welding_Robot_4"
        }
    ]
},

```

```

{
  "assetExternalId": "External_Id_Site_Chicago",
  "assetName": "Sample_Chicago",
  "assetModelExternalId": "External_Id_Site",
  "assetHierarchies": [
    {
      "externalId": "External_Id_Site_T0_Line",
      "childAssetExternalId": "External_Id_Welding_Line_1"
    },
    {
      "externalId": "External_Id_Site_T0_Line",
      "childAssetExternalId": "External_Id_Welding_Line_2"
    }
  ]
}
]
}

```

以下屏幕截图显示了运行上一个代码示例 AWS IoT SiteWise 控制台 之后显示在中的模型、资产和层次结构。

The screenshot shows the AWS IoT SiteWise console interface for the 'Assets' section. At the top, there is a search bar with the text 'Filter top level assets' and a 'Create asset' button. Below the search bar is a table with columns: Name, Description, Status, Date created, and Date modified. The table contains a list of assets, including 'Sample\_AnyCompany Motor', 'Sample\_Chicago', 'Sample\_Welding Line 1', 'Sample\_Welding Robot 2', 'Sample\_Welding Robot 3', 'Sample\_Welding Robot 1', 'Sample\_Welding Line 2', and 'Sample\_Welding Robot 4'. All assets are listed as 'ACTIVE'.

Name	Description	Status	Date created	Date modified
Sample_AnyCompany Motor		ACTIVE	November 09, 2023 at 19:18:05 (UTC-5:00)	November 09, 2023 at 19:18:05 (UTC-5:00)
Sample_Chicago		ACTIVE	November 09, 2023 at 19:17:56 (UTC-5:00)	November 09, 2023 at 19:17:56 (UTC-5:00)
Sample_Welding Line 1		ACTIVE	November 09, 2023 at 19:17:48 (UTC-5:00)	November 09, 2023 at 19:17:48 (UTC-5:00)
Sample_Welding Robot 2		ACTIVE	November 09, 2023 at 19:17:39 (UTC-5:00)	November 09, 2023 at 19:51:05 (UTC-5:00)
Sample_Welding Robot 3		ACTIVE	November 09, 2023 at 20:40:02 (UTC-5:00)	November 09, 2023 at 20:40:02 (UTC-5:00)
Sample_Welding Robot 1		ACTIVE	November 09, 2023 at 19:17:30 (UTC-5:00)	November 09, 2023 at 19:51:05 (UTC-5:00)
Sample_Welding Line 2		ACTIVE	November 09, 2023 at 20:40:20 (UTC-5:00)	November 09, 2023 at 20:40:20 (UTC-5:00)
Sample_Welding Robot 4		ACTIVE	November 09, 2023 at 20:40:11 (UTC-5:00)	November 09, 2023 at 20:40:11 (UTC-5:00)

## 新房产上线示例

此示例定义了现有资产模型上的新属性。请参阅[新增额外资产的示例](#)，以载入其他资产和模型。

- **Joint 1 Temperature**— 此属性已添加到Sample\_Welding Robot资产模型中。此新属性还将传播到根据资产模型创建的每个Sample\_Welding Robot资产。

要向现有资产模型添加新属性，请参阅以下 JSON 元数据文件示例。如 JSON 所示，必须提供完整的现有 Sample\_Welding Robot 资产模型定义以及新属性。如果未提供现有定义中的整个属性列表，则 AWS IoT SiteWise 删除省略的属性。

### 用于加载新属性的 JSON 文件

此示例向资产模型添加了一个新属性 Joint 1 Temperature。

```
{
  "assetModels": [
    {
      "assetModelExternalId": "External_Id_Welding_Robot",
      "assetModelName": "Sample_Welding Robot",
      "assetModelProperties": [
        {
          "dataType": "STRING",
          "externalId": "External_Id_Welding_Robot_Serial_Number",
          "name": "Serial Number",
          "type": {
            "attribute": {
              "defaultValue": "-"
            }
          },
          "unit": "-"
        },
        {
          "dataType": "DOUBLE",
          "externalId": "External_Id_Welding_Robot_Cycle_Count",
          "name": "CycleCount",
          "type": {
            "measurement": {}
          },
          "unit": "EA"
        },
        {
          "dataType": "DOUBLE",
          "externalId": "External_Id_Welding_Robot_Joint_1_Current",
          "name": "Joint 1 Current",
          "type": {
            "measurement": {}
          },
          "unit": "Amps"
        }
      ]
    }
  ]
}
```

```

    },
    {
      "dataType": "DOUBLE",
      "externalId": "External_Id_Welding_Robot_Joint_1_Max_Current",
      "name": "Max Joint 1 Current",
      "type": {
        "metric": {
          "expression": "max(joint1current)",
          "variables": [
            {
              "name": "joint1current",
              "value": {
                "propertyExternalId":
"External_Id_Welding_Robot_Joint_1_Current"
              }
            }
          ],
          "window": {
            "tumbling": {
              "interval": "5m"
            }
          }
        }
      },
      "unit": "Amps"
    },
    {
      "dataType": "DOUBLE",
      "externalId": "External_Id_Welding_Robot_Joint_1_Temperature",
      "name": "Joint 1 Temperature",
      "type": {
        "measurement": {}
      },
      "unit": "degC"
    }
  ]
}

```

## 导出元数据示例

将 AWS IoT SiteWise 内容批量导出到 Amazon S3 时，您可以指定筛选条件来限制要导出的特定资产模型和资产。

您可以在请求正 `iotSiteWiseConfiguration` 文部分的某个 `sources` 部分中指定过滤器。

### Note

您可以包含多个过滤器。批量操作将导出与任何筛选条件匹配的任何资产模型或资产。如果您不提供任何筛选条件，则该操作将导出您的所有资产模型和资产。

```
{
  "metadataTransferJobId": "your-transfer-job-id",
  "sources": [{
    "type": "iotsitewise",
    "iotSiteWiseConfiguration": {
      "filters": [{
        list of filters
      }]
    }
  ]},
  "destination": {
    "type": "s3",
    "s3Configuration": {
      "location": "arn:aws:s3:::your-S3-bucket-location"
    }
  }
}
```

## 按资产模型筛选

您可以筛选特定的资产模型。您还可以使用该模型的所有资产或其层次结构中的所有资产模型。您不能同时包含资产和层次结构。

有关层次结构的更多信息，请参阅[定义资产模型层次结构](#)。

### Asset model

此筛选器包括指定的资产模型：

```
"filterByAssetModel": {  
  "assetModelId": "asset model ID"  
}
```

### Asset model and its assets

此筛选器包括指定的资产模型以及使用该资产模型的所有资产：

```
"filterByAssetModel": {  
  "assetModelId": "asset model ID",  
  "includeAssets": true  
}
```

### Asset model and its hierarchy

此筛选器包括指定的资产模型及其层次结构中的所有关联资产模型：

```
"filterByAssetModel": {  
  "assetModelId": "asset model ID",  
  "includeOffspring": true  
}
```

## 按资产筛选

您可以筛选特定资产。您还可以将其资产模型或其层次结构中的所有关联资产包括在内。您不能同时包含资产模型和层次结构。

有关层次结构的更多信息，请参阅[定义资产模型层次结构](#)。

### Asset

此筛选器包括指定的资产：

```
"filterByAsset": {  
  "assetId": "asset ID"  
}
```

### Asset and its asset model

此筛选器包括指定的资产及其使用的资产模型：

```
"filterByAsset": {
  "assetId": "asset ID",
  "includeAssetModel": true
}
```

## Asset and its hierarchy

此筛选器包括指定的资产及其层次结构中的所有关联资产：

```
"filterByAsset": {
  "assetId": "asset ID",
  "includeOffspring": true
}
```

## AWS IoT SiteWise 元数据传输任务架构

执行自己的批量导入和导出操作时，请使用 AWS IoT SiteWise 元数据传输任务架构作为参考：

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "IoTSiteWise",
  "description": "Metadata transfer job resource schema for IoTSiteWise",
  "definitions": {
    "Name": {
      "type": "string",
      "minLength": 1,
      "maxLength": 256,
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    },
    "Description": {
      "type": "string",
      "minLength": 1,
      "maxLength": 2048,
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    },
    "ID": {
      "type": "string",
      "minLength": 36,
      "maxLength": 36,
      "pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$"
    },
  },
}
```

```

"ExternalId": {
  "type": "string",
  "minLength": 2,
  "maxLength": 128,
  "pattern": "[a-zA-Z0-9_][a-zA-Z_\\-0-9.:]*[a-zA-Z0-9_]+"
},
"AttributeValue": {
  "description": "The value of the property attribute.",
  "type": "string",
  "minLength": 1,
  "maxLength": 1024,
  "pattern": "[^\\u0000-\\u001F\\u007F]+"
},
"PropertyUnit": {
  "description": "The unit of measure (such as Newtons or RPM) of the asset
property.",
  "type": "string",
  "minLength": 1,
  "maxLength": 256,
  "pattern": "[^\\u0000-\\u001F\\u007F]+"
},
"PropertyAlias": {
  "description": "The property alias that identifies the property.",
  "type": "string",
  "minLength": 1,
  "maxLength": 1000,
  "pattern": "[^\\u0000-\\u001F\\u007F]+"
},
"AssetProperty": {
  "description": "The asset property's definition, alias, unit, and notification
state.",
  "type": "object",
  "additionalProperties": false,
  "anyOf": [
    {
      "required": [
        "id"
      ]
    },
    {
      "required": [
        "externalId"
      ]
    }
  ]
}

```



```
  ],
  "properties": {
    "id": {
      "description": "The ID of the asset property.",
      "$ref": "#/definitions/ID"
    },
    "externalId": {
      "description": "The ExternalID of the asset property.",
      "$ref": "#/definitions/ExternalId"
    },
    "alias": {
      "$ref": "#/definitions/PropertyAlias"
    },
    "unit": {
      "$ref": "#/definitions/PropertyUnit"
    },
    "attributeValue": {
      "$ref": "#/definitions/AttributeValue"
    },
    "retainDataOnAliasChange": {
      "type": "string",
      "default": "TRUE",
      "enum": [
        "TRUE",
        "FALSE"
      ]
    },
    "propertyNotificationState": {
      "description": "The MQTT notification state (ENABLED or DISABLED) for this
asset property.",
      "type": "string",
      "enum": [
        "ENABLED",
        "DISABLED"
      ]
    }
  }
},
"AssetHierarchy": {
  "description": "A hierarchy specifies allowed parent/child asset relationships.",
  "type": "object",
  "additionalProperties": false,
  "anyOf": [
    {
```

```
    "required": [
      "id",
      "childAssetId"
    ],
  },
  {
    "required": [
      "externalId",
      "childAssetId"
    ],
  },
  {
    "required": [
      "id",
      "childAssetExternalId"
    ],
  },
  {
    "required": [
      "externalId",
      "childAssetExternalId"
    ],
  }
],
"properties": {
  "id": {
    "description": "The ID of a hierarchy in the parent asset's model.",
    "$ref": "#/definitions/ID"
  },
  "externalId": {
    "description": "The ExternalID of a hierarchy in the parent asset's model.",
    "$ref": "#/definitions/ExternalID"
  },
  "childAssetId": {
    "description": "The ID of the child asset to be associated.",
    "$ref": "#/definitions/ID"
  },
  "childAssetExternalId": {
    "description": "The ExternalID of the child asset to be associated.",
    "$ref": "#/definitions/ExternalID"
  }
}
},
"Tag": {
```

```
"type": "object",
"additionalProperties": false,
"required": [
  "key",
  "value"
],
"properties": {
  "key": {
    "type": "string"
  },
  "value": {
    "type": "string"
  }
}
},
"AssetModelType": {
  "type": "string",
  "default": null,
  "enum": [
    "ASSET_MODEL",
    "COMPONENT_MODEL"
  ]
},
"AssetModelCompositeModel": {
  "description": "Contains a composite model definition in an asset model. This composite model definition is applied to all assets created from the asset model.",
  "type": "object",
  "additionalProperties": false,
  "anyOf": [
    {
      "required": [
        "id"
      ]
    },
    {
      "required": [
        "externalId"
      ]
    }
  ],
  "required": [
    "name",
    "type"
  ],
}
```

```
"properties": {
  "id": {
    "description": "The ID of the asset model composite model.",
    "$ref": "#/definitions/ID"
  },
  "externalId": {
    "description": "The ExternalID of the asset model composite model.",
    "$ref": "#/definitions/ExternalId"
  },
  "parentId": {
    "description": "The ID of the parent asset model composite model.",
    "$ref": "#/definitions/ID"
  },
  "parentExternalId": {
    "description": "The ExternalID of the parent asset model composite model.",
    "$ref": "#/definitions/ExternalId"
  },
  "composedAssetModelId": {
    "description": "The ID of the composed asset model.",
    "$ref": "#/definitions/ID"
  },
  "composedAssetModelExternalId": {
    "description": "The ExternalID of the composed asset model.",
    "$ref": "#/definitions/ExternalId"
  },
  "description": {
    "description": "A description for the asset composite model.",
    "$ref": "#/definitions/Description"
  },
  "name": {
    "description": "A unique, friendly name for the asset composite model.",
    "$ref": "#/definitions/Name"
  },
  "type": {
    "description": "The type of the composite model. For alarm composite models,
this type is AWS/ALARM.",
    "$ref": "#/definitions/Name"
  },
  "properties": {
    "description": "The property definitions of the asset model.",
    "type": "array",
    "items": {
      "$ref": "#/definitions/AssetModelProperty"
    }
  }
}
```

```
    }
  }
},
"AssetModelProperty": {
  "description": "Contains information about an asset model property.",
  "type": "object",
  "additionalProperties": false,
  "anyOf": [
    {
      "required": [
        "id"
      ]
    },
    {
      "required": [
        "externalId"
      ]
    }
  ],
  "required": [
    "name",
    "dataType",
    "type"
  ],
  "properties": {
    "id": {
      "description": "The ID of the asset model property.",
      "$ref": "#/definitions/ID"
    },
    "externalId": {
      "description": "The ExternalID of the asset model property.",
      "$ref": "#/definitions/ExternalId"
    },
    "name": {
      "description": "The name of the asset model property.",
      "$ref": "#/definitions/Name"
    },
    "dataType": {
      "description": "The data type of the asset model property.",
      "$ref": "#/definitions/DataType"
    },
    "dataTypeSpec": {
      "description": "The data type of the structure for this property.",
      "$ref": "#/definitions/Name"
    }
  }
}
```

```

    },
    "unit": {
      "description": "The unit of the asset model property, such as Newtons or
RPM.",
      "type": "string",
      "minLength": 1,
      "maxLength": 256,
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    },
    "type": {
      "description": "The property type",
      "$ref": "#/definitions/PropertyType"
    }
  }
},
"DataType": {
  "type": "string",
  "enum": [
    "STRING",
    "INTEGER",
    "DOUBLE",
    "BOOLEAN",
    "STRUCT"
  ]
},
"PropertyType": {
  "description": "Contains a property type, which can be one of attribute,
measurement, metric, or transform.",
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "attribute": {
      "$ref": "#/definitions/Attribute"
    },
    "transform": {
      "$ref": "#/definitions/Transform"
    },
    "metric": {
      "$ref": "#/definitions/Metric"
    },
    "measurement": {
      "$ref": "#/definitions/Measurement"
    }
  }
}
}

```

```

    },
    "Attribute": {
      "type": "object",
      "additionalProperties": false,
      "properties": {
        "defaultValue": {
          "type": "string",
          "minLength": 1,
          "maxLength": 1024,
          "pattern": "[^\\u0000-\\u001F\\u007F]+"
        }
      }
    },
    },
    "Transform": {
      "type": "object",
      "additionalProperties": false,
      "required": [
        "expression",
        "variables"
      ],
      "properties": {
        "expression": {
          "description": "The mathematical expression that defines the transformation function.",
          "type": "string",
          "minLength": 1,
          "maxLength": 1024
        },
        "variables": {
          "description": "The list of variables used in the expression.",
          "type": "array",
          "items": {
            "$ref": "#/definitions/ExpressionVariable"
          }
        }
      },
      "processingConfig": {
        "$ref": "#/definitions/TransformProcessingConfig"
      }
    }
  },
  "TransformProcessingConfig": {
    "description": "The processing configuration for the given transform property.",
    "type": "object",
    "additionalProperties": false,

```

```

    "required": [
      "computeLocation"
    ],
    "properties": {
      "computeLocation": {
        "description": "The compute location for the given transform property.",
        "$ref": "#/definitions/ComputeLocation"
      },
      "forwardingConfig": {
        "description": "The forwarding configuration for a given property.",
        "$ref": "#/definitions/ForwardingConfig"
      }
    }
  },
  "Metric": {
    "type": "object",
    "additionalProperties": false,
    "required": [
      "expression",
      "variables",
      "window"
    ],
    "properties": {
      "expression": {
        "description": "The mathematical expression that defines the metric aggregation function.",
        "type": "string",
        "minLength": 1,
        "maxLength": 1024
      },
      "variables": {
        "description": "The list of variables used in the expression.",
        "type": "array",
        "items": {
          "$ref": "#/definitions/ExpressionVariable"
        }
      },
      "window": {
        "description": "The window (time interval) over which AWS IoT SiteWise computes the metric's aggregation expression",
        "$ref": "#/definitions/MetricWindow"
      },
      "processingConfig": {
        "$ref": "#/definitions/MetricProcessingConfig"
      }
    }
  }
}

```



```
    }
  }
},
"MetricProcessingConfig": {
  "description": "The processing configuration for the metric.",
  "type": "object",
  "additionalProperties": false,
  "required": [
    "computeLocation"
  ],
  "properties": {
    "computeLocation": {
      "description": "The compute location for the given metric property.",
      "$ref": "#/definitions/ComputeLocation"
    }
  }
},
"ComputeLocation": {
  "type": "string",
  "enum": [
    "EDGE",
    "CLOUD"
  ]
},
"ForwardingConfig": {
  "type": "object",
  "additionalProperties": false,
  "required": [
    "state"
  ],
  "properties": {
    "state": {
      "type": "string",
      "enum": [
        "ENABLED",
        "DISABLED"
      ]
    }
  }
},
"MetricWindow": {
  "description": "Contains a time interval window used for data aggregate
computations (for example, average, sum, count, and so on).",
  "type": "object",
```

```
"additionalProperties": false,
"properties": {
  "tumbling": {
    "description": "The tumbling time interval window.",
    "type": "object",
    "additionalProperties": false,
    "required": [
      "interval"
    ],
    "properties": {
      "interval": {
        "description": "The time interval for the tumbling window.",
        "type": "string",
        "minLength": 2,
        "maxLength": 23
      },
      "offset": {
        "description": "The offset for the tumbling window.",
        "type": "string",
        "minLength": 2,
        "maxLength": 25
      }
    }
  }
}
},
"ExpressionVariable": {
  "type": "object",
  "additionalProperties": false,
  "required": [
    "name",
    "value"
  ],
  "properties": {
    "name": {
      "description": "The friendly name of the variable to be used in the
expression.",
      "type": "string",
      "minLength": 1,
      "maxLength": 64,
      "pattern": "^[a-z][a-z0-9_]*$"
    },
    "value": {
```

```

    "description": "The variable that identifies an asset property from which to
use values.",
    "$ref": "#/definitions/VariableValue"
  }
}
},
"VariableValue": {
  "type": "object",
  "additionalProperties": false,
  "anyOf": [
    {
      "required": [
        "propertyId"
      ]
    },
    {
      "required": [
        "propertyExternalId"
      ]
    }
  ],
  "properties": {
    "propertyId": {
      "$ref": "#/definitions/ID"
    },
    "propertyExternalId": {
      "$ref": "#/definitions/ExternalId"
    },
    "hierarchyId": {
      "$ref": "#/definitions/ID"
    },
    "hierarchyExternalId": {
      "$ref": "#/definitions/ExternalId"
    }
  }
},
"Measurement": {
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "processingConfig": {
      "$ref": "#/definitions/MeasurementProcessingConfig"
    }
  }
}

```

```

},
"MeasurementProcessingConfig": {
  "type": "object",
  "additionalProperties": false,
  "required": [
    "forwardingConfig"
  ],
  "properties": {
    "forwardingConfig": {
      "description": "The forwarding configuration for the given measurement
property.",
      "$ref": "#/definitions/ForwardingConfig"
    }
  }
},
"AssetModelHierarchy": {
  "description": "Contains information about an asset model hierarchy.",
  "type": "object",
  "additionalProperties": false,
  "anyOf": [
    {
      "required": [
        "id",
        "childAssetModelId"
      ]
    },
    {
      "required": [
        "id",
        "childAssetModelExternalId"
      ]
    },
    {
      "required": [
        "externalId",
        "childAssetModelId"
      ]
    },
    {
      "required": [
        "externalId",
        "childAssetModelExternalId"
      ]
    }
  ]
}

```

```

    ],
    "required": [
      "name"
    ],
  },
  "properties": {
    "id": {
      "description": "The ID of the asset model hierarchy.",
      "$ref": "#/definitions/ID"
    },
    "externalId": {
      "description": "The ExternalID of the asset model hierarchy.",
      "$ref": "#/definitions/ExternalId"
    },
    "name": {
      "description": "The name of the asset model hierarchy.",
      "$ref": "#/definitions/Name"
    },
    "childAssetModelId": {
      "description": "The ID of the asset model. All assets in this hierarchy must
be instances of the child AssetModelId asset model.",
      "$ref": "#/definitions/ID"
    },
    "childAssetModelExternalId": {
      "description": "The ExternalID of the asset model. All assets in this
hierarchy must be instances of the child AssetModelId asset model.",
      "$ref": "#/definitions/ExternalId"
    }
  }
},
"AssetModel": {
  "type": "object",
  "additionalProperties": false,
  "anyOf": [
    {
      "required": [
        "assetModelId"
      ]
    },
    {
      "required": [
        "assetModelExternalId"
      ]
    }
  ]
},
],

```

```
"required": [
  "assetModelName"
],
"properties": {
  "assetModelId": {
    "description": "The ID of the asset model.",
    "$ref": "#/definitions/ID"
  },
  "assetModelExternalId": {
    "description": "The ID of the asset model.",
    "$ref": "#/definitions/ExternalId"
  },
  "assetModelName": {
    "description": "A unique, friendly name for the asset model.",
    "$ref": "#/definitions/Name"
  },
  "assetModelDescription": {
    "description": "A description for the asset model.",
    "$ref": "#/definitions/Description"
  },
  "assetModelType": {
    "description": "The type of the asset model.",
    "$ref": "#/definitions/AssetModelType"
  },
  "assetModelProperties": {
    "description": "The property definitions of the asset model.",
    "type": "array",
    "items": {
      "$ref": "#/definitions/AssetModelProperty"
    }
  },
  "assetModelCompositeModels": {
    "description": "The composite asset models that are part of this asset model. Composite asset models are asset models that contain specific properties.",
    "type": "array",
    "items": {
      "$ref": "#/definitions/AssetModelCompositeModel"
    }
  },
  "assetModelHierarchies": {
    "description": "The hierarchy definitions of the asset model. Each hierarchy specifies an asset model whose assets can be children of any other assets created from this asset model.",
    "type": "array",
```

```
    "items": {
      "$ref": "#/definitions/AssetModelHierarchy"
    }
  },
  "tags": {
    "description": "A list of key-value pairs that contain metadata for the asset
model.",
    "type": "array",
    "items": {
      "$ref": "#/definitions/Tag"
    }
  }
},
"Asset": {
  "type": "object",
  "additionalProperties": false,
  "anyOf": [
    {
      "required": [
        "assetId",
        "assetModelId"
      ]
    },
    {
      "required": [
        "assetExternalId",
        "assetModelId"
      ]
    },
    {
      "required": [
        "assetId",
        "assetModelExternalId"
      ]
    },
    {
      "required": [
        "assetExternalId",
        "assetModelExternalId"
      ]
    }
  ],
  "required": [
```

```

    "assetName"
  ],
  "properties": {
    "assetId": {
      "description": "The ID of the asset",
      "$ref": "#/definitions/ID"
    },
    "assetExternalId": {
      "description": "The external ID of the asset",
      "$ref": "#/definitions/ExternalId"
    },
    "assetModelId": {
      "description": "The ID of the asset model from which to create the asset.",
      "$ref": "#/definitions/ID"
    },
    "assetModelExternalId": {
      "description": "The ExternalID of the asset model from which to create the
asset.",
      "$ref": "#/definitions/ExternalId"
    },
    "assetName": {
      "description": "A unique, friendly name for the asset.",
      "$ref": "#/definitions/Name"
    },
    "assetDescription": {
      "description": "A description for the asset",
      "$ref": "#/definitions/Description"
    },
    "assetProperties": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/AssetProperty"
      }
    },
    "assetHierarchies": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/AssetHierarchy"
      }
    },
    "tags": {
      "description": "A list of key-value pairs that contain metadata for the
asset.",
      "type": "array",

```



```
        "uniqueItems": false,
        "items": {
          "$ref": "#/definitions/Tag"
        }
      }
    },
    "additionalProperties": false,
    "properties": {
      "assetModels": {
        "type": "array",
        "uniqueItems": false,
        "items": {
          "$ref": "#/definitions/AssetModel"
        }
      },
      "assets": {
        "type": "array",
        "uniqueItems": false,
        "items": {
          "$ref": "#/definitions/Asset"
        }
      }
    }
  }
}
```

# 通过警报监控数据

您可以为您的数据配置警报，以便在设备或流程运行不佳时提醒您的团队。计算机或进程的最佳性能意味着特定指标的值应在上限和下限的范围内。当这些指标超出其运作范围时，必须通知设备操作员以便修复问题。使用警报功能快速识别问题并通知操作员，以最大限度地提高设备和进程的性能。

## 主题

- [警报类型](#)
- [警报状态](#)
- [警报状态属性](#)
- [在资产模型上定义警报](#)
- [为资产配置警报](#)
- [响应警报](#)
- [获取外部警报状态](#)

## 警报类型

您可以定义在 AWS 云端检测到的警报和通过外部进程检测到的警报。AWS IoT SiteWise 支持以下类型的警报：

- AWS IoT Events 警报

AWS IoT Events 警报是检测到的警报 AWS IoT Events。AWS IoT SiteWise 将资产属性值发送到中的警报模型 AWS IoT Events。然后，AWS IoT Events 将警报状态发送到 AWS IoT SiteWise。您可以配置选项，例如警报何时检测到以及在警报状态发生变化时通知谁。您还可以定义警报状态发生变化时发生的[AWS IoT Events 操作](#)。

中的警报 AWS IoT Events 是警报模型的实例。警报模型指定了警报的阈值和严重性、警报状态发生变化时该怎么做等等。配置警报模型的每个特征时，您可以指定警报监控的资产模型的特征属性。所有基于资产模型的资产在 AWS IoT Events 评估警报的特征时都使用该属性的值。有关更多信息，请参阅《[开发人员指南](#)》中的AWS IoT Events 使用警报。

当 AWS IoT Events 警报状态发生变化时，您可以对其做出响应。例如，您可以在警报激活时确认或暂停警报。您还可以启用、禁用和重置警报。

SiteWise 监控用户可以在监控器门户中对 AWS IoT Events 警报进行可 SiteWise 可视化、配置和响应。有关更多信息，请参阅《AWS IoT SiteWise Monitor 应用程序指南》中的[使用警报进行监控](#)。

#### Note

AWS IoT Events 评估这些警报并在 AWS IoT SiteWise 和之间传输数据需要付费 AWS IoT Events。有关更多信息，请参阅[AWS IoT Events 定价](#)。

#### • 外部警报

外部警报是您在外部评估的警报 AWS IoT SiteWise。如果您有报告警报状态的数据来源，请使用外部警报。外部警报包含一个测量属性，您可以从中提取警报状态数据。

当外部警报改变状态时，您无法确认或暂停该警报。

SiteWise 监控用户可以在 SiteWise 监控器门户中看到外部警报的状态，但他们无法配置或响应这些警报。

AWS IoT SiteWise 不评估外部警报的状态。

## 警报状态

工业警报包括有关其监控的设备或流程状态的信息，以及有关操作员对警报状态的响应的信息（可选）。

定义 AWS IoT Events 警报时，您可以指定是否启用确认流。默认情况下，启用确认流。启用此选项后，操作员可以确认警报并留下备注，详细说明警报或其为解决警报采取的措施。如果操作员在活动警报变为非活动状态之前未确认该活动警报，则该警报将被锁定。锁定状态表示警报已激活且未被确认，因此操作员需要检查设备或流程，并确认已锁定的警报。

警报具有以下状态：

- 正常 (Normal) - 警报已启用但处于非活动状态。工业流程或设备按预期运行。
- 活动 (Active) - 警报处于活动状态。工业流程或设备超出了其操作范围，需要引起注意。
- 已确认 (Acknowledged) - 操作员已确认警报的状态。

此状态仅适用于您启用确认流的警报。

- 已锁定 (Latched) - 警报恢复正常，但处于活动状态，没有操作员确认。工业流程或设备需要操作员注意才能将警报重置为正常。

此状态仅适用于您启用确认流的警报。

- 已暂停 (SnoozeDisabled) - 警报被停用，因为操作员暂停了警报。操作员定义警报暂停的持续时间。在该持续时间之后，警报将恢复正常状态。
- 已禁用 (Disabled) - 警报已禁用，无法检测。

## 警报状态属性

AWS IoT SiteWise 将警报状态数据存储为序列化为字符串的 JSON 对象。此对象包含警报的状态和其他信息，例如操作员的响应操作和警报评估的规则。

您可以通过名称和结构类型 `AWS/ALARM_STATE` 来识别警报状态属性。有关更多信息，请参阅 [在资产模型上定义警报](#)。

警报状态数据对象包含以下信息：

`stateName`

警报的状态。有关更多信息，请参阅 [警报状态](#)。

数据类型：STRING

`customerAction`

( 可选 ) 包含有关操作员响应警报信息的对象。操作员可以启用、禁用、确认和暂停警报。当他们这样做时，警报状态数据包括他们的响应以及他们响应时可以留下的备注。该对象包含以下信息：

`actionName`

操作员为响应警报而采取的操作名称。此值包含以下字符串之一：

- ENABLE
- DISABLE
- SNOOZE
- ACKNOWLEDGE
- RESET

数据类型：STRING

## enable

( 可选 ) 操作员启用警报时 `customerAction` 中存在的对象。当操作员启用警报时，警报状态将变为 `Normal`。该对象包含以下信息：

### note

( 可选 ) 客户在启用警报时留下的备注。

数据类型：STRING

最大长度：128 个字符

## disable

( 可选 ) 操作员禁用警报时 `customerAction` 中存在的对象。当操作员禁用警报时，警报状态将变为 `Disabled`。该对象包含以下信息：

### note

( 可选 ) 客户在禁用警报时留下的备注。

数据类型：STRING

最大长度：128 个字符

## acknowledge

( 可选 ) 操作员确认警报时 `customerAction` 中存在的对象。当操作员确认警报时，警报状态将变为 `Acknowledged`。该对象包含以下信息：

### note

( 可选 ) 客户在确认警报时留下的备注。

数据类型：STRING

最大长度：128 个字符

## snooze

( 可选 ) 操作员暂停警报时 `customerAction` 中存在的对象。当操作员暂停警报时，警报状态将变为 `SnoozeDisabled`。该对象包含以下信息：

### snoozeDuration

操作员暂停警报的持续时间 ( 以秒为单位 )。在此持续时间之后，警报将变为 `Normal` 状态。

数据类型：INTEGER

note

( 可选 ) 客户在暂停警报时留下的备注。

数据类型：STRING

最大长度：128 个字符

ruleEvaluation

( 可选 ) 包含有关警报评估规则信息的对象。该对象包含以下信息：

simpleRule

包含有关简单规则信息的对象，该规则使用比较运算符将属性值与阈值进行比较。该对象包含以下信息：

inputProperty

此警报评估属性的值。

数据类型：DOUBLE

operator

此警报用于将属性与阈值进行比较的比较运算符。此值包含以下字符串之一：

- < - 小于
- <= - 小于或等于
- == - 等于
- != - 不等于
- >= - 大于或等于
- > - 大于

数据类型：STRING

threshold

此警报与属性值进行比较的阈值。

数据类型：DOUBLE

## 在资产模型上定义警报

资产模型可推动您的工业数据和警报实现标准化。您可以在资产模型上规定警报定义，以标准化基于资产模型的所有资产的警报。

您可以使用组合资产模型来定义资产模型上的警报。复合资产模型是在另一种资产模型上标准化一组特定属性的资产模型。复合资产模型可确保资产模型上存在某些属性。警报具有类型、状态和（可选）源属性，因此警报复合模型强制这些属性存在。

每个复合模型都有一个类型，用于定义复合模型的属性。警报复合模型定义警报类型、警报状态和（可选）警报源的属性。当您使用复合模型从资产模型创建资产时，该资产将包括复合模型中的属性以及您在资产模型中指定的属性。

复合模型中的每个属性都必须具有用于标识其复合模型类型的名称。复合模型属性支持具有复杂数据类型的属性。这些属性具有 STRUCT 数据类型和 dataTypeSpec 特征，用于指定属性的复杂数据类型。复杂数据类型属性包含序列化为字符串的 JSON 数据。

警报复合模型具有以下属性。对于此类复合模型，每个属性都必须具有标识该属性的名称。

### 警报类型

警报的类型。指定下列项之一：

- IOT\_EVENTS— AWS IoT Events 警报。AWS IoT SiteWise 向发送数据 AWS IoT Events 以评估此警报的状态。必须指定警报源属性才能定义此 AWS IoT Events 警报定义的警报模型。
- EXTERNAL – 外部警报。您摄取警报状态作为测量值。

属性名称：AWS/ALARM\_TYPE

属性类型：[属性](#)

数据类型：STRING

### 警报状态

警报状态的时间序列数据。这是一个序列化为字符串的对象，其中包含警报的状态和其他信息。有关更多信息，请参阅 [警报状态属性](#)。

属性名称：AWS/ALARM\_STATE

属性类型：[测量值](#)

数据类型：STRUCT

数据结构类型：AWS/ALARM\_STATE

## 警报源

( 可选 ) 评估警报状态的资源的 Amazon 资源名称 ( ARN )。对于 AWS IoT Events 警报，这是警报模型的 ARN。

属性名称：AWS/ALARM\_SOURCE

属性类型：[属性](#)

数据类型：STRING

## Example 警报复合模型示例

以下资产模型表示带有警报以监控其温度的锅炉。AWS IoT SiteWise 将温度数据发送到 AWS IoT Events 以检测警报。

```
{
  "assetModelName": "Boiler",
  "assetModelDescription": "A boiler that alarms when its temperature exceeds its
limit.",
  "assetModelProperties": [
    {
      "name": "Temperature",
      "dataType": "DOUBLE",
      "unit": "Celsius",
      "type": {
        "measurement": {}
      }
    },
    {
      "name": "High Temperature",
      "dataType": "DOUBLE",
      "unit": "Celsius",
      "type": {
        "attribute": {
          "defaultValue": "105.0"
        }
      }
    }
  ],
  "assetModelCompositeModels": [
```



```
{
  "name": "BoilerTemperatureHighAlarm",
  "type": "AWS/ALARM",
  "properties": [
    {
      "name": "AWS/ALARM_TYPE",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "IOT_EVENTS"
        }
      }
    },
    {
      "name": "AWS/ALARM_STATE",
      "dataType": "STRUCT",
      "dataTypeSpec": "AWS/ALARM_STATE",
      "type": {
        "measurement": {}
      }
    },
    {
      "name": "AWS/ALARM_SOURCE",
      "dataType": "STRING",
      "type": {
        "attribute": {}
      }
    }
  ]
}
```

## 主题

- [定义 AWS IoT Events 警报](#)
- [定义外部警报](#)

## 定义 AWS IoT Events 警报

创建 AWS IoT Events 警报时，AWS IoT SiteWise 会将资产属性值发送 AWS IoT Events 到以评估警报的状态。AWS IoT Events 警报定义取决于您在资产模型中定义的警报模型 AWS IoT Events。要在资产模型

上定义 AWS IoT Events 警报，您需要定义一个警报组合模型，该模型将 AWS IoT Events 警报模型指定为其警报源属性。

AWS IoT Events 警报取决于警报阈值和警报通知设置等输入。您可以将这些输入定义为资产模型的属性。然后，您可以根据模型在每个资产上自定义这些输入。AWS IoT SiteWise 控制台可以为您创建这些属性。如果您使用 AWS CLI 或 API 定义警报，则必须在资产模型上手动定义这些属性。

您还可以定义检测到警报时发生的其他操作，例如自定义警报通知操作。例如，您可以配置向 Amazon SNS 主题发送推送通知的操作。有关您可以定义的操作的更多信息，请参阅《AWS IoT Events 开发人员指南》中的[使用其他 AWS 服务](#)。

更新或删除资产模型时，AWS IoT SiteWise 可以检查中的警报模型是否 AWS IoT Events 正在监视与该资产模型关联的资产属性。这可以防止您删除 AWS IoT Events 警报当前正在使用的资产属性。要在中启用此功能 AWS IoT SiteWise，您必须拥有 `iotevents:ListInputRoutings` 权限。此权限 AWS IoT SiteWise 允许调用支持的 [ListInputRouting](#) s API 操作。AWS IoT Events 有关更多信息，请参阅 [\( 可选 \) ListInputRoutings 权限](#)。

#### Note

警报通知功能在中国（北京）区域中不可用。

## 主题

- [警报通知的要求](#)
- [定义 AWS IoT Events 警报 \( AWS IoT SiteWise 控制台 \)](#)
- [定义 AWS IoT Events 警报 \( AWS IoT Events 控制台 \)](#)
- [定义 AWS IoT Events 警报 \(AWS CLI\)](#)

## 警报通知的要求

AWS IoT Events 使用您 AWS 账户中的 AWS Lambda 功能发送警报通知。您必须在警报所在的 AWS 区域创建此 Lambda 函数才能启用警报通知。此 Lambda 函数使用 [Amazon Simple Notification Service \(Amazon SNS\)](#) 发送短信通知，使用 [Amazon Simple Email Service \(Amazon SES\)](#) 发送电子邮件通知。创建 AWS IoT Events 警报时，您需要配置警报用于发送通知的协议和设置。

AWS IoT Events 提供了一个 AWS CloudFormation 堆栈模板，您可以使用该模板在您的账户中创建此 Lambda 函数。有关更多信息，请参阅《AWS IoT Events 开发人员指南》中的[警报通知 Lambda 函数](#)。

## 定义 AWS IoT Events 警报 ( AWS IoT SiteWise 控制台 )

您可以使用 AWS IoT SiteWise 控制台在现有资产模型上定义 AWS IoT Events 警报。要在新资产模型上定义 AWS IoT Events 警报，请创建资产模型，然后完成以下步骤。有关更多信息，请参阅 [创建资产模型](#)。

### Important

每个警报都需要一个属性，来指定要与警报进行比较的阈值。必须先在资产模型上定义阈值属性，然后才能定义警报。


举例说明，如果您想定义一个警报，用于检测风力涡轮机何时超过其最大额定风速 50 英里/小时。在定义警报之前，您必须定义一个默认值为 50 的属性（最大风速）。

### 在资产模型上定义 AWS IoT Events 警报

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择模型。
3. 选择要为其定义警报的资产模型。
4. 选择警报选项卡。
5. 选择添加警报。
6. 在警报类型选项部分，选择 AWS IoT Events 警报。
7. 在基本详细信息部分中，执行以下操作：
  - a. 输入警报的名称。
  - b. （可选）输入警报的描述。
8. 在阈值定义部分，您可以定义警报何时检测以及警报的严重程度。执行以下操作：
  - a. 选择警报检测的属性。每次此属性收到新值时，都会 AWS IoT SiteWise 将该值发送 AWS IoT Events 以评估警报的状态。
  - b. 选择用于将属性与阈值进行比较的运算符。从以下选项中进行选择：
    - 小于
    - 小于或等于
    - = (等于)
    - != 不等于


- 大于或等于
  - 大于
- c. 对于“值”，选择要用作阈值的属性属性。AWS IoT Events 将属性的值与该属性的值进行比较。
  - d. 输入警报的严重程度。使用您的团队能理解的数字来反映此警报的严重程度。
9. (可选) 在通知设置-可选部分中，执行以下操作：

- a. 选择活动。

 Note


如果您选择非活动，则您和您的团队将不会收到任何警报通知。

- b. 在接收人中，选择接收人。

 Important

您可以向 AWS IAM Identity Center 用户发送警报通知。要使用此功能，您必须启用 IAM Identity Center。一次只能在一个 AWS 区域启用 IAM 身份中心。这意味着您只能在启用 IAM Identity Center 的区域中定义警报通知。有关更多信息，请参阅《AWS IAM Identity Center 用户指南》中的[入门](#)。

- c. 对于协议，请从以下选项中选择：
  - 电子邮件和短信 - 警报通过短信和电子邮件通知 IAM Identity Center 用户。
  - 电子邮件 - 警报通过电子邮件通知 IAM Identity Center 用户。
  - 短信 - 警报通过短信通知 IAM Identity Center 用户。
- d. 对于发送人，请选择发送人。

 Important

您必须在 Amazon Simple Email Service (Amazon SES) 中验证发送人的电子邮件地址。有关更多信息，请参阅 Amazon Simple Email Service 开发人员指南中的[在 Amazon SES 中验证电子邮件地址](#)。

10. 在默认资产状态部分，您可以设置从此资产模型创建的警报的默认状态。

**Note**

您可以在后续步骤中为从此资产模型创建的资产激活或停用此警报。

11. 在高级设置部分，您可以配置权限、其他通知设置、警报状态操作、SiteWise 监控器中的警报模式以及确认流。

**Note**

AWS IoT Events 警报需要以下服务角色：

- AWS IoT Events 假定向发送警报状态值的角色 AWS IoT SiteWise。
- 一个 AWS IoT Events 假定向 Lambda 发送数据的角色。只有当您的警报发送通知时，才需要此角色。

在附加权限策略页面上，执行以下操作：

- a. 对于 AWS IoT Events 角色，请使用现有角色或创建具有所需权限的角色。此角色需要允许 `iotevents.amazonaws.com` 担任角色的 `iotsitewise:BatchPutAssetPropertyValue` 许可和信任关系。
  - b. 对于 AWS IoT Events Lambda 角色，请使用现有角色或创建具有所需权限的角色。此角色需要 `lambda:InvokeFunction` 和 `sso-directory:DescribeUser` 权限以及允许担任 `iotevents.amazonaws.com` 角色的信任关系。
12. ( 可选 ) 在其他通知设置部分，执行以下操作：
    - a. 在接收人属性中，您可以定义一个属性，其值指定通知的接收人。您可以选择 IAM Identity Center 用户作为接收人。

您可以在资产模型上创建属性或使用现有属性。

- 如果选择创建新的接收人属性，请指定该属性的接收人属性名称和接收人默认值 - 可选。
- 如果选择使用现有接收人属性，请在接收人属性名称中选择该属性。警报使用您选择属性的默认值。

您可以改写从此资产模型创建的每个资产的默认值。

- b. 对于自定义消息属性，您可以定义一个属性，该属性的值指定除了默认状态更改消息之外还要发送的自定义消息。例如，您可以指定一条消息，以帮助您的团队了解如何处理此警报。

您可以选择在资产模型上创建属性或使用现有属性。

- 如果您选择创建新的自定义消息属性，请指定该属性的自定义消息属性名称和自定义消息默认值 - 可选。
- 如果您选择使用现有的自定义消息属性，请在自定义消息属性名称中选择该属性。警报使用您选择属性的默认值。

您可以改写从此资产模型创建的每个资产的默认值。

- c. 对于 Lambda 函数，请执行下列操作之一：
  - 要 AWS IoT SiteWise 创建新的 Lambda 函数，请选择从 AWS 托管模板创建新的 Lambda。
  - 要使用现有 Lambda 函数，请选择使用现有 Lambda 并选择该函数的名称。

有关详细信息，请参阅 AWS IoT Events 开发人员指南 中的 [管理警报通知](#)。

13. ( 可选 ) 在设置状态操作部分，执行以下操作：

- a. 选择编辑操作。
- b. 在添加警报状态操作下，添加操作。然后选择保存。

您最多可以添加 10 个操作。

AWS IoT Events 可以在警报处于活动状态时执行操作。您可以定义内置操作以使用计时器或设置变量，或者将数据发送到其他 AWS 资源。有关更多信息，请参阅 [AWS IoT Events 开发人员指南](#) 中的 支持的操作。

14. ( 可选 ) 在 “ SiteWise 监控器-可选 ” 中的 “管理警报模式” 下，选择 “活动” 或 “非活动”。

使用此选项可以在 M SiteWise onitorss 中更新警报模型。默认情况下，此选项处于启用状态。

15. 在确认流下，选择活动或非活动。有关确认流的更多信息，请参阅 [警报状态](#)。

16. 选择添加警报。

**Note**

AWS IoT SiteWise 控制台发出多个 API 请求，将警报添加到资产模型中。当您选择添加警报时，控制台会打开一个对话框，显示这些 API 请求的进度。在每个 API 请求成功或一个 API 请求失败之前，请保持停留在此页面。如果请求失败，请关闭对话框，修复问题，然后选择添加警报重试。

## 定义 AWS IoT Events 警报 ( AWS IoT Events 控制台 )

您可以使用 AWS IoT Events 控制台在现有资产模型上定义 AWS IoT Events 警报。要在新资产模型上定义 AWS IoT Events 警报，请创建资产模型，然后完成以下步骤。有关更多信息，请参阅 [创建资产模型](#)。


**Important**

每个警报都需要一个属性，来指定要与警报进行比较的阈值。必须先在资产模型上定义阈值属性，然后才能定义警报。

举例说明，如果您想定义一个警报，用于检测风力涡轮机何时超过其最大额定风速 50 英里/小时。在定义警报之前，您必须定义一个默认值为 50 的属性 ( 最大风速 )。

### 在资产模型上定义 AWS IoT Events 警报

1. 导航到 [AWS IoT Events 控制台](#)。
2. 在导航窗格中，选择警报模型。
3. 选择创建警报模型。
4. 输入警报的名称。
5. ( 可选 ) 输入警报的描述。
6. 在 **警报目标** 部分，执行以下操作：
  - a. 对于目标选项，选择 AWS IoT SiteWise 资产属性。
  - b. 选择要为其添加警报的资产模型。
7. 在 **阈值定义** 部分，您可以定义警报何时检测以及警报的严重程度。执行以下操作：

- a. 选择警报检测的属性。每次此属性收到新值时，都会 AWS IoT SiteWise 将该值发送 AWS IoT Events 到以评估警报的状态。
  - b. 选择用于将属性与阈值进行比较的运算符。从以下选项中进行选择：
    - 小于
    - 小于或等于
    - = (等于)
    - != 不等于
    - 大于或等于
    - 大于
  - c. 对于“值”，选择要用作阈值的属性属性。AWS IoT Events 将属性的值与该属性的值进行比较。
  - d. 输入警报的严重程度。使用您的团队能理解的数字来反映此警报的严重程度。
8. ( 可选 ) 在通知设置-可选部分中，执行以下操作：
- a. 对于协议，请从以下选项中选择：
    - 电子邮件和短信 - 警报通过短信和电子邮件通知 IAM Identity Center 用户。
    - 电子邮件 - 警报通过电子邮件通知 IAM Identity Center 用户。
    - 短信 - 警报通过短信通知 IAM Identity Center 用户。
  - b. 对于 发送人，请选择发送人。
-  **Important**

您必须在 Amazon Simple Email Service (Amazon SES) 中验证发送人的电子邮件地址。有关更多信息，请参阅 Amazon Simple Email Service 开发人员指南 中的 [在 Amazon SES 中验证电子邮件地址](#)。
- c. 在接收人属性 - 可选中选择属性。警报使用您选择属性的默认值。
  - d. 在自定义消息属性 - 可选中选择属性。警报使用您选择属性的默认值。
9. 在实例部分，指定此警报的默认状态。您可以在稍后的步骤中为通过该资产模型创建的所有资产激活或停用此警报。
10. 在高级设置设置中，您可以配置权限、其他通知设置、警报状态操作、SiteWise 监控器中的警报模式以及确认流。



**Note**

AWS IoT Events 警报需要以下服务角色：

- AWS IoT Events 假定向发送警报状态值的角色 AWS IoT SiteWise。
- 一个 AWS IoT Events 假定向 Lambda 发送数据的角色。只有当您的警报发送通知时，才需要此角色。

- a. 在确认流部分，选择已启用或已禁用。有关确认流的更多信息，请参阅 [警报状态](#)。
- b. 在附加权限策略页面上，执行以下操作：
  - i. 对于 AWS IoT Events 角色，请使用现有角色或创建具有所需权限的角色。此角色需要允许 `iotevents.amazonaws.com` 担任角色的 `iotsitewise:BatchPutAssetPropertyValue` 许可和信任关系。
  - ii. 对于 Lambda 角色，请使用现有角色或创建具有所需权限的角色。此角色需要 `lambda:InvokeFunction` 和 `sso-directory:DescribeUser` 权限以及允许担任 `iotevents.amazonaws.com` 角色的信任关系。
- c. ( 可选 ) 在其他通知设置窗格中，执行以下操作：
  - 对于 Lambda 函数，请执行下列操作之一：
    - 要 AWS IoT Events 创建新的 Lambda 函数，请选择创建新的 Lambda 函数。
    - 要使用现有 Lambda 函数，请选择使用现有 Lambda 函数并选择该函数的名称。

有关详细信息，请参阅 AWS IoT Events 开发人员指南 中的 [管理警报通知](#)。
- d. ( 可选 ) 在设置状态操作 - 可选部分中，执行以下操作：
  - 在警报状态操作下，添加操作，然后选择保存。

您最多可以添加 10 个操作。

AWS IoT Events 可以在警报处于活动状态时执行操作。您可以定义内置操作以使用计时器或设置变量，或者将数据发送到其他 AWS 资源。有关更多信息，请参阅 [AWS IoT Events 开发人员指南](#) 中的 支持的操作。

## 11. 选择创建。

**Note**

AWS IoT Events 控制台发出多个 API 请求，将警报添加到资产模型中。当您选择添加警报时，控制台会打开一个对话框，显示这些 API 请求的进度。在每个 API 请求成功或一个 API 请求失败之前，请保持停留在此页面。如果请求失败，请关闭对话框，修复问题，然后选择添加警报重试。

## 定义 AWS IoT Events 警报 (AWS CLI)

您可以使用 AWS Command Line Interface (AWS CLI) 来定义监控资产属性的 AWS IoT Events 警报。您可以在新的或现有的资产模型上定义警报。在资产模型上定义警报后，可以在中创建警报 AWS IoT Events 并将其连接到资产模型。在此过程中，您将执行以下操作：

### 步骤

- [步骤 1：在资产模型上定义警报](#)
- [步骤 2：定义 AWS IoT Events 警报模型](#)
- [步骤 3：启用 AWS IoT SiteWise 和之间的数据流 AWS IoT Events](#)

### 步骤 1：在资产模型上定义警报

向新的或现有的资产模型添加警报定义和关联属性。

#### 在资产模型上定义警报 (CLI)

1. 创建名为 `asset-model-payload.json` 的文件。按照其他部分中的步骤将资产模型的详细信息添加到文件中，但不要提交创建或更新资产模型的请求。在本节中，您将向 `asset-model-payload.json` 文件中的资产模型详细信息添加警报定义。
  - 有关如何创建资产模型的更多信息，请参阅[创建资产模型 \(AWS CLI\)](#)。
  - 有关如何更新现有资产模型的更多信息，请参阅[更新资产或组件模型 \(AWS CLI\)](#)。

**Note**

您的资产模型必须定义至少一个资产属性，包括要通过警报监控的资产属性。


2. 向资产模型添加警报复合模型 (assetModelCompositeModels)。AWS IoT Events 警报组合模型指定警报IOT\_EVENTS类型并指定警报源属性。在中创建警报模型后，您可以添加警报源属性AWS IoT Events。

**⚠ Important**

警报组合模型必须与您稍后创建的 AWS IoT Events 警报模型同名。警报模型名称只能包含字母数字字符。指定一个唯一的字母数字名称，这样您就可以对警报模型使用相同的名称。

```
{
  ...
  "assetModelCompositeModels": [
    {
      "name": "BoilerTemperatureHighAlarm",
      "type": "AWS/ALARM",
      "properties": [
        {
          "name": "AWS/ALARM_TYPE",
          "dataType": "STRING",
          "type": {
            "attribute": {
              "defaultValue": "IOT_EVENTS"
            }
          }
        },
        {
          "name": "AWS/ALARM_STATE",
          "dataType": "STRUCT",
          "dataTypeSpec": "AWS/ALARM_STATE",
          "type": {
            "measurement": {}
          }
        }
      ]
    }
  ]
}
```

3. 向资产模型添加警报阈值属性。指定用于此阈值的默认值。您可以基于此模型覆盖每个资产的此默认值。

 Note

警报阈值属性必须为 INTEGER 或 DOUBLE。

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "Temperature Max Threshold",
      "dataType": "DOUBLE",
      "type": {
        "attribute": {
          "defaultValue": "105.0"
        }
      }
    }
  ]
}
```

4. (可选) 向资产模型添加警报通知属性。这些属性指定了 IAM Identity Center 的收件人和其他输入，这些输入 AWS IoT Events 用于在警报状态发生变化时发送通知。您可以基于此模型覆盖每个资产的这些默认值。

 Important

您可以向 AWS IAM Identity Center 用户发送警报通知。要使用此功能，您必须启用 IAM Identity Center。一次只能在一个 AWS 区域启用 IAM 身份中心。这意味着您只能在启用 IAM Identity Center 的区域中定义警报通知。有关更多信息，请参阅《AWS IAM Identity Center 用户指南》中的[入门](#)。

执行以下操作：

- a. 添加一个属性来指定您的 IAM Identity Center 身份存储的 ID。您可以使用 IAM 身份中心 [ListInstances](#) API 操作列出您的身份存储。此操作仅适用于您启用 IAM Identity Center 的区域。

```
aws sso-admin list-instances
```

然后，将身份存储 ID（例如 `d-123EXAMPLE`）指定为该属性的默认值。

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "identityStoreId",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "d-123EXAMPLE"
        }
      }
    }
  ]
}
```

- b. 添加一个属性，指定接收通知的 IAM Identity Center 用户 ID。要定义默认通知接收人，请将 IAM Identity Center 用户 ID 添加为默认值。执行以下任一操作以获取 IAM Identity Center 用户 ID：
  - i. 您可以使用 IAM Identity Center [ListUsers](#) API 来获取您知道其用户名的用户的 ID。将 `d-123EXAMPLE` 替换为您的身份存储的 ID，并将 `##` 替换为用户的用户名。

```
aws identitystore list-users \
  --identity-store-id d-123EXAMPLE \
  --filters AttributePath=UserName,AttributeValue=Name
```

- ii. 使用 [IAM Identity Center 控制台](#) 浏览您的用户并查找用户 ID。

然后，将用户 ID（例如 `123EXAMPLE-a1b2c3d4-5678-90ab-cdef-33333EXAMPLE`）指定为属性的默认值，或者定义不带默认值的属性。

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "userId",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "123EXAMPLE-a1b2c3d4-5678-90ab-cdef-33333EXAMPLE"
        }
      }
    }
  ]
}
```

- c. (可选) 添加一个属性，指定短信 (文本) 消息通知的默认发送人 ID。发送人 ID 在 Amazon Simple Notification Service (Amazon SNS) 发送的消息上显示为消息发送人。有关更多信息，请参阅 Amazon Simple Notification Service 开发人员指南 中的 [使用 Amazon SNS 请求短信发送者 ID](#)。

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "senderId",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "MyFactory"
        }
      }
    }
  ]
}
```

- d. (可选) 添加一个属性，指定用作电子邮件通知中发送人地址的默认电子邮件地址。

```
{
  ...
  "assetModelProperties": [
```

```

...
{
  "name": "fromAddress",
  "dataType": "STRING",
  "type": {
    "attribute": {
      "defaultValue": "my.factory@example.com"
    }
  }
}
]
}

```

- e. (可选) 添加一个属性，指定电子邮件通知中使用的默认主题。

```

{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "emailSubject",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "[ALERT] High boiler temperature"
        }
      }
    }
  ]
}

```

- f. (可选) 添加一个属性，指定要包含在通知中的其他消息。默认情况下，通知消息包含有关警报的信息。您还可以添加一条额外的消息，为用户提供更多信息。

```

{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "additionalMessage",
      "dataType": "STRING",
      "type": {
        "attribute": {

```

```

        "defaultValue": "Turn off the power before you check the alarm."
    }
}
]
}

```

## 5. 创建资产模型或更新现有资产模型。请执行以下操作之一：

- 要创建资源模型，请运行以下命令。

```
aws iotsitewise create-asset-model --cli-input-json file://asset-model-payload.json
```

- 要更新现有资源模型，请运行以下命令。将 *asset-model-id* 替换为资产模型的 ID。

```
aws iotsitewise update-asset-model \
  --asset-model-id asset-model-id \
  --cli-input-json file://asset-model-payload.json
```

运行命令之后，请记下响应中的 `assetModelId`。

### 示例：锅炉资产模型

以下资产模型表示报告温度数据的锅炉。该资产模型定义了一个警报，用于检测锅炉何时过热。

```

{
  "assetModelName": "Boiler Model",
  "assetModelDescription": "Represents a boiler.",
  "assetModelProperties": [
    {
      "name": "Temperature",
      "dataType": "DOUBLE",
      "unit": "C",
      "type": {
        "measurement": {}
      }
    },
    {
      "name": "Temperature Max Threshold",
      "dataType": "DOUBLE",
      "type": {
        "attribute": {

```



```
        "defaultValue": "105.0"
      }
    },
    {
      "name": "identityStoreId",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "d-123EXAMPLE"
        }
      }
    },
    {
      "name": "userId",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "123EXAMPLE-a1b2c3d4-5678-90ab-cdef-33333EXAMPLE"
        }
      }
    },
    {
      "name": "senderId",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "MyFactory"
        }
      }
    },
    {
      "name": "fromAddress",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "my.factory@example.com"
        }
      }
    },
    {
      "name": "emailSubject",
      "dataType": "STRING",
      "type": {
```

```

        "attribute": {
            "defaultValue": "[ALERT] High boiler temperature"
        }
    },
    {
        "name": "additionalMessage",
        "dataType": "STRING",
        "type": {
            "attribute": {
                "defaultValue": "Turn off the power before you check the alarm."
            }
        }
    }
],
"assetModelHierarchies": [

],
"assetModelCompositeModels": [
    {
        "name": "BoilerTemperatureHighAlarm",
        "type": "AWS/ALARM",
        "properties": [
            {
                "name": "AWS/ALARM_TYPE",
                "dataType": "STRING",
                "type": {
                    "attribute": {
                        "defaultValue": "IOT_EVENTS"
                    }
                }
            },
            {
                "name": "AWS/ALARM_STATE",
                "dataType": "STRUCT",
                "dataTypeSpec": "AWS/ALARM_STATE",
                "type": {
                    "measurement": {}
                }
            }
        ]
    }
]
]

```

```
}
```

## 步骤 2：定义 AWS IoT Events 警报模型

在中创建警报模型 AWS IoT Events。在中 AWS IoT Events，您可以使用表达式来指定警报模型中的值。您可以使用表达式来指定 AWS IoT SiteWise 要评估的值，并将其用作警报的输入。向警报模型 AWS IoT SiteWise 发送资产属性值时，AWS IoT Events 会评估表达式以获取该属性的值或资产的 ID。您可以在警报模型中使用以下表达式：

- 资产属性值

要获取资产属性的值，请使用以下表达式。将 `##ModelId` 替换为资产模型的 ID，并将 `propertyID #####` 的 ID。

```
$sitewise.assetModel.`assetModelId`.`propertyId`.propertyValue.value
```

- 资产 ID

要获取资产的 ID，请使用以下表达式。将 `##ModelId` 替换为资产模型的 ID，并将 `propertyID # #####` 的 ID。

```
$sitewise.assetModel.`assetModelId`.`propertyId`.assetId
```

### Note

创建警报模型时，您可以定义字面值，而不是计算结果为 AWS IoT SiteWise 值的表达式。这可以减少您在资产模型上定义的属性数量。但是，如果您将值定义为文字，则无法根据资产模型自定义资产的该值。您的 AWS IoT SiteWise Monitor 用户也无法自定义警报，因为他们只能在资产上配置警报设置。

## 创建 AWS IoT Events 警报模型 (CLI)

1. 在中创建警报模型时 AWS IoT Events，必须指定警报使用的每个属性的 ID，其中包括以下内容：
  - 复合资产模型中的警报状态属性
  - 警报监控的属性
  - 阈值属性


- ( 可选 ) IAM Identity Center 身份存储 ID 属性
- ( 可选 ) IAM Identity Center 用户 ID 属性
- ( 可选 ) 短信发送人 ID 属性
- ( 可选 ) 电子邮件发送人地址属性
- ( 可选 ) 电子邮件主题属性
- ( 可选 ) 附加消息属性

运行以下命令以检索资产模型上这些属性的 ID。将 *asset-model-id* 替换为上一步骤中的资产模型的 ID。

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

此操作将返回包含资产模型详细信息的响应。记下警报使用的每个属性的 ID。在下一步中创建 AWS IoT Events 警报模型时，您将使用这些 ID。

2. 在中创建警报模型 AWS IoT Events。执行以下操作：
  - a. 创建名为 `alarm-model-payload.json` 的文件。
  - b. 将以下 JSON 对象复制到文件中。
  - c. 为您的警报输入名称 (`alarmModelName`)、说明 (`alarmModelDescription`) 和严重性 (`severity`)。对于严重性，请指定一个反映贵公司严重性级别的整数。

 Important

警报模型必须与您之前在资产模型上定义的警报复合模型同名。  
警报模型名称只能包含字母数字字符。

```
{  
  "alarmModelName": "BoilerTemperatureHighAlarm",  
  "alarmModelDescription": "Detects when the boiler temperature is high.",  
  "severity": 3  
}
```

- d. 将比较规则 (`alarmRule`) 添加到警报中。此规则定义了要监控的属性 (`inputProperty`)、要比较的阈值 (`threshold`) 和要使用的比较运算符 (`comparisonOperator`)。

- 将 `##ModelId` 替换为资产模型的 ID。
- 将 `##PropertyId` 替换为警报监控的属性的 ID。
- 将 `##AttributeId` 替换为阈值属性属性的 ID。
- 将 `GREATER` 替换为运算符，用于将属性值与阈值进行比较。从以下选项中进行选择：
  - LESS
  - LESS\_OR\_EQUAL
  - EQUAL
  - NOT\_EQUAL
  - GREATER\_OR\_EQUAL
  - GREATER

```
{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
    }
  }
}
```

- e. 添加一个操作 (alarmEventActions)，当警报状态发生更改时将警报状态发送给 AWS IoT SiteWise。

#### Note

对于高级配置，您可以定义警报状态更改时要执行的其他操作。例如，您可以调用 AWS Lambda 函数或发布到 MQTT 主题。有关更多信息，请参阅《AWS IoT Events 开发人员指南》中的[使用其他 AWS 服务](#)。

- 将 `##ModelId` 替换为资产模型的 ID。
- 将 `##PropertyId` 替换为警报监控的属性的 ID。
- 将 `## StateProperty ID` 替换为警报复合模型中警报状态属性的 ID。

```
{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.assetId",
          "propertyId": "`alarmStatePropertyId`"
        }
      }
    ]
  }
}
```

- f. (可选) 配置警报通知设置。警报通知操作使用您账户中的 Lambda 函数发送警报通知。有关更多信息，请参阅 [警报通知的要求](#)。在警报通知设置中，您可以将短信和电子邮件通知配置为发送给 IAM Identity Center 用户。执行以下操作：
  - i. 将警报通知配置 (alarmNotification) 添加到 alarm-model-payload.json 中的有效负载。
    - 将 `## NotificationFunction Arn` 替换为处理警报通知的 Lambda 函数的 ARN。

```

{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId` .propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId` .propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId` .assetId",
          "propertyId": "'alarmStatePropertyId'"
        }
      }
    ]
  },
  "alarmNotification": {
    "notificationActions": [
      {
        "action": {
          "lambdaAction": {
            "functionArn": "alarmNotificationFunctionArn"
          }
        }
      }
    ]
  }
}

```

- ii. ( 可选 ) 将短信通知 (smsConfigurations) 配置为在警报状态发生变化时发送给 IAM Identity Center 用户。

- 将 `##StoreIdAttributeId` 替换为包含 IAM Identity Center 身份存储 ID 的属性的 ID。
- 将 `## IdAttribute ID` 替换为包含 IAM 身份中心用户 ID 的属性的 ID。
- 将 `###IdAttribute##` 替换为包含 Amazon SNS 发件人编号的属性 ID，或者 `senderId` 从有效负载中删除。
- 将 `附# MessageAttribute ID` 替换为包含附加消息的属性的 ID，或者 `additionalMessage` 从有效负载中移除。

```
{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId` .propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId` .propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId` .assetId",
          "propertyId": "alarmStatePropertyId"
        }
      }
    ]
  },
  "alarmNotification": {
    "notificationActions": [
      {
        "action": {
          "lambdaAction": {
            "functionArn": "alarmNotificationFunctionArn"
          }
        }
      }
    ]
  }
}
```



```

    "smsConfigurations": [
      {
        "recipients": [
          {
            "ssoIdentity": {
              "identityStoreId":
                "$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.value",
              "userId":
                "$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
            }
          }
        ],
        "senderId":
          "$sitewise.assetModel.`assetModelId`.`senderIdAttributeId`.propertyValue.value",
        "additionalMessage":
          "$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
      }
    ]
  }
}

```

- iii. (可选) 将电子邮件通知 (emailConfigurations) 配置为在警报状态发生变化时发送给 IAM Identity Center 用户。
- 将 `##StoreIdAttributeId` 替换为 IAM Identity Center 身份存储 ID 属性属性的 ID。
  - 将 `## IdAttribute ID` 替换为 IAM 身份中心用户 ID 属性属性的 ID。
  - 将 `from AddressAttribute mId` 替换为“发件人”地址属性的 ID，或者 `from` 从负载中移除。
  - 将 `#### SubjectAttribute ID` 替换为电子邮件主题属性的 ID，或者 `subject` 从有效负载中移除。
  - 将 `附# MessageAttribute ID` 替换为附加消息属性的 ID，或者 `additionalMessage` 从有效负载中移除。

```

{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {

```

```

    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.assetId",
          "propertyId": "'alarmStatePropertyId'"
        }
      }
    ]
  },
  "alarmNotification": {
    "notificationActions": [
      {
        "action": {
          "lambdaAction": {
            "functionArn": "alarmNotificationFunctionArn"
          }
        },
        "smsConfigurations": [
          {
            "recipients": [
              {
                "ssoIdentity": {
                  "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.value",
                  "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
                }
              }
            ],
            "senderId":
"$sitewise.assetModel.`assetModelId`.`senderIdAttributeId`.propertyValue.value",
            "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
          }
        ]
      }
    ]
  }
}

```



```

    "threshold":
    "$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
  }
},
"alarmEventActions": {
  "alarmActions": [
    {
      "iotSiteWise": {
        "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.assetId",
        "propertyId": "`alarmStatePropertyId`"
      }
    }
  ]
},
"alarmNotification": {
  "notificationActions": [
    {
      "action": {
        "lambdaAction": {
          "functionArn": "alarmNotificationFunctionArn"
        }
      },
      "smsConfigurations": [
        {
          "recipients": [
            {
              "ssoIdentity": {
                "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.value",
                "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
              }
            }
          ],
          "senderId":
"$sitewise.assetModel.`assetModelId`.`senderIdAttributeId`.propertyValue.value",
          "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
        }
      ],
      "emailConfigurations": [
        {

```

```

        "from":
"$sitewise.assetModel.`assetModelId`.`fromAddressAttributeId`.propertyValue.value",
        "recipients": {
            "to": [
                {
                    "ssoIdentity": {
                        "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.value"
                        "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
                    }
                }
            ],
            "content": {
                "subject":
"$sitewise.assetModel.`assetModelId`.`emailSubjectAttributeId`.propertyValue.value",
                "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
            }
        }
    ],
    },
    "alarmCapabilities": {
        "initializationConfiguration": {
            "disabledOnInitialization": false
        },
        "acknowledgeFlow": {
            "enabled": true
        }
    }
}

```

- h. 添加 AWS IoT Events 可以假设向其发送数据的 IAM 服务角色 (roleArn) AWS IoT SiteWise。此角色需要 `iotsitewise:BatchPutAssetPropertyValue` 权限以及允许 `iotevents.amazonaws.com` 担任角色的信任关系。要发送通知，此角色还需要 `lambda:InvokeFunction` 和 `sso-directory:DescribeUser` 权限。有关更多信息，请参阅 AWS IoT Events 开发人员指南中的 [警报服务角色](#)。

- 将 `roleArn` 替换为 AWS IoT Events 可以代入执行这些操作的角色的 ARN。

```

{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId` .propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId` .propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId` .assetId",
          "propertyId": "'alarmStatePropertyId'"
        }
      }
    ]
  },
  "alarmNotification": {
    "notificationActions": [
      {
        "action": {
          "lambdaAction": {
            "functionArn": "alarmNotificationFunctionArn"
          }
        },
        "smsConfigurations": [
          {
            "recipients": [
              {
                "ssoIdentity": {
                  "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId` .propertyValue.value",
                  "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId` .propertyValue.value"
                }
              }
            ]
          }
        ]
      }
    ]
  }
}

```

```

    }
  ],
  "senderId":
"$sitewise.assetModel.`assetModelId`.`senderIdAttributeId`.propertyValue.value",
  "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
}
],
"emailConfigurations": [
{
  "from":
"$sitewise.assetModel.`assetModelId`.`fromAddressAttributeId`.propertyValue.value",
  "recipients": {
    "to": [
      {
        "ssoIdentity": {
          "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.value"
          "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
        }
      }
    ]
  },
  "content": {
    "subject":
"$sitewise.assetModel.`assetModelId`.`emailSubjectAttributeId`.propertyValue.value",
    "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
  }
}
]
},
"alarmCapabilities": {
  "initializationConfiguration": {
    "disabledOnInitialization": false
  },
  "acknowledgeFlow": {
    "enabled": false
  }
},
"roleArn": "arn:aws:iam::123456789012:role/MyIoTEventsAlarmRole"

```

```
}
```

- i. 运行以下命令从中的有效负载创建 AWS IoT Events 警报模型 `alarm-model-payload.json`。

```
aws iotevents create-alarm-model --cli-input-json file://alarm-model-payload.json
```

- j. 操作将返回响应，其中包含警报模型的 ARN，`alarmModelArn`。复制此 ARN，以便下一步在资产模型的警报定义中进行设置。

### 步骤 3：启用 AWS IoT SiteWise 和之间的数据流 AWS IoT Events

在 AWS IoT SiteWise 和中创建所需资源后 AWS IoT Events，您可以启用资源之间的数据流以启用警报。在本节中，您将更新资产模型中的警报定义以使用您在上一步中创建的警报模型。

#### 启用 AWS IoT SiteWise 和之间的数据流 AWS IoT Events (CLI)

- 在资产模型中将警报模型设置为警报的来源。执行以下操作：
  - a. 运行以下命令以检索现有资产模型定义。将 `asset-model-id` 替换为资产模型的 ID。

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

此操作将返回包含资产模型详细信息的响应。

- b. 创建一个名为 `update-asset-model-payload.json` 的文件，并将上一命令的响应复制到该文件中。
- c. 从 `update-asset-model-payload.json` 文件中删除以下键/值对：
  - `assetModelId`
  - `assetModelArn`
  - `assetModelCreationDate`
  - `assetModelLastUpdateDate`
  - `assetModelStatus`
- d. 将警报源属性 (`AWS/ALARM_SOURCE`) 添加到您之前定义的警报复合模型中。将 `##ModelArn` 替换为警报模型的 ARN，后者设置警报源属性的值。



```

...
"assetModelCompositeModels": [
  ...
  {
    "name": "BoilerTemperatureHighAlarm",
    "type": "AWS/ALARM",
    "properties": [
      {
        "id": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "name": "AWS/ALARM_TYPE",
        "dataType": "STRING",
        "type": {
          "attribute": {
            "defaultValue": "IOT_EVENTS"
          }
        }
      },
      {
        "id": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
        "name": "AWS/ALARM_STATE",
        "dataType": "STRUCT",
        "dataTypeSpec": "AWS/ALARM_STATE",
        "type": {
          "measurement": {}
        }
      },
      {
        "name": "AWS/ALARM_SOURCE",
        "dataType": "STRING",
        "type": {
          "attribute": {
            "defaultValue": "alarmModelArn"
          }
        }
      }
    ]
  }
]
}

```

- e. 运行以下命令，使用存储在 `update-asset-model-payload.json` 文件中的定义更新资产模型。将 `asset-model-id` 替换为资产模型的 ID。

```
aws iotsitewise update-asset-model \  
  --asset-model-id asset-model-id \  
  --cli-input-json file://update-asset-model-payload.json
```

现在，您的资产模型定义了一个在 AWS IoT Events 中检测到的警报。报警根据该资产模型对所有资产中的目标属性进行监控。您可以为每项资产配置警报，以自定义每项资产的阈值或 IAM Identity Center 接收人等属性。有关更多信息，请参阅 [为资产配置警报](#)。

## 定义外部警报

外部警报包含您在 AWS IoT SiteWise 外部检测到的警报状态。

### 定义外部警报（控制台）

您可以使用 AWS IoT SiteWise 控制台在现有资产模型上定义外部警报。要在新资产模型上定义外部警报，请创建资产模型，然后完成以下步骤。有关更多信息，请参阅 [创建资产模型](#)。

在资产模型上定义警报

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择模型。
3. 选择要为其定义警报的资产模型。
4. 选择警报定义选项卡。
5. 选择添加警报。
6. 在警报类型选项中，选择外部警报。
7. 输入警报的名称。
8. （可选）输入警报的描述。
9. 选择添加警报。

### 定义外部警报 (CLI)

您可以使用在新资产模型或现有资产模型上定义外部警报。AWS CLI

要向资产模型添加外部警报，请向资产模型添加警报复合模型。外部警报复合模型指定 EXTERNAL 类型，但不指定警报源属性。以下复合警报示例定义了外部温度警报。

```
{
  ...
  "assetModelCompositeModels": [
    {
      "name": "BoilerTemperatureHighAlarm",
      "type": "AWS/ALARM",
      "properties": [
        {
          "name": "AWS/ALARM_TYPE",
          "dataType": "STRING",
          "type": {
            "attribute": {
              "defaultValue": "EXTERNAL"
            }
          }
        },
        {
          "name": "AWS/ALARM_STATE",
          "dataType": "STRUCT",
          "dataTypeSpec": "AWS/ALARM_STATE",
          "type": {
            "measurement": {}
          }
        }
      ]
    }
  ]
}
```

有关如何向新的或现有的资产模型添加复合模型的更多信息，请参阅以下内容：

- [创建资产模型 \(AWS CLI\)](#)
- [更新资产或组件模型 \(AWS CLI\)](#)

定义外部警报后，您可以根据资产模型将警报状态推送到资产。有关更多信息，请参阅 [获取外部警报状态](#)。

## 为资产配置警报

在资产模型上定义 AWS IoT Events 警报后，您可以根据资产模型为每项资产配置警报。您可以编辑警报的阈值和通知设置。其中每个值都是资产的一个属性，因此您可以更新属性的默认值来配置这些值。

**Note**

您可以为 AWS IoT Events 警报配置这些值，但不能为外部警报配置这些值。

**主题**

- [配置阈值 \(控制台\)](#)
- [配置阈值 \(AWS CLI\)](#)
- [配置通知设置 \(控制台\)](#)
- [配置通知设置 \(CLI\)](#)

## 配置阈值 (控制台)

您可以使用 AWS IoT SiteWise 控制台更新用于指定警报阈值的属性的值。

### 更新警报阈值 (控制台)

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 资产。
3. 选择要更新警报阈值的资产。

**Tip**

您可以选择箭头图标来展开资产层次结构以查找资产。

4. 选择编辑。
5. 找到警报用于其阈值的属性，然后输入其新值。
6. 选择保存。

## 配置阈值 (AWS CLI)

您可以使用 AWS Command Line Interface (AWS CLI) 更新指定警报阈值的属性的值。

您必须知道资产的 `assetId` 和属性的 `propertyId` 才能完成此过程。您也可以使用外部 ID。如果您创建了资产但不知道该资产 `assetId`，请使用 [ListAssets](#) API 列出特定模型的所有资产。使用该 [DescribeAsset](#) 操作查看资产的属性，包括财产 ID。

使用“[BatchPutAssetPropertyValue](#)”操作为您的资产分配属性值。您可以使用此操作一次设置多个属性。此操作的负载包含条目列表，每个条目都包含资产 ID、属性 ID 和属性值。

### 更新属性的值 (AWS CLI)

1. 创建一个名为 `batch-put-payload.json` 的文件，并将以下 JSON 对象复制到该文件中。此示例负载演示如何设置风力涡轮机的纬度和经度。更新 ID、值和时间戳以修改用例的负载。

```
{
  "entries": [
    {
      "entryId": "windfarm3-turbine7-latitude",
      "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
      "propertyValues": [
        {
          "value": {
            "doubleValue": 47.6204
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          }
        }
      ]
    },
    {
      "entryId": "windfarm3-turbine7-longitude",
      "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE",
      "propertyValues": [
        {
          "value": {
            "doubleValue": 122.3491
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          }
        }
      ]
    }
  ]
}
```

- 负载中的每个条目都包含一个 `entryId`，您可以将其定义为任何唯一的字符串。如果任何请求条目失败，则每个错误都将包含相应请求的 `entryId`，以便您知道要重试的请求。
- 要设置属性值，可以在每个属性属性的列表中包含一个 `timestamp-quality-value (TQV)` 结构。`propertyValues` 此结构必须包含新的 `value` 和当前的 `timestamp`。
  - `value` – 包含下列字段之一的结构，具体取决于要设置的属性的类型：
    - `booleanValue`
    - `doubleValue`
    - `integerValue`
    - `stringValue`
  - `timestamp`— 一种包含当前 Unix 纪元时间（以秒为单位）的结构，`timeInSeconds`。AWS IoT SiteWise 拒绝任何时间戳在过去存在超过 7 天或将来超过 5 分钟的数据点。

有关如何为 [BatchPutAssetPropertyValue](#) 准备负载的更多信息，请参阅 [使用 API 摄取数据 AWS IoT SiteWise](#)。

2. 运行以下命令将属性值发送到 AWS IoT SiteWise：

```
aws iotsitewise batch-put-asset-property-value --cli-input-json file://batch-put-payload.json
```

## 配置通知设置（控制台）

您可以使用 AWS IoT SiteWise 控制台更新为警报指定通知设置的属性的值。

### 更新警报的通知设置（控制台）

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 资产。
3. 选择要更新其警报设置的资产。
4. 选择编辑。
5. 找到警报用于要更改的通知设置的属性，然后输入其新值。
6. 选择保存。

## 配置通知设置 (CLI)

您可以使用 AWS Command Line Interface (AWS CLI) 更新为警报指定通知设置的属性的值。

您必须知道资产的 `assetId` 和属性的 `propertyId` 才能完成此过程。您也可以使用外部 ID。如果您创建了资产但不知道该资产 `assetId`，请使用 [ListAssets](#) API 列出特定模型的所有资产。使用该 [DescribeAsset](#) 操作查看资产的属性，包括财产 ID。

使用 “[BatchPutAssetProperty](#)值” 操作为您的资产分配属性值。您可以使用此操作一次设置多个属性。此操作的负载包含条目列表，每个条目都包含资产 ID、属性 ID 和属性值。

### 更新属性的值 (AWS CLI)

1. 创建一个名为 `batch-put-payload.json` 的文件，并将以下 JSON 对象复制到该文件中。此示例负载演示如何设置风力涡轮机的纬度和经度。更新 ID、值和时间戳以修改用例的负载。

```
{
  "entries": [
    {
      "entryId": "windfarm3-turbine7-latitude",
      "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
      "propertyValues": [
        {
          "value": {
            "doubleValue": 47.6204
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          }
        }
      ]
    },
    {
      "entryId": "windfarm3-turbine7-longitude",
      "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE",
      "propertyValues": [
        {
          "value": {
            "doubleValue": 122.3491
          },
          "timestamp": {
```

```

        "timeInSeconds": 1575691200
      }
    }
  ]
}

```

- 负载中的每个条目都包含一个 `entryId`，您可以将其定义为任何唯一的字符串。如果任何请求条目失败，则每个错误都将包含相应请求的 `entryId`，以便您知道要重试的请求。
- 要设置属性值，可以在每个属性属性的列表中包含一个 `timestamp-quality-value (TQV)` 结构。 `propertyValues` 此结构必须包含新的 `value` 和当前的 `timestamp`。
  - `value` – 包含下列字段之一的结构，具体取决于要设置的属性的类型：
    - `booleanValue`
    - `doubleValue`
    - `integerValue`
    - `stringValue`
  - `timestamp`— 一种包含当前 Unix 纪元时间（以秒为单位）的结构， `timeInSeconds`。  
AWS IoT SiteWise 拒绝任何时间戳在过去存在超过 7 天或将来超过 5 分钟的数据点。

有关如何为 [BatchPutAssetPropertyValue](#) 准备负载的更多信息，请参阅 [使用 API 摄取数据 AWS IoT SiteWise](#)。

2. 运行以下命令将属性值发送到 AWS IoT SiteWise：

```
aws iotsitewise batch-put-asset-property-value -\-cli-input-json file://batch-put-payload.json
```

## 响应警报

当 AWS IoT Events 警报状态发生变化时，您可以执行以下操作来响应警报：

- 确认警报以表明您正在处理问题。
- 暂停警报以将其暂时停用。
- 禁用警报可将其永久禁用，直到您再次启用警报。
- 启用已禁用的警报以检测警报状态。



- [重置警报](#)以清除其状态和最新值。

您可以使用 AWS IoT SiteWise 控制台或 AWS IoT Events API 来响应警报。

#### Note

您可以响应 AWS IoT Events 警报，但不能响应外部警报。

#### 主题

- [响应警报 \(控制台\)](#)
- [响应警报 \(API\)](#)

## 响应警报 (控制台)

您可以使用 AWS IoT SiteWise 控制台确认、暂停、禁用或启用警报。

#### 主题

- [确认警报 \(控制台\)](#)
- [暂停警报 \(控制台\)](#)
- [禁用警报 \(控制台\)](#)
- [启用警报 \(控制台\)](#)
- [重置警报 \(控制台\)](#)

## 确认警报 (控制台)

您可以确认警报以表明您正在处理问题。

#### Note

您必须在警报上启用确认流，这样才能确认警报。如果您从 AWS IoT SiteWise 控制台定义警报，则默认情况下会启用此选项。

## 确认警报 ( 控制台 )

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 资产。
3. 选择要确认警报的资产。

### Tip

您可以选择箭头图标来展开资产层次结构以查找资产。

4. 选择 警报 选项卡。
5. 选择要确认的警报，然后选择 操作 以打开响应操作菜单。
6. 选择确认。警报的状态更改为 已确认。

## 暂停警报 ( 控制台 )

您可以暂停警报以将其暂时停用。指定暂停警报的持续时间。

### 对警报设置稍后提醒 (控制台)

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 资产。
3. 选择要对警报设置稍后提醒的资产。

### Tip

您可以选择箭头图标来展开资产层次结构以查找资产。

4. 选择 警报 选项卡。
5. 选择要设置稍后提醒的警报，然后选择操作以打开响应操作菜单。
6. 选择暂停。将打开一个模型，您可以在其中指定暂停的持续时间。
7. 选择 暂停时长或输入 自定义暂停时长。
8. 选择保存。警报的状态更改为 已暂停。

## 禁用警报 ( 控制台 )

您可以禁用警报，使其不再检测。禁用警报后，如果您希望警报检测，则必须重新启用警报。

### 禁用警报 ( 控制台 )

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 资产。
3. 选择要禁用其警报的资产。

#### Tip

您可以选择箭头图标来展开资产层次结构以查找资产。

4. 选择 警报 选项卡。
5. 选择要禁用的警报，然后选择操作以打开响应操作菜单。
6. 选择 禁用。警报的状态更改为 已禁用。

## 启用警报 ( 控制台 )

您可以启用警报，使其在禁用或暂停后再次检测。

### 若要启用警报 ( 控制台 )

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 资产。
3. 选择要启用警报的资产。

#### Tip

您可以选择箭头图标来展开资产层次结构以查找资产。

4. 选择 警报 选项卡。
5. 选择要启用的警报，然后选择操作以打开响应操作菜单。
6. 请选择 启用。警报的状态更改为 正常。

## 重置警报 (控制台)

您可以重置警报以清除其状态和最新值。

若要重置警报 (控制台)

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 资产。
3. 选择要重置警报的资产。

### Tip

您可以选择箭头图标来展开资产层次结构以查找资产。

4. 选择 警报 选项卡。
5. 选择要启用的警报，然后选择操作以打开响应操作菜单。
6. 选择 重置。警报的状态更改为 正常。

## 响应警报 (API)

您可以使用 AWS IoT Events API 来确认、暂停、禁用、启用或重置警报。有关更多信息，请参阅 [AWS IoT Events API 参考](#) 中的以下操作：

- [BatchAcknowledge警报](#)
- [BatchSnooze警报](#)
- [BatchDisable警报](#)
- [BatchEnable警报](#)
- [BatchReset警报](#)

有关更多信息，请参阅 [AWS IoT Events 开发人员指南](#) 中的 [响应警报](#)。

## 获取外部警报状态

外部警报是您在外部评估的警报 AWS IoT SiteWise。当您有一个数据来源报告您想要获取到 AWS IoT SiteWise的警报状态时，您可以使用外部警报。

警报状态属性要求警报状态数据值采用特定的格式。每个数据值都必须是序列化为字符串的 JSON 对象。然后，将序列化的字符串作为字符串值提取。有关更多信息，请参阅 [警报状态属性](#)。

Example 警报状态数据值示例（未序列化）

```
{
  "stateName": "Active"
}
```

Example 警报状态数据值示例（序列化）

```
{"stateName\":\"Active\"}
```

### Note

如果您的数据来源无法报告这种格式的数据，或者您无法在采集数据之前将其转换为此格式，则可以选择不使用警报属性。相反，例如，您可以将数据作为具有字符串数据类型的测量属性进行获取。有关更多信息，请参阅 [定义来自设备的数据流（测量）](#) 和 [正在将数据摄取到 AWS IoT SiteWise](#)。

## 映射外部警报状态流

您可以定义属性别名，将数据流映射到警报状态属性。这有助于您在获取或检索数据时轻松识别警报状态属性。有关属性别名的更多信息，请参阅 [将工业数据流映射到资产属性](#)。

主题

- [映射外部警报状态流（控制台）](#)
- [映射外部警报状态流（AWS CLI）](#)

### 映射外部警报状态流（控制台）

您可以定义属性别名，将数据流映射到警报状态属性。这有助于您在获取或检索数据时轻松识别警报状态属性。有关属性别名的更多信息，请参阅 [将工业数据流映射到资产属性](#)。

您可以使用 AWS IoT SiteWise 控制台为警报状态属性设置别名。

## 为警报状态属性设置属性别名 (控制台)

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 资产。
3. 选择要为其设置属性别名的资产。

### Tip

您可以选择箭头图标来展开资产层次结构以查找资产。

4. 选择编辑。
5. 滚动到“警报”并展开该部分。
6. 在“外部警报”下，在“属性别名 — 可选”中输入别名。
7. 选择保存。

## 映射外部警报状态流 (AWS CLI)

您可以定义属性别名，将数据流映射到警报状态属性。这有助于您在获取或检索数据时轻松识别警报状态属性。有关属性别名的更多信息，请参阅 [将工业数据流映射到资产属性](#)。

您可以使用 AWS Command Line Interface (AWS CLI) 为警报状态属性设置别名。

您必须知道资产的 `assetId` 和属性的 `propertyId` 才能完成此过程。您也可以使用外部 ID。如果您创建了资产但不知道该资产 `assetId`，请使用 [ListAssets](#) API 列出特定模型的所有资产。使用该 [DescribeAsset](#) 操作查看资产的属性，包括财产 ID。

### Note

[DescribeAsset](#) 响应包括该资产的复合资产模型列表。每个警报都是一个复合模型。要查找 `propertyId`，请找到警报的复合模型，然后在该复合模型中找到 `AWS/ALARM_STATE` 属性。

有关如何设置属性别名的更多信息，请参阅 [设置属性别名 \(AWS CLI\)](#)。

## 获取警报状态数据

警报状态属性期望警报状态为序列化的 JSON 字符串。要将警报状态摄取到中的外部警报 AWS IoT SiteWise，请将此序列化字符串作为带时间戳的字符串值提取。以下示例演示了活动警报的状态数据值。

```
{\"stateName\": \"Active\"}
```

要标识警报状态属性，您可以指定以下项之一：

- 您要向其发送数据的警报属性的 `assetId` 和 `propertyId`。
- `propertyAlias`，这是一个数据流别名（例如，`/company/windfarm/3/turbine/7/temperature/high`）。要使用此选项，您必须首先设置警报属性的别名。要了解如何为警报状态属性设置属性别名，请参阅 [映射外部警报状态流](#)。

以下示例 [BatchPutAssetPropertyValue](#) API 负载演示了如何格式化外部警报的状态。当风力涡轮机的每分钟转速 (RPM) 读数过高时，此外部警报会报告。

Example 警报状态数据的 BatchPutAssetPropertyValue 有效负载示例

```
{
  "entries": [
    {
      "entryId": "unique entry ID",
      "propertyAlias": "/company/windfarm/3/turbine/7/temperature/high",
      "propertyValues": [
        {
          "value": {
            "stringValue": "{\"stateName\": \"Active\"}"
          },
          "timestamp": {
            "timeInSeconds": 1607550262
          }
        }
      ]
    }
  ]
}
```

有关如何使用 BatchPutAssetPropertyValue API 获取数据的更多信息，请参阅 [使用 API 摄取数据 AWS IoT SiteWise](#)。

要进一步了解有关采集数据的其他方式，请参阅 [正在将数据摄取到 AWS IoT SiteWise](#)。

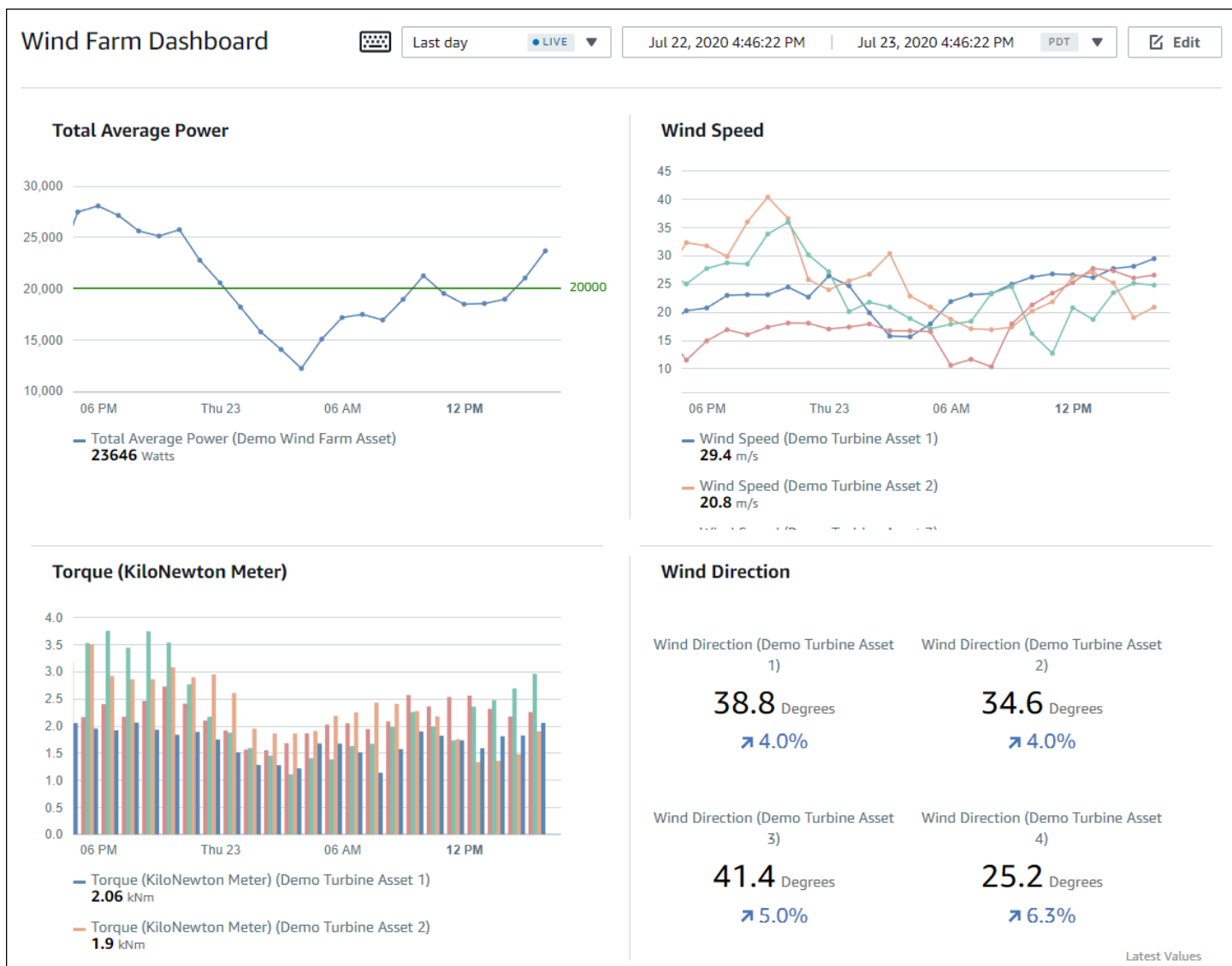


# 使用监控数据 AWS IoT SiteWise Monitor

您可以通过 AWS IoT SiteWise 创建 Monitor Web 门户来 SiteWise 监控来自流程、设备和设备的数据。SiteWise Monitor 是 AWS IoT SiteWise 项功能，可用于以托管 Web 应用程序的形式创建门户。然后，您可以使用这些门户查看和共享您的操作数据。您可以使用控制面板创建项目来可视化流程、设备以及连接到 AWS IoT 的装置中的数据。

领域专家（如流程工程师）可以使用这些门户快速查看操作数据，进而了解设备和装置的行为。

以下是显示风力发电厂数据的示例控制面板。



由于可以 AWS IoT SiteWise 捕获一段时间内的数据，因此您可以使用 M SiteWise onitor 查看一段时间内的操作数据，或者查看特定时间点上次报告的值。这样，您就可以发掘以其他方式可能很难发现的情况。

## SiteWise 监控角色

四个角色与 SiteWise 监视器交互：

### AWS 管理员

AWS 管理员使用 AWS IoT SiteWise 控制台创建门户。AWS 管理员还可以分配门户管理员和添加门户用户。门户管理员可在以后将门户用户分配到项目，作为拥有者或查看者。AWS 管理员只能在 AWS 控制台中工作。

### 门户管理员

每个 SiteWise 监视器门户都有一个或多个门户管理员。门户管理员使用门户创建包含资产和控制面板集合的项目。门户管理员随后会将资产和拥有者分配给每个项目。通过控制对项目的访问，门户管理员可以指定项目拥有者和查看者可以查看哪些资产。

### 项目拥有者

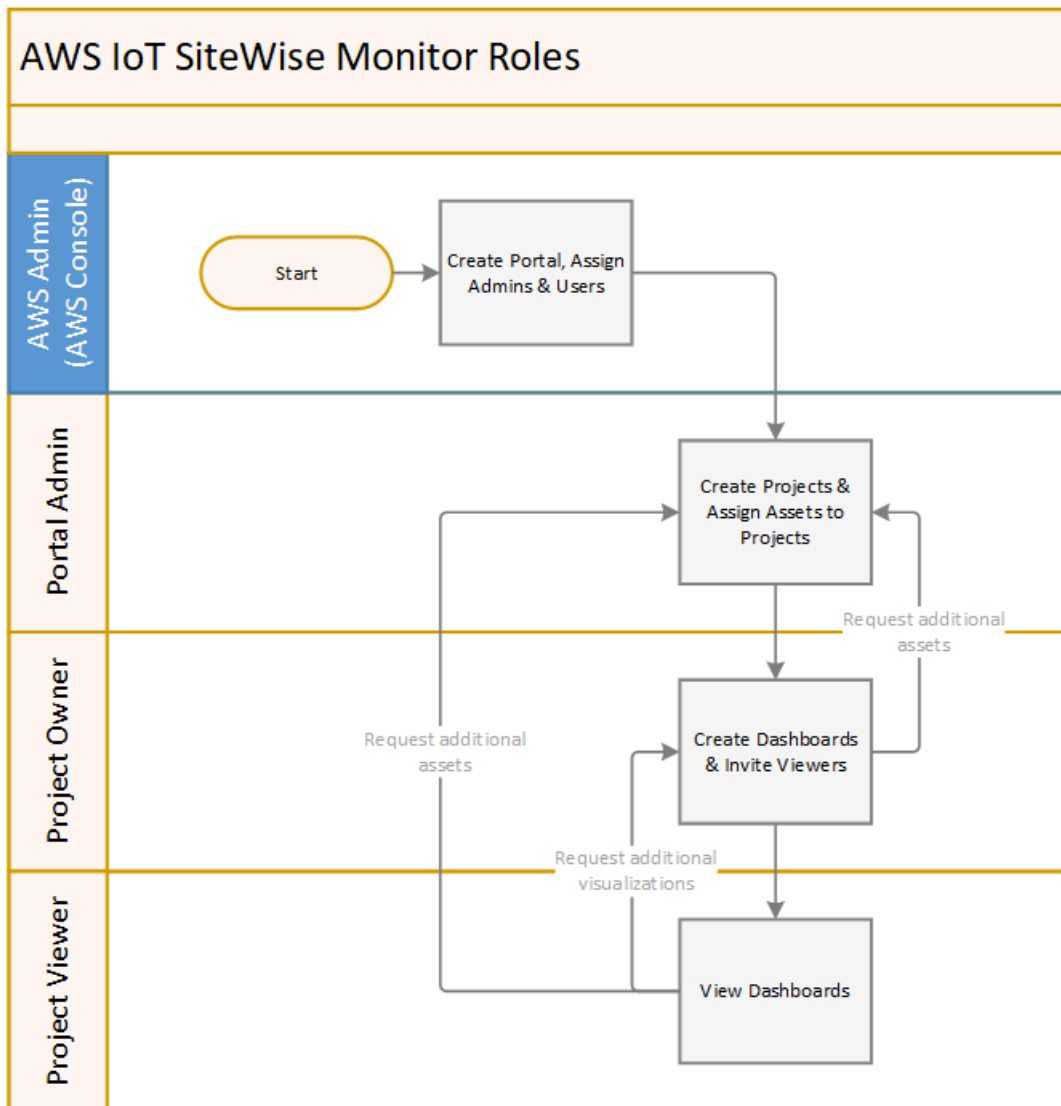
每个 M SiteWise onitor 项目都有所有者。项目所有者以控制面板形式创建可视化效果，来一致地呈现您的操作数据。当控制面板准备好共享时，项目所有者可以邀请查看者加入项目。项目所有者还可以将其他所有者分配给项目。项目所有者可以为警报配置阈值和通知设置。

### 项目查看者

每个 M SiteWise onitor 项目都有查看器。项目查看者可以连接到门户以查看项目拥有者创建的控制面板。在每个控制面板中，项目查看者可以调整时间范围以更好地了解运营数据。项目查看者只能查看其具有访问权限的项目中的控制面板。项目查看者可以确认和暂停警报。

根据您的组织，同一人员可能承担多种角色。

下图说明了这四个角色在 Monito SiteWise r 门户中的交互方式。



您可以使用 AWS IAM Identity Center 或 IAM 来管理谁有权访问您的数据。您的数据用户可以使用他们的 IAM 身份中心或 IAM 凭证从桌面或移动浏览器登录 M SiteWise onitor。

## SAML 联合身份验证

IAM Identity Center 和 IAM 通过 [SAML \(安全断言标记语言\) 2.0](#) 支持身份联合验证。SAML 2.0 是一个开放标准，许多外部身份提供商 (IdPs) 使用它来对用户进行身份验证并将其身份和安全信息传递给服务提供商 (SP)。SP 通常是应用程序或服务。SAML 联合允许您的 M SiteWise onitor 门户管理员和用户使用外部凭据 (例如公司用户名和密码) 登录他们分配的门户。

您可以将 IAM 身份中心和 IAM 配置为使用基于 SAML 的联合身份验证来访问您的 SiteWise 监控器门户。

## IAM Identity Center

您的门户管理员和用户可以使用其公司用户名和密码登录 AWS 访问门户。然后，他们可以导航到分配给他们的 SiteWise 监视器入口。IAM Identity Center 使用证书在您的身份提供商和 AWS 之间建立 SAML 信任关系。有关更多信息，请参阅 AWS IAM Identity Center 用户指南中的 [SCIM 配置文件](#) 和 [SAML 2.0 实施](#)。

## IAM

您的门户管理员和用户可以申请临时安全证书，以访问他们分配的 Monitor SiteWise 门户。您可以在 IAM 中创建 SAML 身份提供商身份，以便在您的身份提供商与 AWS 之间建立信任关系。有关更多信息，请参阅 IAM 用户指南中的 [使用基于 SAML 的联合身份 AWS 验证进行 API 访问](#)。

您的门户管理员和用户可以登录贵公司的门户，然后选择进入 AWS 管理控制台的选项。然后，他们可以导航到分配给他们的 SiteWise 监视器入口。贵公司的门户网站负责处理您的身份提供商与之间的信任交换 AWS。有关更多信息，请参阅 IAM 用户指南中的允许 SAML 2.0 联合用户 [访问 AWS 管理控制台](#)。

### Note

向门户添加用户或管理员时，请避免创建限制用户权限（例如受限 IP）的 IAM policy。任何权限受限的附加策略都将无法连接到 AWS IoT SiteWise 门户。

## SiteWise 显示器概念

要使用 SiteWise 监视器，您应该熟悉以下概念：

### 门户

AWS IoT SiteWise Monitor 门户是一个 Web 应用程序，可用于可视化和共享 AWS IoT SiteWise 数据。门户具有一个或多个管理员，并且包含零个或多个项目。

### 项目

每个 Monitor SiteWise 门户都包含一组项目。每个项目都有一个与之关联的 AWS IoT SiteWise 资产子集。项目所有者创建一个或多个控制面板，以提供一致的方式来查看与这些资产关联的数据。项目所有者可以邀请查看者加入项目，以允许他们查看项目中的资产和控制面板。该项目是 Monitor SiteWise 内部共享的基本单元。项目所有者可以邀请 AWS 管理员授予门户访问权限的用户。用户必须具有门户访问权限，然后才能与该用户共享该门户中的项目。

## 资产

当 AWS IoT SiteWise 从您的工业设备中摄取数据时，您的设备、设备和流程将分别表示为资产。每项资产都有与之关联的属性和警报。门户管理员还将资产组分配给每个项目。

## 属性

属性是与资产关联的时间序列数据。例如，一套设备可能具有序列号、位置、品牌和型号以及安装日期。它还可能具有可用性、性能、质量、温度、压力等时间序列值。

## 警报

警报监控属性，以识别设备何时超出其工作范围。每个警报都定义了一个阈值和一个要监控的属性。当属性超过阈值时，警报将变为活动状态，并表示您或您的团队中的某人应解决问题。项目所有者可以自定义警报的阈值和通知设置。项目查看者可以确认和暂停警报，他们可以留言，详细说明警报或他们为解决警报而采取的措施。

## 控制面板

每个项目都包含一组控制面板。控制面板为一组资产的值提供一组可视化效果。项目所有者创建控制面板及其包含的可视化效果。当项目所有者准备好共享控制面板集时，所有者可以邀请查看者加入项目，从而使他们能够访问项目中的所有控制面板。如果希望为不同的控制面板使用一组不同的查看器，则必须在项目之间划分控制面板。当查看者查看仪表板时，他们可以自定义时间范围以查看特定数据。

## 可视化

在每个控制面板中，项目所有者决定如何显示与项目关联的资产的属性和警报。可用性可能表示为折线图，而其他值可能显示为条形图或关键绩效指标 (KPI)。警报最好以状态网格和状态时间轴的形式显示。项目所有者可以自定义每个可视化效果，以便更好地了解该资产的数据。

# 入门 AWS IoT SiteWise Monitor

如果您是组织的 AWS 管理员，则可以通过 AWS IoT SiteWise 控制台创建门户。完成以下步骤以创建门户，以便您的组织成员可以查看您的 AWS IoT SiteWise 数据：

1. 配置和创建门户
2. 添加门户管理员并发送邀请电子邮件
3. 添加门户用户

创建门户后，门户管理员可以查看您的 AWS IoT SiteWise 资产并将其分配给门户中的项目。然后，项目拥有者可以创建控制面板来显示资产的属性，以帮助项目查看者了解设备、流程和装置的运行情况。

#### Note

向门户添加用户或管理员时，请避免创建限制用户权限 AWS Identity and Access Management（例如受限 IP）的 (IAM) 策略。任何具有受限权限的附加策略都将无法连接到 AWS IoT SiteWise 门户。

您可以按照教程的说明，逐步设置具有项目、控制面板和多个用户的门户，用于使用风电场数据的特定场景。有关更多信息，请参阅 [在 Monitor 中可视化和共享风电场 SiteWise 数据](#)。

#### 主题

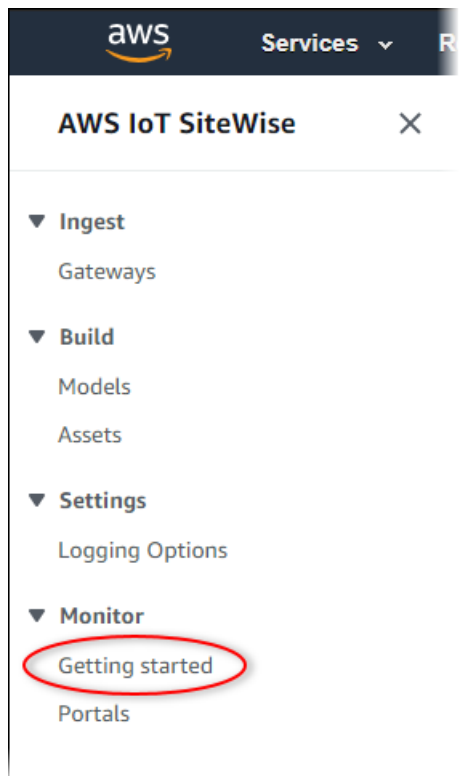
- [创建门户](#)
- [配置门户](#)
- [邀请管理员](#)
- [添加门户用户](#)

## 创建门户

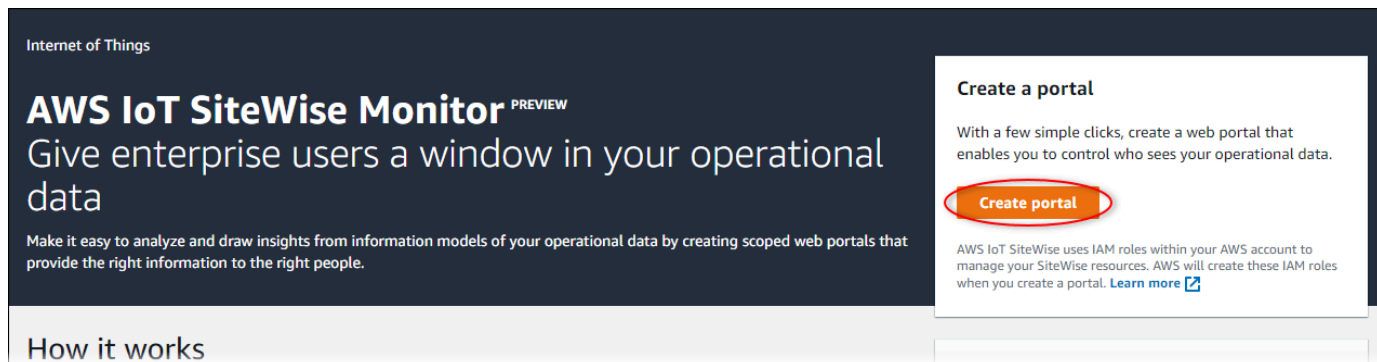
您可以在 AWS IoT SiteWise 控制台中创建 SiteWise 监控器门户。

#### 创建门户

1. 登录 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 **监控、入门**。



### 3. 选择 创建门户。



接下来，您必须提供一些基本信息来配置门户。

## 配置门户

用户使用门户查看您的数据。您可以自定义门户的名称、描述、品牌、用户验证、支持联系人电子邮件和权限。

AWS IoT SiteWise &gt; Monitor &gt; Portals &gt; Create portal

Step 1  
Portal configurationStep 2 - optional  
Additional featuresStep 3  
Invite administratorsStep 4  
Assign users

## Portal configuration

Each web portal provides enterprise users with access to your IoT SiteWise assets. [Learn more](#)

### Portal details

#### Portal name

Choose a portal name to identify the web portal to your users. Company name is recommended.

example-factory-1

Name should be 1-128 characters and only contain A-Z a-z 0-9 \_ and -.

#### Description - optional

Create a description of your portal

Example Corp Factory #1 in Renton, WA

Description should contain a maximum of 2048 characters.

### Portal branding

You can provide your logo image to display your brand in this web portal.

#### Logo image

Upload a square, high-resolution .png file. The image is displayed on a dark background.

Choose file

The file size must be less than 1 MB.

### User authentication

Your users can sign in to this portal with their AWS Single Sign-On (AWS SSO) or AWS Identity and Access Management (IAM) credentials. If you choose AWS SSO, you must enable the service for your AWS account.

 You haven't enabled AWS SSO in your account yet. When you create your first portal user, this automatically enables AWS SSO in your AWS account.

Create user

#### AWS SSO

Your users can sign in to the portal with their corporate usernames and passwords.

#### IAM

Your users can sign in to the portal with their IAM credentials.

### Support contact email

You can provide an email address for cases where there's a problem or issue with this portal and your users need to contact support to resolve.

#### Email

support@example.com

### Tags

This resource doesn't have any tags.

Add tag

You can add up to 50 more tags.

### Permissions

SiteWise Monitor assumes this role to give permissions to your federated users to access AWS IoT SiteWise resources. [Learn](#)



## 配置门户

1. 输入门户的名称。
2. (可选) 输入门户的描述。如果您有多个门户，请使用有意义的描述来帮助您跟踪每个门户包含的内容。
3. (可选) 上传图片以在门户中显示您的品牌。选择一个正方形 PNG 图片。如果上传非方形图片，门户会将图片缩减为正方形。
4. 请选择以下选项之一：
  - 如果您的门户用户使用其公司用户名和密码登录此门户，请选择 IAM Identity Center。

如果您尚未在账户中启用 IAM Identity Center，请执行以下操作：

- a. 选择 创建用户。
- b. 在 创建用户页面上，要创建第一个门户，请输入用户的电子邮件地址、名字和姓氏，然后选择创建用户。

The screenshot shows a 'Create user' dialog box. It contains the following text and fields:

- Title: Create user
- Message: When you create your first portal user, this automatically enables AWS SSO in your AWS account.
- Form fields:
  - Email address: janedoe@example.com
  - First name: Jane
  - Last name: Doe
- Buttons: Cancel, Create user

### Note

- AWS 当您创建第一个门户网站用户时，会自动在您的账户中启用 IAM 身份中心。
- 您一次只能在一个区域配置 IAM Identity Center。SiteWise Monitor 连接到您为 IAM 身份中心配置的区域。这意味着您可以使用一个区域访问 IAM Identity Center，但可以在任何区域创建门户。

- 如果您的门户用户使用其 IAM 凭证登录此门户，请选择 IAM。

**⚠ Important**

用户或角色必须具有登录门户的 `iotsitewise:DescribePortal` 权限。

5. 输入门户用户在遇到门户问题需要帮助时可以联系的电子邮件地址。
6. (可选) 为门户添加标签。有关更多信息，请参阅 [为资源添加 AWS IoT SiteWise 标签](#)。
7. 请选择以下选项之一：
  - 请选择创建和使用新的服务角色。默认情况下，M SiteWise monitor 会自动为每个门户创建一个服务角色。此角色允许您的门户用户访问您的 AWS IoT SiteWise 资源。有关更多信息，请参阅 [将服务角色用于 AWS IoT SiteWise Monitor](#)。
  - 选择使用现有服务角色，然后选择目标角色。
8. 选择 下一步。
9. 或者，为您的门户启用警报。有关更多信息，请参阅 [为门户启用警报](#)。
10. 选择“创建”。AWS IoT SiteWise 将创建您的门户。

**i Note**

如果关闭控制台，您可以通过添加管理员和用户来完成设置过程。有关更多信息，请参阅 [添加或删除门户管理员](#)。如果您不想保留此门户，请将其删除，以免使用资源。有关更多信息，请参阅 [删除门户](#)。

状态列可以是以下值之一。

- C@@ RE AT AWS IoT SiteWise ING-正在处理您创建门户的请求。此过程可能需要几分钟才能完成。
- 更新- AWS IoT SiteWise 正在处理您更新门户的请求。此过程可能需要几分钟才能完成。
- 待处理 —— AWS IoT SiteWise 正在等待 DNS 记录传播完成。此过程可能需要几分钟完成。当门户的状态为 待处理时，您可以将其删除。
- 正在删除- AWS IoT SiteWise 正在处理您删除门户的请求。此过程可能需要几分钟才能完成。
- 活动- 当门户变为活动状态时，您的门户用户可以访问它。
- 失败- AWS IoT SiteWise 无法处理您创建、更新或删除门户的请求。如果您启用 AWS IoT SiteWise 向 Amazon Logs 发送 CloudWatch 日志，则可以使用这些日志来解决问题。有关更多信息，请参阅 [AWS IoT SiteWise 使用 CloudWatch 日志进行监控](#)。

门户创建时将显示一条消息。

A green banner with a white checkmark icon on the left and a white 'X' icon on the right. The text in the center reads: "Successfully created portal URL at https://a1b2c3d4-5678-90ab-cdef-11111EXAMPLE.app.iotsitewise.aws".

Successfully created portal URL at https://a1b2c3d4-5678-90ab-cdef-11111EXAMPLE.app.iotsitewise.aws

接下来，您必须为门户邀请一个或多个门户管理员。到目前为止，您创建了一个门户，但没有人可以访问它。

## 邀请管理员

要开始使用新门户，您必须分配门户管理员。门户管理员创建项目、选择项目拥有者并将资产分配给项目。门户管理员可以查看您的所有 AWS IoT SiteWise 资产。

根据用户身份验证服务，请选择以下选项之一：

### IAM Identity Center

如果您是首次使用 M SiteWise onitor，则可以选择之前创建的用户作为门户管理员。如果您想添加其他用户作为门户管理员，可以从此页面创建 IAM Identity Center 用户。或者，您可以将外部身份提供商连接到 IAM Identity Center。有关更多信息，请参阅 [《AWS IAM Identity Center 用户指南》](#)。

#### 邀请管理员

1. 选中您希望作为门户管理员的用户对应的复选框。这会将用户添加到门户管理员列表中。

#### Note

如果您使用 IAM Identity Center 作为身份存储，并且已登录 AWS Organizations 管理账户，则可以选择创建用户来创建 IAM Identity Center 用户。IAM Identity Center 会向新用户发送一封电子邮件，让他们设置密码。然后，您可以将该用户作为管理员分配到门户。有关更多信息，请参阅[管理 IAM Identity Center 中的身份](#)。

2. (可选) 选择 向选定用户发送邀请。您的电子邮件客户端将打开，邮件正文中将填充邀请内容。

在将电子邮件发送给门户管理员之前您可以对其进行自定义。您也可以稍后再向门户管理员发送电子邮件。如果您是第一次尝试 M SiteWise onitor，并添加新的 IAM 身份中心或 IAM 用户或角色作为门户管理员，则无需给自己发送电子邮件。

3. 如果您添加了不希望其作为管理员的用户，请清除该用户对应的复选框。
4. 邀请门户管理员步骤完成后，选择 下一步。

## IAM

您可以选择用户或角色作为门户管理员。如果您想添加其他用户或角色作为门户管理员，可以在 IAM 控制台中创建用户或角色。有关更多信息，请参阅 IAM 用户指南中的[在您的 AWS 账户中创建 IAM 用户](#)和[创建 IAM 角色](#)。

### 邀请管理员

1. 执行以下操作：
  - 选择 IAM 用户以添加一个 IAM 用户作为您的门户管理员。
  - 选择 IAM 角色以将 IAM 角色添加为门户管理员。
2. 选中您希望作为门户管理员的用户或角色对应的复选框。这会将用户或角色添加到门户管理员列表中。
3. 如果您添加了不希望其作为管理员的用户或角色，请清除该用户或角色对应的复选框。
4. 邀请门户管理员步骤完成后，选择 下一步。

#### Important

用户或角色必须具有登录门户的 `iotsitewise:DescribePortal` 权限。

#### Note

如果您使用 IAM Identity Center 作为身份存储，并且已登录 AWS Organizations 管理账户，则可以选择创建用户来创建 IAM Identity Center 用户。IAM Identity Center 会向新用户发送一封电子邮件，让他们设置密码。然后，您可以将该用户作为管理员分配到门户。有关更多信息，请参阅[管理 IAM Identity Center 中的身份](#)。

您可以在以后更改门户管理员列表。有关更多信息，请参阅[添加或删除门户管理员](#)。

#### Note

由于只有门户管理员可以创建项目并将资产分配给项目，因此在您指定至少一个门户管理员之前，门户的用途非常有限。

在最后一步中，您添加可以访问新门户的用户。

## 添加门户用户

您控制哪些用户可以访问您的门户。在每个门户中，门户管理员创建一个或多个项目，并分配门户用户作为每个项目的拥有者或查看者。每个项目的拥有者可以邀请其他门户用户来拥有或查看项目。

根据用户身份验证服务，请选择以下选项之一：

### IAM Identity Center

如果您要在用户列表中添加用户，请完成以下步骤。

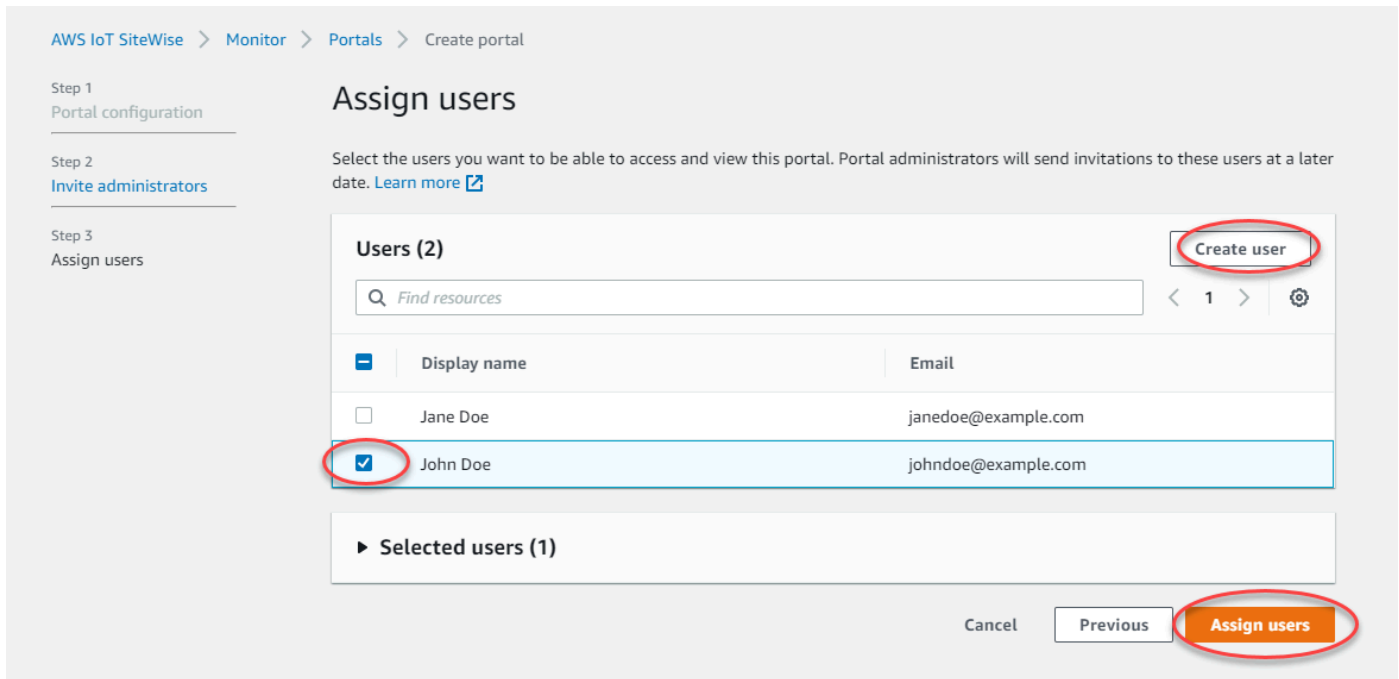
#### 添加门户用户

1. 从用户列表中选择要添加到门户的用户。这会将用户添加到门户用户列表中。如果您是首次使用 M SiteWise onitor，则无需将门户管理员添加为门户用户。

#### Note

如果您使用 IAM Identity Center 作为身份存储，并且已登录 AWS Organizations 管理账户，则可以选择创建用户来创建 IAM Identity Center 用户。IAM Identity Center 会向新用户发送一封电子邮件，让他们设置密码。然后，您可以将用户作为用户分配给门户。有关更多信息，请参阅[管理 IAM Identity Center 中的身份](#)。

2. 如果您添加了不希望其访问门户的用户，请清除该用户对应的复选框。
3. 选择完用户后，选择分配用户。



## IAM

如果您看到要在 IAM 用户或 IAM 角色列表中添加的用户或角色，请完成以下步骤。

### 添加门户用户

#### 1. 执行以下操作：

- 选择 IAM 用户以将 IAM 用户添加为门户用户。
- 选择 IAM 角色以将 IAM 角色添加为门户用户。

如果您是首次使用 M SiteWise onitor，则无需将门户管理员添加为门户用户。

2. 选中要作为门户用户的用户或角色所对应的复选框。这会将用户或角色添加到门户用户列表中。
3. 如果您添加了不希望其访问门户的用户，请清除该用户对应的复选框。
4. 选择完用户后，选择分配用户。

#### **⚠ Important**

用户或角色必须具有登录门户的 `iotsitewise:DescribePortal` 权限。

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1  
Portal configuration

Step 2  
[Invite administrators](#)

Step 3  
Assign users

## Assign users

Select the users you want to be able to access and view this portal. Portal administrators will send invitations to these users at a later date. [Learn more](#)

**Users** Roles

**IAM users (1)** [Manage users in IAM console](#)

Find user name

<input checked="" type="checkbox"/>	Name	Date created
<input checked="" type="checkbox"/>	raspberrypi-testing	11-08-2019

Portal users (1) [Remove](#)

Cancel Previous **Assign users**

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1  
Portal configuration

Step 2  
Invite administrators

Step 3  
Assign users

## Assign users

Select the users you want to be able to access and view this portal. Portal administrators will send invitations to these users at a later date. [Learn more](#)

Users | **Roles**

**IAM roles (66)** [Manage roles in IAM console](#)

Find role name

<input type="checkbox"/>	Name	Date created
<input type="checkbox"/>	[REDACTED]	
<input type="checkbox"/>	[REDACTED]	
<input checked="" type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_4wZigNpA1	03-16-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_EcKt-2Oar	03-11-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_GTnd0O4Wr	03-16-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_rHINLNCS-	03-11-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_XB330QUIO	03-10-2021
<input type="checkbox"/>	[REDACTED]	
<input type="checkbox"/>	[REDACTED]	
<input type="checkbox"/>	[REDACTED]	

► Portal users (2) [Remove](#)

Cancel Previous **Assign users**

恭喜您！您已成功创建门户、分配门户管理员并分配在收到邀请时可以使用该门户的用户。门户管理员现在可以创建项目并将资产添加到这些项目。然后，项目所有者可以创建控制面板来可视化各个项目资产的数据。

您可以在以后更改门户用户列表。有关更多信息，请参阅 [添加或删除门户用户](#)。

如果需要对门户进行更改，请参阅 [管理您的 SiteWise 监控器门户](#)。

要开始使用门户，请参阅《SiteWise 监控器应用程序指南》中的“[入门](#)”。



## 创建控制面板 (AWS Command Line Interface)

当您使用 AWS CLI 在控制面板中定义可视化内容 ( 或小部件 ) 时，必须在 `dashboardDefinition` JSON 文档中指定以下信息。此定义是 [CreateDashboard](#) 和 [UpdateDashboard](#) 操作的参数。

### `widgets`

小部件定义结构的列表，其中每个结构均包含以下信息：

#### `type`

小部件的类型。AWS IoT SiteWise 提供了以下小部件类型：

- `sc-line-chart` – 折线图。有关更多信息，请参阅 AWS IoT SiteWise Monitor 应用指南中的 [折线图](#)。
- `sc-scatter-chart` – 散点图。有关更多信息，请参阅 AWS IoT SiteWise Monitor 应用程序指南中的 [散点图](#)。
- `sc-bar-chart` – 条形图。有关更多信息，请参阅 AWS IoT SiteWise Monitor 应用指南中的 [条形图](#)。
- `sc-status-grid` – 一个状态控件，以网格形式显示资产属性的最新值。有关更多信息，请参阅 AWS IoT SiteWise Monitor 应用程序指南中的 [状态控件](#)。
- `sc-status-timeline` – 一个状态控件，以时间轴的形式显示资产属性的历史值。有关更多信息，请参阅 AWS IoT SiteWise Monitor 应用程序指南中的 [状态控件](#)。
- `sc-kpi` – 关键绩效指标 (KPI) 可视化。有关更多信息，请参阅 AWS IoT SiteWise Monitor 应用指南中的 [KPI 控件](#)。
- `sc-table` – 表格控件。有关更多信息，请参阅 AWS IoT SiteWise Monitor 应用指南中的 [表格控件](#)。

#### `title`

小部件的标题。

#### `x`

小部件的水平位置，从网格的左侧开始。此值表示小部件在控制面板网格中的位置。

`y`

小部件的垂直位置，从网格的顶部开始。此值表示小部件在控制面板网格中的位置。

`width`

小部件的宽度，用控制面板网格上的空格数表示。

`height`

小部件的高度，用控制面板网格上的空格数表示。

`metrics`

指标结构的列表，其中每个结构均为此小部件定义一个数据流。此列表中的每个结构都必须包含以下信息：

`label`

要为此指标显示的标签。

`type`

此指标的数据源类型。AWS IoT SiteWise 提供以下指标类型：

- `iotsitewise` – 控制面板提取 AWS IoT SiteWise 中的资产属性的数据。如果选择此选项，则必须为此指标定义 `assetId` 和 `propertyId`。

`assetId`

( 可选 ) AWS IoT SiteWise 中的资产的 ID。

如果您在该指标中为 `type` 选择 `iotsitewise`，则此字段是必填字段。

`propertyId`

( 可选 ) AWS IoT SiteWise 中的资产属性的 ID。

如果您在该指标中为 `type` 选择 `iotsitewise`，则此字段是必填字段。

`analysis`

或者，一种结构，用于定义要为小组件显示的分析（例如趋势线）。有关更多信息，请参阅 AWS IoT SiteWise Monitor 应用指南中的 [配置趋势线](#)。您可以在小组件中为每个属性添加每种类型的趋势线。分析结构包含以下信息：

`trends`

或者，趋势结构列表，每个趋势结构都为该小组件定义了趋势分析。此列表中的每个结构包含以下信息：

## type

趋势线的类型。选择以下选项：

- `linear-regression`— 显示线性回归线。SiteWise Monitor 使用[最小二乘法](#)计算线性回归。

## annotations

或者，定义小组件阈值的注释结构。有关更多信息，请参阅<https://docs.aws.amazon.com/iot-sitewise/latest/appguide/configure-thresholds.html>版本指南中的AWS IoT SiteWise Monitor配置阈值。每个控件可以添加最多 6 个注释。注释结构包含以下信息：

## y

或者，注释结构列表，每个注释结构都定义了此小组件的水平阈值。此列表中的每个结构包含以下信息：

## comparisonOperator

阈值的比较运算符。选择以下操作之一：

- `LT` – 突出显示至少有一个数据点小于 `value` 的属性。
- `GT` – 突出显示至少有一个数据点大于 `value` 的属性。
- `LTE` – 突出显示至少有一个数据点小于或等于 `value` 的属性。
- `GTE` – 突出显示至少有一个数据点大于或等于 `value` 的属性。
- `EQ` – 突出显示至少有一个数据点等于 `value` 的属性。

## value

用于将数据点与 `comparisonOperator` 进行比较的阈值。

## color

( 可选 ) 阈值颜色的 6 位十六进制代码。对于至少有一个数据点符合阈值规则的属性，可视化效果会以这种颜色显示属性图例。默认值为黑色 ( `#000000` )。

## showValue

或者，是否在控件的边距中显示阈值的值。默认值为 `true`。

## properties

( 可选 ) 小部件属性的平面字典。该结构的成员取决于上下文。AWS IoT SiteWise 提供了以下使用 `properties` 的小部件：

- [折线图](#)、[散点图](#)和[条形图](#)具有以下属性：

## colorDataAcrossThresholds

或者，是否更改此小组件中跨越阈值的数据的颜色。启用此选项后，超过阈值的数据将以您选择的颜色显示。默认值为 `true`。

- [状态网格](#)具有以下属性：

## labels

或者，一种结构，用于定义要在状态网格上显示的标签。标签结构包含以下信息：

## showValue

或者，是否在此小组件中显示每个资产属性的单位和值。默认值为 `true`。

## Example 示例控制面板定义

以下示例从存储在 JSON 文件中的负载定义控制面板。

```
aws iotsitewise create-dashboard \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE \  
  --dashboard-name "Wind Farm Dashboard" \  
  --dashboard-definition file://dashboard-definition.json
```

针对 `dashboard-definition.json` 的以下 JSON 示例定义具有以下可视化小部件的控制面板：

- 控制面板左上角的直观显示风电场总功率的折线图。该折线图包括一个阈值，该阈值指示风电场何时输出功率少于其最低预期输出。该折线图还包括一条线性回归趋势线。
- 控制面板右上角的直观显示四个涡轮机的风速的折线图。

### Note

此示例表示控制面板上的折线图和条形图可视化效果。此控制面板与[示例风电场控制面板](#)类似。

```
{  
  "widgets": [  
    {  
      "type": "sc-line-chart",  
      "title": "Total Average Power",
```

```
"x": 0,
"y": 0,
"height": 3,
"width": 3,
"metrics": [
  {
    "label": "Power",
    "type": "iotsitewise",
    "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
    "analysis": {
      "trends": [
        {
          "type": "linear-regression"
        }
      ]
    }
  }
],
"annotations": {
  "y": [
    {
      "comparisonOperator": "LT",
      "value": 20000,
      "color": "#D13212",
      "showValue": true
    }
  ]
}
},
{
  "type": "sc-bar-chart",
  "title": "Wind Speed",
  "x": 3,
  "y": 3,
  "height": 3,
  "width": 3,
  "metrics": [
    {
      "label": "Turbine 1",
      "type": "iotsitewise",
      "assetId": "a1b2c3d4-5678-90ab-cdef-2a2a2EXAMPLE",
      "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE"
    }
  ],
}
```

```
{
  "label": "Turbine 2",
  "type": "iotsitewise",
  "assetId": "a1b2c3d4-5678-90ab-cdef-2b2b2EXAMPLE",
  "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE"
},
{
  "label": "Turbine 3",
  "type": "iotsitewise",
  "assetId": "a1b2c3d4-5678-90ab-cdef-2c2c2EXAMPLE",
  "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE"
},
{
  "label": "Turbine 4",
  "type": "iotsitewise",
  "assetId": "a1b2c3d4-5678-90ab-cdef-2d2d2EXAMPLE",
  "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE"
}
]
}
]
```

## 为门户启用警报

您可以使用 AWS IoT Events 为门户启用支持的警报功能，以便门户管理员可以在您的 SiteWise 监控器入口中创建、编辑和删除 AWS IoT Events 警报模型。项目所有者可以配置警报。项目查看者可以查看警报详情。本节介绍如何使用 AWS IoT SiteWise 控制台为门户启用警报功能。

### Important

- 您无法在门户中创建外部警报。
- 如果要发送警报通知，则必须为用户身份验证服务选择 IAM Identity Center。
- 警报通知功能在中国（北京）不可用 AWS 区域。

配置和创建门户时，可以在步骤 2 其他功能中启用警报和警报通知。根据用户身份验证服务，请选择以下选项之一：

## IAM Identity Center

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1  
Portal configuration

Step 2- optional  
**Additional features**

Step 3  
Invite administrators

Step 4  
Assign users

## Additional features - *optional*

### Alarms

Your portal users can create alarms in the portal to monitor equipment or processes. They can also get notified when the equipment or processes perform outside specified range.

**Enable alarms**  
If enabled, your portal users can define AWS IoT Events alarms in SiteWise Monitor.

**AWS IoT SiteWise access role**  
Choose an IAM role that allows AWS IoT Events to send data to AWS IoT SiteWise. To edit the role, go to the [IAM console](#).

Create a role from an AWS managed template

Use an existing role

**Enable alarm notifications**  
If enabled, alarms can send email or SMS notifications.

**Sender**  
Specify the email address that sends alarm notifications. To edit or add a sender, go to the [Amazon SES console](#).

**AWS Lambda role**  
Choose an IAM role that allows AWS Lambda to send data to Amazon SES and Amazon SNS. To edit the role, go to the [IAM console](#).

Create a role from an AWS managed template

Use an existing role

**AWS Lambda function**  
Choose an AWS Lambda function to manage alarm notifications. To edit the function, go to the [AWS Lambda console](#).

Create a lambda from an AWS managed template

Use an existing lambda

Previous **Create**

## 为入口启用警报

1. 或者，选择启用警报。
  - 对于 AWS IoT SiteWise 访问角色，请使用现有角色或创建具有所需权限的角色。此角色需要 `iotevents:BatchPutMessage` 权限和信任关系，以允许 `iot.amazonaws.com` 和 `iotevents.amazonaws.com` 代入该角色。
2. 或者，选择启用警报通知。
  - a. 对于发送人，请选择发送人。

**⚠ Important**

您必须在 Amazon SES 中验证发送人的电子邮件地址。有关更多信息，请参阅 Amazon Simple Email Service 开发人员指南 中的 [在 Amazon SES 中验证电子邮件地址](#)。

- b. 对于 AWS Lambda 角色，请使用现有角色或创建具有所需权限的角色。此角色需要 `lambda:InvokeFunction` 和 `sso-directory:DescribeUser` 权限以及允许 `iotevents.amazonaws.com` 和 `lambda.amazonaws.com` 代入该角色的信任关系。
- c. 对于 AWS Lambda 函数，请选择现有 Lambda 函数或创建用于管理警报通知的函数。有关详细信息，请参阅 AWS IoT Events 开发人员指南 中的 [管理警报通知](#)。

## IAM

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1  
Portal configuration

Step 2- optional  
Additional features

Step 3  
Invite administrators

Step 4  
Assign users

### Additional features - optional

#### Alarms

Your portal users can create alarms in the portal to monitor equipment or processes. They can also get notified when the equipment or processes perform outside specified range.

Enable alarms  
If enabled, your portal users can define AWS IoT Events alarms in SiteWise Monitor.

**AWS IoT SiteWise access role**  
Choose an IAM role that allows AWS IoT Events to send data to AWS IoT SiteWise. To edit the role, go to the [IAM console](#).

Create a role from an AWS managed template  
 Use an existing role

**ⓘ** Alarms created in the portal can't send notifications. If you want to send alarm notifications, choose **Previous**. Then, on the **Portal configuration** page, choose **AWS SSO** for **User authentication**.

Previous **Create**

## 为入口启用警报

- 或者，选择启用警报。
  - 对于 AWS IoT SiteWise 访问角色，请使用现有角色或创建具有所需权限的角色。此角色需要 `iotevents:BatchPutMessage` 权限和信任关系，以允许 `iot.amazonaws.com` 和 `iotevents.amazonaws.com` 代入该角色。



有关 Monitor SiteWise r 中警报的更多信息，请参阅《AWS IoT SiteWise 应用指南》中的“[使用警报进行监控](#)”。

## 在边缘启用您的门户

在边缘启用门户后，在您的账户中启用了数据处理包的所有 SiteWise Edge 网关上均可使用此门户。

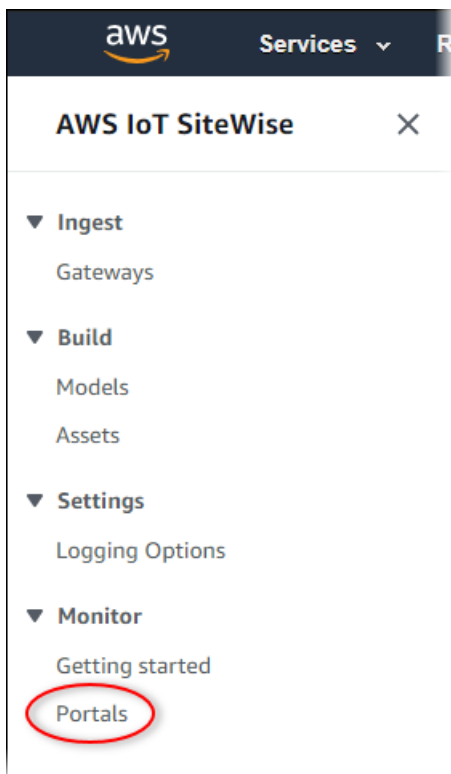
在边缘启用传送门

1. 在边缘配置部分，开启在边缘启用此门户。
2. 选择 Create ( 创建 )。

## 管理您的 SiteWise 监控器门户

您可能需要更新门户详细信息、更改管理员或将用户添加到您的门户。本节介绍如何完成 SiteWise 监控器门户的这些基本管理任务。

1. 登录 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 监控、门户。



3. 选择门户，然后选择 查看详细信息 ( 或选择门户的 名称 )。

#### 4. 您可以执行以下任一管理任务：

- [更改门户的名称、描述、品牌、支持电子邮件和权限](#)
- [添加或删除门户管理员](#)
- [向门户管理员发送电子邮件邀请](#)
- [添加或删除门户用户](#)
- [删除门户](#)

有关如何创建门户的信息，请参阅[入门 AWS IoT SiteWise Monitor](#)。

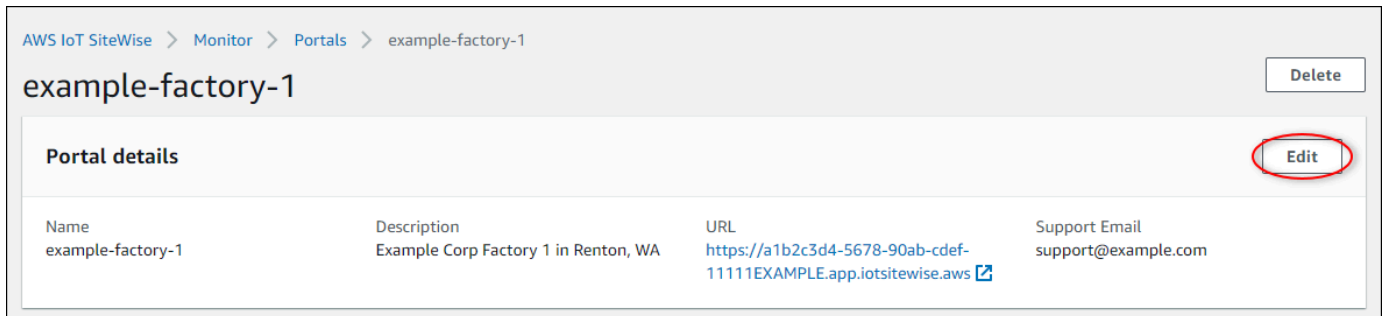
#### 主题

- [更改门户的名称、描述、品牌、支持电子邮件和权限](#)
- [添加或删除门户管理员](#)
- [向门户管理员发送电子邮件邀请](#)
- [添加或删除门户用户](#)
- [删除门户](#)

## 更改门户的名称、描述、品牌、支持电子邮件和权限

您可以更改门户的名称、描述、品牌、支持电子邮件和权限。

1. 在门户详细信息页面的门户详细信息部分，选择 **编辑**。

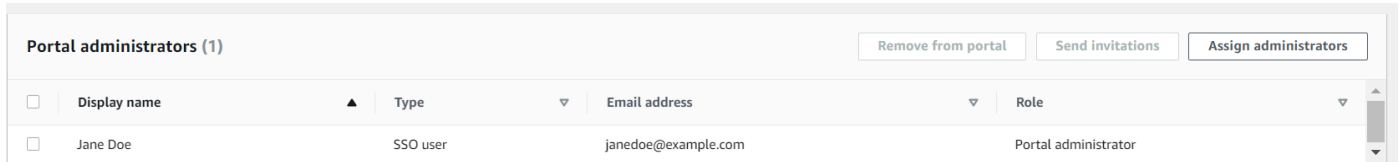


2. 更新 名称、描述、门户品牌、支持联系人电子邮件 或 权限。
3. 完成后，选择 **保存**。

## 添加或删除门户管理员

通过几个步骤，您可以添加或删除作为门户管理员的用户。根据用户身份验证服务，请选择以下选项之一。

### IAM Identity Center



The screenshot shows a table titled "Portal administrators (1)". At the top right, there are three buttons: "Remove from portal", "Send invitations", and "Assign administrators". The table has columns for "Display name", "Type", "Email address", and "Role". One row is visible with the following data:

<input type="checkbox"/>	Display name	Type	Email address	Role
<input type="checkbox"/>	Jane Doe	SSO user	janedoe@example.com	Portal administrator

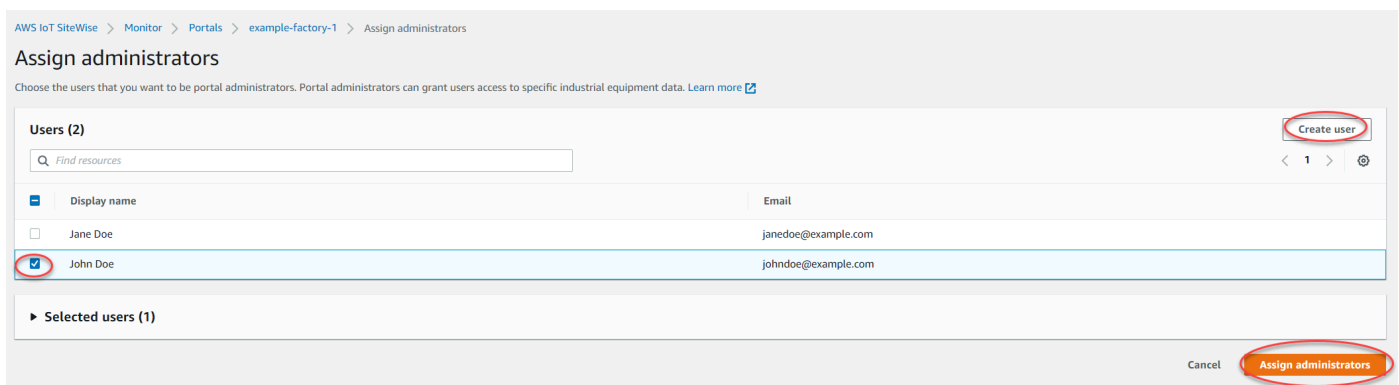
### 添加门户管理员

1. 在门户详细信息页面的门户管理员部分，选择分配管理员。
2. 在分配管理员页面上，选中作为管理员添加到门户的用户对应的复选框。

#### Note

如果您使用 IAM Identity Center 作为身份存储，并且已登录 AWS Organizations 管理账户，则可以选择创建用户来创建 IAM Identity Center 用户。IAM Identity Center 会向新用户发送一封电子邮件，让他们设置密码。然后，您可以将该用户作为管理员分配到门户。有关更多信息，请参阅[管理 IAM Identity Center 中的身份](#)。

3. 选择分配管理员。



The screenshot shows the "Assign administrators" page. At the top, there is a breadcrumb trail: "AWS IoT SiteWise > Monitor > Portals > example-factory-1 > Assign administrators". Below the breadcrumb, there is a heading "Assign administrators" and a sub-heading "Choose the users that you want to be portal administrators. Portal administrators can grant users access to specific industrial equipment data. [Learn more](#)".

The main content area is titled "Users (2)" and contains a search bar with the placeholder text "Find resources". Below the search bar, there is a table with columns "Display name" and "Email". Two rows are visible:

<input type="checkbox"/>	Display name	Email
<input type="checkbox"/>	Jane Doe	janedoe@example.com
<input checked="" type="checkbox"/>	John Doe	johndoe@example.com

Below the table, there is a section titled "Selected users (1)". At the bottom right of the page, there are two buttons: "Cancel" and "Assign administrators".

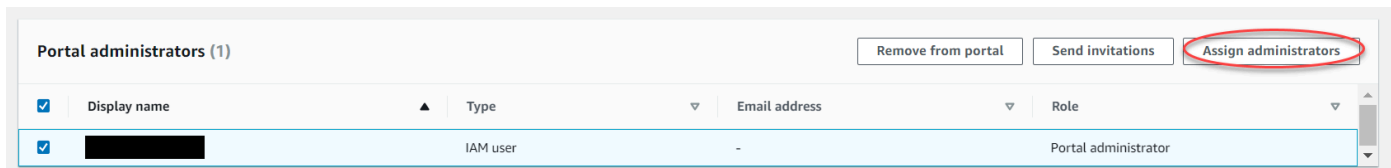
### 删除门户管理员

- 在门户详细信息页面的门户管理员部分，选中要删除的各个用户的复选框，然后选择从门户中删除。

**Note**

我们建议您至少选择一名门户管理员。

## IAM



### 添加门户管理员

1. 在门户详细信息页面的门户管理员部分，选择分配管理员。
2. 在分配管理员页面上，执行以下操作：
  - 如果您想添加IAM 用户作为门户管理员，请选择 IAM 用户。
  - 如果您想以门户管理员身份添加 IAM 角色，请选择 IAM 角色。
3. 选中您希望作为门户管理员的用户或角色对应的复选框。这会将用户或角色添加到门户管理员列表中。
4. 选择分配管理员。

**Important**

用户或角色必须具有登录门户的 `iotsitewise:DescribePortal` 权限。

AWS IoT SiteWise > Monitor > Portals > example-factory-2 > Assign administrators

## Assign administrators

Choose the users that you want to be portal administrators. Portal administrators can grant users access to specific industrial equipment data. [Learn more](#)

ⓘ IAM users or roles must have the `iotsitewise:DescribePortal` permission to sign in to the portal.

**Users** Roles

**IAM users (1)** [Manage users in IAM console](#)

Find user name

<input checked="" type="checkbox"/>	Name	Date created
<input checked="" type="checkbox"/>	raspberrypi-testing	11-08-2019

▶ **Portal administrators (1)** [Remove](#)

Cancel [Assign administrators](#)

AWS IoT SiteWise > Monitor > Portals > example-factory-2 > Assign administrators

## Assign administrators

Choose the users that you want to be portal administrators. Portal administrators can grant users access to specific industrial equipment data. [Learn more](#)

ⓘ IAM users or roles must have the `iotsitewise:DescribePortal` permission to sign in to the portal.

Users **Roles**

**IAM roles (66)** [Manage roles in IAM console](#)

Find role name

<input type="checkbox"/>	Name	Date created
<input type="checkbox"/>	[REDACTED]	
<input checked="" type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_4wZigNpA1	03-16-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_ECKT-2Oar	03-11-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_GTnd0O4Wr	03-16-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_rHINLNC5-	03-11-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_XB330QUIO	03-10-2021
<input type="checkbox"/>	[REDACTED]	
<input type="checkbox"/>	[REDACTED]	
<input type="checkbox"/>	[REDACTED]	

▶ **Portal administrators (2)** [Remove](#)

Cancel [Assign administrators](#)

## 删除门户管理员

- 在门户详细信息页面的门户管理员部分，选中要删除的各个用户的复选框，然后选择从门户中删除。

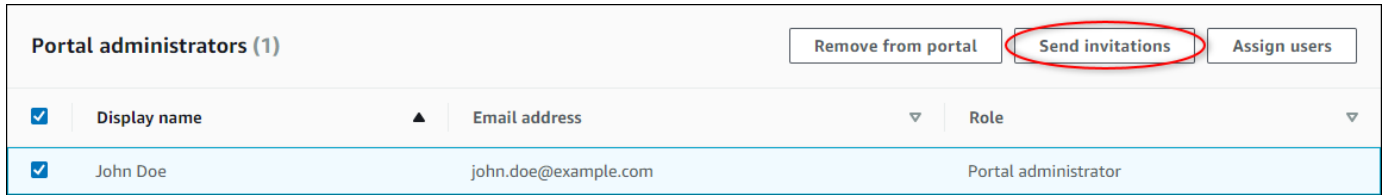
### Note

不建议不为门户添加管理员。

## 向门户管理员发送电子邮件邀请

向门户管理员发送电子邮件邀请。

1. 在门户详细信息页面的 门户管理员 部分，选中门户管理员对应的复选框。



2. 选择 发送邀请。您的电子邮件客户端将打开，邮件正文中将填充邀请内容。

在将电子邮件发送给门户管理员之前您可以对其进行自定义。

## 添加或删除门户用户

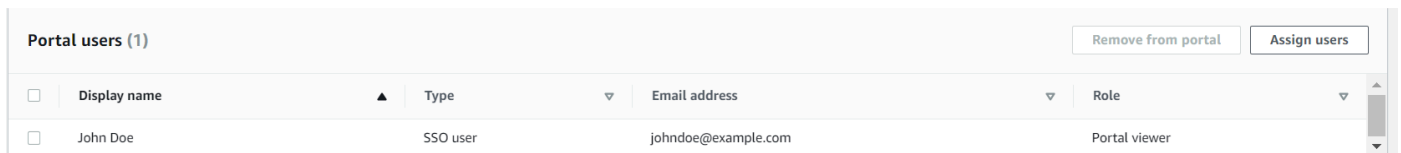
您选择哪些用户可以访问您的门户。门户用户显示在 Monito SiteWise r 门户中的用户列表中。从此列表，门户管理员可以添加项目拥有者，而项目拥有者可以添加项目查看者。

### Note

如果您的门户管理员和门户用户需要添加或删除用户，可能会通过门户的支持电子邮件与您联系。

根据用户身份验证服务，请选择以下选项之一。

### IAM Identity Center



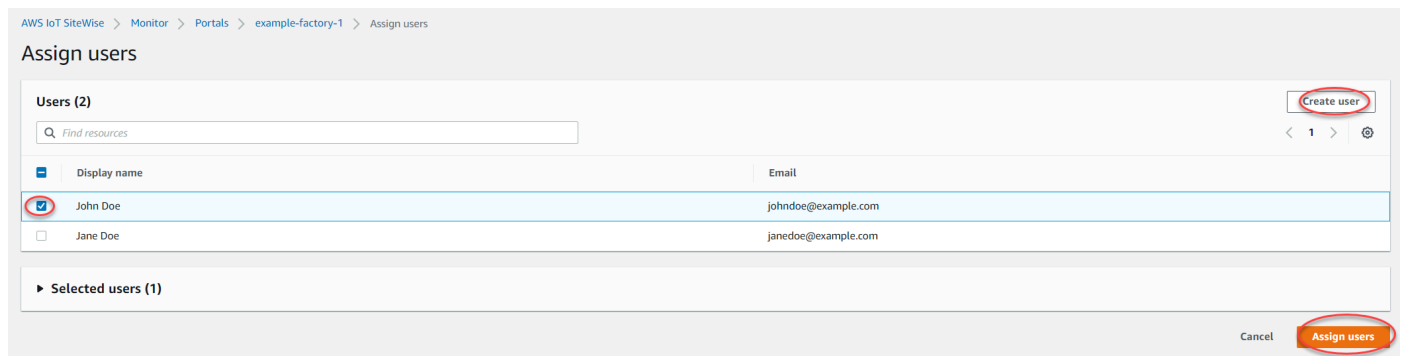
### 添加门户用户

1. 在门户详细信息页面的 门户用户 部分，选择 分配用户。
2. 在分配用户页面上，选中要添加到门户的用户的复选框。

### Note

如果您使用 IAM Identity Center 作为身份存储，并且已登录 AWS Organizations 管理账户，则可以选择创建用户来创建 IAM Identity Center 用户。IAM Identity Center 会向新用户发送一封电子邮件，让他们设置密码。然后，您可以将用户作为用户分配给门户。有关更多信息，请参阅[管理 IAM Identity Center 中的身份](#)。

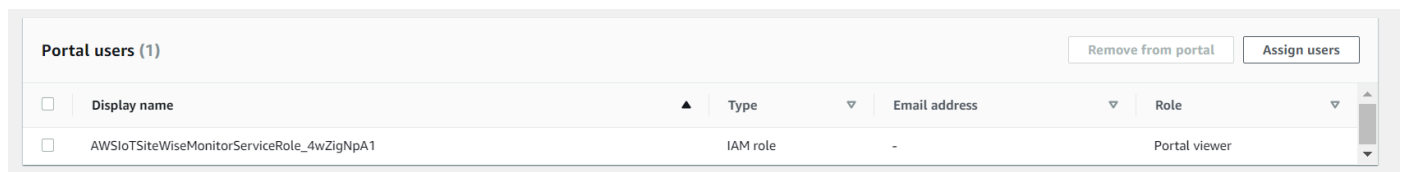
### 3. 选择 分配用户。



### 删除门户用户

- 在门户详细信息页面的门户用户部分，选中要从门户中删除的用户对应的复选框，然后选择从门户中删除。

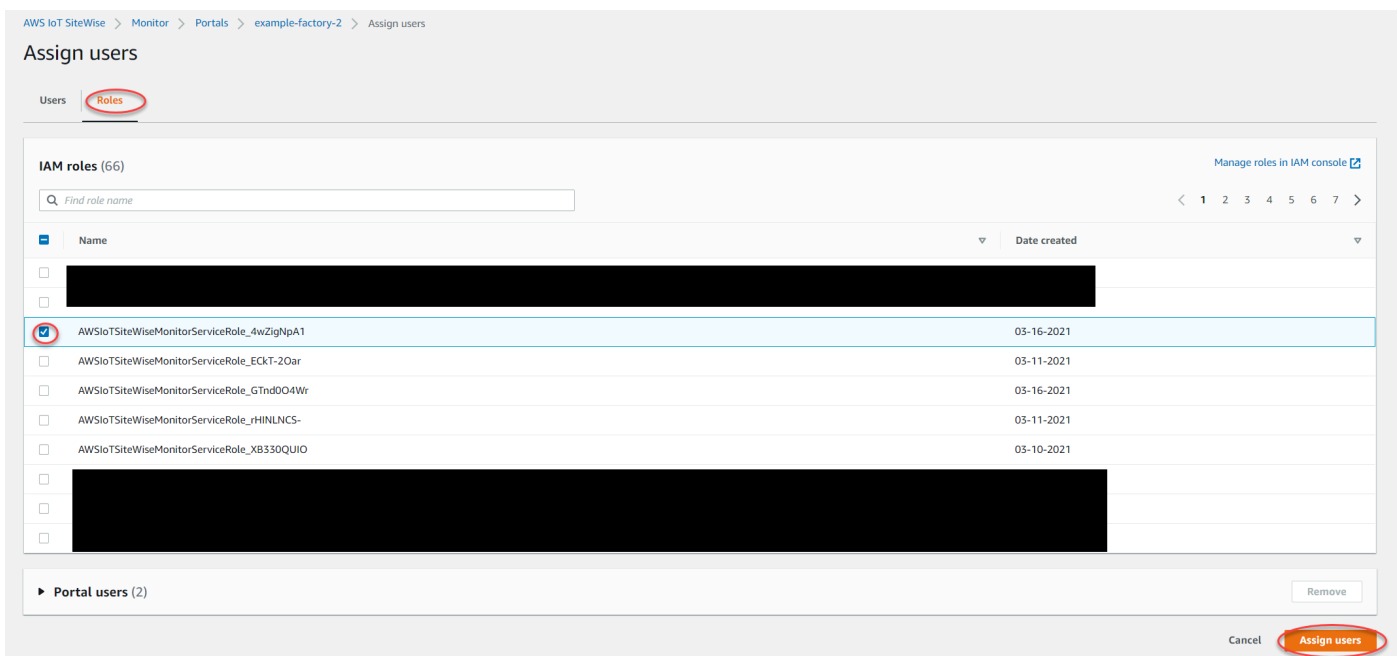
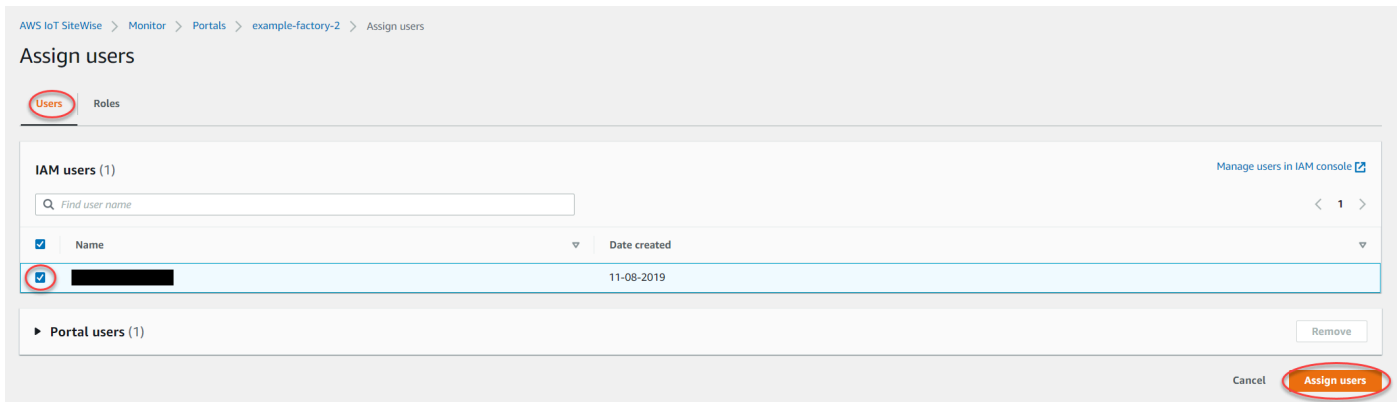
## IAM



### 添加门户用户

- 在门户详细信息页面的门户用户部分，选择分配用户。
- 在分配用户页面上，执行以下操作：
  - 选择 IAM 用户以将 IAM 用户添加为您的门户用户。
  - 选择 IAM 角色以将 IAM 角色添加为您的门户用户。

- 选中您希望作为门户用户添加的用户或角色对应的复选框。这会将用户或角色添加到门户用户列表中。
- 选择 分配用户。



## 删除门户用户

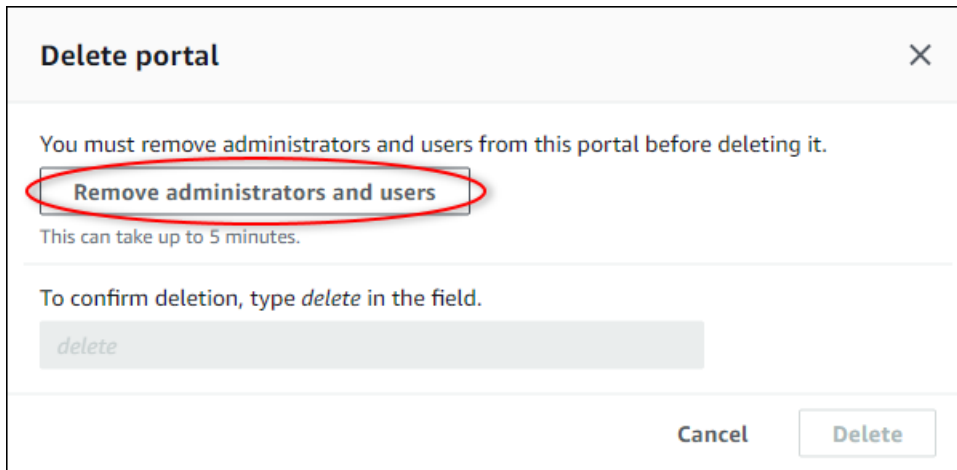
- 在门户详细信息页面的门户用户部分，选中要从门户中删除的用户对应的复选框，然后选择从门户中删除。

### Important

用户或角色必须具有登录门户的 `iotsitewise:DescribePortal` 权限。







**Delete portal** ×

You must remove administrators and users from this portal before deleting it.

**Remove administrators and users**

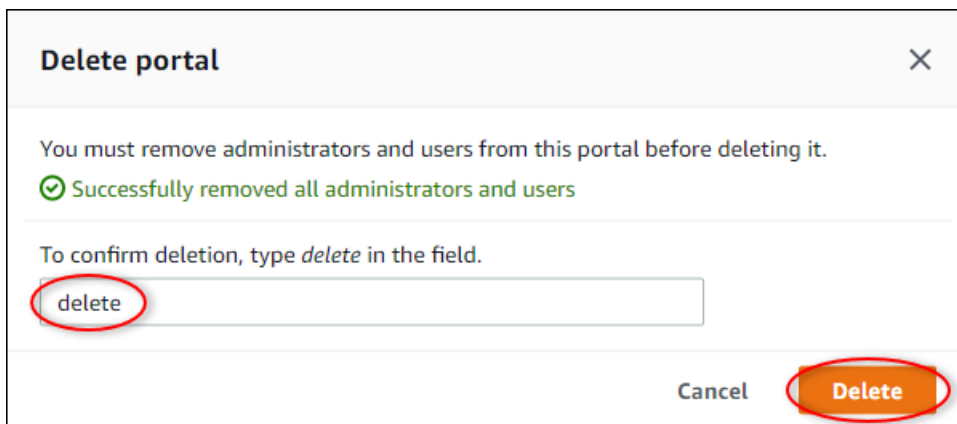
This can take up to 5 minutes.

To confirm deletion, type *delete* in the field.

*delete*

Cancel Delete

3. 如果您确定要删除整个门户，请在字段中输入 **delete** 以确认删除。



**Delete portal** ×

You must remove administrators and users from this portal before deleting it.

✔ Successfully removed all administrators and users

To confirm deletion, type *delete* in the field.

*delete*

Cancel Delete

4. 选择 删除。

## 使用 IoT 控制面板应用程序监控数据

物联网仪表板应用程序是一个开源仪表板应用程序，您可以在其中可视化操作数据并与之交互。您可以利用AWS Cloud Development Kit (AWS CDK)来部署 IoT 控制面板应用程序。

以下是 IoT 控制面板应用程序中可自定义的数据可视化功能的示例：

- Support 支持在单个折线图中使用多个属性。
- 增强了对资产和财产搜索。

来自制造、物流、能源和其他行业的客户可以使用物联网仪表板应用程序来应对特定挑战，例如跟踪设备性能、优化运营效率和数据驱动的决策。有关更多信息，请参阅 [IoT 控制面板应用程序GitHub 存储库](#)。

# 从中查询数据 AWS IoT SiteWise

您可以使用 AWS IoT SiteWise API 操作来查询资产属性的当前值、历史值以及特定时间间隔内的聚合。

使用这些功能深入了解您的数据。例如，发现具有给定属性值的所有资产，或者创建数据的自定义表示形式。您还可以使用 API 操作来开发与 AWS IoT SiteWise 资产中存储的工业数据集成的软件解决方案。您还可以在 AWS IoT SiteWise Monitor 中实时浏览您的资产数据。要了解如何配置 SiteWise 监控器，请参阅[使用监控数据 AWS IoT SiteWise Monitor](#)。

本节中描述的操作返回包含时间戳、质量、值 (TQV) 结构的属性值对象：

- timestamp 包含当前的 Unix 纪元时间（以秒为单位），偏移量为纳秒。
- quality 包含以下字符串之一，用于指示数据点的质量：
  - GOOD – 数据不受任何问题的影响。
  - BAD – 数据受传感器故障等问题的影响。
  - UNCERTAIN – 数据受传感器不准确等问题的影响。
- value 包含下列字段之一，具体取决于属性的类型：
  - booleanValue
  - doubleValue
  - integerValue
  - stringValue

## 主题

- [查询当前资产的属性值](#)
- [查询历史资产属性值](#)
- [查询资产属性聚合](#)
- [AWS IoT SiteWise 查询语言](#)

## 查询当前资产的属性值

本教程展示了两种获取资产属性的当前值的方法。您可以使用 AWS IoT SiteWise 控制台或在 AWS Command Line Interface (AWS CLI) 中使用 API。

## 主题

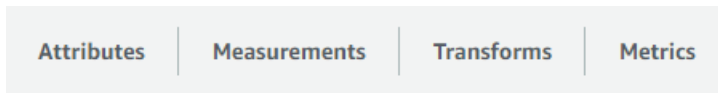
- [查询资产属性的当前值 \(控制台\)](#)
- [查询资产属性的当前价值 \(AWS CLI\)](#)

## 查询资产属性的当前值 (控制台)

您可以使用 AWS IoT SiteWise 控制台查看资产属性的当前值。

### 获取资产属性的当前值 (控制台)

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 资产。
3. 选择具有要查询的属性的资产。
4. 选择箭头图标展开资产层次结构以查找您的资产。
5. 选择属性类型对应的选项卡。例如，选择 测量值 可以查看测量值属性的当前值。



6. 找到要查看的属性。当前值将显示在 最新值 列中。

## 查询资产属性的当前价值 (AWS CLI)

您可以使用 AWS Command Line Interface (AWS CLI) 来查询资产属性的当前值。

使用 [GetAssetPropertyValue](#) 操作查询资产属性的当前值。

要标识资产属性，请指定以下选项之一：

- 数据发送到 `propertyId` 的资产属性的 `assetId` 和。
- `propertyAlias`，这是一个数据流别名（例如，`/company/windfarm/3/turbine/7/temperature`）。要使用此选项，您必须首先设置资产属性的别名。要设置属性别名，请参见 [将工业数据流映射到资产属性](#)。

### 获取资产属性的当前价值 (AWS CLI)

- 运行以下命令以获取资产属性的当前值。将 `asset-id` 替换为资产的 ID，将 `property-id` 替换为属性的 ID。

```
aws iotsitewise get-asset-property-value \  
  --asset-id asset-id \  
  --property-id property-id
```

此操作将按以下格式返回包含属性当前的 TQV 的响应。

```
{  
  "propertyValue": {  
    "value": {  
      "booleanValue": Boolean,  
      "doubleValue": Number,  
      "integerValue": Number,  
      "stringValue": "String"  
    },  
    "timestamp": {  
      "timeInSeconds": Number,  
      "offsetInNanos": Number  
    },  
    "quality": "String"  
  }  
}
```

## 查询历史资产属性值

您可以使用 AWS IoT SiteWise API [GetAssetPropertyValueHistory](#) 操作来查询资产属性的历史值。

要标识资产属性，请指定以下选项之一：

- 数据发送到 `propertyId` 的资产属性的 `assetId` 和。
- `propertyAlias`，这是一个数据流别名（例如，`/company/windfarm/3/turbine/7/temperature`）。要使用此选项，您必须首先设置资产属性的别名。要设置属性别名，请参见 [将工业数据流映射到资产属性](#)。

传递以下参数以优化结果：

- `startDate`— 查询历史数据范围的唯一起始点，以 Unix 纪元时间（以秒为单位）表示。
- `endDate` – 查询历史数据范围的结束端，以 Unix 纪元时间（以秒为单位）表示。
- `maxResults` – 一个请求中要返回的最大结果数。默认为 20 结果。

- `nextToken` – 从此操作的上一次调用返回的分页标记。
- `timeOrdering` – 应用于返回值的顺序：ASCENDING 或 DESCENDING。
- `qualities` – 筛选结果的质量分为：GOOD、BAD 或 UNCERTAIN。

## 主题

- [查询资产属性的价值历史记录 \(AWS CLI\)](#)

## 查询资产属性的价值历史记录 (AWS CLI)

### 要查询资产属性的价值历史记录 (AWS CLI)

1. 运行以下命令以获取资产属性的值历史记录。此命令将查询特定 10 分钟间隔内的属性历史记录。将 `asset-id` 替换为资产的 ID，将 `property-id` 替换为属性的 ID。将日期参数替换为要查询的时间间隔。

```
aws iotsitewise get-asset-property-value-history \  
  --asset-id asset-id \  
  --property-id property-id \  
  --start-date 1575216000 \  
  --end-date 1575216600
```

该操作返回的响应包含该属性的历史 TQV，格式如下：

```
{  
  "assetPropertyValueHistory": [  
    {  
      "value": {  
        "booleanValue": Boolean,  
        "doubleValue": Number,  
        "integerValue": Number,  
        "stringValue": "String"  
      },  
      "timestamp": {  
        "timeInSeconds": Number,  
        "offsetInNanos": Number  
      },  
      "quality": "String"  
    }  
  ],  
}
```

```
"nextToken": "String"  
}
```

2. 如果存在更多值条目，则可以将分页令牌从nextToken字段传递给后续的[GetAssetPropertyValueHistory](#)操作调用。

## 查询资产属性聚合

AWS IoT SiteWise 自动计算聚合的资产属性值，这些值是在多个时间间隔内计算出来的一组基本指标。AWS IoT SiteWise 每分钟、每小时和每天计算您的资产属性的以下聚合：

- 平均值 – 属性在一个时间间隔内的平均值。
- 计数 – 属性在一个时间间隔内的数据点数。
- 最大值 – 属性在一个时间间隔内的最大值。
- 最小值 – 属性在一个时间间隔内的最小值。
- 标准偏差 – 属性值在一个时间间隔内的标准偏差。
- 总和 – 属性值在一个时间间隔内的总和。

对于非数字属性，例如字符串和布尔值，仅 AWS IoT SiteWise 计算计数聚合。

您还可以计算资产数据的自定义指标。使用指标属性，您可以定义特定于您的操作的聚合。指标属性提供了额外的聚合函数和时间间隔，这些函数和时间间隔不是为 AWS IoT SiteWise API 预先计算的。有关更多信息，请参阅 [聚合来自属性和其他资产（指标）的数据](#)。

### 主题

- [资产属性的聚合 \(API\)](#)
- [资产属性的聚合 \(\)AWS CLI](#)

## 资产属性的聚合 (API)

您可以使用 AWS IoT SiteWise API 获取资产属性的聚合。

使用[GetAssetPropertyAggregates](#)操作来查询资产属性的聚合。

要识别资产属性，请指定以下选项之一：

- 数据发送到propertyId的资产属性的assetId和。



- `propertyAlias`，这是一个数据流别名（例如，`/company/windfarm/3/turbine/7/temperature`）。要使用此选项，您必须首先设置资产属性的别名。要设置属性别名，请参见[将工业数据流映射到资产属性](#)。

您还必须传递以下必需参数：

- `aggregateTypes` – 要检索的聚合的列表。您可以指定 `AVERAGE`、`COUNT`、`MAXIMUM`、`MINIMUM`、`STANDARD_DEVIATION` 和 `SUM` 中的任何一个。
- `resolution` – 检索指标的时间间隔：`1m`（1分钟）、`1h`（1小时）或 `1d`（1天）。
- `startDate` – 查询历史数据范围的唯一起始点，以 Unix 纪元时间（以秒为单位）表示。
- `endDate` – 查询历史数据范围的结束端，以 Unix 纪元时间（以秒为单位）表示。

还可以传递以下任何参数来优化结果：

- `maxResults` – 一个请求中要返回的最大结果数。默认为 20 结果。
- `nextToken` – 从此操作的上一次调用返回的分页标记。
- `timeOrdering` – 应用于返回值的顺序：`ASCENDING` 或 `DESCENDING`。
- `qualities` – 筛选结果的质量分为：`GOOD`、`BAD` 或 `UNCERTAIN`。

### Note

该 [GetAssetPropertyAggregates](#) 操作返回的 TQV 格式与本节中描述的其他操作不同。`value` 结构包含请求中每个 `aggregateTypes` 的字段。`timestamp` 包含聚合发生的时间，以 Unix 纪元时间表示（以秒为单位）。

## 资产属性的聚合 ()AWS CLI

### 查询资产属性的聚合 ()AWS CLI

1. 运行以下命令以获取资产属性的聚合。此命令会查询特定 1 小时间隔内的平均值和总和（采用 1 小时分辨率）。将 `asset-id` 替换为资产的 ID，将 `property-id` 替换为属性的 ID。将参数替换为要查询的聚合和时间间隔。

```
aws iotsitewise get-asset-property-aggregates \
```

```
--asset-id asset-id \  
--property-id property-id \  
--start-date 1575216000 \  
--end-date 1575219600 \  
--aggregate-types AVERAGE SUM \  
--resolution 1h
```

此操作将按以下格式返回包含属性的历史 TQV 的响应。响应中仅包括请求的聚合。

```
{  
  "aggregatedValues": [  
    {  
      "timestamp": Number,  
      "quality": "String",  
      "value": {  
        "average": Number,  
        "count": Number,  
        "maximum": Number,  
        "minimum": Number,  
        "standardDeviation": Number,  
        "sum": Number  
      }  
    }  
  ],  
  "nextToken": "String"  
}
```

2. 如果存在更多值条目，则可以将分页令牌从nextToken字段传递给后续的[GetAssetPropertyAggregates](#)操作调用。

## AWS IoT SiteWise 查询语言

通过 AWS IoT SiteWise 数据检索 [ExecuteQuery](#) API 操作，您可以从以下位置检索有关声明式结构定义以及与之关联的时间序列数据的信息：

- 模型
- 资产
- 测量
- 指标
- 变换

- 总量

这可以在单个 API 请求中使用类似查询语句的 SQL 来完成。

#### Note

除了 AWS GovCloud ( 美国西部 ) 之外，所有同时提供 AWS IoT SiteWise 和的地区 AWS IoT TwinMaker 均提供此功能。

## 主题

- [先决条件](#)
- [查询语言参考](#)

## 先决条件

AWS IoT SiteWise 需要权限才能与之集成，AWS IoT TwinMaker 这样它才能组织和建模工业数据。

在检索有关模型、资产、测量、指标、变换和聚合的信息之前，请确保满足以下先决条件：

- 两者都有服务相关角色 AWS IoT SiteWise，并在您的 AWS 账户中 AWS IoT TwinMaker 设置。有关服务相关角色的更多信息，请参见 IAM 用户指南中的[使用服务相关角色](#)。
- 为您的 IAM 角色启用的 AWS IoT SiteWise 集成。有关更多信息，请参见[集成 AWS IoT SiteWise 和 AWS IoT TwinMaker](#)。
- IoTSiteWiseDefaultWorkspace 在您所在地区的账户中带有 ID AWS IoT TwinMaker 的工作空间。有关更多信息，请参见[IoTSiteWiseDefaultWorkspace 用户指南](#)中的使用 AWS IoT TwinMaker。
- AWS IoT TwinMaker 启用的标准或分层捆绑定价模式。有关更多信息，请参见《AWS IoT TwinMaker 用户指南》中的[切换 AWS IoT TwinMaker 定价模式](#)。

## 查询语言参考

AWS IoT SiteWise 支持丰富的查询语言来处理您的数据。以下主题描述了可用的数据类型、运算符、函数和结构。

[示例查询](#)要使用查询语言编写 AWS IoT SiteWise 查询，请参阅。

## 主题

- [了解视图](#)
- [支持的数据类型](#)
- [使用 SELECT 语句检索数据](#)
- [逻辑运算符](#)
- [比较运算符](#)
- [示例查询](#)

## 了解视图

本节提供的信息可帮助您理解中的视图 AWS IoT SiteWise ，例如流程元数据和遥测数据。

下表提供了视图的视图名称和描述。

### 数据模型

视图名称	视图描述
asset	包含有关资产和模型派生的信息。
资产_财产	包含有关资产属性结构的信息。
raw_time_series	包含时间序列的历史数据。
latest_value_time_serie	包含时间序列的最新值。
预先计算的聚合	包含自动计算的聚合资产属性值。它们是一组按多个时间间隔计算的基本指标。

以下视图列出了查询的列名以及示例数据。

### 视图:资产

资产_id	资产名称	资产描述	资产_模型_id
88898498-0b8b-42b5-bf57-16180bc3d3a0	WindTurbine A	WindTurbine 资产 A	17847250-5bf0-4f74-b775-cc03f05e7cb8

资产_id	资产名称	资产描述	资产_模型_id
17847250-5bf0-4f74-b775-cc03f05e7cb8	风力涡轮机资产模型	代表风力发电场中的涡轮机。	

查看：资产\_属性

属性_id	资产_id	属性_名称	属性_数据_类型	属性_别名	asset_composite_model_id
b29be434-b000-4d74-b809-75287d83bcd6	88898498-0b8b-42b5-bf57-16180bc3d3a0	电机温度	Double	Rochester2/44///Line-5/Bus-2/Machine-5/Temperature	
3b458f00-24e7-458a-b4e8-c6026eff654a	88898498-0b8b-42b5-bf57-16180bc3d3a0	风向	Double	/company/windfarm/3/turbine/7/winddirection	2f458n00-56e7-458h-b4e8-c6026eff985g

查看：raw\_time\_series

资产_id	属性_id	属性_别名	事件时间戳	质量	布尔值	int_value	双精度值	字符串值
88898498-0b8b-42b5-bf57-16180bc3d3a0	b29be434-b000-4d74-b809-75287d83bcd6	Rochester2/44///Line-5/Bus-2/Machine-5/	157521960	好			115.0	

资产_id	属性_id	属性_别名	事件时间戳	质量	布尔值	int_value	双精度值	字符串值
		Temperature						
88898498-0b8b-42b1-bf57-16180bc3d3a	3b458f00-24e7-458c-b4e8-c6026eff654a	/ company, windfarm / 3/ turbine / 7/ winddirection	157521937	好			348.75	

### Note

您必须在event\_timestamp列中包含筛选子句才能查询raw\_time\_series视图。这是必需的过滤器，如果没有它，查询将失败。

#### Example 查询

```
SELECT event_timestamp, double_value FROM raw_time_series WHERE event_timestamp > 1234567890
```

#### 查看 : latest\_Value\_time\_series

资产_id	属性_id	属性_别名	事件时间戳	质量	布尔值	int_value	双精度值	字符串值
88898498-0b8b-42b1-bf57-16180bc3d3a	3b458f00-24e7-458c-b4e8-c6026eff654a	/ company, windfarm / 3/ turbine / 7/	157521960	好			355.39	

资产_id	属性_id	属性_别名	事件时间戳	质量	布尔值	int_value	双精度值	字符串值
		winddirection						

## 视图：预计算\_聚合

资产_id	属性_id	属性_别名	事件时间戳	resolution	总和值	计数值	平均值	最大值	最小值	stdev_value
8889840b8b-42b000-4c2bf57-1680bc3d	b29be4b000-4c2bf57-1680bc3d	Rochester-5/Bus-Machine-5/Temperature	1575210	15m	1105.48	15	73.4	80.6	68	3.64

## 支持的数据类型

AWS IoT SiteWise 查询语言支持以下数据类型。

视图:资产

数据类型	描述
STRING	最大长度为 1024 字节的字符串。
INTEGER	一个有符号的 32 位整数，范围为 $-2,147,483,648$ to $2,147,483,647$
DOUBLE	一个浮点数，其范围为 $-10^{100}$ to $10^{100}$ 和 IEEE 754 双精度。

数据类型	描述
BOOLEAN	true 或 false。

### Note

双精度数据不准确。由于精度有限，有些值无法精确转换，也无法代表所有实数。查询中的浮点数据可能与内部表示的值不同。如果输入数字的精度过高，则该值将被四舍五入。

## 使用 SELECT 语句检索数据

该SELECT语句用于从一个或多个视图中检索数据。AWS IoT SiteWise 支持隐式JOIN的观点。您可以列出要加入的视图（在SELECT语FROM句的子句中），使用逗号分隔它们。

### Example

使用以下SELECT语句：

```
SELECT select_expr [, ...]  
[ FROM from_item [, ...] ]  
[ WHERE [LIKE condition ESCAPE condition] ]
```

在前面的示例中，LIKE子句使用通配符指定搜索和筛选条件。AWS IoT SiteWise 支持percentage (%)作为通配符。

Example 要%在某种条件下使用：

```
Prefix search: String%  
Infix search: %String%  
Suffix search: %String
```

Example 要搜索资产，请执行以下操作：

```
SELECT asset_name, asset_description FROM asset WHERE asset_name LIKE 'Wind%'
```



**Example** 要使用 ESCAPE 条件搜索资产，请执行以下操作：

```
SELECT asset_name, asset_description FROM asset WHERE asset_name LIKE 'room\%' ESCAPE
'\'
```

## 逻辑运算符

AWS IoT SiteWise 支持以下逻辑运算符。

### 逻辑运算符

操作员	描述	示例
AND	TRUE如果两个值都为真	a AND b

如果 a 或 b 为 FALSE，则前一个表达式的计算结果为 false。要使 AND 运算符的计算结果为真，a 和 b 都必须为真。

### Example

```
SELECT a.asset_name
FROM asset as a, latest_value_time_series as t
WHERE t.int_value > 30 AND t.event_timestamp > 1234567890
```

## 比较运算符

AWS IoT SiteWise 支持以下比较运算符。

### 逻辑运算符

操作员	描述
<	Less than
>	Greater than
<=	小于或等于
>=	大于或等于

操作员	描述
=	等于
!=	Not equal

## 示例查询

### 元数据筛选

以下示例是使用带有 AWS IoT SiteWise 查询语言的语SELECT句进行元数据筛选：

```
SELECT a.asset_name, p.property_name
FROM asset a, asset_property p
WHERE a.asset_id = p.asset_id AND a.asset_name LIKE '%windmill%'
```

### 值筛选

以下是使用带有 AWS IoT SiteWise 查询语言的语SELECT句进行值筛选的示例：

```
SELECT a.asset_name FROM asset a, raw_time_series r
WHERE a.asset_id = r.asset_id AND r.int_value > 30 AND r.event_timestamp > 1234567890
AND r.event_timestamp < 1234567891
```

## 与其他 AWS 服务互动

AWS IoT SiteWise 可以将资产数据发布到 AWS IoT MQTT 发布-订阅消息代理，以便您可以与其他服务中的资产数据进行交互。AWS IoT SiteWise 为每个资产属性分配一个唯一的 MQTT 主题，您可以使用该主题使用 AWS IoT 核心规则将资产数据路由到其他 AWS 服务。例如，您可以将 AWS IoT Core 规则配置为执行以下任务：

- 识别设备故障，并通过向 [AWS IoT Events](#) 发送数据通知适当的人员。
- 在历史记录中选择要在外部软件解决方案中使用的资产数据，然后将其发送到 [Amazon DynamoDB](#)。
- 通过触发 [AWS Lambda](#) 函数生成每周报告。

您可以按照教程进行操作，该教程逐步执行设置在 DynamoDB 中存储属性值的规则所需的步骤。有关更多信息，请参阅 [将属性值更新发布到 Amazon DynamoDB](#)。

有关如何配置规则的更多信息，请参阅 AWS IoT 开发人员指南中的[规则](#)。

您也可以将来自其他 AWS 服务的数据重新使用到 AWS IoT SiteWise。要通过 AWS IoT SiteWise 规则操作摄取数据，请参阅[使用规则摄取数据 AWS IoT Core](#)。

### 主题

- [了解资产属性的 MQTT 主题](#)
- [处理资产属性通知](#)
- [使用资产属性通知将数据导出到 Amazon S3](#)
- [与 Grafana 集成](#)
- [集成 AWS IoT SiteWise 和 AWS IoT TwinMaker](#)
- [使用 Amazon Lookout for Equipment 检测设备异常](#)

## 了解资产属性的 MQTT 主题

每个资产属性都具有以下格式的唯一 MQTT 主题路径。

```
$aws/sitewise/asset-models/assetModelId/assets/assetId/properties/propertyId
```

**Note**

AWS IoT SiteWise 不支持 C AWS IoT Core 规则引擎中的 # ( 多级 ) 主题筛选通配符。您可以使用 + ( 单级 ) 通配符。例如，您可以使用以下主题筛选器来匹配特定资产模型的所有更新。

```
$aws/sitewise/asset-models/assetModelId/assets/+/properties/+
```

要了解有关主题筛选通配符的更多信息，请参阅 AWS IoT Core 开发人员指南中的 [主题](#)。

## 处理资产属性通知

您可以启用属性通知以向其发布资产数据更新 AWS IoT Core，然后对您的数据进行查询。通过资产属性通知，AWS IoT SiteWise 提供了一个可用于将 AWS IoT SiteWise 数据导出到 Amazon S3 的 AWS CloudFormation 模板。

**Note**

资产数据在 AWS IoT Core 每次收到时都会发送到 AWS IoT SiteWise，无论其值是否发生变化。

### 主题

- [启用资产属性通知 \( 控制台 \)](#)
- [启用资产属性通知 \(AWS CLI\)](#)
- [查询资产属性通知消息](#)

## 启用资产属性通知 ( 控制台 )

默认情况下，AWS IoT SiteWise 不发布属性值更新。您可以使用 AWS IoT SiteWise 控制台为资产属性启用通知。

### 启用或禁用资产属性的通知 ( 控制台 )

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 资产。
3. 选择资产以启用属性的通知。

**i** Tip

您可以选择箭头图标来展开资产层次结构以查找资产。

4. 选择编辑。
5. 对于资产属性的通知状态，选择 已启用。

"Wind Speed"

Enter a property alias

Must be less than 2048 characters.

Notification status

ENABLED

Notification will be published to topic \$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE

您还可以选择 已禁用 以禁用资产属性的通知。

6. 选择 保存。

## 启用资产属性通知 (AWS CLI)

默认情况下，AWS IoT SiteWise 不发布属性值更新。您可以使用 AWS Command Line Interface (AWS CLI) 来启用或禁用资产属性的通知。

您必须知道资产的 `assetId` 和属性的 `propertyId` 才能完成此过程。您也可以使用外部 ID。如果您创建了资产但不知道该资产 `assetId`，请使用 [ListAssets](#) API 列出特定模型的所有资产。使用该 [DescribeAsset](#) 操作查看资产的属性，包括财产 ID。

使用 [UpdateAssetProperty](#) 操作来启用或禁用资产属性的通知。指定以下参数：

- `assetId` – 资产的 ID。
- `propertyId` – 资产属性的 ID。
- `propertyNotificationState` – 属性值通知状态：ENABLED 或 DISABLED。
- `propertyAlias` – 属性的别名。在更新通知状态时指定属性的现有别名。如果省略此参数，则将删除属性的现有别名。

### 启用或禁用资产属性的通知 (CLI)

1. 运行以下命令以检索资产属性的别名。将 `asset-id` 替换为资产的 ID，将 `property-id` 替换为属性的 ID。

```
aws iotsitewise describe-asset-property \  
  --asset-id asset-id \  
  --property-id property-id
```

此操作将按以下格式返回包含资产属性的详细信息的响应。属性别名位于 JSON 对象中的 `assetProperty.alias` 中。

```
{  
  "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",  
  "assetName": "Wind Turbine 7",  
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
  "assetProperty": {  
    "id": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",  
    "name": "Wind Speed",  
    "alias": "/company/windfarm/3/turbine/7/windspeed",  
    "notification": {  
      "topic": "$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/  
assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-  
cdef-33333EXAMPLE",  
      "state": "DISABLED"  
    },  
    "dataType": "DOUBLE",  
    "unit": "m/s",  
    "type": {  
      "measurement": {}  
    }  
  }  
}
```

2. 运行以下命令以启用资产属性的通知。使用上一命令响应中的属性别名替换 *property-alias*，或者省略 `--property-alias` 以更新不带别名的属性。

```
aws iotsitewise update-asset-property \  
  --asset-id asset-id \  
  --property-id property-id \  
  --property-notification-state ENABLED \  
  --property-alias property-alias
```

您还可以传递 `--property-notification-state DISABLED` 以禁用资产属性的通知。

## 查询资产属性通知消息

要查询资产属性通知，请创建由 SQL 语句组成的 AWS IoT Core 规则。

AWS IoT SiteWise 按以下格式向 AWS IoT Core 发布资产属性数据更新。

```
{
  "type": "PropertyValueUpdate",
  "payload": {
    "assetId": "String",
    "propertyId": "String",
    "values": [
      {
        "timestamp": {
          "timeInSeconds": Number,
          "offsetInNanos": Number
        },
        "quality": "String",
        "value": {
          "booleanValue": Boolean,
          "doubleValue": Number,
          "integerValue": Number,
          "stringValue": "String"
        }
      }
    ]
  }
}
```

values 列表中的每个结构都是一个 timestamp-quality-value (TQV) 结构。

- timestamp 包含当前的 Unix 纪元时间（以秒为单位），偏移量为纳秒。
- quality 包含以下字符串之一，用于指示数据点的质量：
  - GOOD – 数据不受任何问题的影响。
  - BAD – 数据受传感器故障等问题的影响。
  - UNCERTAIN – 数据受传感器不准确等问题的影响。
- value 包含下列字段之一，具体取决于属性的类型：
  - booleanValue
  - doubleValue

- integerValue
- stringValue

要解析 values 数组中的值，您需要在规则的 SQL 语句中使用复杂的嵌套对象查询。有关更多信息，请参阅 AWS IoT 开发人员指南中的[嵌套对象查询](#)，或参阅[将属性值更新发布到 Amazon DynamoDB 教程](#)，了解解析资产属性通知消息的具体示例。

#### Example 提取值数组的示例查询

以下语句演示如何针对具有特定 double 类型属性的所有资产查询该属性的已更新属性值的数组。

```
SELECT
  (SELECT VALUE (value.doubleValue) FROM payload.values) AS windspeed
FROM
  '$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/+/
properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE'
WHERE
  type = 'PropertyValueUpdate'
```

上一个规则查询语句采用以下格式输出数据。

```
{
  "windspeed": [
    26.32020195042838,
    26.282584572975477,
    26.352566977372508,
    26.283084346171442,
    26.571883739599322,
    26.60684140743005,
    26.628738636715045,
    26.273486932802125,
    26.436379105473964,
    26.600590095377303
  ]
}
```

#### Example 提取单个值的示例查询

以下语句演示如何针对具有特定 double 类型属性的所有资产查询该属性的属性值数组中的第一个值。

```
SELECT
```



```
get((SELECT VALUE (value.doubleValue) FROM payload.values), 0) AS windspeed
FROM
'$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/+
properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE'
WHERE
type = 'PropertyValueUpdate'
```

上一个规则查询语句采用以下格式输出数据。

```
{
  "windspeed": 26.32020195042838
}
```

### Important

该规则查询语句忽略每个批次中的第一个值更新以外的更新。每个批次最多可以包含 10 个值。如果需要包含其余值，您必须设置更复杂的解决方案，以将资产属性值输出到其他服务。例如，您可以设置一条规则，其 AWS Lambda 操作是将数组中的每个值重新发布到另一个主题，然后设置另一个规则来查询该主题并将每个值发布到所需的规则操作。

## 使用资产属性通知将数据导出到 Amazon S3

您可以将传入的数据导出 AWS IoT SiteWise 到您账户中的 Amazon S3 存储桶。您可以按照一种格式备份数据，该格式可用于创建历史报告或使用复杂方法分析数据。

### Note

AWS IoT SiteWise 还支持冷层存储，允许您将数据保存在客户管理的 Amazon S3 存储桶中。有关支持的存储层的更多信息，请参阅 [管理数据存储](#)。

AWS IoT SiteWise 将此功能作为 AWS CloudFormation 模板提供。当您根据模板创建堆栈时，AWS CloudFormation 会创建将传入数据流式传输 AWS IoT SiteWise 到 S3 存储桶所需的 AWS 资源。

然后，S3 存储桶将接收从属性值更新消息发送的所有资产 AWS IoT SiteWise 属性数据。S3 存储桶还会接收您的资产元数据，其中包括资产和属性名称以及其他信息。

有关如何为导出到 Amazon S3 的资产属性启用属性值更新消息的更多信息，请参阅 [与其他 AWS 服务互动](#)。

此功能在 AmazonS3 中以 [ApacheParquet](#) 格式存储资产属性数据和资产元数据。Parquet 是列式数据格式，与 JSON 等行式格式相比，前者可节省空间并更快地启用查询。

#### Note

此功能在检索资产元数据时，最多支持大约 1500 个资产属性。此限制仅适用于资产元数据。在该功能导出资产属性数据时，此限制不应用到支持的资产数。

每个资源的名称都包含一个前缀，您可以在创建堆栈时自定义该前缀。资源包括以下内容：

- 一个 Amazon S3 存储桶
- AWS Lambda 函数
- 一条 AWS IoT Core 规则
- AWS Identity and Access Management 角色
- 亚马逊 Data Firehose 直播
- 一个 AWS Glue 数据库

有关完整列表，请参阅[从模板创建的资源](#)。

#### Important

您需要为该 AWS CloudFormation 模板创建和使用的资源付费。这些费用包括多项 AWS 服务的数据存储和数据传输。

#### 主题

- [创建堆 AWS CloudFormation 栈](#)
- [在 Amazon S3 中查看您的数据](#)
- [使用 Amazon Athena 分析导出的数据](#)
- [从模板创建的资源](#)

## 创建堆 AWS CloudFormation 栈

您必须在中创建堆栈才能 AWS CloudFormation 将您的资产数据导出到 Amazon S3。

## 要将数据导出到 Amazon S3

1. 打开 [AWS CloudFormation 模板](#) 并登录 AWS Management Console。
2. 在创建堆栈 页面，选择页面底部的 下一步。
3. 在指定堆栈详细信息页面上，输入此模板BucketName为接收资产数据而创建的 S3 存储桶。此存储桶名称必须全局唯一。有关更多信息，请参阅 Amazon Simple Storage Service 用户指南中的 [存储桶命名规则](#)。
4. ( 可选 ) 更改模板的任何其他参数：
  - GlobalResourcePrefix - 从此模板创建的全局资源 ( 如 IAM 角色 ) 名称的前缀。
  - LocalResourcePrefix - 在当前区域中从此模板创建的资源的名称前缀。

### Note

如果您多次创建此模板，则要更改存储桶名称和资源前缀参数，以避免资源名称冲突。

5. 选择 Next(下一步)。
6. 在 配置堆栈选项 页面上，请选择 下一步。
7. 在页面底部，选中显示我确认，AWS CloudFormation 可能会创建 IAM 资源的复选框。
8. 选择创建堆栈。

堆栈需要几分钟的时间来创建。如果堆栈创建失败，则您的账户可能没有足够的权限，或者您可能输入了已存在的存储桶名称。使用以下步骤删除堆栈并重试：

- a. 选择右上角的 删除。

删除堆栈需要几分钟时间。

### Note

AWS CloudFormation 不会删除 S3 存储桶或 CloudWatch 日志组。您可以在相应服务的控制台中删除这些资源。

- b. 如果无法删除堆栈，请再次选择删除。
- c. 如果无法再次删除堆栈，请按照 AWS CloudFormation 控制台中的步骤跳过删除失败的资源，然后重试。

- 成功创建 AWS CloudFormation 堆栈后，请按照以下步骤在 Amazon S3 中浏览您的资产属性数据。

### Important

创建堆栈后，您可以看到 AWS 账户中的新资源。如果您删除或修改这些资源，该功能可能会停止正常工作。我们建议，除非您希望停止向存储桶发送数据或要自定义此功能，否则不要修改这些资源。

## 在 Amazon S3 中查看您的数据

创建功能后，您可以在 Amazon S3 中查看资产属性数据和资产元数据。

### Note

资产元数据每 6 小时更新一次。您可能需要等待最多 6 小时才能看到资产元数据显示在 S3 存储桶中。

此功能将资产属性数据存储在下述列表中，每行都包含一个数据点：

- type - 属性通知的类型 (PropertyValueUpdate)。
- asset\_id - 接收数据点的资产的 ID。
- asset\_property\_id - 接收资产数据点的属性的 ID。
- time\_in\_seconds - 接收数据的时间，以 Unix 纪元时间（以秒为单位）表示。
- offset\_in\_nanos - 与 timeInSeconds 的纳秒偏移量。
- asset\_property\_quality - 数据点的质量：GOOD、UNCERTAIN 或 BAD。
- asset\_property\_value - 数据点的值。
- asset\_property\_data\_type - 资产属性的数据类型：boolean、double、integer 或 string。

此功能将资产元数据存储在下述列表中，每行都包含一个资产属性：

- asset\_id - 资产的 ID。
- asset\_name - 资产的名称。
- asset\_model\_id - 资产的模型的 ID。

- `asset_property_id` - 资产属性的 ID。
- `asset_property_name` - 资产属性的名称。
- `asset_property_data_type` - 资产属性的数据类型：BOOLEAN、DOUBLE、INTEGER 或 STRING。
- `asset_property_unit` - 资产属性的单位。
- `asset_property_alias` - 资产属性的别名。

在 Amazon S3 中查看您的 AWS IoT SiteWise 数据

1. 导航到 [Amazon S3 控制台](#)。
2. 从存储桶的列表中，选择具有您在创建模板时所选名称的存储桶。
3. 在存储桶中，选择以下文件夹之一：
  - `asset-property-updates`— 此文件夹包含从中导出的资产属性数据 AWS IoT SiteWise。
  - `asset-metadata`— 此文件夹包含从中导出的资产详细信息 AWS IoT SiteWise。
4. 选择要查看的对象。
5. 在对象页面上，执行以下操作：
  - a. 选择选择范围选项卡。

在此面板中，您可以预览 Parquet 文件中的记录。
  - b. 对于文件格式，选择 Parquet。
  - c. 选择显示文件预览以 JSON 格式显示文件的内容。

#### Note

如果新数据未显示在存储桶中，请检查您是否为资产属性启用了属性值更新通知。有关更多信息，请参阅 [与其他 AWS 服务互动](#)。

有关如何分析存储在 S3 存储桶中的资产数据的更多信息，请参阅[使用 Amazon Athena 分析导出的数据](#)。

## 使用 Amazon Athena 分析导出的数据

在 Amazon S3 中保存资产属性数据后，您可以使用多种 AWS 服务来生成报告或分析和查询您的数据：

- 使用 [Amazon Athena](#) 对您的数据运行 SQL 查询。
- 使用 [Amazon EMR](#) 执行大数据分析。
- 使用 [亚马逊 OpenSearch 服务](#) 搜索和分析您的数据。

您可以在 Amazon S3 中找到可以与您的数据交互的其他 AWS 服务，这些服务列在“分析”下 [AWS Management Console](#)。

#### Note

堆栈会创建一个 AWS Glue 数据库来格式化资产属性数据。您无法查询此数据库中的资产数据。按照本节中的步骤创建可以查询 AWS Glue 的数据库。

在本教程中，您将学习如何配置使用 Amazon Athena 的先决条件，以及如何使用 Athena 对导出的资产数据运行 SQL 查询。AWS IoT SiteWise 要使用 Athena 查询数据，必须先使用您的资产数据填充 AWS Glue Data Catalog 到。数据目录包含数据库和表格，Athena 可以访问数据目录中的数据。您可以创建一个 AWS Glue Crawler，使用导出的资产数据定期更新数据目录。

#### 主题

- [配置爬网程序以填充 AWS Glue Data Catalog](#)
- [使用 Athena 查询数据](#)

## 配置爬网程序以填充 AWS Glue Data Catalog

AWS Glue 抓取工具会抓取数据存储以填充中的表。AWS Glue Data Catalog 在此过程中，您将为包含导出资产数据的 S3 存储桶创建并运行 AWS Glue 爬虫。爬网程序会创建一个用于资产属性更新的表和一个用于资产元数据的表。然后，您可以使用 Athena 对这些表执行 SQL 查询。有关更多信息，请参阅 AWS Glue 开发人员指南中的 [填充 AWS Glue Data Catalog](#) 和 [定义爬网程序](#)。

#### 创建 AWS Glue 爬虫

1. 导航到 [AWS Glue 控制台](#)。
2. 在导航窗格中，选择 爬网程序。
3. 选择 添加爬网程序。
4. 在 添加爬网程序 页面上，执行以下操作：
  - a. 输入爬网程序的名称（例如，**IoTSiteWiseDataCrawler**），然后选择 下一步。

- b. 对于爬网程序源类型，选择 数据存储，然后选择 下一步。
- c. 在 添加数据存储 页面上，执行以下操作：
  - i. 对于 选择数据存储，请选择 S3。
  - ii. 在 包括路径 中，输入 **s3://DOC-EXAMPLE-BUCKET1** 以将资产数据存储桶添加为数据存储。将 DOC-EXAMPLE-BUCKET1 替换为您在创建堆栈时选择的存储桶名称。
  - iii. 选择下一步。

**Add a data store**

**Choose a data store**

S3

**Connection**

Select a connection

Optionally include a Network connection to use with this S3 target. Note that each crawler is limited to one Network connection so any future S3 targets will also use the same connection (or none, if left blank).

Add connection

**Crawl data in**

Specified path in my account

Specified path in another account

**Include path**

s3://AWSDOC-EXAMPLE-BUCKET1

All folders and files contained in the include path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

Exclude patterns (optional)

Back Next

- d. 在 添加另一个数据存储 页面上，选择 否，然后选择 下一步。
- e. 在选择 IAM 角色页面上，执行以下操作：
  - i. 要创建允许 AWS Glue 访问 S3 存储桶的新服务角色，请选择创建 IAM 角色。
  - ii. 输入角色名称的后缀，例如 **IoTSiteWiseDataCrawler**。
  - iii. 选择下一步。
- f. 对于 频率，选择 每小时，然后选择 下一步。爬网程序每次运行时都会使用新数据更新表，因此您可以选择适合您使用案例的任何频率。
- g. 在 配置爬网程序的输出 页面上，执行以下操作：
  - i. 选择“添加数据库”，为您的资产数据创建 AWS Glue 数据库。

- ii. 输入数据库的名称，例如 **iot\_sitewise\_asset\_database**。
  - iii. 选择创建。
  - iv. 选择下一步。
- h. 检查爬网程序详细信息，然后选择 完成。

The screenshot shows the configuration page for a crawler in AWS IoT SiteWise. It is organized into five main sections, each with a title and a list of configuration items:

- Crawler info**: Name: `IoTSiteWiseDataCrawler`, Tags: `-`
- Data stores**: Data store: `S3`, Include path: `s3://AWSDOC-EXAMPLE-BUCKET1`, Connection: (empty), Exclude patterns: (empty)
- IAM role**: IAM role: `arn:aws:iam::123456789012:role/service-role/AWSGlueServiceRole-IoTSiteWiseDataCrawler`
- Schedule**: Schedule: `At 00 minutes past the hour`
- Output**: Database: `iot_sitewise_asset_database`, Prefix added to tables (optional): (empty), Create a single schema for each S3 path: `false`, Configuration options: (expandable)

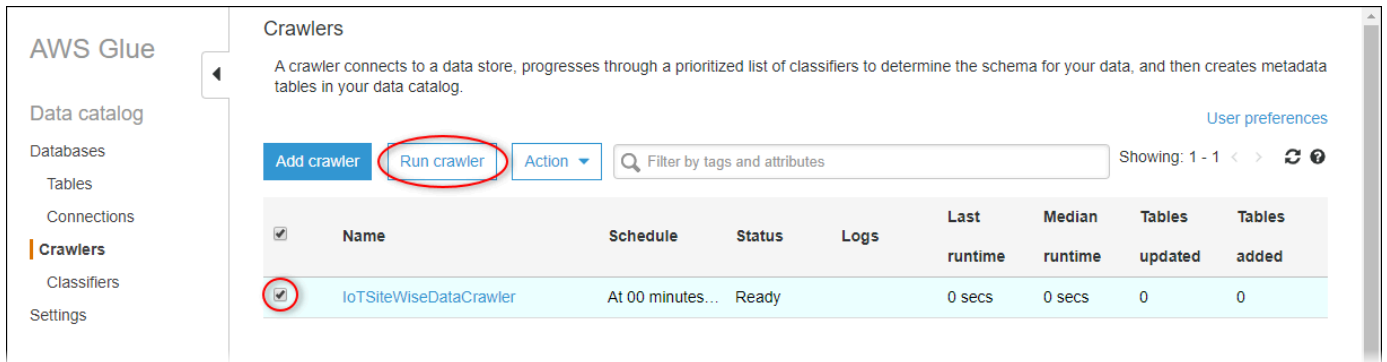
At the bottom of the page, there are two buttons: `Back` and `Finish`. The `Finish` button is highlighted with a red circle.

默认情况下，新爬网程序不会立即运行。您必须手动运行它，或等待它按照配置的时间表运行。

## 运行爬网程序

1. 在爬网程序页面上，选中新爬网程序的复选框，然后选择 运行爬网程序。





2. 等到爬网程序完成并且状态为 就绪。

运行爬网程序可能需要几分钟时间，并且其状态会自动更新。

3. 在导航窗格中，选择表。

您应该看到两个新表：asset\_metadata 和 asset\_property\_updates。

## 使用 Athena 查询数据

Athena 会在中自动发现您的资产数据表。AWS Glue Data Catalog 要对这些表的交集执行查询，您可以创建一个视图，即逻辑数据表。有关更多信息，请参阅 Amazon Athena 用户指南中的[使用视图](#)。

创建结合了资产属性数据和元数据的视图后，您可以运行查询，输出附加了资产和属性名称的属性值。有关更多信息，请参阅 Amazon Athena 用户指南中的[使用 Amazon Athena 运行 SQL 查询](#)。

### 使用 Athena 查询资产数据

1. 导航到 [Athena 控制台](#)。

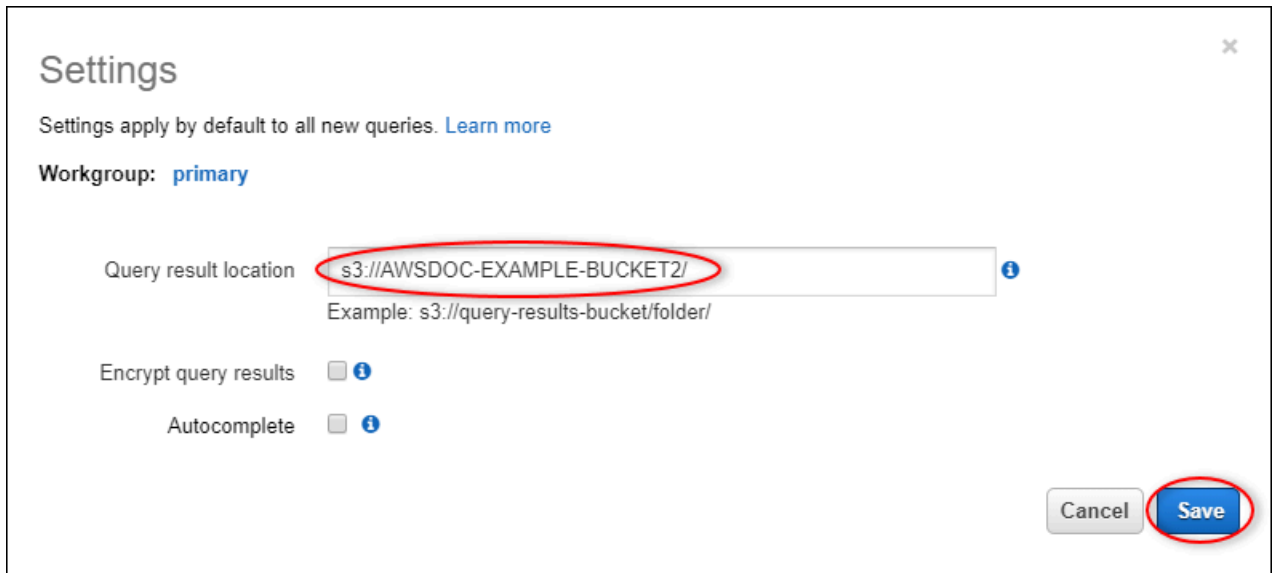
如果出现 入门 页面，请选择 开始”。

2. 如果您是第一次使用 Athena，请完成以下步骤来配置 S3 存储桶以获得查询结果。Athena 将您的查询结果保存在此存储桶中。

#### **⚠ Important**

使用与资产数据存储桶不同的存储桶，以便您之前创建的爬网程序不会抓取查询结果。我们建议您创建仅用于 Athena 查询结果的存储桶。有关更多信息，请参阅 Amazon Simple Storage Service 用户指南中的[如何创建 S3 存储桶？](#)

- a. 选择 设置。
- b. 在查询结果位置中，输入 Athena 查询结果的 S3 存储桶。存储桶必须以 / 结尾。



Settings

Settings apply by default to all new queries. [Learn more](#)

Workgroup: **primary**

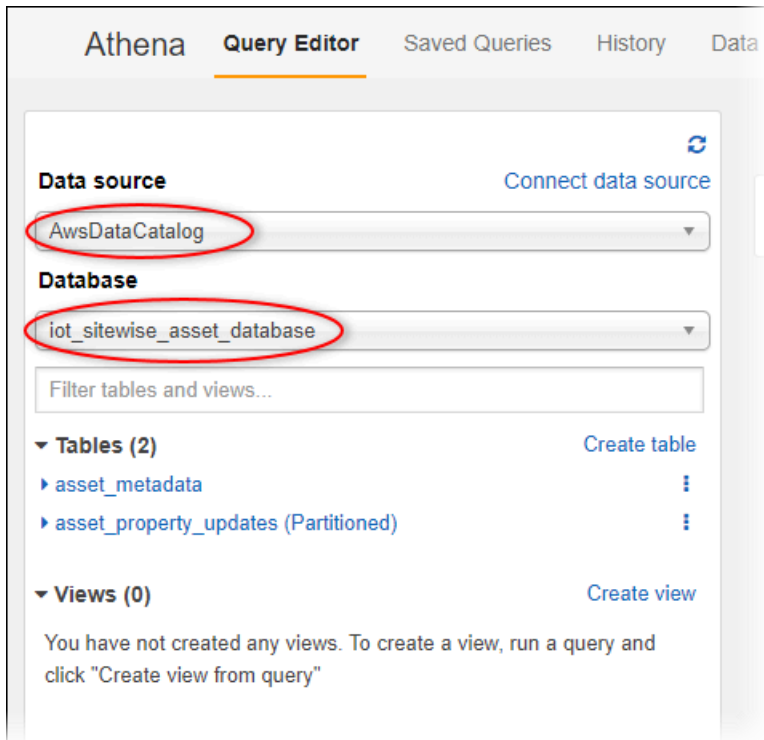
Query result location  ⓘ  
Example: s3://query-results-bucket/folder/

Encrypt query results  ⓘ

Autocomplete  ⓘ

Cancel Save

- c. 选择保存。
3. 左侧面板包含要查询的数据源。执行以下操作：
    - a. 对于数据源，AwsDataCatalog请选择使用 AWS Glue Data Catalog。
    - b. 对于数据库，选择您使用 Crawler 创建 AWS Glue 的数据库。



您应该看到两个表：asset\_metadata 和 asset\_property\_updates。

4. 要根据资产属性数据和元数据的组合创建视图，请输入以下查询，然后选择 运行查询。

```
CREATE
  OR REPLACE VIEW iot_sitewise_asset_data AS
SELECT "from_unixtime"("time_in_seconds" + ("offset_in_nanos" / 1000000000))
  "timestamp",
      "metadata"."asset_name",
      "metadata"."asset_property_name",
      "data"."asset_property_value",
      "metadata"."asset_property_unit",
      "metadata"."asset_property_alias"
FROM ( "iot_sitewise_asset_database".asset_property_updates data
INNER JOIN "iot_sitewise_asset_database".asset_metadata metadata
  ON ( ("data"."asset_id" = "metadata"."asset_id")
      AND ("data"."asset_property_id" = "metadata"."asset_property_id") ) );
```

此查询连接与资产 ID 和属性 ID 相关的资产属性数据和元数据表以创建视图。您可以多次运行此查询，如果视图已存在，它将替换现有视图。

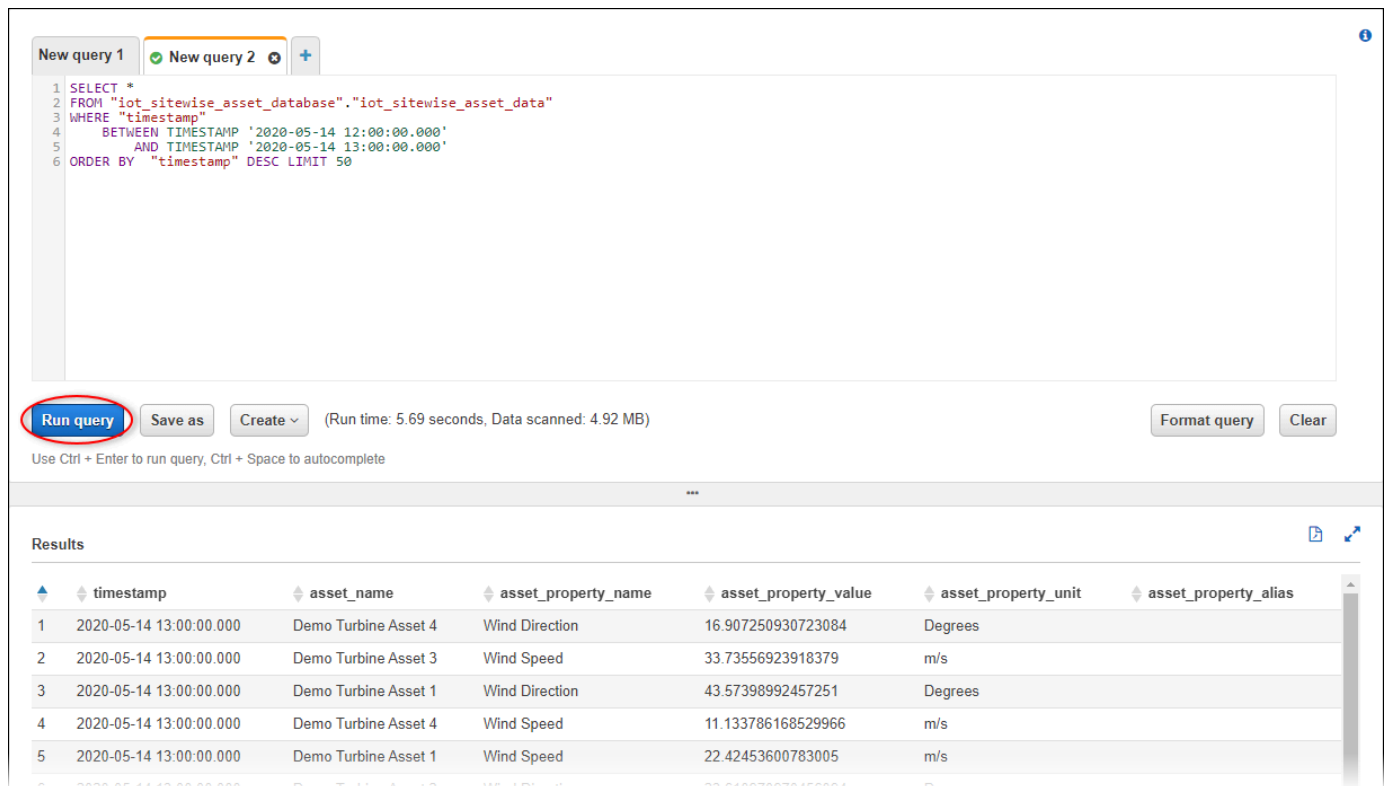
5. 选择 + 图标以添加新查询。

6. 要查看资产数据示例，请输入以下查询，然后选择 Run query (运行查询)。将时间戳替换为存储桶包含数据的时间间隔。

```
SELECT *
FROM "iot_sitewise_asset_database"."iot_sitewise_asset_data"
WHERE "timestamp"
    BETWEEN TIMESTAMP '2020-05-14 12:00:00.000'
    AND TIMESTAMP '2020-05-14 13:00:00.000'
ORDER BY "timestamp" DESC LIMIT 50;
```

此查询在两个时间戳之间输出多达 50 个数据点，先显示最近的条目。

您的查询输出可能与以下结果类似。



The screenshot shows the AWS IoT SiteWise console interface. At the top, there are two tabs: "New query 1" and "New query 2". The "New query 1" tab is active, displaying a SQL query in a text editor. The query is: `SELECT * FROM "iot_sitewise_asset_database"."iot_sitewise_asset_data" WHERE "timestamp" BETWEEN TIMESTAMP '2020-05-14 12:00:00.000' AND TIMESTAMP '2020-05-14 13:00:00.000' ORDER BY "timestamp" DESC LIMIT 50`. Below the editor, there are buttons for "Run query" (highlighted with a red circle), "Save as", "Create", "Format query", and "Clear". A status bar indicates "(Run time: 5.69 seconds, Data scanned: 4.92 MB)". Below the editor, there is a "Results" section showing a table of data. The table has six columns: "timestamp", "asset\_name", "asset\_property\_name", "asset\_property\_value", "asset\_property\_unit", and "asset\_property\_alias". The first five rows of data are visible:

timestamp	asset_name	asset_property_name	asset_property_value	asset_property_unit	asset_property_alias
2020-05-14 13:00:00.000	Demo Turbine Asset 4	Wind Direction	16.907250930723084	Degrees	
2020-05-14 13:00:00.000	Demo Turbine Asset 3	Wind Speed	33.73556923918379	m/s	
2020-05-14 13:00:00.000	Demo Turbine Asset 1	Wind Direction	43.57398992457251	Degrees	
2020-05-14 13:00:00.000	Demo Turbine Asset 4	Wind Speed	11.133786168529966	m/s	
2020-05-14 13:00:00.000	Demo Turbine Asset 1	Wind Speed	22.42453600783005	m/s	

现在，您可以运行对您的 AWS IoT SiteWise 应用程序有用的查询。有关更多信息，请参阅 Amazon Athena 用户指南中的 [Amazon Athena SQL 参考](#)。

## 从模板创建的资源

使用模板创建堆栈时，AWS CloudFormation 会创建以下资源。大多数资源的名称都包含一个前缀，您可以在创建堆栈时自定义该前缀。

## 资源名称参数

- **BucketName** - 从此模板创建的 S3 存储桶的名称，该存储桶接收资产数据。
- **GlobalResourcePrefix** - 从此模板创建的全局资源名称的前缀。默认值为 `sitewise-export-to-s3`。
- **LocalResourcePrefix** - 在当前区域中从此模板创建的资源的名称前缀。默认值为 `sitewise_export_to_s3`。

## AWS CloudFormation 模板创建的资源

资源	描述	名称
<a href="#">S3</a> 存储桶，用于已处理数据	此存储桶包含两个文件夹。一个文件夹接收来自 Firehose 交付流的扁平化格式化数据，另一个文件夹接收资产元数据。	<code>\${BucketName}</code>
<a href="#">AWS Glue</a> 数据库	此数据库包含此堆栈创建的 AWS Glue 表。	<code>\${LocalResourcePrefix}_firehose_glue_database</code>
<a href="#">AWS Glue</a> 表	Firehose 交付流使用此表将数据格式化为 Parquet 格式。	<code>\${LocalResourcePrefix}_firehose_glue_table</code>
<a href="#">AWS Lambda</a> 函数，用于转换数据	此函数对从中发送的属性值通知消息中的值数组进行扁平化。AWS IoT SiteWise	<code>\${LocalResourcePrefix}_lambda_transform_function</code>
<a href="#">IAM</a> 角色，用于转换 Lambda 函数	此角色允许 Lambda 存储转换函数的运行时系统日志。	<code>\${GlobalResourcePrefix}-lambda-transform-role</code>
<a href="#">IAM</a> policy，用于转换 Lambda 函数	此策略允许 Lambda 存储转换函数的执行日志。	<code>\${GlobalResourcePrefix}-lambda-transform-policy</code>

资源	描述	名称
<a href="#">CloudWatch 记录</a> 转换函数的日志组	此日志组包含转换函数的日志。	/aws/lambda/\${LocalResourcePrefix}_lambda_transform_function
<a href="#">Lambda</a> 函数，用于收集资产元数据	此函数检索中资产的详细信息，AWS IoT SiteWise 并将详细信息存储在此堆栈创建的 Amazon S3 存储桶中。	\${LocalResourcePrefix}_lambda_metadata_function
<a href="#">Lambda</a> 层，用于元数据函数	该层提供了一个 AWS SDK，其中包含元数据函数使用的 AWS IoT SiteWise 操作。	\${LocalResourcePrefix}_lambda_metadata_layer
<a href="#">IAM</a> 角色，用于元数据 Lambda 函数	此角色允许 Lambda 检索中资产的详细信息。AWS IoT SiteWise	\${GlobalResourcePrefix}-lambda-metadata-role
<a href="#">IAM</a> policy 角色，用于元数据 Lambda 函数角色	此策略允许 Lambda 检索中资产的详细信息。AWS IoT SiteWise	\${GlobalResourcePrefix}-lambda-metadata-policy
<a href="#">EventBridge</a> 元数据 Lambda 函数的预定事件	此计划的事件每 6 小时运行一次元数据 Lambda，以更新资产元数据存储桶。	\${LocalResourcePrefix}-metadata-event
CloudWatch 元数据函数的 @@ <a href="#">日志</a> 日志组	此日志组包含元数据函数的日志。	/aws/lambda/\${LocalResourcePrefix}_lambda_metadata_function
<a href="#">AWS IoT</a> 规则	此规则查询属性值通知消息，并将资产数据发送到 Amazon Data Firehose 传输流。	\${LocalResourcePrefix}_iot_topic_rule

资源	描述	名称
AWS IoT 规则@@ 的 IAM 角色	此角色 AWS IoT 允许向 Firehose 传输流发送数据。	<code>\${GlobalResourcePrefix}-core-firehose-role</code>
AWS IoT 规则角色的 IAM 策略	此策略 AWS IoT 允许向 Firehose 传输流发送数据。	<code>\${GlobalResourcePrefix}-core-firehose-policy</code>
<a href="#">Firehose 传送直播</a>	此传输流使用 AWS IoT 规则中的数据，使用 Lambda 函数对数据进行扁平化，然后将数据传输到 Amazon S3。	<code>\${LocalResourcePrefix}_firehose_delivery_stream</code>
传输流 IAM 角色	此角色允许 Firehose 对 S3 存储桶、AWS Glue 表、Lambda 函数和 CloudWatch 日志组执行操作。	<code>\${GlobalResourcePrefix}-firehose-delivery-role</code>
<a href="#">CloudWatch 记录</a> 传输流的日志组	此日志组包含一个日志流 S3 Delivery，用于接收有关 Firehose 传输流的日志。	<code>/aws/kinesisfirehose/\${LocalResourcePrefix}_firehose_delivery_stream</code>

## 与 Grafana 集成

Grafana 是一个数据可视化平台，可用于可视化和监控仪表板中的数据。在 Grafana 版本 7.3.0 及更高版本中，您可以使用 AWS IoT SiteWise 插件在 Grafana 仪表板中可视化 AWS IoT SiteWise 资产数据。您可以使用单个 Grafana 控制面板可视化来自多个 AWS 来源（例如 AWS IoT SiteWise Amazon Timestream 和亚马逊 CloudWatch）和其他数据源的数据。

使用 AWS IoT SiteWise 插件有两种选择：

- 本地 Grafana 服务器

您可以在您管理的 Grafana 服务器上设置 AWS IoT SiteWise 插件。有关如何添加和使用插件的更多信息，请参阅网站上的 [AWS IoT SiteWise Datasource 自述](#) 文件。GitHub

- AWS Managed Service for Grafana

您可以在 AWS Managed Service for Grafana (AMG) 中使用 AWS IoT SiteWise 插件。AMG 为您管理 Grafana 服务器，以便您可以可视化您的数据，而无需构建、打包或部署任何硬件或任何其他 Grafana 基础设施。有关更多信息，请参阅 AWS Managed Service for Grafana 用户指南中的以下主题：

- [什么是 Amazon Managed Service for Grafana \(AMG\) ?](#)
- [使用 AWS IoT SiteWise 数据来源](#)

### Example Grafana 仪表板示例

以下 Grafana 仪表板可视化 [演示风电场](#)。您可以在 [Grafana Play](#) 网站上访问此演示仪表板。





## 集成 AWS IoT SiteWise 和 AWS IoT TwinMaker

与集成后AWS IoT TwinMaker，用户可以访问中的强大功能AWS IoT SiteWise，例如AWS IoT SiteWise控制台中的AWS IoT SiteWise数据检索 ExecuteQuery API 和高级资产搜索。要集成服务并使用这些功能，必须先启用集成。

### 主题

- [启用集成](#)
- [集成 AWS IoT SiteWise 和 AWS IoT TwinMaker](#)

## 启用集成

管理员可以使用 AWS JSON 策略来指定谁有权访问什么内容。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。JSON 策略的 Action 元素描述可用于在策略中允许或拒绝访问的操作。有关 AWS IoT SiteWise 支持操作的更多信息，请参阅《服务授权参考》中的[AWS IoT SiteWise 定义的操作](#)。

有关 AWS IoT TwinMaker 服务关联角色的更多信息，请参阅《AWS IoT TwinMaker 用户指南》AWS IoT TwinMaker 中的[服务关联角色](#)。

在集成 AWS IoT SiteWise 和之前 AWS IoT TwinMaker，必须授予 AWS IoT SiteWise 允许与 AWS IoT TwinMaker 链接工作区集成的以下权限：

- `iotsitewise:EnableSiteWiseIntegration`— AWS IoT SiteWise 允许与链接 AWS IoT TwinMaker 的工作区集成。这种集成 AWS IoT TwinMaker 允许 AWS IoT SiteWise 通过 AWS IoT TwinMaker 与服务相关的角色读取所有建模信息。要启用此权限，请将以下策略添加到您的 IAM 角色：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:EnableSiteWiseIntegration"
      ],
      "Resource": "*"
    }
  ]
}
```

## 集成 AWS IoT SiteWise 和 AWS IoT TwinMaker

要整合 AWS IoT SiteWise 和 AWS IoT TwinMaker，您必须具备以下条件：

- AWS IoT SiteWise 在您的账户中设置服务相关角色
- AWS IoT TwinMaker 在您的账户中设置服务相关角色
- AWS IoT TwinMakerIoTSiteWiseDefaultWorkspace 在您所在区域的账户中使用 ID 的工作空间。

## 使用AWS IoT SiteWise控制台进行集成

当您在控制台中看到“与集成”AWS IoT TwinMaker 横幅时，选择“授予权限”。先决条件是在您的账户中创建的。

### 要通过使用进行集成 AWS CLI

要AWS IoT TwinMaker进行集成AWS IoT SiteWise并使用AWS CLI，请输入以下命令：

1. 拨打 CreateServiceLinkedRole o AWSServiceName fiotsitewise.amazonaws.com.

```
aws iam create-service-linked-role --aws-service-name iotsitewise.amazonaws.com
```

2. 拨打 CreateServiceLinkedRole o AWSServiceName f iottwinmaker.amazonaws.com.

```
aws iam create-service-linked-role --aws-service-name iottwinmaker.amazonaws.com
```

3. 拨打 CreateWorkspace o ID fIoTSiteWiseDefaultWorkspace.

```
aws iottwinmaker create-workspace --workspace-id IoTSiteWiseDefaultWorkspace
```

## 使用 Amazon Lookout for Equipment 检测设备异常

### Note

异常检测仅在提供 Amazon Lookout for Equipment 的地区可用。

您可以 AWS IoT SiteWise 与 Amazon Lookout for Equipment 集成，通过工业设备的异常检测和预测性维护，深入了解您的工业设备。Lookout for Equipment 是一项机器学习 (ML) 服务，用于监控工业设备，检测设备异常行为并识别潜在故障。借助 Lookout for Equipment，您可以实施预测性维护计划并识别次优的设备流程。有关 Lookout for Equipment 的更多信息，[请参阅 Amazon Lookout for Equipment 是什么？](#) 在《亚马逊 Lookout for Equipment 用户指南》中。

当您创建预测以训练机器学习模型以检测异常设备行为时，AWS IoT SiteWise 会将资产属性值发送到 Lookout for Equipment，以训练机器学习模型来检测异常设备行为。要定义资产模型的预测定义，您需要指定 Lookout for Equipment 访问您的数据所需的 IAM 角色，以及要发送到 Lookout for Equipment 并将处理过的数据发送到 Amazon S3 的属性。有关更多信息，请参阅 [创建资产模型](#)。

要集成 AWS IoT SiteWise 和 Lookout for Equipment，您需要执行以下高级步骤：

- 在资产模型上添加预测定义，概述您要跟踪的属性。预测定义是可重复使用的测量值、变换和指标集合，用于根据该资产模型对资产进行预测。
- 根据您提供的历史数据训练预测。
- 计划推断，它 AWS IoT SiteWise 告诉运行特定预测的频率。

计划推理后，Lookout for Equipment 模型会监控它从您的设备接收的数据，并查找设备行为的异常情况。您可以使用 GET API 操作或 Lookout for SiteWise r Equip AWS IoT SiteWise ment 控制台在 Monitor 中查看和分析结果。您还可以使用资产模型中的警报探测器创建警报，提醒您注意设备异常行为。

## 主题

- [添加预测定义 \(控制台\)](#)
- [训练预测 \(控制台\)](#)
- [启动或停止对预测的推理 \(控制台\)](#)
- [添加预测定义 \(CLI\)](#)
- [训练预测并开始推理 \(CLI\)](#)
- [训练预测 \(CLI\)](#)
- [开始或停止对预测的推理 \(CLI\)](#)

## 添加预测定义 (控制台)

要开始将收集的数据发送 AWS IoT SiteWise 到 Lookout for Equipment，您必须向资产模型添加 AWS IoT SiteWise 预测定义。

向 AWS IoT SiteWise 资产模型添加预测定义

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择模型，然后选择要向其添加预测定义的资产模型。
3. 选择“预测”。
4. 选择添加预测定义。
5. 定义有关预测定义的详细信息。

- a. 为您的预测定义输入唯一的名称和描述。请谨慎选择名称，因为在创建预测定义之后，您无法更改其名称。
- b. 创建或选择允许与 Amazon Lookout for Equipment 共享您的资产数据的 IAM 权限角色。该角色应具有以下 IAM 和信任策略。有关创建角色的帮助，请参阅[使用自定义信任策略创建角色 \(控制台\)](#)。

### IAM policy

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "L4EPermissions",
    "Effect": "Allow",
    "Action": [
      "lookoutequipment:CreateDataset",
      "lookoutequipment:CreateModel",
      "lookoutequipment:CreateInferenceScheduler",
      "lookoutequipment:DescribeDataset",
      "lookoutequipment:DescribeModel",
      "lookoutequipment:DescribeInferenceScheduler",
      "lookoutequipment:ListInferenceExecutions",
      "lookoutequipment:StartDataIngestionJob",
      "lookoutequipment:StartInferenceScheduler",
      "lookoutequipment:UpdateInferenceScheduler",
      "lookoutequipment:StopInferenceScheduler"
    ],
    "Resource": [
      "arn:aws:lookoutequipment:Region:Account_ID:inference-scheduler/IoTSiteWise_*",
      "arn:aws:lookoutequipment:Region:Account_ID:model/IoTSiteWise_*",
      "arn:aws:lookoutequipment:Region:Account_ID:dataset/IoTSiteWise_*"
    ]
  },
  {
    "Sid": "L4EPermissions2",
    "Effect": "Allow",
    "Action": [
      "lookoutequipment:DescribeDataIngestionJob"
    ],
    "Resource": "*"
  }
}
```

```

    },
    {
      "Sid": "S3Permissions",
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:ListBucket",
        "s3:PutObject",
        "s3:GetObject"
      ],
      "Resource": ["arn:aws:s3:::iotsitewise-*"]
    },
    {
      "Sid": "IAMPermissions",
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::Account_ID:role/Role_name"
    }
  ]
}

```

## 信任策略

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "Service": "iotsitewise.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "Account_ID"
      },
      "ArnEquals": {
        "aws:SourceArn":
"arn:aws:iotsitewise:Region:Account_ID:asset/*"
      }
    }
  ]
}

```

```
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lookoutequipment.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "Account_ID"
        },
        "ArnEquals": {
          "aws:SourceArn":
            "arn:aws:lookoutequipment:Region:Account_ID:*"
        }
      }
    }
  ]
}
```

- c. 选择下一步。
6. 选择要发送到 Lookout for Equipment 的数据属性（测量、变换和指标）。
    - a. （可选）选择测量值。
    - b. （可选）选择变换。
    - c. （可选）选择指标。
    - d. 选择下一步。
  7. 检视您的选择。要将预测定义添加到资产模型中，请在摘要页面上选择添加预测定义。

您也可以编辑或删除附加了有效预测的现有预测定义。

## 训练预测（控制台）

在向资产模型添加预测定义后，您可以对资产的预测进行训练。

在中训练预测 AWS IoT SiteWise

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择资产，然后选择要监控的资产。
3. 选择“预测”。

4. 选择要训练的预测。
5. 在“操作”下，选择“开始训练”，然后执行以下操作：
  - a. 在“预测详情”下，选择允许 AWS IoT SiteWise 与 Lookout for Equipment 共享您的资产数据的 IAM 权限角色。如果需要创建新角色，请选择创建新角色。
  - b. 在训练数据设置中，输入训练数据时间范围以选择用于训练预测的数据。
  - c. （可选）选择后处理后数据的采样率。
  - d. （可选）对于数据标签，请提供用于保存标签数据的 Amazon S3 存储桶和前缀。有关为数据添加标签的更多信息，请参阅 Amazon Lookout for Equipment 用户指南中的为[数据添加](#)标签。
  - e. 选择下一步。
6. （可选）如果您希望预测在训练完成后立即生效，请在“高级设置”下，选择“训练后自动激活预测”，然后执行以下操作：
  - a. 在输入数据下，对于数据上传频率，定义上传数据的频率，对于偏移延迟时间，定义要使用的缓冲区量。
  - b. 选择下一步。
7. 查看预测的详细信息，然后选择保存并开始。

## 启动或停止对预测的推理（控制台）

### Note

Lookout for Equipment 费用适用于在 Lookout for Equipment AWS IoT SiteWise 之间传输数据的预定推断。有关更多信息，请参阅 [Amazon Lookout for Equipment 定价](#)。

如果你添加了预测 b “lookoutequipment:CreateDataset”，但它在训练后没有选择激活它，你必须激活它才能开始监控你的资产。

### 开始对预测进行推理

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择资产，然后选择要添加预测的资产。
3. 选择“预测”。
4. 选择要激活的预测。



5. 在“操作”下，选择“开始推理”，然后执行以下操作：
  - a. 在输入数据下，对于数据上传频率，定义上传数据的频率，对于偏移延迟时间，定义要使用的缓冲区量。
  - b. 选择“保存并启动”。

### 停止对预测进行推理

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择资产，然后选择要添加预测的资产。
3. 选择“预测”。
4. 选择要停止的预测。
5. 在“操作”下，选择“停止推理”。

## 添加预测定义 (CLI)

要在新资产模型或现有资产模型上定义预测定义，可以使用 AWS Command Line Interface (AWS CLI)。在资产模型上定义预测定义后，您可以使用 Lookout for Equipment 对资产的预测进行训练和计划推理，AWS IoT SiteWise 以便进行异常检测。

### 先决条件

要完成这些步骤，您必须拥有一个资产模型并至少创建一个资产。有关更多信息，请参阅 [创建资产模型 \(AWS CLI\)](#) 和 [创建资产 \(AWS CLI\)](#)。

如果您不熟悉 AWS IoT SiteWise，则必须调用 CreateBulkImportJob API 操作将资产属性值导入其中 AWS IoT SiteWise，这将用于训练模型。有关更多信息，请参阅 [创建批量导入作业\(AWS CLI\)](#)。

### 添加预测定义

1. 创建名为 `asset-model-payload.json` 的文件。按照其他部分中的步骤将资产模型的详细信息添加到文件中，但不要提交创建或更新资产模型的请求。
  - 有关如何创建资产模型的更多信息，请参阅 [创建资产模型 \(AWS CLI\)](#)
  - 有关如何更新现有资产模型的更多信息，请参阅 [更新资产或组件模型 \(AWS CLI\)](#)
2. 通过添加以下代码，将 Lookout for Equipment 复合模型 `assetModelCompositeModels ()` 添加到资产模型中。

- *Property* 替换为要包含的属性的 ID。要获取这些身份证，请致电 [DescribeAssetModel](#)。
- *RoleARN* 替换为允许 Lookout for Equipment 访问您的数据的 IAM 角色的 ARN。AWS IoT SiteWise

```
{
  ...
  "assetModelCompositeModels": [
    {
      "name": "L4Epredictiondefinition",
      "type": "AWS/L4E_ANOMALY",
      "properties": [
        {
          "name": "AWS/L4E_ANOMALY_RESULT",
          "dataType": "STRUCT",
          "dataTypeSpec": "AWS/L4E_ANOMALY_RESULT",
          "unit": "none",
          "type": {
            "measurement": {}
          }
        },
        {
          "name": "AWS/L4E_ANOMALY_INPUT",
          "dataType": "STRUCT",
          "dataTypeSpec": "AWS/L4E_ANOMALY_INPUT",
          "type": {
            "attribute": {
              "defaultValue": "{\"properties\": [\"Property1\", \"Property2\"]}"
            }
          }
        },
        {
          "name": "AWS/L4E_ANOMALY_PERMISSIONS",
          "dataType": "STRUCT",
          "dataTypeSpec": "AWS/L4E_ANOMALY_PERMISSIONS",
          "type": {
            "attribute": {
              "defaultValue": "{\"roleArn\": \"RoleARN\"}"
            }
          }
        }
      ]
    }
  ]
}
```

```
    "name": "AWS/L4E_ANOMALY_DATASET",
    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/L4E_ANOMALY_DATASET",
    "type": {
      "attribute": {}
    }
  },
  {
    "name": "AWS/L4E_ANOMALY_MODEL",
    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/L4E_ANOMALY_MODEL",
    "type": {
      "attribute": {}
    }
  },
  {
    "name": "AWS/L4E_ANOMALY_INFERENCE",
    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/L4E_ANOMALY_INFERENCE",
    "type": {
      "attribute": {}
    }
  },
  {
    "name": "AWS/L4E_ANOMALY_TRAINING_STATUS",
    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/L4E_ANOMALY_TRAINING_STATUS",
    "type": {
      "attribute": {
        "defaultValue": "{}"
      }
    }
  },
  {
    "name": "AWS/L4E_ANOMALY_INFERENCE_STATUS",
    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/L4E_ANOMALY_INFERENCE_STATUS",
    "type": {
      "attribute": {
        "defaultValue": "{}"
      }
    }
  }
]
```

```
}
```

### 3. 创建资产模型或更新现有资产模型。请执行以下操作之一：

- 要创建资产模型，请运行以下命令：

```
aws iotsitewise create-asset-model --cli-input-json file://asset-model-payload.json
```

- 要更新现有资源模型，请运行以下命令。*asset-model-id*替换为要更新的资产模型的 ID。

```
aws iotsitewise update-asset-model \  
  --asset-model-id asset-model-id \  
  --cli-input-json file://asset-model-payload.json
```

运行命令之后，请记住响应中的 `assetModelId`。

## 训练预测并开始推理 (CLI)

现在，预测定义已经定义，您可以根据它训练资产并开始推理。如果您想训练预测但不想开始推理，请跳至。[训练预测 \(CLI\)](#)要训练预测并开始对资产进行推理，您需要目标资源`assetId`的。

### 训练并开始对预测进行推断

1. 运行以下命令查找`assetModelCompositeModelId`下方`assetModelCompositeModelSummaries`。*asset-model-id*替换为您在中创建的资产模型的 ID [更新资产或组件模型 \(AWS CLI\)](#)。

```
aws iotsitewise describe-asset-model \  
  --asset-model-id asset-model-id \  
  --cli-input-json file://asset-model-payload.json
```

2. 运行以下命令以查找`actionDefinitionId`动`TrainingWithInference`作。*asset-model-id*替换为上一步中使用的 ID，*asset-model-composite-model-id*替换为上一步中返回的 ID。

```
aws iotsitewise describe-asset-model-composite-model \  
  --asset-model-id asset-model-id \  
  --asset-model-composite-model-id asset-model-composite-model-id \  
  --cli-input-json file://asset-model-payload.json
```

3. 创建一个名为的文件`train-start-inference-prediction.json`并添加以下代码，替换以下代码：

- *asset-id*使用目标资产的 ID
- *action-definition-id*带有动 TrainingWithInference 作的 ID
- *StartTime*以训练数据的开头为单位，以纪元秒为单位提供
- *EndTime*训练数据结束，以纪元秒为单位提供
- *TargetSamplingRate*使用 Lookout for Equipment 进行后期处理后的数据采样率。允许的值为:PT1S | PT5S | PT10S | PT15S | PT30S | PT1M | PT5M | PT10M | PT15M | PT30M | PT1H。

```
{
  "targetResource": {
    "assetId": "asset-id"
  },
  "actionDefinitionId": "action-definition-Id",
  "actionPayload":{
    "stringValue": "{\"l4ETrainingWithInference\":{\"trainingWithInferenceMode\": \"START\", \"trainingPayload\": {\"exportDataStartTime\": StartTime, \"exportDataEndTime\": EndTime}, \"targetSamplingRate\": \"TargetSamplingRate\"}, \"inferencePayload\": {\"dataDelayOffsetInMinutes\": 0, \"dataUploadFrequency\": \"PT5M\"}}}"
  }
}
```

4. 运行以下命令开始训练和推理：

```
aws iotsitewise execute-action --cli-input-json file://train-start-inference-prediction.json
```

## 训练预测 (CLI)

现在，预测定义已经定义，您可以根据它训练资产。要训练资产的预测，您需要目标`assetId`资源的预测。



```
"actionPayload":{ "stringValue": "{\\"l4ETraining\\": {\\"trainingMode\\":
\\"START\\",\\"exportDataStartTime\\": StartTime, \\"exportDataEndTime\\": EndTime,
\\"targetSamplingRate\\":\\"TargetSamplingRate\"}, \\"labelInputConfiguration\\":
{\\"bucketName\\": \\"BucketName\\", \\"prefix\\": \\"Prefix\"}}}"
}
}
```

#### 4. 运行以下命令开始训练：

```
aws iotsitewise execute-action --cli-input-json file://train-prediction.json
```

在开始推理之前，必须完成训练。要查看培训状态，请执行以下任一操作：

- 在控制台中，导航到预测所在的资产。
- 从中 AWS CLI，BatchGetAssetPropertyValue使用该propertyIdtrainingStatus属性的调用。

## 开始或停止对预测的推理 (CLI)

训练完预测后，你可以开始推理，告诉 Lookout for Equipment 开始监控你的资产。要开始或停止推理，您需要目标资源assetId的。

### 开始推理

1. 运行以下命令查找assetModelCompositeModelId下方assetModelCompositeModelSummaries。*asset-model-id*替换为您在中创建的资产模型的 ID [更新资产或组件模型 \(AWS CLI\)](#)。

```
aws iotsitewise describe-asset-model \
  --asset-model-id asset-model-id \
```

2. 运行以下命令以查找actionDefinitionId动Inference作。*asset-model-id*替换为上一步中使用的 ID，*asset-model-composite-model-id*替换为上一步中返回的 ID。

```
aws iotsitewise describe-asset-model-composite-model \
  --asset-model-id asset-model-id \
  --asset-model-composite-model-id asset-model-composite-model-id \
```

3. 创建一个名为的文件start-inference.json并添加以下代码，替换以下代码：

- *asset-id* 使用目标资产的 ID
- *action-definition-id* 使用开始推理操作的 ID
- *Offset* 附上要使用的缓冲量
- *Frequency* 以及数据上传的频率

```
{
  "targetResource": {
    "assetId": "asset-id"
  },
  "actionDefinitionId": "action-definition-Id",
  "actionPayload": { "stringValue": "{\"\u0014EInference\": {\"inferenceMode\": \"START\", \"dataDelayOffsetInMinutes\": Offset, \"dataUploadFrequency\": \"Frequency\"}}"}
}
```

#### 4. 运行以下命令开始推理：

```
aws iotsitewise execute-action --cli-input-json file://start-inference.json
```

### 停止推理

1. 运行以下命令查找 `assetModelCompositeModelId` 下方 `assetModelCompositeModelSummaries`。 *asset-model-id* 替换为您在中创建的资产模型的 ID [更新资产或组件模型 \(AWS CLI\)](#)。

```
aws iotsitewise describe-asset-model \
  --asset-model-id asset-model-id \
```

2. 运行以下命令以查找 `actionDefinitionId` 启动 Inference 作。 *asset-model-id* 替换为上一步中使用的 ID， *asset-model-composite-model-id* 替换为上一步中返回的 ID。

```
aws iotsitewise describe-asset-model-composite-model \
  --asset-model-id asset-model-id \
  --asset-model-composite-model-id asset-model-composite-model-id \
```

3. 创建一个名为 `stop-inference.json` 的文件并添加以下代码，替换以下代码：

- *asset-id* 使用目标资产的 ID



- *action-definition-id* 使用开始推理操作的 ID

```
{
  "targetResource": {
    "assetId": "asset-id"
  },
  "actionDefinitionId": "action-definition-Id",
  "actionPayload": { "stringValue": "{\"l4EInference\":{\"inferenceMode\":\"STOP\"}}"
  }
}
```

4. 运行以下命令停止推理：

```
aws iotsitewise execute-action --cli-input-json file://stop-inference.json
```

# 管理数据存储

您可以配置 AWS IoT SiteWise 为将数据保存在以下存储层中：

## 热层

热存储层是 AWS IoT SiteWise 托管的时间序列存储。Hot tier 对频繁访问的数据最有效，write-to-read 延迟时间很短。存储在热层中的数据供需要快速访问设备中最新测量值的工业应用程序使用。这包括使用交互式仪表板可视化实时指标的应用程序，或监控操作并启动警报以识别性能问题的应用程序。

默认情况下，采集到的数据存储 AWS IoT SiteWise 在热门层中。您可以为热层定义保留期，之后根据您的配置将热层中的数据 AWS IoT SiteWise 移动到温层或冷层存储。为了获得最佳性能和成本效益，请将热层保留期设置为比经常检索数据所花费的时间长。这用于实时指标、警报和监控场景。如果未设置保留期，则您的数据将无限期地存储在热门层中。

## 温暖等级

温存储层是一个 AWS IoT SiteWise 托管层，可以有效地经济高效地存储历史数据。它最适合用于检索具有中等 write-to-read 延迟特征的大量数据。使用暖层存储大型工作负载所需的历史数据。例如，它用于分析、商业智能应用程序 (BI)、报告工具和机器学习 (ML) 模型训练的数据检索。如果您启用冷存储层，则可以定义温层保留期。保留期结束后，从温层中 AWS IoT SiteWise 删除数据。

## 冷层

冷存储层使用 Amazon S3 存储桶来存储很少使用的数据。启用冷层后，每 AWS IoT SiteWise 6 小时复制一次时间序列，包括测量、指标、变换和聚合以及资产模型定义。Cold tier 用于存储可容忍历史报告和备份的高读取延迟的数据。

## 主题

- [配置存储设置](#)
- [存储设置疑难解答](#)
- [冷层中保存的数据的文件路径和架构](#)

## 配置存储设置

您可以配置存储设置以选择使用托管的温层存储，也可以将数据复制到冷层。要详细了解温暖和热门等级的保留期，请参阅[数据保留的影响](#)。配置存储设置时，请执行以下操作：

- 热层保留 — 设置保留期，规定数据在删除之前在热层中存储多长时间，然后根据您的存储设置将其移至服务托管的温层存储或冷层存储。AWS IoT SiteWise 将删除在保留期结束之前存在的热门层中的所有数据。如果您未设置保留期，则您的数据将无限期地存储在热门层中。
- 热层保留 — 设置保留期，规定您的数据在从存储中删除并移至客户管理的冷层 AWS IoT SiteWise 存储之前在温层中存储多长时间。AWS IoT SiteWise 从保留期结束之前存在的暖层中删除所有数据。如果未设置保留期，则您的数据将无限期地存储在温层中。

### Note

要提高查询性能，请使用温层存储设置热层保留期。

## 在热层和温层存储中保留数据的影响

- 当您缩短热层存储的保留期时，数据将从热层永久移动到温层或冷层。当您缩短温层的保留期时，数据将移至冷层，并从温层中永久删除。
- 当您延长热层或温层存储的保留期时，更改会影响 AWS IoT SiteWise 从那时起发送到的数据。AWS IoT SiteWise 不会从温存储或冷存储中检索数据以填充热层。例如，如果热层存储的保留期最初设置为 30 天，然后增加到 60 天，则热层存储需要 30 天才能包含价值 60 天的数据。

### 主题

- [配置暖层（控制台）的存储设置](#)
- [为温层配置存储设置 \(AWS CLI\)](#)
- [配置冷层（控制台）的存储设置](#)
- [配置冷层的存储设置 \(AWS CLI\)](#)

## 配置暖层（控制台）的存储设置

以下过程向您展示如何在 AWS IoT SiteWise 控制台中配置存储设置以将数据复制到温层。

若要配置控制台中的存储设置

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中的设置下，选择存储。
3. 在右上角，选择 编辑。

4. 在编辑存储页面中，执行以下操作：
5. 对于热门等级设置，请执行以下操作：
  - 如果要为数据在删除之前在热层中存储的时间设置保留期，并移至服务托管的温层存储，请选择启用保留期。
  - 要配置保留期，请输入整数并选择单位。保留期必须大于或等于 30 天。

AWS IoT SiteWise 删除热门层中所有早于保留期的数据。如果未设置保留期，则数据将无限期存储。

6. (推荐) 对于 Warm 等级设置，请执行以下操作：
  - 要选择使用温层存储，请选择我确认加入温层存储，选择加入温层存储。
  - (可选) 要配置保留期，请输入整数并选择一个单位。保留期必须大于或等于 365 天。

AWS IoT SiteWise 删除在保留期之前存在的温层中的数据。如果未设置保留期，则数据将无限期存储。

#### Note

- 当您选择使用暖层时，配置仅显示一次。
- 要设置热层保留期，您必须有温层或冷层存储。为了提高成本效益和检索历史数据，AWS IoT SiteWise 建议您将长期数据存储存储在温层。
- 要设置热层保留，必须有冷层存储。

7. 选择“保存”以保存您的存储设置。

在AWS IoT SiteWise 存储部分中，温层存储处于以下状态之一：

- 已启用 — 如果您的数据在热层保留期之前存在，则会将数据 AWS IoT SiteWise 移至热层。”
- 已禁用-禁用暖层存储。

## 为温层配置存储设置 (AWS CLI)

您可以使用 AWS CLI 和以下命令配置存储设置以将数据移动到温层。

为防止覆盖现有配置，请运行以下命令来检索当前的存储配置信息：

```
aws iotsitewise describe-storage-configuration
```

Example 在没有现有冷层配置的情况下进行响应

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "disassociatedDataStorage": "ENABLED",
  "configurationStatus": {
    "state": "ACTIVE"
  },
  "lastUpdateDate": "2021-10-14T15:53:35-07:00",
  "warmTier": "DISABLED"
}
```

Example 使用现有冷层配置进行响应

```
{
  "storageType": "MULTI_LAYER_STORAGE",
  "multiLayerStorage": {
    "customerManagedS3Storage": {
      "s3ResourceArn": "arn:aws:s3:::bucket-name/prefix/",
      "roleArn": "arn:aws:iam::aws-account-id:role/role-name"
    }
  },
  "disassociatedDataStorage": "ENABLED",
  "retentionPeriod": {
    "numberOfDays": retention-in-days
  },
  "configurationStatus": {
    "state": "ACTIVE"
  },
  "lastUpdateDate": "2023-10-25T15:59:46-07:00",
  "warmTier": "DISABLED"
}
```

## 使用配置暖层的存储设置 AWS CLI

运行以下命令来配置存储设置。file-name 替换为包含 AWS IoT SiteWise 存储配置的文件名称。

```
aws iotsitewise put-storage-configuration --cli-input-json file://file-name.json
```

## Example AWS IoT SiteWise 带热层和温层的配置

```
{
    "storageType": "SITEWISE_DEFAULT_STORAGE",
    "disassociatedDataStorage": "ENABLED",
    "warmTier": "ENABLED",
    "retentionPeriod": {
        "numberOfDays": hot-tier-retention-in-days
    }
}
```

*hot-tier-retention-in-days* 必须是大于或等于 30 天的整数。

## Example 响应

```
{
    "storageType": "SITEWISE_DEFAULT_STORAGE",
    "configurationStatus": {
        "state": "UPDATE_IN_PROGRESS"
    }
}
```

如果您启用了冷层存储，请参阅[使用 AWS CLI 和现有冷层配置存储设置](#)。

## 使用 AWS CLI 和现有冷层配置存储设置

### 使用 AWS CLI 现有冷层存储配置存储设置


- 运行以下命令来配置存储设置。将 *file-name* 替换为包含 AWS IoT SiteWise 存储配置的文件名称。

```
aws iotsitewise put-storage-configuration --cli-input-json file://file-name.json
```

### Example AWS IoT SiteWise 存储配置

- 将 *bucket-name* 替换为您的 Amazon S3 存储桶的名称。
- 将 *##* 替换为您的 Amazon S3 前缀。
- aws-account-id* 替换为您的 AWS 账户 ID。

- 将####替换为允许 AWS IoT SiteWise 向 Amazon S3 发送数据的 Amazon S3 访问角色的名称。
- 用大于或等于 30 *hot-tier-retention-in#####*- days。
- 用大于或等于 365 *warm-tier-retention-in#####*- days。

 Note

AWS IoT SiteWise 将删除温层中所有早于冷层保留期的数据。如果未设置保留期，则数据将无限期存储。

```
{
  "storageType": "MULTI_LAYER_STORAGE",
  "multiLayerStorage": {
    "customerManagedS3Storage": {
      "s3ResourceArn": "arn:aws:s3:::bucket-name/prefix/",
      "roleArn": "arn:aws:iam::aws-account-id:role/role-name"
    }
  },
  "disassociatedDataStorage": "ENABLED",
  "retentionPeriod": {
    "numberOfDays": hot-tier-retention-in-days
  },
  "warmTier": "ENABLED",
  "warmTierRetentionPeriod": {
    "numberOfDays": warm-tier-retention-in-days
  }
}
```

### Example 响应

```
{
  "storageType": "MULTI_LAYER_STORAGE",
  "configurationStatus": {
    "state": "UPDATE_IN_PROGRESS"
  }
}
```

## 配置冷层（控制台）的存储设置

以下过程向您展示如何在 AWS IoT SiteWise 控制台中配置存储设置以将数据复制到冷层。

若要配置控制台中的存储设置

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中的设置下，选择存储。
3. 在右上角，选择 编辑。
4. 在编辑存储页面中，执行以下操作：
  - a. 对于 存储设置，选择 启用冷层存储。默认情况下，冷层存储处于禁用状态。
  - b. 对于 S3 存储桶位置，输入现有 Amazon S3 存储桶的名称和前缀。

### Note

- Amazon S3 使用前缀作为 Amazon S3 存储桶中的文件夹名称。前缀必须包含 1-255 个字符，并以正斜杠 (/) 结尾。您的 AWS IoT SiteWise 数据将保存在此文件夹中。
- 如果您没有 Amazon S3 存储桶，请选择 查看，然后在 Amazon S3 控制台中创建一个存储桶。有关更多信息，请参阅 Amazon S3 用户指南中的 [创建您的第一个 S3 存储桶](#)。

- c. 对于 S3 访问角色，执行下列操作之一：
  - 选择从 AWS 托管模板创建角色，AWS 自动创建允许向 Amazon S3 AWS IoT SiteWise 发送数据的 IAM 角色。
  - 选择使用现有角色，然后从列表中选择您创建的角色。

### Note

- 您必须对 S3 存储桶位置使用与您在上一步和 IAM Policy 中使用的相同的 Amazon S3 存储桶名称。
- 确保您的角色具有以下示例中显示的权限。



## Example 权限策略 :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:GetBucketLocation",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}
```

将 *bucket-name* 替换为您的 Amazon S3 存储桶的名称。

- d. 要设置热门等级，请参阅中的步骤 5 [配置暖层（控制台）的存储设置](#)。
- e. （可选）对于 AWS IoT Analytics 集成，请执行以下操作。
  - i. 如果要使用 AWS IoT Analytics 来查询数据，请选择已启用 AWS IoT Analytics 数据存储。
  - ii. AWS IoT SiteWise 为您的数据存储生成一个名称，也可以输入其他名称。

AWS IoT SiteWise 自动在中创建数据存储 AWS IoT Analytics 以保存您的数据。要查询数据，您可以使用 AWS IoT Analytics 创建数据集。有关更多信息，请参阅《AWS IoT Analytics 用户指南》中的[使用 AWS IoT SiteWise 数据](#)。

- f. 选择保存。

在 AWS IoT SiteWise 存储部分中，冷层存储可以是以下值之一：

- 已启用- AWS IoT SiteWise 将您的数据复制到指定的 Amazon S3 存储桶。
- 启用- AWS IoT SiteWise 正在处理您的启用冷层存储的请求。此过程可能需要几分钟才能完成。
- Enable\_Failed — AWS IoT SiteWise 无法处理您的启用冷层存储的请求。如果您启用 AWS IoT SiteWise 向 Amazon Logs 发送 CloudWatch 日志，则可以使用这些日志来解决问题。有关更多信息，请参阅 [使用 Amazon CloudWatch 日志进行监控](#)。
- 已禁用 – 冷层存储已禁用。

## 配置冷层的存储设置 (AWS CLI)

以下过程演示如何配置存储设置以使用 AWS CLI 将数据复制到冷层。

### 使用配置存储设置 AWS CLI

1. 要将数据导出到您账户中的 Amazon S3 存储桶，请运行以下命令以配置存储设置。将###替换为包含 AWS IoT SiteWise 存储配置的文件名称。

```
aws iotsitewise put-storage-configuration --cli-input-json file://file-name.json
```

### Example AWS IoT SiteWise 存储配置

- 将 *bucket-name* 替换为您的 Amazon S3 存储桶的名称。
- 将##替换为您的 Amazon S3 前缀。
- *aws-account-id*替换为您的 AWS 账户 ID。
- 将####替换为允许 AWS IoT SiteWise 向 Amazon S3 发送数据的 Amazon S3 访问角色的名称。
- *retention-in-days*替换为大于或等于 30 天的整数。

```
{
  "storageType": "MULTI_LAYER_STORAGE",
  "multiLayerStorage": {
    "customerManagedS3Storage": {
      "s3ResourceArn": "arn:aws:s3::bucket-name/prefix/",
      "roleArn": "arn:aws:iam::aws-account-id:role/role-name"
    }
  },
  "retentionPeriod": {
    "numberOfDays": retention-in-days,
  }
}
```

```

    "unlimited": false
  }
}

```

### Note

- 您必须在存储配置和 IAM 策略中使用相同的 Amazon S3 AWS IoT SiteWise 存储桶名称。
- 确保您的角色具有以下示例中显示的权限。

Example 权限策略：

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:GetBucketLocation",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}

```

将 *bucket-name* 替换为您的 Amazon S3 存储桶的名称。

### Example 响应

```

{
  "storageType": "MULTI_LAYER_STORAGE",
  "retentionPeriod": {

```

```
    "numberOfDays": 100,  
    "unlimited": false  
  },  
  "configurationStatus": {  
    "state": "UPDATE_IN_PROGRESS"  
  }  
}
```

### Note

更新存储配置可能需要几分钟。AWS IoT SiteWise

- 若要检索存储配置信息，请运行以下命令。

```
aws iotsitewise describe-storage-configuration
```

### Example 响应

```
{  
  "storageType": "MULTI_LAYER_STORAGE",  
  "multiLayerStorage": {  
    "customerManagedS3Storage": {  
      "s3ResourceArn": "arn:aws:s3::DOC-EXAMPLE-BUCKET/torque/",  
      "roleArn": "arn:aws:iam::123456789012:role/SWAccessS3Role"  
    }  
  },  
  "retentionPeriod": {  
    "numberOfDays": 100,  
    "unlimited": false  
  },  
  "configurationStatus": {  
    "state": "ACTIVE"  
  },  
  "lastUpdateDate": "2021-03-30T15:54:14-07:00"  
}
```

- 要停止将数据导出到 Amazon S3 存储桶，请运行以下命令以配置存储设置。

```
aws iotsitewise put-storage-configuration --storage-type SITEWISE_DEFAULT_STORAGE
```

**Note**

默认情况下，您的数据仅存储在热门层中 AWS IoT SiteWise。

**Example 响应**

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "configurationStatus": {
    "state": "UPDATE_IN_PROGRESS"
  }
}
```

- 若要检索存储配置信息，请运行以下命令。

```
aws iotsitewise describe-storage-configuration
```

**Example 响应**

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "configurationStatus": {
    "state": "ACTIVE"
  },
  "lastUpdateDate": "2021-03-30T15:57:14-07:00"
}
```

**( 可选 ) 创建 AWS IoT Analytics 数据存储 (AWS CLI)**

AWS IoT Analytics 数据存储是一种可扩展且可查询的存储库，用于接收和存储数据。您可以使用 AWS IoT SiteWise 控制台或 AWS IoT Analytics API 创建 AWS IoT Analytics 数据存储以保存 AWS IoT SiteWise 数据。要查询数据，请使用创建数据集 AWS IoT Analytics。有关更多信息，请参阅《AWS IoT Analytics 用户指南》中的[使用 AWS IoT SiteWise 数据](#)。

以下步骤 AWS CLI 用于在中创建数据存储 AWS IoT Analytics。

若要创建数据集，运行以下命令。将 *file-name* 替换为包含数据存储配置的文件名称。

```
aws iotanalytics create-datastore --cli-input-json file://file-name.json
```

### Note

- 您必须指定现有 Amazon S3 存储桶的名称。如果没有 Amazon S3 存储桶，请先创建一个。有关更多信息，请参阅 Amazon S3 用户指南中的[创建您的第一个 S3 存储桶](#)。
- 您必须在存储配置、IAM 策略和 AWS IoT Analytics 数据 AWS IoT SiteWise 存储配置中使用相同的 Amazon S3 存储桶名称。

### Example AWS IoT Analytics 数据存储配置

将 *s3 #####data-store-name### AWS IoT Analytics #####* 和 Amazon S3 存储桶名称。

```
{
  "datastoreName": "data-store-name",
  "datastoreStorage": {
    "iotSiteWiseMultiLayerStorage": {
      "customerManagedS3Storage": {
        "bucket": "s3-bucket-name"
      }
    }
  },
  "retentionPeriod": {
    "numberOfDays": 90
  }
}
```

### Example 响应

```
{
  "datastoreName": "datastore_IoTSiteWise_demo",
  "datastoreArn": "arn:aws:iotanalytics:us-west-2:123456789012:datastore/
datastore_IoTSiteWise_demo",
  "retentionPeriod": {
    "numberOfDays": 90,
    "unlimited": false
  }
}
```

```
}
```

## 存储设置疑难解答

使用以下信息对存储配置进行故障诊断并解决问题。

### 问题

- [错误：存储桶不存在](#)
- [错误：对 Amazon S3 路径的访问被拒绝](#)
- [错误：无法代入角色 ARN](#)
- [错误：无法访问跨区域 Amazon S3 存储桶](#)

### 错误：存储桶不存在

解决方案：找 AWS IoT SiteWise 不到您的 Amazon S3 存储桶。确保您输入当前区域中现有 Amazon S3 存储桶的名称。

### 错误：对 Amazon S3 路径的访问被拒绝

解决方案：AWS IoT SiteWise 无法访问您的 Amazon S3 存储桶。执行以下操作：

- 确保您使用的 Amazon S3 存储桶与您在 IAM Policy 中指定的相同。
- 确保您的角色具有以下示例中显示的权限。

#### Example 权限策略

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:GetBucketLocation",
        "s3:ListBucket"
      ]
    }
  ],
}
```

```
    "Resource": [  
      "arn:aws:s3:::bucket-name",  
      "arn:aws:s3:::bucket-name/*"  
    ]  
  }  
]
```

将 *bucket-name* 替换为您的 Amazon S3 存储桶的名称。

## 错误：无法代入角色 ARN

解决方案：AWS IoT SiteWise 无法代表您担任 IAM 角色。确保您的角色信任以下服务：`iotsitewise.amazonaws.com`。有关更多信息，请参阅《IAM 用户指南》中的[我无法带入角色](#)。

## 错误：无法访问跨区域 Amazon S3 存储桶

解决方案：您指定的 Amazon S3 存储桶位于不同的 AWS 区域。确保您的 Amazon S3 存储桶和 AWS IoT SiteWise 资产位于同一区域。

## 冷层中保存的数据的文件路径和架构

AWS IoT SiteWise 通过复制时间序列（包括度量、指标、变换和聚合，以及资产和资产模型定义）将您的数据存储到冷层中。下面介绍发送到冷层的数据的文件路径和架构。

### 主题

- [设备数据（测量）](#)
- [指标、转换和聚合](#)
- [资产元数据](#)
- [资产层次结构元数据](#)
- [存储数据索引文件](#)

## 设备数据（测量）

AWS IoT SiteWise 每六小时将设备数据（测量值）导出到冷层一次。原始数据以 [Apache AVRO](#)（.avro）格式保存在冷层中。



## 文件路径

AWS IoT SiteWise 使用以下模板将设备数据（测量值）存储在冷层中。

```
{keyPrefix}/raw/startYear={startYear}/startMonth={startMonth}/startDay={startDay}/
seriesBucket={seriesBucket}/raw_{timeseriesId}_{startTimestamp}_{quality}.avro
```

Amazon S3 中原始数据的每个文件路径都包含以下组件。

路径组件	描述
keyPrefix	您在 AWS IoT SiteWise 存储配置中指定的 Amazon S3 前缀。Amazon S3 使用前缀作为存储桶中的文件夹名称。
raw	存储设备（测量）中的时间序列数据的文件夹。raw 文件夹保存在前缀文件夹中。
seriesBucket	<p>介于 00 和 ff 之间的十六进制数。此数字派生自 timeSeriesId 。此分区用于在 AWS IoT SiteWise 写入冷层时增加吞吐量。当您使用 Amazon Athena 运行查询时，您可以使用该分区进行精细分区以提高查询性能。</p> <p>资产元数据中的 seriesBucket 和 timeSeriesBucket 是相同的数字。</p>
startYear	与时间序列数据关联的独占开始时间的年份。
startMonth	与时间序列数据关联的独占开始时间的月份。
startDay	与时间序列数据关联的独占开始时间的月份中的某一天。
fileName	<p>文件名使用下划线 (_) 字符作为分隔符来分隔以下内容：</p> <ul style="list-style-type: none"> <li>raw 前缀。</li> <li>timeSeriesId 值。</li> </ul>

路径组件	描述
	<ul style="list-style-type: none"> <li>与时间序列数据关联的独占开始时间的纪元时间戳。</li> <li>数据质量。有效值：GOOD、BAD 和 UNCERTAIN。有关更多信息，请参阅 AWS IoT SiteWise API 参考中的 <a href="#">AssetProperty</a> 值。</li> </ul> <p>该文件通过使用 <a href="#">Snappy</a> 压缩以 .avro 格式保存。</p>

Example 冷层中原始数据的文件路径

```
keyPrefix/raw/startYear=2021/startMonth=1/startDay=2/seriesBucket=a2/
raw_7020c8e2-e6db-40fa-9845-ed0dd4c77d_95e63da7-d34e-43e1-
bc6f-1b490154b07a_1609577700_G00D.avro
```

## 字段

导出到冷层的原始数据的架构包含以下字段。

字段名称	支持的类型	默认类型	描述
seriesId	string	不适用	标识设备（度量）中的时间序列数据的 ID。您可以使用此字段在查询中联接原始数据和资产元数据。
timeInSeconds	long	不适用	采用 Unix 纪元时间格式的时间戳日期（以秒为单位）。分数纳秒数据由 offsetInNanos 提供。
offsetInNanos	long	不适用	与 timeInSeconds 的纳秒偏移量。

字段名称	支持的类型	默认类型	描述
quality	string	不适用	时间序列值的质量。
doubleValue	double 或 null	null	双精度 (浮点数) 类型的时间序列数据。
stringValue	string 或 null	null	字符串类型 (字符序列) 的时间序列数据。
integerValue	int 或 null	null	整数类型 (整数) 的时间序列数据。
booleanValue	boolean 或 null	null	布尔类型的时间序列数据 (true 或 false)。
jsonValue	string 或 null	null	JSON 类型的时间序列数据 (存储为字符串的复杂数据类型)。
recordVersion	long 或 null	null	记录的版本号。您可以使用版本号来选择最新记录。较新的记录具有较大的版本号。

### Example 冷层中的原始数据

```
{
  "seriesId": "e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a",
  "timeInSeconds": 1625675887,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "doubleValue": 0.75,
  "stringValue": null,
  "integerValue": null,
  "booleanValue": null,
  "jsonValue": null,
  "recordVersion": 1
},
{
  "seriesId": "e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a",
  "timeInSeconds": 1625675889,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "doubleValue": 0.69,
  "stringValue": null,
  "integerValue": null,
  "booleanValue": null,
  "jsonValue": null,
  "recordVersion": 2
},
{
  "seriesId": "e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a",
  "timeInSeconds": 1625675890,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "doubleValue": 0.66,
  "stringValue": null,
  "integerValue": null,
  "booleanValue": null,
  "jsonValue": null,
  "recordVersion": 3
}
```

```

{"seriesId":"e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-
bc6f-1b490154b07a","timeInSeconds":1625675891,"offsetInNanos":0,"quality":"GOOD","doubleValue":
{"double":0.92},"stringValue":null,"integerValue":null,"booleanValue":null,"jsonValue":null,"re
{"seriesId":"e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-
bc6f-1b490154b07a","timeInSeconds":1625675892,"offsetInNanos":0,"quality":"GOOD","doubleValue":
{"double":0.73},"stringValue":null,"integerValue":null,"booleanValue":null,"jsonValue":null,"re

```

## 指标、转换和聚合

AWS IoT SiteWise 每六小时将指标、转换和聚合导出到冷层一次。指标、转换和聚合以 [Apache AVRO](#) (.avro) 格式保存在冷层中。

### 文件路径

AWS IoT SiteWise 使用以下模板将指标、转换和聚合存储在冷层中。

```

{keyPrefix}/agg/startYear={startYear}/startMonth={startMonth}/startDay={startDay}/
seriesBucket={seriesBucket}/agg_{timeseriesId}_{startTimestamp}_{quality}.avro

```

Amazon S3 中指标、转换和聚合的每个文件路径都包含以下组件。

路径组件	描述
keyPrefix	您在 AWS IoT SiteWise 存储配置中指定的 Amazon S3 前缀。Amazon S3 使用前缀作为存储桶中的文件夹名称。
agg	用于存储来自指标的时间序列数据的文件夹。agg 文件夹保存在前缀文件夹中。
seriesBucket	介于 00 和 ff 之间的十六进制数。此数字派生自 timeSeriesId 。此分区用于在 AWS IoT SiteWise 写入冷层时增加吞吐量。当您使用 Amazon Athena 运行查询时，您可以使用该分区进行精细分区以提高查询性能。  资产元数据中的 seriesBucket 和 timeSeriesBucket 是相同的数字。
startYear	与时间序列数据关联的独占开始时间的年份。

路径组件	描述
startMonth	与时间序列数据关联的独占开始时间的月份。
startDay	与时间序列数据关联的独占开始时间的月份中的某一天。
fileName	<p>文件名使用下划线 (_) 字符作为分隔符来分隔以下内容：</p> <ul style="list-style-type: none"> <li>raw 前缀。</li> <li>timeSeriesId 值。</li> <li>与时间序列数据关联的独占开始时间的纪元时间戳。</li> <li>数据质量。有效值：GOOD、BAD 和 UNCERTAIN 。有关更多信息，请参阅 AWS IoT SiteWise API 参考中的 <a href="#">AssetProperty</a> 值。</li> </ul> <p>该文件通过使用 <a href="#">Snappy</a> 压缩以 .avro 格式保存。</p>

### Example 冷层中指标的文件路径

```
keyPrefix/agg/startYear=2021/startMonth=1/startDay=2/seriesBucket=a2/agg_7020c8e2-e6db-40fa-9845-ed0dddd4c77d_95e63da7-d34e-43e1-bc6f-1b490154b07a_1609577700_GOOD.avro
```

### 字段

导出到冷层的指标、转换和聚合的架构包含以下字段。

字段名称	支持的类型	默认类型	描述
seriesId	string	不适用	标识来自设备、指标或转换的时序数据的 ID。您可以使用此字

字段名称	支持的类型	默认 类型	描述
			段在查询中联接原始数据和资产元数据。
timeInSeconds	long	不适用	采用 Unix 纪元时间格式的时间戳日期 ( 以秒为单位 )。分数纳秒数据由 offsetInNanos 提供。
offsetInNanos	long	不适用	与 timeInSeconds 的纳秒偏移量。
quality	string	不适用	筛选资产数据所依据的质量。
resolution	string	不适用	聚合数据的时间间隔。
count	double 或 null	null	给定变量在当前时间间隔内的数据点总数。
average	double 或 null	null	给定变量在当前时间间隔内的平均值。
min	double 或 null	null	给定变量在当前时间间隔内的最小值。
max	boolean 或 null	null	给定变量在当前时间间隔内的最大值。
sum	string 或 null	null	给定变量在当前时间间隔内的值总和。

字段名称	支持的类型	默认类型	描述
recordVersion	long 或 null	null	记录的版本号。您可以使用版本号来选择最新记录。较新的记录具有较大的版本号。

### Example 冷层中的指标数据

```
{
  "seriesId": "f74c2828-5317-4df3-ba16-6d41b5bcb531",
  "timeInSeconds": 1637334060,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "resolution": "16.0",
  "min": {
    "double": 1.0
  },
  "max": {
    "double": 31.0
  },
  "sum": {
    "double": 496.0
  },
  "recordVersion": null
}
{"seriesId": "f74c2828-5317-4df3-ba16-6d41b5bcb531",
timeInSeconds": 1637334120,
offsetInNanos": 0,
quality": "GOOD",
resolution": "16.0",
min": {
  double": 32.0
},
max": {
  double": 60.0
},
sum": {
  double": 1334.0
},
recordVersion": null
}
{"seriesId": "f74c2828-5317-4df3-ba16-6d41b5bcb531",
timeInSeconds": 1637334540,
offsetInNanos": 0,
quality": "GOOD",
resolution": "16.0",
min": {
  double": 1.0
},
max": {
  double": 31.0
},
sum": {
  double": 496.0
},
recordVersion": null
}
{"seriesId": "f74c2828-5317-4df3-ba16-6d41b5bcb531",
timeInSeconds": 1637334600,
offsetInNanos": 0,
quality": "GOOD",
resolution": "16.0",
min": {
  double": 32.0
},
max": {
  double": 60.0
},
sum": {
  double": 1334.0
},
recordVersion": null
}
{"seriesId": "f74c2828-5317-4df3-ba16-6d41b5bcb531",
timeInSeconds": 1637335020,
offsetInNanos": 0,
quality": "GOOD",
resolution": "16.0",
min": {
  double": 1.0
},
max": {
  double": 31.0
},
sum": {
  double": 496.0
},
recordVersion": null
}
```

## 资产元数据

首次启用 AWS IoT SiteWise 将数据导出到冷层时，资产元数据将导出到冷层。初始配置完成后，只有在您更改资产模型定义或资产定义时，才会将资产元数据 AWS IoT SiteWise 导出到该层。资产元数据以换行符分隔的 JSON (.ndjson) 格式保存在冷层中。

### 文件路径

AWS IoT SiteWise 使用以下模板将资产元数据存储于冷层中。

```
{keyPrefix}/asset_metadata/asset_{assetId}.ndjson
```

冷层中资产元数据的每个文件路径都包含以下组件。

路径组件	描述
keyPrefix	您在的存储配置中指定的 Amazon AWS IoT SiteWise S3 前缀。Amazon S3 使用前缀作为存储桶中的文件夹名称。
asset_metadata	存储资产元数据的文件夹。asset_metadata 文件夹保存在前缀文件夹中。
fileName	文件名使用下划线 (_) 字符作为分隔符来分隔以下内容： <ul style="list-style-type: none"> <li>asset 前缀。</li> <li>assetId 值。</li> </ul> <p>该文件以 .ndjson 格式保存。</p>

Example 较冷层中资产元数据的文件路径

```
keyPrefix/asset_metadata/asset_35901915-d476-4dca-8637-d9ed4df939ed.ndjson
```

## 字段

导出到冷层的资产元数据的模式包含以下字段。

字段名称	描述
assetId	资产的 ID。
assetName	资产的名称。
assetExternalId	资产的外部 ID。
assetModelId	用于创建该资产的资产模型的 ID。



字段名称	描述
assetModelName	资产模型的名称。
assetModelExternalId	资产模型的外部 ID。
assetPropertyId	资产属性的 ID。
assetPropertyName	资产属性的名称。
assetPropertyExternalId	资产属性的外部 ID。
assetPropertyDataType	资产属性的数据类型。
assetPropertyUnit	资产属性的单位 ( 例如 , Newtons 和 RPM ) 。
assetPropertyAlias	标识资产属性的别名 , 例如 OPC-UA 服务器数据流路径 ( 例如 , /company/windfarm/3/turbine/7/temperature ) 。
timeSeriesId	标识来自设备、指标或转换的时序数据的 ID。您可以使用此字段在查询中联接原始数据和资产元数据。
timeSeriesBucket	<p>介于 00 和 ff 之间的十六进制数。此数字派生自 timeSeriesId 。此分区用于在 AWS IoT SiteWise 写入冷层时增加吞吐量。当您使用 Amazon Athena 运行查询时 , 您可以使用该分区进行精细分区以提高查询性能。</p> <p>原始数据的文件路径中的 timeSeriesBucket 和 seriesBucket 是相同的数字。</p>
assetCompositeModelId	复合模型的 ID。
assetCompositeModelExternalId	复合模型的外部 ID。
assetCompositeModelDescription	复合模型的描述。

字段名称	描述
assetCompositeModelName	复合模型的名称。
assetCompositeModelType	复合模型的类型。对于警报复合模型，这种类型是 AWS/ALARM 。
assetCreationDate	资产的创建日期，以 Unix 纪元时间为单位。
assetLastUpdateDate	资产上次更新的日期，以 Unix 纪元为单位。
assetStatusErrorCode	错误代码。
assetStatusErrorMessage	错误消息。
assetStatusState	资产的当前状态。

### Example 冷层中的资产元数据

```

{"assetId":"7020c8e2-e6db-40fa-9845-
ed0dddd4c77d","assetExternalId":null,"assetName":"Wind Turbine Asset
  2","assetModelId":"ec1d924f-f07d-444f-b072-
e2994c165d35","assetModelExternalId":null,"assetModelName":"Wind
  Turbine Asset Model","assetPropertyId":"95e63da7-d34e-43e1-
bc6f-1b490154b07a","assetPropertyExternalId":null,"assetPropertyName":"Temperature","assetPrope
Washington/Seattle/WT2/temp","timeSeriesId":"7020c8e2-e6db-40fa-9845-
ed0dddd4c77d_95e63da7-d34e-43e1-
bc6f-1b490154b07a","timeSeriesBucket":"f6","assetArn":null,"assetCompositeModelDescription":null
  {"assetId":"7020c8e2-e6db-40fa-9845-
ed0dddd4c77d","assetExternalId":null,"assetName":"Wind Turbine Asset
  2","assetModelId":"ec1d924f-f07d-444f-b072-
e2994c165d35","assetModelExternalId":null,"assetModelName":"Wind Turbine Asset
  Model","assetPropertyId":"c706d54d-4c11-42dc-9a01-63662fc697b4","assetPropertyExternalId":null
Washington/Seattle/WT2/pressure","timeSeriesId":"7020c8e2-e6db-40fa-9845-
ed0dddd4c77d_c706d54d-4c11-42dc-9a01-63662fc697b4","timeSeriesBucket":"1e","assetArn":null,"ass
  {"assetId":"7020c8e2-e6db-40fa-9845-
ed0dddd4c77d","assetExternalId":null,"assetName":"Wind Turbine Asset
  2","assetModelId":"ec1d924f-f07d-444f-b072-
e2994c165d35","assetModelExternalId":null,"assetModelName":"Wind
  Turbine Asset Model","assetPropertyId":"8cf1162f-dead-4fbe-b468-

```

```
c8e24cde9f50", "assetPropertyExternalId": null, "assetPropertyName": "Max
  Temperature", "assetPropertyDataType": "DOUBLE", "assetPropertyUnit": null, "assetPropertyAlias": nu
e6db-40fa-9845-ed0ddd4c77d_8cf1162f-dead-4fbe-b468-
c8e24cde9f50", "timeSeriesBucket": "d7", "assetArn": null, "assetCompositeModelDescription": null, "as

  {"assetId": "3a5f2a22-3b37-4332-9c1c-404ea1d73fab", "assetExternalId": null, "assetName": "BatchAss
ebc75e75e827", "assetModelExternalId": null, "assetModelName": "FlashTestAssetModelDouble", "assetPr
b410-
ab401a9176ed", "assetPropertyExternalId": null, "assetPropertyName": "measurementProperty", "assetPr
ae89-
ff316f5ff8aa", "timeSeriesBucket": "af", "assetArn": null, "assetCompositeModelDescription": null, "as
```

## 资产层次结构元数据

当您首次启用 AWS IoT SiteWise 在冷层中保存数据时，资产层次结构元数据将导出到冷层。初始配置完成后，只有在对资产模型或资产定义进行更改时，才会将资产层次结构元数据 AWS IoT SiteWise 导出到冷层。资产层次结构元数据以换行符分隔的 JSON (.ndjson) 格式保存在冷层中。

通过调用 [DescribeAsset](#) API 可以检索层次结构、目标资产或源资产的外部标识符。

### 文件路径

AWS IoT SiteWise 使用以下模板将资产层次结构元数据存储到冷层中。

```
{keyPrefix}/asset_hierarchy_metadata/{parentAssetId}_{hierarchyId}.ndjson
```

冷层中资产层次结构元数据的每个文件路径都包含以下组件。

路径组件	描述
keyPrefix	您在 AWS IoT SiteWise 存储配置中指定的 Amazon S3 前缀。Amazon S3 使用前缀作为存储桶中的文件夹名称。
asset_hierarchy_metadata	存储资产层次结构元数据的文件夹。asset_hierarchy_metadata 文件夹保存在前缀文件夹中。

路径组件	描述
fileName	<p>文件名使用下划线 (_) 字符作为分隔符来分隔以下内容：</p> <ul style="list-style-type: none"> <li>parentAssetId 值。</li> <li>hierarchyId 值。</li> </ul> <p>该文件以 .ndjson 格式保存。</p>

Example 冷层中资产层次结构元数据的文件路径

keyPrefix/asset\_hierarchy\_metadata/35901915-d476-4dca-8637-d9ed4df939ed\_c5b3ced8-589a-48c7-9998-cdcccfc9747a0.ndjson

## 字段

导出到冷层的资产层次结构元数据的架构包含以下字段。

字段名称	描述
sourceAssetId	此资产关系中源资产的 ID。
targetAssetId	此资产关系中目标资产的 ID。
hierarchyId	层次结构的 ID。
associationType	<p>此资产关系的关联类型。</p> <p>值必须为 CHILD。目标资产是源资产的子资产。</p>

Example 冷层中的资产层次结构元数据

```
{
  "sourceAssetId": "80388e72-2284-44fb-9c89-bfbaf0dfedd2",
  "targetAssetId": "2b866c25-0c74-4750-bdf5-b73683c8a2a2",
  "hierarchyId": "bbed9f59-0412-4585-a61d-6044db526aee",
  "associationType": "CHILD"
}
```

```
{"sourceAssetId":"80388e72-2284-44fb-9c89-bfbaf0dfedd2","targetAssetId":"6b51246e-984d-460d-bc0b-470ea47d1e31","hierarchyId":"bbed9f59-0412-4585-a61d-6044db526aee","associationType":"CHILD"}
```

若要在冷层中查看您的数据

1. 导航到 [Amazon S3 控制台](#)。
2. 在导航窗格中，选择存储桶，然后选择您的 Amazon S3 存储桶。
3. 导航到包含原始数据、资产元数据或资产层次结构元数据的文件夹。
4. 选择文件，然后从操作中选择下载。

## 存储数据索引文件

AWS IoT SiteWise 使用这些文件来优化数据查询性能。它们显示在您的 Amazon S3 存储桶中，但您不需要使用它们。

### 文件路径

AWS IoT SiteWise 使用以下模板将数据索引文件存储在冷层中。

```
keyPrefix/index/series=timeseriesId/startYear=startYear/startMonth=startMonth/  
startDay=startDay/index_timeseriesId_startTimestamp_quality
```

Example 数据存储索引文件的文件路径

```
keyPrefix/index/series=7020c8e2-e6db-40fa-9845-ed0dddd4c77d_95e63da7-d34e-43e1-bc6f-1b490154b07a/startYear=2022/startMonth=02/startDay=03/  
index_7020c8e2-e6db-40fa-9845-ed0dddd4c77d_95e63da7-d34e-43e1-bc6f-1b490154b07a_1643846400_G00D
```

# 安全性 AWS IoT SiteWise

云安全 AWS 是重中之重。作为 AWS 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。

安全是双方共同承担 AWS 的责任。[责任共担模式](#)将其描述为云的安全性和云中的安全性：

- 云安全 — AWS 负责保护在 AWS 云中运行 AWS 服务的基础架构。AWS 还为您提供可以安全使用的服务。作为[AWS 合规计划](#)的一部分，第三方审计师定期测试和验证我们安全的有效性。要了解适用的合规计划 AWS IoT SiteWise，请参阅[AWS 按合规计划划分的范围内 AWS 服务 \(按合分\)](#)。
- 云端安全-您的责任由您使用的 AWS 服务决定。您还需要对其他因素负责，包括您的数据的敏感性、您的公司的要求以及适用的法律法规。

本文档可帮助您了解在使用时如何应用分担责任模型 AWS IoT SiteWise。以下主题向您介绍如何进行配置 AWS IoT SiteWise 以满足您的安全和合规性目标。您还将学习如何使用其他 AWS 服务来帮助您监控和保护您的 AWS IoT SiteWise 资源。

## 主题

- [中的数据保护 AWS IoT SiteWise](#)
- [数据加密](#)
- [的身份和访问管理 AWS IoT SiteWise](#)
- [合规性验证 AWS IoT SiteWise](#)
- [韧性在 AWS IoT SiteWise](#)
- [基础设施安全 AWS IoT SiteWise](#)
- [配置和漏洞分析](#)
- [VPC 端点](#)
- [以下方面的安全最佳实践 AWS IoT SiteWise](#)

## 中的数据保护 AWS IoT SiteWise

分 AWS [担责任模型](#)适用于中的数据保护 AWS IoT SiteWise。如本模型所述 AWS，负责保护运行所有内容的全球基础架构 AWS Cloud。您负责维护对托管在此基础设施上的内容的控制。您还负责您所

使用的 AWS 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅[数据隐私常见问题](#)。有关欧洲数据保护的信息，请参阅 AWS 安全性博客 上的 [AWS 责任共担模式和 GDPR](#) 博客文章。

出于数据保护目的，我们建议您保护 AWS 账户 凭证并使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 设置个人用户。这样，每个用户只获得履行其工作职责所需的权限。我们还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证 (MFA)。
- 使用 SSL/TLS 与资源通信。AWS 我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用设置 API 和用户活动日志 AWS CloudTrail。
- 使用 AWS 加密解决方案以及其中的所有默认安全控件 AWS 服务。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果您在 AWS 通过命令行界面或 API 进行访问时需要经过 FIPS 140-2 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅 [《美国联邦信息处理标准 \( FIPS \) 第 140-2 版》](#)。

我们强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如名称字段）。这包括您使用控制台、API AWS IoT SiteWise 或 SDK 或以其他 AWS 服务 方式使用控制台 AWS CLI、API 或 AWS SDK 的情况。在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供网址，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

## 主题

- [互连网络流量隐私保护](#)

## 互连网络流量隐私保护

AWS IoT SiteWise 与本地应用程序（例如 SiteWise Edge 网关）之间的连接通过传输层安全 (TLS) 连接进行保护。有关更多信息，请参阅 [传输中加密](#)。

AWS IoT SiteWise 不支持 AWS 区域内可用区之间的连接或 AWS 账户之间的连接。

您一次只能在一个区域配置 IAM Identity Center。SiteWise Monitor 连接到您为 IAM 身份中心配置的区域。这意味着您可以使用一个区域访问 IAM Identity Center，但在任何区域创建门户。

# 数据加密

数据加密是指保护传输中的数据（当数据在 SiteWise 边缘网关和服务器之间往返传输时）和静态数据（存储在本地设备或 AWS 服务中时）。AWS IoT SiteWise 您可以使用传输层安全性 (TLS) 保护传输中的数据，或使用客户端加密保护静态数据。

## Note

AWS IoT SiteWise 边缘处理会公开托管在 E SiteWise dge 网关中并可通过本地网络访问的 API。这些 API 通过由 AWS IoT SiteWise Edge 连接器拥有的服务器证书支持的 TLS 连接公开。对于客户端身份验证，这些 API 使用访问控制密码。服务器证书私钥和访问控制密码都存储在磁盘上。AWS IoT SiteWise 边缘处理依赖文件系统加密来确保这些静态凭证的安全。

有关服务器端加密和客户端加密的更多信息，请查看下面列出的主题。

## 主题

- [静态加密](#)
- [传输中加密](#)
- [密钥管理](#)

## 静态加密

AWS IoT SiteWise 将您的数据存储存储在 AWS 云端和 AWS IoT SiteWise 边缘网关上。

### AWS 云端静态数据

AWS IoT SiteWise 将数据存储存储在默认情况下对静态数据进行加密的其他 AWS 服务中。静态加密与 AWS Key Management Service (AWS KMS) 集成，用于管理加密密钥，该密钥用于加密您的资产属性值和聚合中的值 AWS IoT SiteWise。您可以选择使用客户托管密钥加密 AWS IoT SiteWise 中的资产属性值和聚合值。您可以通过 AWS KMS 创建、管理和查看您的加密密钥。

您可以选择一个 AWS 拥有的密钥来加密您的数据，也可以选择客户管理的密钥来加密您的资产属性值和汇总值：

### 工作方式

静态加密与集成，AWS KMS 用于管理用于加密数据的加密密钥。



- AWS 拥有的密钥 — 默认加密密钥。AWS IoT SiteWise 拥有这把钥匙。您无法在 AWS 账户中查看此密钥。您也无法在 AWS CloudTrail 日志中查看对密钥的操作。使用此密钥不会产生额外的费用。
- 客户托管密钥 - 此密钥存储在您的账户中，由您创建、拥有和管理。您对 KMS 密钥拥有完全控制权。需 AWS KMS 支付额外费用。

## AWS 拥有的密钥

AWS 拥有的密钥 未存储在您的账户中。它们是 KMS 密钥集合的一部分，这些密钥 AWS 拥有并管理，可在多个 AWS 账户中使用。AWS 服务可以 AWS 拥有的密钥 用来保护您的数据。

您无法查看、管理 AWS 拥有的密钥、使用或审核其使用情况。但是无需执行任何工作或更改任何计划即可保护用于加密数据的密钥。

如果您使用，则无需支付月费或使用费 AWS 拥有的密钥，也不会计入您账户的 AWS KMS 配额。

## 客户管理密钥

客户管理密钥是在您的账户中由您创建、拥有和管理的 KMS 密钥。您可以完全控制这些 KMS 密钥，例如：

- 制定和维护密钥策略、IAM 策略和授权
- 启用和禁用 KMS 密钥
- 轮换 KMS 密钥的加密材料
- 添加标签
- 创建引用 KMS 密钥的别名
- 安排删除 KMS 密钥

您还可以使用 CloudTrail 和 Amaz CloudWatch on Logs 来跟踪 AWS KMS 代表您 AWS IoT SiteWise 发送的请求。

如果您使用的是客户托管密钥，则需要授予对存储在您账户中的 KMS 密钥的 AWS IoT SiteWise 访问权限。AWS IoT SiteWise 使用信封加密和密钥层次结构来加密数据。您的 AWS KMS 加密密钥用于加密此密钥层次结构的根密钥。有关更多信息，请参阅 [开发人员指南](#) 中的 AWS Key Management Service 信封加密。

以下示例策略授予代表您创建客户托管密钥的 AWS IoT SiteWise 权限。创建密钥时，需要允许 `kms:CreateGrant` 和 `kms:DescribeKey` 操作。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1603902045292",
      "Action": [
        "kms:CreateGrant",
        "kms:DescribeKey"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

您创建的授权的加密上下文使用您的 `aws:iotsitewise:subscriberId` 和账户 ID。

## SiteWise 边缘网关上的静态数据

AWS IoT SiteWise 网关在本地文件系统中存储以下数据：

- OPC-UA 源配置信息
- 来自已连接 OPC-UA 源的 OPC-UA 数据流路径集
- 当 SiteWise Edge 网关失去与互联网的连接时，工业数据会被缓存

SiteWise 边缘网关在上运行 AWS IoT Greengrass。AWS IoT Greengrass 依靠 Unix 文件权限和全盘加密（如果启用）来保护核心上的静态数据。您负责确保文件系统和设备的安全性。

但是，AWS IoT Greengrass 确实会加密从 Secrets Manager 检索到的 OPC-UA 服务器密钥的本地副本。有关更多信息，请参阅 AWS IoT Greengrass Version 1 开发人员指南中的[密钥加密](#)。

有关 AWS IoT Greengrass 内核静态加密的更多信息，请参阅《AWS IoT Greengrass Version 1 开发人员指南》中的[静态加密](#)。

## 传输中加密

AWS IoT SiteWise 在传输数据时有三种通信模式：

- [通过互联网](#)-本地设备（包括 SiteWise 边缘网关）之间的通信 AWS IoT SiteWise 已加密。

- [通过本地网络](#) — SiteWise 应用程序网关和 SiteWise Edge 网关之间的 OpsHub 通信始终处于加密状态。浏览器中运行的 SiteWise 监控应用程序与 SiteWise Edge 网关之间的通信始终处于加密状态。可以对 SiteWise 边缘网关和 OPC-UA 源之间的通信进行加密。
- [SiteWise 边缘网关上的组件之间](#)-边 SiteWise 边缘网关上 AWS IoT Greengrass 组件之间的通信未加密。

## 主题

- [通过 Internet 传输的数据](#)
- [通过本地网络传输的数据](#)
- [SiteWise 边缘网关上的本地组件之间传输的数据](#)

## 通过 Internet 传输的数据

AWS IoT SiteWise 使用传输层安全 (TLS) 加密互联网上的所有通信。发送到 AWS 云端的所有数据均使用 MQTT 或 HTTPS 协议通过 TLS 连接发送，因此默认情况下它是安全的。SiteWise 在上 AWS IoT Greengrass 运行的 Edge 网关和属性值通知使用 AWS IoT 传输安全模型。有关更多信息，请参阅 AWS IoT 开发人员指南中的[传输安全](#)。

## 通过本地网络传输的数据

SiteWise 边缘网关遵循 OPC-UA 规范，用于与本地 OPC-UA 源进行通信。您负责将源配置为使用加密传输中数据的消息安全模式。

如果您选择签名消息安全模式，则在 SiteWise Edge 网关和源之间传输的数据将经过签名但不加密。如果您选择签名并加密消息安全模式，则会对在 SiteWise Edge 网关和源之间传输的数据进行签名和加密。有关配置源的更多信息，请参阅[配置数据源](#)。

边缘控制台应用程序与 Edge SiteWise 网关之间的通信始终通过 TLS 加密。SiteWise Edge 网关上的 SiteWise Edge 连接器生成并存储自签名证书，以便能够与边缘控制台建立 TLS 连接以供 AWS IoT SiteWise 应用程序使用。在将应用程序连接到 SiteWise Edge 网关之前，您需要将此证书从 Edge 网关复制到 SiteWise Edge 控制台以供 AWS IoT SiteWise 应用程序。这可确保 AWS IoT SiteWise 应用程序的边缘控制台能够验证其是否已连接到您可信的 SiteWise Edge 网关。

除了用于保密和服务器真实性的 TLS 之外，SiteWise Edge 还使用 Sigv4 协议来确定 Edge 控制台应用程序的真实性。SiteWise Edge 网关上的 SiteWise Edge 连接器接受并存储密码，以便能够验证来自边缘控制台应用程序、浏览器中运行的 SiteWise 监控应用程序以及其他基于 AWS IoT SiteWise SDK 的客户端的传入连接。

有关生成密码和服务器证书的更多信息，请参阅 [the section called “管理 SiteWise Edge 网关”](#)

## SiteWise 边缘网关上的本地组件之间传输的数据

SiteWise 边缘网关在上运行 AWS IoT Greengrass，它不会加密在 AWS IoT Greengrass 核心上本地交换的数据，因为数据不会离开设备。这包括 AWS IoT SiteWise 连接器等 AWS IoT Greengrass 组件之间的通信。想要了解更多信息，请参阅 AWS IoT Greengrass Version 1 开发人员指南中的 [核心设备上的数据](#)。

## 密钥管理

### AWS IoT SiteWise 云密钥管理

默认情况下，AWS IoT SiteWise AWS 托管式密钥 用于保护您在 AWS 云中的数据。您可以更新设置，使用客户托管密钥加密 AWS IoT SiteWise 中的某些数据。您可以通过 AWS Key Management Service (AWS KMS) 创建、管理和查看您的加密密钥。

AWS IoT SiteWise 支持服务器端加密，使用存储在中的客户托管密钥 AWS KMS 来加密以下数据：

- 资产属性值
- 汇总值

#### Note

其他数据和资源使用默认加密进行加密，密钥由管理 AWS IoT SiteWise。此密钥存储在 AWS IoT SiteWise 账户中。

有关更多信息，请参阅 [什么是 AWS Key Management Service ?](#) 在《AWS Key Management Service 开发人员指南》中。

### 使用由客户托管的密钥启用加密

要将客户托管密钥与一起使用 AWS IoT SiteWise，您需要更新 AWS IoT SiteWise 设置。

### 使用 KMS 密钥启用加密

1. 导航到 [AWS IoT SiteWise 控制台](#)。

2. 选择账户设置，选择编辑 打开编辑账户设置页面。
3. 对于加密密钥类型，选择 选择其他 AWS KMS 密钥。这对存储在 AWS KMS 中的客户托管密钥启用加密。

**Note**

目前，您只能对资产属性值和聚合值使用客户托管密钥加密。

4. 从以下选项中选择您的 KMS 密钥：
  - 使用现有 KMS 密钥 - 从列表中选择您的 KMS 密钥别名。
  - 要创建新的 KMS 密钥，请选择创建 AWS KMS 密钥。

**Note**

这将打开 AWS KMS 控制面板。有关创建 KMS 密钥的更多信息，请参阅 AWS Key Management Service 开发人员指南中的 [创建密钥](#)。

5. 选择保存以保存您的设置。

## SiteWise 边缘网关密钥管理

SiteWise 边缘网关在上运行 AWS IoT Greengrass，AWS IoT Greengrass 核心设备使用公钥和私钥向 AWS 云端进行身份验证并加密本地密钥，例如 OPC-UA 身份验证密钥。有关更多信息，请参阅 AWS IoT Greengrass Version 1 开发人员指南中的 [密钥管理](#)。

## 的身份和访问管理 AWS IoT SiteWise

AWS Identity and Access Management (IAM) AWS 服务 可帮助管理员安全地控制对 AWS 资源的访问权限。IAM 管理员控制谁可以进行身份验证（登录）和授权（有权限）使用 AWS IoT SiteWise 资源。您可以使用 IAM AWS 服务，无需支付额外费用。

### 主题

- [受众](#)
- [使用身份进行身份验证](#)
- [如何 AWS IoT SiteWise 与 IAM 配合使用](#)
- [AWS 的托管策略 AWS IoT SiteWise](#)

- [将服务相关角色用于 AWS IoT SiteWise](#)
- [设置 AWS IoT Events 警报权限](#)
- [防止跨服务混淆代理](#)
- [对 AWS IoT SiteWise 身份和访问进行故障排除](#)

## 受众

您的使用方式 AWS Identity and Access Management (IAM) 会有所不同，具体取决于您所做的工作 AWS IoT SiteWise。

**服务用户**-如果您使用 AWS IoT SiteWise 服务完成工作，则管理员会为您提供所需的凭证和权限。当你使用更多 AWS IoT SiteWise 功能来完成工作时，你可能需要额外的权限。了解如何管理访问权限有助于您向管理员请求适合的权限。如果您无法访问 AWS IoT SiteWise 中的特征，请参阅 [对 AWS IoT SiteWise 身份和访问进行故障排除](#)。

**服务管理员**-如果您负责公司的 AWS IoT SiteWise 资源，则可能拥有完全访问权限 AWS IoT SiteWise。您的工作是确定您的服务用户应访问哪些 AWS IoT SiteWise 功能和资源。然后，您必须向 IAM 管理员提交请求以更改服务用户的权限。请查看该页面上的信息以了解 IAM 的基本概念。要详细了解您的公司如何将 IAM 与配合使用 AWS IoT SiteWise，请参阅[如何 AWS IoT SiteWise 与 IAM 配合使用](#)。

**IAM 管理员**：如果您是 IAM 管理员，您可能希望了解如何编写策略以管理对 AWS IoT SiteWise 的访问权限的详细信息。要查看您可以在 IAM 中使用的 AWS IoT SiteWise 基于身份的策略示例，请参阅[AWS IoT SiteWise 基于身份的策略示例](#)

## 使用身份进行身份验证

身份验证是您 AWS 使用身份凭证登录的方式。您必须以 IAM 用户身份或通过担 AWS 账户根用户任 IAM 角色进行身份验证（登录 AWS）。

您可以使用通过身份源提供的凭据以 AWS 联合身份登录。AWS IAM Identity Center（IAM Identity Center）用户、贵公司的单点登录身份验证以及您的 Google 或 Facebook 凭据就是联合身份的示例。当您以联合身份登录时，您的管理员以前使用 IAM 角色设置了身份联合验证。当你使用联合访问 AWS 时，你就是在间接扮演一个角色。

根据您的用户类型，您可以登录 AWS Management Console 或 AWS 访问门户。有关登录的更多信息 AWS，请参阅《AWS 登录 用户指南》[中的如何登录到您 AWS 账户的](#)。

如果您 AWS 以编程方式访问，则会 AWS 提供软件开发套件 (SDK) 和命令行接口 (CLI)，以便使用您的凭据对请求进行加密签名。如果您不使用 AWS 工具，则必须自己签署请求。有关使用推荐的方法自行签署请求的更多信息，请参阅 IAM 用户指南中的[签署 AWS API 请求](#)。

无论使用何种身份验证方法，您可能需要提供其他安全信息。例如，AWS 建议您使用多重身份验证 (MFA) 来提高账户的安全性。要了解更多信息，请参阅《AWS IAM Identity Center 用户指南》中的[多重身份验证](#)和《IAM 用户指南》中的[在 AWS 中使用多重身份验证 \(MFA\)](#)。

## AWS 账户 root 用户

创建时 AWS 账户，首先要有一个登录身份，该身份可以完全访问账户中的所有资源 AWS 服务和资源。此身份被称为 AWS 账户 root 用户，使用您创建账户时使用的电子邮件地址和密码登录即可访问该身份。强烈建议您不要使用根用户执行日常任务。保护好根用户凭证，并使用这些凭证来执行仅根用户可以执行的任务。有关需要您以根用户身份登录的任务的完整列表，请参阅 IAM 用户指南 中的 [需要根用户凭证的任务](#)。

## IAM 用户和群组

[IAM 用户](#)是您 AWS 账户 内部对个人或应用程序具有特定权限的身份。在可能的情况下，我们建议使用临时凭证，而不是创建具有长期凭证（如密码和访问密钥）的 IAM 用户。但是，如果您有一些特定的使用场景需要长期凭证以及 IAM 用户，建议您轮换访问密钥。有关更多信息，请参阅《IAM 用户指南》中的[对于需要长期凭证的使用场景定期轮换访问密钥](#)。

[IAM 组](#)是一个指定一组 IAM 用户的身份。您不能使用组的身份登录。您可以使用组来一次性为多个用户指定权限。如果有大量用户，使用组可以更轻松地管理用户权限。例如，您可能具有一个名为 IAMAdmins 的组，并为该组授予权限以管理 IAM 资源。

用户与角色不同。用户唯一地与某个人员或应用程序关联，而角色旨在让需要它的任何人代入。用户具有永久的长期凭证，而角色提供临时凭证。要了解更多信息，请参阅《IAM 用户指南》中的[何时创建 IAM 用户（而不是角色）](#)。

## IAM 角色

[IAM 角色](#)是您内部具有特定权限 AWS 账户 的身份。它类似于 IAM 用户，但与特定人员不关联。您可以 AWS Management Console 通过[切换角色在中临时担任 IAM 角色](#)。您可以通过调用 AWS CLI 或 AWS API 操作或使用自定义 URL 来代入角色。有关使用角色的方法的更多信息，请参阅《IAM 用户指南》中的 [使用 IAM 角色](#)。

具有临时凭证的 IAM 角色在以下情况下很有用：

- 联合用户访问 – 要向联合身份分配权限，请创建角色并为角色定义权限。当联合身份进行身份验证时，该身份将与角色相关联并被授予由此角色定义的权限。有关联合身份验证的角色的信息，请参阅《IAM 用户指南》中的 [为第三方身份提供商创建角色](#)。如果您使用 IAM Identity Center，则需要配置权限集。为控制您的身份在进行身份验证后可以访问的内容，IAM Identity Center 将权限集与 IAM 中的角色相关联。有关权限集的信息，请参阅《AWS IAM Identity Center 用户指南》中的 [权限集](#)。
- 临时 IAM 用户权限 – IAM 用户可代入 IAM 用户或角色，以暂时获得针对特定任务的不同权限。
- 跨账户存取 – 您可以使用 IAM 角色以允许不同账户中的某个人（可信主体）访问您的账户中的资源。角色是授予跨账户访问权限的主要方式。但是，对于某些资源 AWS 服务，您可以将策略直接附加到资源（而不是使用角色作为代理）。要了解用于跨账户访问的角色和基于资源的策略之间的差别，请参阅《IAM 用户指南》中的 [IAM 角色与基于资源的策略有何不同](#)。
- 跨服务访问 — 有些 AWS 服务 使用其他 AWS 服务服务中的功能。例如，当您在某个服务中进行调用时，该服务通常会在 Amazon EC2 中运行应用程序或在 Amazon S3 中存储对象。服务可能会使用发出调用的主体的权限、使用服务角色或使用服务相关角色来执行此操作。
  - 转发访问会话 (FAS) — 当您使用 IAM 用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS 使用调用委托人的权限以及 AWS 服务 向下游服务发出请求的请求。AWS 服务只有当服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两个操作的权限。有关发出 FAS 请求时的策略详情，请参阅[转发访问会话](#)。
  - 服务角色 - 服务角色是服务代表您在您的账户中执行操作而分派的 [IAM 角色](#)。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的[创建向 AWS 服务委派权限的角色](#)。
  - 服务相关角色-服务相关角色是一种与服务相关联的服务角色。AWS 服务服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户，并且归服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。
- 在 Amazon EC2 上运行的应用程序 — 您可以使用 IAM 角色管理在 EC2 实例上运行并发出 AWS CLI 或 AWS API 请求的应用程序的临时证书。这优先于在 EC2 实例中存储访问密钥。要向 EC2 实例分配 AWS 角色并使其可供其所有应用程序使用，您需要创建附加到该实例的实例配置文件。实例配置文件包含角色，并使 EC2 实例上运行的程序能够获得临时凭证。有关更多信息，请参阅《IAM 用户指南》中的 [使用 IAM 角色为 Amazon EC2 实例上运行的应用程序授予权限](#)。

要了解是使用 IAM 角色还是 IAM 用户，请参阅《IAM 用户指南》中的[何时创建 IAM 角色（而不是用户）](#)。



## 如何 AWS IoT SiteWise 与 IAM 配合使用

在使用 AWS Identity and Access Management (IAM) 管理访问权限之前 AWS IoT SiteWise，您应该了解哪些可用的 IAM 功能 AWS IoT SiteWise。

IAM 功能	支持 AWS IoT SiteWise
<a href="#">具有资源级权限的基于身份的策略</a>	是
<a href="#">策略操作</a>	是
<a href="#">策略资源</a>	是
<a href="#">策略条件键</a>	是
基于资源的策略	否
访问控制列表 (ACL)	否
<a href="#">基于标签的授权 (ABAC)</a>	是
<a href="#">临时凭证</a>	是
<a href="#">转发访问会话 (FAS)</a>	是
<a href="#">服务相关角色</a>	是
<a href="#">服务角色</a>	是

要全面了解如何 AWS IoT SiteWise 和其他 AWS 服务与 IAM 配合使用，请参阅 IAM 用户指南中的与 IAM [配合使用的AWS 服务](#)。

### 目录

- [AWS IoT SiteWise IAM 角色](#)
  - [将临时证书与 AWS IoT SiteWise](#)

- [的转发访问会话 \(FAS\) AWS IoT SiteWise](#)
- [服务相关角色](#)
- [服务角色](#)
- [在 AWS IoT SiteWise 中选择 IAM 角色](#)
- [基于 AWS IoT SiteWise 标签的授权](#)
- [AWS IoT SiteWise 基于身份的策略](#)
  - [策略操作](#)
    - [BatchPutAssetPropertyValue 授权](#)
  - [策略资源](#)
  - [策略条件键](#)
  - [示例](#)
- [AWS IoT SiteWise 基于身份的策略示例](#)
  - [策略最佳实践](#)
  - [使用 AWS IoT SiteWise 控制台](#)
  - [允许用户查看他们自己的权限](#)
  - [允许用户将数据提取到一个层次结构中的资产](#)
  - [根据标签查看 AWS IoT SiteWise 资产](#)
- [使用策略管理访问](#)
  - [基于身份的策略](#)
  - [基于资源的策略](#)
  - [访问控制列表 \(ACL\)](#)
  - [其他策略类型](#)
  - [多个策略类型](#)

## AWS IoT SiteWise IAM 角色

[IAM 角色](#)是 AWS 账户 中具有特定权限的实体。

将临时证书与 AWS IoT SiteWise

可以使用临时凭证进行联合身份验证登录，分派 IAM 角色或分派跨账户角色。您可以通过调用 AWS STS API 操作（例如 [AssumeRole](#) 或 [GetFederation令牌](#)）来获取临时安全证书。

AWS IoT SiteWise 支持使用临时证书。

SiteWise Monitor 支持联合用户访问门户。门户用户使用其 IAM Identity Center 或 IAM 凭证进行身份验证。

**⚠ Important**

用户或角色必须具有登录门户的 `iotsitewise:DescribePortal` 权限。

当用户登录门户时，M SiteWise onitor 会生成一个提供以下权限的会话策略：

- 对您账户 AWS IoT SiteWise 中的资产和资产数据的只读访问权限，该门户的角色为其提供访问权限。
- 对该门户中的项目（用户对这些项目具有管理员（项目拥有者）或只读（项目查看者）访问权限）的访问权限。

有关联合身份门户用户权限的更多信息，请参阅[将服务角色用于 AWS IoT SiteWise Monitor](#)。

的转发访问会话 (FAS) AWS IoT SiteWise

支持转发访问会话 (FAS)

是

当您使用 IAM 用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS 使用调用委托人的权限以及 AWS 服务 向下游服务发出请求的请求。AWS 服务只有当服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两个操作的权限。有关发出 FAS 请求时的策略详情，请参阅[转发访问会话](#)。

服务相关角色

[服务相关角色](#) 允许 AWS 服务访问其他服务中的资源以代表您完成操作。服务相关角色显示在您的 AWS 账户 中，并由该服务拥有。IAM 管理员可以查看但不能编辑服务相关角色的权限。

AWS IoT SiteWise 支持服务相关角色。有关创建或管理 AWS IoT SiteWise 服务相关角色的详细信息，请参阅 [将服务相关角色用于 AWS IoT SiteWise](#)。

## 服务角色

此功能允许服务代表您担任[服务角色](#)。此角色允许服务访问其他服务中的资源以代表您完成操作。服务角色显示在您的 AWS 账户中，并由该账户拥有。这意味着，IAM 管理员可以更改该角色的权限。但是，这样做可能会中断服务的功能。

AWS IoT SiteWise 使用服务角色允许 M SiteWise onitor 门户用户代表您访问您的某些 AWS IoT SiteWise 资源。有关更多信息，请参阅[将服务角色用于 AWS IoT SiteWise Monitor](#)。

在中创建 AWS IoT Events 警报模型之前，您必须拥有所需的权限 AWS IoT SiteWise。有关更多信息，请参阅[设置 AWS IoT Events 警报权限](#)。

在 AWS IoT SiteWise 中选择 IAM 角色

在中创建 portal 资源时 AWS IoT SiteWise，必须选择一个角色才能允许 Monitor SiteWise r 门户的联合用户 AWS IoT SiteWise 代表您进行访问。如果您之前创建过服务角色，则会 AWS IoT SiteWise 为您提供可供选择的角色列表。否则，可以在创建门户时创建具有所需权限的角色。选择一个允许访问资产和资产数据的角色非常重要。有关更多信息，请参阅[将服务角色用于 AWS IoT SiteWise Monitor](#)。

## 基于 AWS IoT SiteWise 标签的授权

您可以为 AWS IoT SiteWise 资源附加标签或在请求中传递标签 AWS IoT SiteWise。要基于标签控制访问，您需要使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 条件键在策略的[条件元素](#)中提供标签信息。有关标记 AWS IoT SiteWise 资源的更多信息，请参阅[为资源添加 AWS IoT SiteWise 标签](#)。

要查看基于身份的策略（用于根据资源上的标签来限制对该资源的访问）的示例，请参阅[根据标签查看 AWS IoT SiteWise 资产](#)。

## AWS IoT SiteWise 基于身份的策略

IAM 策略允许您控制谁能在中做什么 AWS IoT SiteWise。您可以决定允许或不允许哪些操作，并为这些操作设置具体条件。例如，您可以制定规则，规定谁可以查看或更改中的信息 AWS IoT SiteWise。AWS IoT SiteWise 支持特定的操作、资源和条件键。要了解在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的[IAM JSON 策略元素参考](#)。

## 策略操作

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

JSON 策略的 Action 元素描述可用于在策略中允许或拒绝访问的操作。策略操作通常与关联的 AWS API 操作同名。有一些例外情况，例如没有匹配 API 操作的仅限权限 操作。还有一些操作需要在策略中执行多个操作。这些附加操作称为相关操作。

在策略中包含操作以授予执行关联操作的权限。

正在执行的策略操作在操作前 AWS IoT SiteWise 使用以下前缀:iotsitewise:. 例如，要授予某人BatchPutAssetPropertyValue通过 API 操作将资产属性数据上传到 AWS IoT SiteWise 的权限，您可以将该iotsitewise:BatchPutAssetPropertyValue操作包含在他们的策略中。策略声明必须包含Action或NotAction元素。AWS IoT SiteWise 定义了自己的一组操作，这些操作描述了您可以使用此服务执行的任务。

要在单个语句中指定多项 操作，请使用逗号将它们隔开，如下所示。

```
"Action": [  
  "iotsitewise:action1",  
  "iotsitewise:action2"  
]
```

您也可以使用通配符 (\*) 指定多个操作。例如，要指定以单词 Describe 开头的所有操作，请包括以下操作。

```
"Action": "iotsitewise:Describe*"
```

要查看 AWS IoT SiteWise 操作列表，请参阅 IAM 用户指南 AWS IoT SiteWise 中的[定义操作](#)。

### BatchPutAssetPropertyValue 授权

AWS IoT SiteWise 以一种不寻常的方式授权访问 “[BatchPutAssetPropertyValue](#)” 操作。对于大多数操作，当您允许或拒绝访问时，如果未授予权限，则该操作会返回错误。使用BatchPutAssetPropertyValue，您可以在单个 API 请求中向不同的资产和资产属性发送多个数据条目。AWS IoT SiteWise 独立授权每个数据输入。对于请求中未通过授权的任何单个条目，AWS IoT SiteWise 请在返回的错误列表AccessDeniedException中包含一个。AWS IoT SiteWise 接收任何授权并成功的条目的数据，即使同一请求中的另一个条目失败也是如此。

#### Important

在将数据提取到数据流之前，请执行以下操作：

- 如果您使用属性别名来标识数据流，则对time-series资源进行授权。

- 如果您使用资产 ID 来标识包含关联资产属性的资产，则对asset资源进行授权。

## 策略资源

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体 可以对什么资源 执行操作，以及在什么条件下执行。

Resource JSON 策略元素指定要向其应用操作的一个或多个对象。语句必须包含 Resource 或 NotResource 元素。作为最佳实践，请使用其 [Amazon 资源名称 \( ARN \)](#) 指定资源。对于支持特定资源类型 ( 称为资源级权限 ) 的操作，您可以执行此操作。

对于不支持资源级权限的操作 ( 如列出操作 ) ，请使用通配符 (\*) 指示语句应用于所有资源。

```
"Resource": "*" 
```

每个 IAM policy 语句适用于您使用资源的 ARN 指定的资源。ARN 具有以下一般语法。

```
arn:${Partition}:${Service}:${Region}:${Account}:${ResourceType}/${ResourcePath}
```

有关 ARN 格式的更多信息，请参阅 [Amazon 资源名称 \(ARN\) 和 AWS 服务命名空间](#)。

例如，要在语句中指定 ID 为 a1b2c3d4-5678-90ab-cdef-22222EXAMPLE 的资产，请使用以下 ARN。

```
"Resource": "arn:aws:iotsitewise:region:123456789012:asset/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE" 
```

要指定属于特定账户的所有数据流，请使用通配符 (\*)：

```
"Resource": "arn:aws:iotsitewise:region:123456789012:time-series/*" 
```

要指定属于特定账户的所有资产，请使用通配符 (\*)：

```
"Resource": "arn:aws:iotsitewise:region:123456789012:asset/*" 
```

某些 AWS IoT SiteWise 操作 ( 例如创建资源的操作 ) 无法对特定资源执行。在这些情况下，您必须使用通配符 (\*)。

```
"Resource": "*"
```

要在单个语句中指定多个资源，请使用逗号分隔 ARN。

```
"Resource": [  
  "resource1",  
  "resource2"  
]
```

要查看 AWS IoT SiteWise 资源类型及其 ARN 的列表，请参阅 IAM 用户指南 AWS IoT SiteWise 中的[由定义的资源](#)。要了解您可以在哪些操作中指定每个资源的 ARN，请参阅[AWS IoT SiteWise 定义的操作](#)。

### 策略条件键

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

在 Condition 元素 ( 或 Condition 块 ) 中，可以指定语句生效的条件。Condition 元素是可选的。您可以创建使用[条件运算符](#) ( 例如，等于或小于 ) 的条件表达式，以使策略中的条件与请求中的值相匹配。

如果您在一个语句中指定多个 Condition 元素，或在单个 Condition 元素中指定多个键，则 AWS 使用逻辑 AND 运算评估它们。如果您为单个条件键指定多个值，则使用逻辑 OR 运算来 AWS 评估条件。在授予语句的权限之前必须满足所有的条件。

在指定条件时，您也可以使用占位符变量。例如，只有在使用 IAM 用户名标记 IAM 用户时，您才能为其授予访问资源的权限。有关更多信息，请参阅《IAM 用户指南》中的[IAM 策略元素：变量和标签](#)。

AWS 支持全局条件密钥和特定于服务的条件密钥。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的[AWS 全局条件上下文密钥](#)。

#### Important

许多条件键是特定于某个资源的，而某些 API 操作会使用多个资源。如果您使用条件键编写策略语句，请使用该语句的 Resource 元素指定要将该条件键应用于的资源。如果您没有这样做，该策略可能会完全阻止用户执行操作，因为针对未应用条件键的资源的条件检查失败。如果您不想指定资源，或者如果您已将策略的 Action 元素编写为包含多个 API 操作，则必须使用 ...IfExists 条件类型以确保对不使用条件键的资源忽略条件键。有关更多信息，请参阅[... IfExists IAM 用户指南中的条件](#)。

AWS IoT SiteWise 定义自己的条件键集，还支持使用一些全局条件键。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的[AWS 全局条件上下文密钥](#)。

## AWS IoT SiteWise 条件键

条件键	描述	类型
<code>iotsitewise:isAssociatedWithAssetProperty</code>	<p>数据流是否与资产属性相关联。使用此条件键根据数据流的关联资产属性的存在来定义权限。</p> <p>示例值: <code>true</code></p>	String
<code>iotsitewise:assetHierarchyPath</code>	<p>资产的层次结构路径，它是一个资产 ID 字符串，其中每个 ID 由一个正斜杠分隔开。使用此条件键可根据您账户中所有资产的层次结构子集来定义权限。</p> <p>示例值: <code>/a1b2c3d4-5678-90ab-cdef-2222EXAMPLE/a1b2c3d4-5678-90ab-cdef-6666EXAMPLE</code></p>	String
<code>iotsitewise:propertyId</code>	<p>资产属性的 ID。使用此条件键可根据指定的资产模型属性来定义权限。此条件键适用于该模型的所有资产。</p> <p>示例值: <code>a1b2c3d4-5678-90ab-cdef-3333EXAMPLE</code></p>	String
<code>iotsitewise:childAssetId</code>	<p>作为子级关联到其他资产的资产 ID。使用此条件键可根据子资产来定义权限。要根据父资</p>	String



条件键	描述	类型
	<p>产定义权限，请使用策略语句的资源部分。</p> <p>示例值: a1b2c3d4-5678-90ab-cdef-66666EXAMPLE</p>	
<code>iotsitewise:iam</code>	<p>列出访问策略时 IAM 身份的 ARN。使用此条件键可定义 IAM 身份的访问策略权限。</p> <p>示例值: arn:aws:iam::123456789012:user/JohnDoe</p>	字符串，Null
<code>iotsitewise:propertyAlias</code>	<p>标识资产属性或数据流的别名。使用此条件键可根据别名来定义权限。</p>	String
<code>iotsitewise:user</code>	<p>列出访问策略时的 IAM Identity Center 用户 ID。使用此条件键可定义 IAM Identity Center 用户的访问策略权限。</p> <p>示例值: a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE</p>	字符串，Null
<code>iotsitewise:group</code>	<p>列出访问策略时的 IAM Identity Center 组 ID。使用此条件键可定义 IAM Identity Center 组的访问策略权限。</p> <p>示例值: a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbbEXAMPLE</p>	字符串，Null

条件键	描述	类型
<code>iotsitewise:portal</code>	<p>访问策略中的门户 ID。使用此条件键可根据门户来定义访问策略权限。</p> <p>示例值: a1b2c3d4-5678-90ab-cdef-7777EXAMPLE</p>	字符串, Null
<code>iotsitewise:project</code>	<p>访问策略中的项目 ID 或控制面板的项目 ID。使用此条件键可根据项目来定义控制面板或访问策略权限。</p> <p>示例值: a1b2c3d4-5678-90ab-cdef-8888EXAMPLE</p>	字符串, Null

要了解您可以使用条件键的操作和资源，请参阅[操作定义者 AWS IoT SiteWise](#)。

## 示例

要查看 AWS IoT SiteWise 基于身份的策略的示例，请参阅[AWS IoT SiteWise 基于身份的策略示例](#)

### AWS IoT SiteWise 基于身份的策略示例

默认情况下，实体（用户和角色）无权创建或修改 AWS IoT SiteWise 资源。他们也无法使用 AWS Management Console、AWS Command Line Interface (AWS CLI) 或 AWS API 执行任务。要调整权限，AWS Identity and Access Management (IAM) 管理员必须执行以下操作：

1. 创建 IAM 策略，向用户和角色授予对其所需资源执行特定 API 操作的权限。
2. 将这些策略附加到需要这些权限的用户或群组。

要了解如何使用这些示例 JSON 策略文档创建 IAM 基于身份的策略，请参阅《IAM 用户指南》中的[在 JSON 选项卡上创建策略](#)。

## 主题

- [策略最佳实践](#)
- [使用 AWS IoT SiteWise 控制台](#)
- [允许用户查看他们自己的权限](#)
- [允许用户将数据提取到一个层次结构中的资产](#)
- [根据标签查看 AWS IoT SiteWise 资产](#)

## 策略最佳实践

基于身份的策略决定了某人是否可以在您的账户中创建、访问或删除 AWS IoT SiteWise 资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下准则和建议：

- 开始使用 AWS 托管策略并转向最低权限权限 — 要开始向用户和工作负载授予权限，请使用为许多常见用例授予权限的 AWS 托管策略。它们在你的版本中可用 AWS 账户。我们建议您通过定义针对您的用例的 AWS 客户托管策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的 [AWS 托管策略](#) 或 [工作职能的 AWS 托管策略](#)。
- 应用最低权限 – 在使用 IAM 策略设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 [IAM 中的策略和权限](#)。
- 使用 IAM 策略中的条件进一步限制访问权限 – 您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果服务操作是通过特定 AWS 服务的（例如）使用的，则也可以使用条件来授予对服务操作的访问权限 AWS CloudFormation。有关更多信息，请参阅《IAM 用户指南》中的 [IAM JSON 策略元素：条件](#)。
- 使用 IAM Access Analyzer 验证您的 IAM 策略，以确保权限的安全性和功能性 – IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM 策略语言 (JSON) 和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助您制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的 [IAM Access Analyzer 策略验证](#)。
- 需要多重身份验证 (MFA)-如果 AWS 账户您的场景需要 IAM 用户或根用户，请启用 MFA 以提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的 [配置受 MFA 保护的 API 访问](#)。

有关 IAM 中的最佳实践的更多信息，请参阅《IAM 用户指南》中的 [IAM 中的安全最佳实践](#)。

## 使用 AWS IoT SiteWise 控制台

要访问 AWS IoT SiteWise 控制台，您需要一组基本权限。这些权限允许您查看和管理中 AWS IoT SiteWise 资源的详细信息 AWS 账户。

如果您制定了过于严格的策略，那么对于使用该策略的用户或角色（实体），控制台可能无法按预期运行。为确保这些实体仍然可以使用 AWS IoT SiteWise 控制台，请将[AWSIoTSiteWiseConsoleFullAccess](#)托管策略附加到它们或为这些实体定义等效权限。有关更多信息，请参阅《IAM 用户指南》中的[为用户添加权限](#)。

如果实体仅使用 AWS Command Line Interface (CLI) 或 AWS IoT SiteWise API，而不使用控制台，则不需要这些最低权限。在这种情况下，只需让他们访问其 API 任务所需的特定操作即可。

### 允许用户查看他们自己的权限

该示例说明了如何创建策略，以允许 IAM 用户查看附加到其用户身份的内联和托管式策略。此策略包括在控制台上或使用 AWS CLI 或 AWS API 以编程方式完成此操作的权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",

```

```

        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
    ],
    "Resource": "*"
}
]
}

```

允许用户将数据提取到一个层次结构中的资产

在此示例中，您希望授予用户从根资源开始向特定资产层次结构中的所有资产属性写入数据的 AWS 账户 权限 `a1b2c3d4-5678-90ab-cdef-2222EXAMPLE`。此策略向用户授予 `iotsitewise:BatchPutAssetPropertyValue` 权限。此策略使用 `iotsitewise:assetHierarchyPath` 条件键来限制对其层次结构路径与资产或其后代匹配的资产的访问。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PutAssetPropertyValuesForHierarchy",
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "arn:aws:iotsitewise:*:*:asset/*",
      "Condition": {
        "StringLike": {
          "iotsitewise:assetHierarchyPath": [
            "/a1b2c3d4-5678-90ab-cdef-2222EXAMPLE",
            "/a1b2c3d4-5678-90ab-cdef-2222EXAMPLE/*"
          ]
        }
      }
    }
  ]
}

```

根据标签查看 AWS IoT SiteWise 资产

使用基于身份的策略中的条件根据标签控制对 AWS IoT SiteWise 资源的访问权限。此示例说明如何创建允许查看资产的策略。不过，仅在资产标签 `Owner` 具有该用户的用户名值时授予权限。此策略还授予在控制台上完成此操作的权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListAllAssets",
      "Effect": "Allow",
      "Action": [
        "iotsitewise:ListAssets",
        "iotsitewise:ListAssociatedAssets"
      ],
      "Resource": "*"
    },
    {
      "Sid": "DescribeAssetIfOwner",
      "Effect": "Allow",
      "Action": "iotsitewise:DescribeAsset",
      "Resource": "arn:aws:iotsitewise:*:*:asset/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Owner": "${aws:username}"
        }
      }
    }
  ]
}
```

将此策略附加到您账户中的用户。如果名为的用户 richard-roe 尝试查看 AWS IoT SiteWise 资产，则必须对该资产进行标记 `Owner=richard-roe` 或 `owner=richard-roe`。否则，Richard 将被拒绝访问。条件标签密钥名称不区分大小写。因此，同时 `Owner` 匹配 `Owner` 和 `owner`。有关更多信息，请参阅 IAM 用户指南 中的 [IAM JSON 策略元素：条件](#)。

## 使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略是其中的一个对象 AWS，当与身份或资源关联时，它会定义其权限。AWS 在委托人（用户、root 用户或角色会话）发出请求时评估这些策略。策略中的权限确定是允许还是拒绝请求。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的结构和内容的更多信息，请参阅《IAM 用户指南》中的 [JSON 策略概览](#)。

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

默认情况下，用户和角色没有权限。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。管理员随后可以向角色添加 IAM 策略，用户可以代入角色。

IAM 策略定义操作的权限，无关乎您使用哪种方法执行操作。例如，假设您有一个允许 `iam:GetRole` 操作的策略。拥有该策略的用户可以从 AWS Management Console AWS CLI、或 AWS API 获取角色信息。

## 基于身份的策略

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的[创建 IAM 策略](#)。

基于身份的策略可以进一步归类为内联策略或托管式策略。内联策略直接嵌入单个用户、组或角色中。托管策略是独立的策略，您可以将其附加到中的多个用户、群组 and 角色 AWS 账户。托管策略包括 AWS 托管策略和客户托管策略。要了解如何在托管式策略和内联策略之间进行选择，请参阅《IAM 用户指南》中的[在托管式策略与内联策略之间进行选择](#)。

## 基于资源的策略

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Simple Storage Service ( Amazon S3 ) 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中[指定主体](#)。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

基于资源的策略是位于该服务中的内联策略。您不能在基于资源的策略中使用 IAM 中的 AWS 托管策略。

## 访问控制列表 (ACL)

访问控制列表 ( ACL ) 控制哪些主体 ( 账户成员、用户或角色 ) 有权访问资源。ACL 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

Amazon S3 和 Amazon VPC 就是支持 ACL 的服务示例。AWS WAF 要了解有关 ACL 的更多信息，请参阅《Amazon Simple Storage Service 开发人员指南》中的[访问控制列表 \( ACL \) 概览](#)。

## 其他策略类型

AWS 支持其他不太常见的策略类型。这些策略类型可以设置更常用的策略类型向您授予的最大权限。

- 权限边界 - 权限边界是一个高级功能，用于设置基于身份的策略可以为 IAM 实体 ( IAM 用户或角色 ) 授予的最大权限。您可为实体设置权限边界。这些结果权限是实体基于身份的策略及其权限边界的交集。在 Principal 中指定用户或角色的基于资源的策略不受权限边界限制。任一项策略中的显式拒绝将覆盖允许。有关权限边界的更多信息，请参阅《IAM 用户指南》中的 [IAM 实体的权限边界](#)。
- 服务控制策略 (SCP)-SCP 是 JSON 策略，用于指定组织或组织单位 (OU) 的最大权限。AWS Organizations AWS Organizations 是一项用于对您的企业拥有的多 AWS 账户 项进行分组和集中管理的 服务。如果在组织内启用了所有功能，则可对任意或全部账户应用服务控制策略 (SCP)。SCP 限制成员账户中的实体 ( 包括每个 AWS 账户根用户实体 ) 的权限。有关 Organizations 和 SCP 的更多信息，请参阅《AWS Organizations 用户指南》中的 [SCP 的工作原理](#)。
- 会话策略 - 会话策略是当您以编程方式为角色或联合用户创建临时会话时作为参数传递的高级策略。结果会话的权限是用户或角色的基于身份的策略和会话策略的交集。权限也可以来自基于资源的策略。任一项策略中的显式拒绝将覆盖允许。有关更多信息，请参阅《IAM 用户指南》中的 [会话策略](#)。

## 多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解在涉及多种策略类型时如何 AWS 确定是否允许请求，请参阅 IAM 用户指南中的 [策略评估逻辑](#)。

## AWS 的托管策略 AWS IoT SiteWise

使用 AWS 托管策略简化向用户、群组和角色添加权限的过程，而不是自己编写策略。[创建为您的团队提供精确权限的 IAM 客户托管策略](#)需要时间和专业知识。为了更快地进行设置，可以考虑将我们的 AWS 托管策略用于常见用例。在您的中查找 AWS 托管策略 AWS 账户。有关 AWS 托管策略的更多信息，请参阅《IAM 用户指南》中的 [AWS 托管策略](#)。

AWS 服务负责更新和维护 AWS 托管策略，这意味着您无法修改这些策略的权限。有时，AWS IoT SiteWise 可能会添加权限以适应新功能，从而影响附加策略的所有身份。随着新服务或功能的推出，此类更新很常见。但是，永远不会移除权限，从而确保您的设置保持不变。

此外，还 AWS 支持跨多个服务的工作职能的托管策略。例如，ReadOnly访问 AWS 管理策略提供对所有 AWS 服务和资源的只读访问权限。当服务启动一项新功能时，AWS 会为新操作和资源添加只读权限。有关工作职能策略的列表和说明，请参阅 IAM 用户指南中的 [适用于工作职能的 AWS 托管策略](#)。

### AWS 托管策略：AWSIoTSiteWiseReadOnlyAccess

使用AWSIoTSiteWiseReadOnlyAccess AWS 托管策略允许只读访问权限 AWS IoT SiteWise。



您可以将 `AWSIoTSiteWiseReadOnlyAccess` 策略附加到 IAM 身份。

## 服务级别权限

此策略提供对的只读访问权限 AWS IoT SiteWise。此策略中不包含其他服务权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:Describe*",
        "iotsitewise:List*",
        "iotsitewise:BatchGet*",
        "iotsitewise:Get*"
      ],
      "Resource": "*"
    }
  ]
}
```

## AWS 托管策略：AWSServiceRoleForIoTSiteWise

`AWSServiceRoleForIoTSiteWise` 角色获得下列权限使用 `AWSServiceRoleForIoTSiteWise` 策略。本政策：

- AWS IoT SiteWise 允许部署 SiteWise Edge 网关（在上运行 AWS IoT Greengrass）。
- AWS IoT SiteWise 允许执行日志记录。
- AWS IoT SiteWise 允许对 AWS IoT TwinMaker 数据库运行元数据搜索查询。

如果您 AWS IoT SiteWise 使用的是单个用户账户，则该 `AWSServiceRoleForIoTSiteWise` 角色将在您的 IAM 账户中创建 `AWSServiceRoleForIoTSiteWise` 策略，并将其附加到的 `AWSServiceRoleForIoTSiteWise` [服务相关角色](#)。AWS IoT SiteWise

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSiteWiseReadGreenGrass",
```

```

"Effect": "Allow",
"Action": [
  "greengrass:GetAssociatedRole",
  "greengrass:GetCoreDefinition",
  "greengrass:GetCoreDefinitionVersion",
  "greengrass:GetGroup",
  "greengrass:GetGroupVersion"
],
"Resource": "*"
},
{
  "Sid": "AllowSiteWiseAccessLogGroup",
  "Effect": "Allow",
  "Action": [
    "logs:CreateLogGroup",
    "logs:DescribeLogGroups"
  ],
  "Resource": "arn:aws:logs:*:*:log-group:/aws/iotsitewise*"
},
{
  "Sid": "AllowSiteWiseAccessLog",
  "Effect": "Allow",
  "Action": [
    "logs:CreateLogStream",
    "logs:DescribeLogStreams",
    "logs:PutLogEvents"
  ],
  "Resource": "arn:aws:logs:*:*:log-group:/aws/iotsitewise*:log-stream:*"
},
{
  "Sid": "AllowSiteWiseAccessSiteWiseManagedWorkspaceInTwinMaker",
  "Effect": "Allow",
  "Action": [
    "iottwinmaker:GetWorkspace",
    "iottwinmaker:ExecuteQuery"
  ],
  "Resource": "arn:aws:iottwinmaker:*:*:workspace/*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "iottwinmaker:linkedServices": [
        "IOTSITWISE"
      ]
    }
  }
}

```

```

}
]
}

```

## AWS IoT SiteWise AWS 托管策略的更新

您可以查看有关 AWS 托管策略更新的详细信息 [AWS IoT SiteWise](#)，从该服务开始跟踪更改时开始。要获得有关此页面变更的自动提醒，请订阅“[AWS IoT SiteWise 文档历史记录](#)”页面上的 RSS feed。

更改	描述	日期
<a href="#">AWSServiceRoleForIoTSiteWise</a> – 对现有策略的更新	AWS IoT SiteWise 现在可以对 AWS IoT TwinMaker 数据库运行元数据搜索查询。	2023 年 11 月 6 日
<a href="#">AWSIoTSiteWiseReadOnlyAccess</a> – 更新了现有策略	AWS IoT SiteWise 添加了新的策略前缀 BatchGet*，使您可以执行批量读取操作。	2022 年 9 月 16 日
<a href="#">AWSIoTSiteWiseReadOnlyAccess</a> ：新策略	AWS IoT SiteWise 添加了授予只读访问权限的新策略 AWS IoT SiteWise。	2021 年 11 月 24 日
AWS IoT SiteWise 开始跟踪更改	AWS IoT SiteWise 开始跟踪其 AWS 托管策略的更改。	2021 年 11 月 24 日

## 将服务相关角色用于 AWS IoT SiteWise

AWS IoT SiteWise 使用 AWS Identity and Access Management (IAM) [服务相关角色](#)。服务相关角色是一种与之直接关联的 IAM 角色的独特类型。AWS IoT SiteWise 服务相关角色由服务预定义 [AWS IoT SiteWise](#)，包括该服务代表您调用其他 AWS 服务所需的所有权限。

服务相关角色 [AWS IoT SiteWise](#) 通过自动包含所有必要的权限来简化配置。AWS IoT SiteWise 定义其服务相关角色的权限，除非另有定义，否则 AWS IoT SiteWise 只能担任其角色。定义的权限包括信任策略和权限策略。而且该权限策略不能附加到任何其他 IAM 实体。

只有在首先删除相关资源后，您才能删除服务相关角色。这样可以保护您的 [AWS IoT SiteWise](#) 资源，因为您不会无意中删除访问资源的权限。

有关支持服务相关角色的其他服务的信息，请参阅[可与 IAM 搭配使用的AWS 服务](#)，并查找服务相关角色列表中为是的服务。选择是和链接，查看该服务的服务相关角色文档。

## 主题

- [AWS IoT SiteWise的服务相关角色权限](#)
- [为 AWS IoT SiteWise创建服务相关角色](#)
- [为 AWS IoT SiteWise编辑服务相关角色](#)
- [删除 AWS IoT SiteWise的服务相关角色](#)
- [AWS IoT SiteWise 服务相关角色支持的区域](#)
- [将服务角色用于 AWS IoT SiteWise Monitor](#)

## AWS IoT SiteWise的服务相关角色权限

AWS IoT SiteWise 使用名AWSServiceRoleForIoTSiteWise为的服务相关角色。AWS IoT SiteWise 使用此服务相关角色部署 SiteWise Edge 网关（在上运行 AWS IoT Greengrass）和执行日志记录。

AWSServiceRoleForIoTSiteWise服务相关角色使用具有以下权限的AWSServiceRoleForIoTSiteWise策略。本政策：

- AWS IoT SiteWise 允许部署 SiteWise Edge 网关（在上运行AWS IoT Greengrass）。
- AWS IoT SiteWise 允许执行日志记录。
- AWS IoT SiteWise 允许对 AWS IoT TwinMaker 数据库运行元数据搜索查询。

有关中允许的操作的更多信息AWSServiceRoleForIoTSiteWise，请参阅中的[AWS 托管策略 AWS IoT SiteWise](#)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSiteWiseReadGreenGrass",
      "Effect": "Allow",
      "Action": [
        "greengrass:GetAssociatedRole",
        "greengrass:GetCoreDefinition",
        "greengrass:GetCoreDefinitionVersion",
        "greengrass:GetGroup",
        "greengrass:GetGroupVersion"
      ]
    }
  ]
}
```

```

    ],
    "Resource": "*"
  },
  {
    "Sid": "AllowSiteWiseAccessLogGroup",
    "Effect": "Allow",
    "Action": [
      "logs:CreateLogGroup",
      "logs:DescribeLogGroups"
    ],
    "Resource": "arn:aws:logs:*:*:log-group:/aws/iotsitewise*"
  },
  {
    "Sid": "AllowSiteWiseAccessLog",
    "Effect": "Allow",
    "Action": [
      "logs:CreateLogStream",
      "logs:DescribeLogStreams",
      "logs:PutLogEvents"
    ],
    "Resource": "arn:aws:logs:*:*:log-group:/aws/iotsitewise*:log-stream:*"
  },
  {
    "Sid": "AllowSiteWiseAccessSiteWiseManagedWorkspaceInTwinMaker",
    "Effect": "Allow",
    "Action": [
      "iottwinmaker:GetWorkspace",
      "iottwinmaker:ExecuteQuery"
    ],
    "Resource": "arn:aws:iottwinmaker:*:*:workspace/*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "iottwinmaker:linkedServices": [
          "IOTSITewise"
        ]
      }
    }
  }
]
}

```

您可以使用日志来监控 SiteWise Edge 网关并对其进行故障排除。有关更多信息，请参阅 [监控 SiteWise 边缘网关日志](#)。

为允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务相关角色，首先配置权限。有关更多信息，请参阅《IAM 用户指南》中的[服务相关角色权限](#)。

## 为 AWS IoT SiteWise 创建服务相关角色

您无需手动创建服务相关角色。当您在 AWS IoT SiteWise 控制台中执行以下操作时，AWS IoT SiteWise 会为您创建服务相关角色。

- 创建 Greengrass V1 网关。
- 配置日志选项。
- 在执行查询横幅中选择“选择加入”按钮。

如果您删除该服务相关角色，然后需要再次创建，您可以使用相同流程在账户中重新创建此角色。当您在 AWS IoT SiteWise 控制台中执行任何操作时，AWS IoT SiteWise 会再次为您创建服务相关角色。

您也可以使用 IAM 控制台或 API 为 AWS IoT SiteWise 创建服务相关角色。

- 要在 IAM 控制台中执行此操作，请使用 `AWSServiceRoleForIoTSiteWise` 策略创建一个角色并与建立信任关系 `iotsitewise.amazonaws.com`。
- 要使用 AWS CLI 或 IAM API 执行此操作，请使用 `arn:aws:iam::aws:policy/aws-service-role/AWSServiceRoleForIoTSiteWise` 策略创建一个角色并与建立信任关系 `iotsitewise.amazonaws.com`。

有关更多信息，请参阅 IAM 用户指南 中的[创建服务相关角色](#)。

如果您删除了此服务相关角色，可以使用同样的过程再次创建角色。

## 为 AWS IoT SiteWise 编辑服务相关角色

AWS IoT SiteWise 不允许您编辑 `AWSServiceRoleForIoTSiteWise` 服务相关角色。在创建服务相关角色后，您将无法更改角色的名称，因为可能有多种实体引用该角色。不过，您可以使用 IAM 编辑角色的说明。有关更多信息，请参阅《IAM 用户指南》中的[编辑服务相关角色](#)。

## 删除 AWS IoT SiteWise 的服务相关角色

如果不再使用需要服务相关角色的功能或服务，则建议删除关联的角色。这是为了避免出现未被监控或维护的非活动实体。但是，您必须先清除服务相关角色的资源，然后才能手动删除它。

**Note**

如果您尝试删除资源时 AWS IoT SiteWise 服务正在使用该角色，则删除可能会失败。如果发生这种情况，请等待几分钟，然后重试。

## 删除使用的 AWS IoT SiteWise 资源 AWSServiceRoleForIoTSiteWise

1. 禁用登录功能 AWS IoT SiteWise。有关更多信息，请参阅[更改您的日志级别](#)
2. 删除所有活动的 SiteWise Edge 网关。

## 使用 IAM 手动删除服务相关角色

使用 IAM 控制台 AWS CLI、或 AWS API 删除 AWSServiceRoleForIoTSiteWise 服务相关角色。有关更多信息，请参见《IAM 用户指南》中的[删除服务相关角色](#)。

## AWS IoT SiteWise 服务相关角色支持的区域

AWS IoT SiteWise 支持在提供服务的所有区域中使用服务相关角色。有关更多信息，请参阅[AWS IoT SiteWise 终端节点和配额](#)。

## 将服务角色用于 AWS IoT SiteWise Monitor

服务角色是由一项服务担任、代表您执行操作的 [IAM 角色](#)。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的[创建向 AWS 服务委派权限的角色](#)。

要允许联合的 M SiteWise onitor 门户用户访问您的 AWS IoT SiteWise 和 AWS IAM Identity Center 资源，您必须将服务角色附加到您创建的每个门户。服务角色必须将 M SiteWise onitor 指定为可信实体，并包含[AWSIoTSiteWiseMonitorPortalAccess](#) 托管策略或定义[等效权限](#)。此策略由 M SiteWise onitor 维护 AWS 并定义了用于访问您 AWS IoT SiteWise 和 IAM Identity Center 资源的一组权限。

创建 M SiteWise onitor 门户时，必须选择允许该门户的用户访问您 AWS IoT SiteWise 和 IAM Identity Center 资源的角色。AWS IoT SiteWise 控制台可以为您创建和配置角色。您可以稍后在 IAM 中编辑该角色。如果您从角色中移除所需的权限或删除该角色，您的门户用户在使用其 SiteWise 监控器门户时会遇到问题。

**Note**

2020 年 4 月 29 日之前创建的门户不需要服务角色。如果您是在此日期之后创建的门户，则必须附加服务角色才能继续使用。为此，请导航到 [AWS IoT SiteWise 控制台](#) 的门户页面，再选择迁移所有门户以使用 IAM 角色。

以下各节介绍如何在 AWS Management Console 或中创建和管理 M SiteWise onitor 服务角色 AWS Command Line Interface。

**目录**

- [SiteWise 监控器的服务角色权限](#)
- [管理 SiteWise 监控服务角色 \(控制台\)](#)
  - [查找门户的服务角色 \(控制台\)](#)
  - [创建 SiteWise 监控服务角色 \(AWS IoT SiteWise 控制台\)](#)
  - [创建 SiteWise 监控服务角色 \(IAM 控制台\)](#)
  - [更改门户的服务角色 \(控制台\)](#)
- [管理 SiteWise 监控服务角色 \(CLI\)](#)
  - [查找门户的服务角色 \(CLI\)](#)
  - [创建 SiteWise 监控服务角色 \(CLI\)](#)
- [SiteWise 监视更新 AWSIoTSiteWiseMonitorServiceRole](#)

**SiteWise 监控器的服务角色权限**

创建门户时，AWS IoT SiteWise 允许您创建名称以开头的角色 `AWSIoTSiteWiseMonitorServiceRole`。此角色允许联合 SiteWise 监视器用户访问您的门户配置、资产、资产数据以及 IAM Identity Center 配置。

该角色信任以下服务来代入角色：

- `monitor.iotsitewise.amazonaws.com`

该角色使用以下权限策略（名称以开头）来允许 M SiteWise onitor 用户对您账户中的资源完成操作。AWSIoTSiteWiseMonitorServicePortalPolicy [AWSIoTSiteWiseMonitorPortalAccess](#) 托管策略定义了等效权限。



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribePortal",
        "iotsitewise:CreateProject",
        "iotsitewise:DescribeProject",
        "iotsitewise:UpdateProject",
        "iotsitewise>DeleteProject",
        "iotsitewise:ListProjects",
        "iotsitewise:BatchAssociateProjectAssets",
        "iotsitewise:BatchDisassociateProjectAssets",
        "iotsitewise:ListProjectAssets",
        "iotsitewise:CreateDashboard",
        "iotsitewise:DescribeDashboard",
        "iotsitewise:UpdateDashboard",
        "iotsitewise>DeleteDashboard",
        "iotsitewise:ListDashboards",
        "iotsitewise:CreateAccessPolicy",
        "iotsitewise:DescribeAccessPolicy",
        "iotsitewise:UpdateAccessPolicy",
        "iotsitewise>DeleteAccessPolicy",
        "iotsitewise:ListAccessPolicies",
        "iotsitewise:DescribeAsset",
        "iotsitewise:ListAssets",
        "iotsitewise:ListAssociatedAssets",
        "iotsitewise:DescribeAssetProperty",
        "iotsitewise:GetAssetPropertyValue",
        "iotsitewise:GetAssetPropertyValueHistory",
        "iotsitewise:GetAssetPropertyAggregates",
        "iotsitewise:BatchPutAssetPropertyValue",
        "iotsitewise:ListAssetRelationships",
        "iotsitewise:DescribeAssetModel",
        "iotsitewise:ListAssetModels",
        "iotsitewise:UpdateAssetModel",
        "iotsitewise:UpdateAssetModelPropertyRouting",
        "sso-directory:DescribeUsers",
        "sso-directory:DescribeUser",
        "iotevents:DescribeAlarmModel",
        "iotevents:ListTagsForResource"
      ],
    }
  ],
}
```

```

    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "iotevents:BatchAcknowledgeAlarm",
      "iotevents:BatchSnoozeAlarm",
      "iotevents:BatchEnableAlarm",
      "iotevents:BatchDisableAlarm"
    ],
    "Resource": "*",
    "Condition": {
      "Null": {
        "iotevents:keyValue": "false"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "iotevents:CreateAlarmModel",
      "iotevents:TagResource"
    ],
    "Resource": "*",
    "Condition": {
      "Null": {
        "aws:RequestTag/iotsitewisemonitor": "false"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "iotevents:UpdateAlarmModel",
      "iotevents>DeleteAlarmModel"
    ],
    "Resource": "*",
    "Condition": {
      "Null": {
        "aws:ResourceTag/iotsitewisemonitor": "false"
      }
    }
  },
  {

```

```
    "Effect": "Allow",
    "Action": [
      "iam:PassRole"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": [
          "iotevents.amazonaws.com"
        ]
      }
    }
  ]
}
```

有关警报所需权限的更多信息，请参阅 [设置 AWS IoT Events 警报权限](#)。

当门户用户登录时，M SiteWise onitor [会根据服务角色与该用户的访问策略](#)的交叉点创建会话策略。访问策略定义 身份对门户和项目的访问级别。有关门户权限和访问策略的更多信息，请参阅[管理您的 SiteWise 监控器门户](#)和[CreateAccess策略](#)。

### 管理 SiteWise 监控服务角色 ( 控制台 )

AWS IoT SiteWise 控制台 便于管理门户的 SiteWise 监控服务角色。创建入口后，控制台会检查是否存在适合附加的现有角色。如果没有可用的服务角色，则控制台可以为您创建和配置服务角色。有关更多信息，请参阅 [创建门户](#)。

### 主题

- [查找门户的服务角色 \( 控制台 \)](#)
- [创建 SiteWise 监控服务角色 \( AWS IoT SiteWise 控制台 \)](#)
- [创建 SiteWise 监控服务角色 \( IAM 控制台 \)](#)
- [更改门户的服务角色 \( 控制台 \)](#)

### 查找门户的服务角色 ( 控制台 )

使用以下步骤查找附加到 SiteWise 监控器门户的服务角色。

### 查找门户的服务角色

1. 导航到 [AWS IoT SiteWise 控制台](#)。

2. 在左侧导航窗格中，选择 门户。
3. 选择要查找其服务角色的门户。

附加到门户的角色显示在权限、服务角色 下。

### 创建 SiteWise 监控服务角色 ( AWS IoT SiteWise 控制台 )

创建 M SiteWise onitor 门户时，可以为门户创建服务角色。有关更多信息，请参阅 [创建门户](#)。

您也可以在 AWS IoT SiteWise 控制台中为现有门户创建服务角色。这将取代门户的现有服务角色。

#### 为现有门户创建服务角色

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 门户。
3. 选择要为其创建新服务角色的门户。
4. 在门户详细信息下，选择编辑。
5. 在权限下，从列表中选择创建和使用新的服务角色。
6. 输入新角色的名称。
7. 选择保存。

### 创建 SiteWise 监控服务角色 ( IAM 控制台 )

您可以在 IAM 控制台中从服务角色模板创建服务角色。此角色模板包括 [AWSIoTSiteWiseMonitorPortalAccess](#) 托管策略，并将 Monito SiteWise r 指定为可信实体。

#### 通过门户服务角色模板创建服务角色

1. 导航到 [IAM 控制台](#)。
2. 在导航窗格中，选择 角色。
3. 选择 创建角色。
4. 在选择用例中，选择物联网 SiteWise。
5. 在选择您的使用案例中，选择 IoT SiteWise Monitor - Portal。
6. 选择下一步: 权限。
7. 选择下一步: 标签。
8. 选择下一步：审核。

9. 在角色名称中，输入新服务角色名称。
10. 选择 **创建角色**。

### 更改门户的服务角色 (控制台)

使用以下步骤为门户选择不同的 SiteWise 监控服务角色。

#### 更改门户的服务角色

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在导航窗格中，选择 **门户**。
3. 选择要更改其服务角色的门户。
4. 在门户详细信息下，选择 **编辑**。
5. 在权限下，选择使用现有角色。
6. 选择要附加到此门户的现有角色。
7. 选择 **保存**。

### 管理 SiteWise 监控服务角色 (CLI)

您可以将 AWS CLI 用于以下门户服务角色管理任务：

#### 主题

- [查找门户的服务角色 \(CLI\)](#)
- [创建 SiteWise 监控服务角色 \(CLI\)](#)

### 查找门户的服务角色 (CLI)

要查找附加到 SiteWise 监控器门户的服务角色，请运行以下命令列出当前区域中的所有门户。

```
aws iotsitewise list-portals
```

此操作将按照以下格式返回包含门户摘要的响应。

```
{
  "portalSummaries": [
    {
```

```

    "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
    "name": "WindFarmPortal",
    "description": "A portal that contains wind farm projects for Example Corp.",
    "roleArn": "arn:aws:iam::123456789012:role/service-role/role-name",
    "startUrl": "https://a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE.app.iotsitewise.aws",
    "creationDate": "2020-02-04T23:01:52.90248068Z",
    "lastUpdateDate": "2020-02-04T23:01:52.90248078Z"
  }
]
}

```

如果您知道门户的 ID，也可以使用该[DescribePortal](#)操作来查找门户的角色。

## 创建 SiteWise 监控服务角色 (CLI)

使用以下步骤创建新的 SiteWise 监控服务角色。

### 创建 SiteWise 监控服务角色

1. 使用允许 Monitor SiteWise r 担任该角色的信任策略创建角色。此示例根据存储在 JSON 字符串中的信任策略创建名为 **MySiteWiseMonitorPortalRole** 的角色。

Linux, macOS, or Unix

```

aws iam create-role --role-name MySiteWiseMonitorPortalRole --assume-role-
policy-document '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "monitor.iotsitewise.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}'

```

Windows command prompt

```

aws iam create-role --role-name MySiteWiseMonitorPortalRole --assume-role-
policy-document "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Effect\": \"Allow

```

```
\",\"Principal\":{\"Service\": \"monitor.iotsitewise.amazonaws.com\"},\"Action\": \"sts:AssumeRole\"}]}]\"
```

2. 从输出中的角色元数据复制角色 ARN。创建门户时，您可以使用此 ARN 将角色与门户关联。有关创建门户的更多信息，请参阅 AWS IoT SiteWise API 参考 [CreatePortal](#) 中的。
3. 将 `AWSIoTSiteWiseMonitorPortalAccess` 策略附加到角色，或附加定义等效权限的策略。

```
aws iam attach-role-policy --role-name MySiteWiseMonitorPortalRole --policy-arn arn:aws:iam::aws:policy/service-role/AWSIoTSiteWiseMonitorPortalAccess
```

## 将服务角色附加到现有门户

1. 要检索门户的现有详细信息，请运行以下命令。将 `portal-id` 替换为该门户的 ID。

```
aws iotsitewise describe-portal --portal-id portal-id
```

该操作返回一个响应，其中包含如下格式的门户详细信息。

```
{
  "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "portalArn": "arn:aws:iotsitewise:region:account-id:portal/a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "portalName": "WindFarmPortal",
  "portalDescription": "A portal that contains wind farm projects for Example Corp.",
  "portalClientId": "E-1a2b3c4d5e6f_sn6tbqHVzLWVEXAMPLE",
  "portalStartUrl": "https://a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE.app.iotsitewise.aws",
  "portalContactEmail": "support@example.com",
  "portalStatus": {
    "state": "ACTIVE"
  },
  "portalCreationDate": "2020-04-29T23:01:52.90248068Z",
  "portalLastUpdateDate": "2020-04-29T00:28:26.103548287Z",
  "roleArn": "arn:aws:iam::123456789012:role/service-role/AWSIoTSiteWiseMonitorServiceRole_1aEXAMPLE"
}
```

2. 要将服务角色附加到门户，请运行以下命令。将 `role-arn` 替换为服务角色 ARN，将其余参数替换为门户的现有值。

```
aws iotsitewise update-portal \
  --portal-id portal-id \
  --role-arn role-arn \
  --portal-name portal-name \
  --portal-description portal-description \
  --portal-contact-email portal-contact-email
```

## SiteWise 监视更新 AWSIoTSiteWiseMonitorServiceRole

您可以查看有关 for M SiteWise onit AWSIoTSiteWiseMonitorServiceRoleor 更新的详细信息，从该服务开始跟踪更改时开始。要获得有关此页面变更的自动提醒，请订阅“AWS IoT SiteWise 文档历史记录”页面上的 RSS feed。

更改	描述	日期
<a href="#">AWSIoTSiteWiseMonitorPortal Access</a> - 更新的策略	AWS IoT SiteWise 更新了警报功能的 <a href="#">AWSIoTSiteWiseMonitorPortalAccess</a> 托管策略。	2021 年 5 月 27 日
AWS IoT SiteWise 开始跟踪更改	AWS IoT SiteWise 开始跟踪其服务角色的变化。	2020 年 12 月 15 日

## 设置 AWS IoT Events 警报权限

当您使用 AWS IoT Events 警报模型监控 AWS IoT SiteWise 资产属性时，您必须拥有以下 IAM 权限：

- 允许 AWS IoT Events 向发送数据的 AWS IoT Events 服务角色 AWS IoT SiteWise。有关更多信息，请参阅 AWS IoT Events 开发人员指南中的适用于 AWS IoT Events 的 [身份和访问管理](#)。
- 您必须拥有以下 AWS IoT SiteWise 操作权限：`iotsitewise:DescribeAssetModel`和*iot*`sitewise:UpdateAssetModelPropertyRouting`。这些权限允许 AWS IoT SiteWise 向 AWS IoT Events 警报模型发送资产属性值。

有关更多信息，请参阅 IAM 用户指南的[基于资源的策略](#)。



## 所需的操作权限

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。JSON 策略的 Action 元素描述可用于在策略中允许或拒绝访问的操作。

在定义 AWS IoT Events 警报模型之前，必须授予以下权限，允许 AWS IoT SiteWise 向警报模型发送资产属性值。

- `iotsitewise:DescribeAssetModel`— AWS IoT Events 允许检查资产属性是否存在。
- `iotsitewise:UpdateAssetModelPropertyRouting`— AWS IoT SiteWise 允许自动创建允许 AWS IoT SiteWise 向其发送数据的订阅 AWS IoT Events。

有关 AWS IoT SiteWise 支持的操作的更多信息，请参阅《服务授权参考》AWS IoT SiteWise 中[定义的操作](#)。

### Example 权限策略示例 1

以下策略允许 AWS IoT SiteWise 向任何 AWS IoT Events 警报模型发送资产属性值。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotevents:CreateAlarmModel",
        "iotevents:UpdateAlarmModel"
      ],
      "Resource": "arn:aws:iotevents:us-east-1:123456789012:alarmModel/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribeAssetModel",
        "iotsitewise:UpdateAssetModelPropertyRouting"
      ],
      "Resource": "arn:aws:iotsitewise:us-east-1:123456789012:asset-model/*"
    }
  ]
}
```

## Example 权限策略示例 2

以下策略 AWS IoT SiteWise 允许将指定资产属性的值发送到指定的 AWS IoT Events 警报模型。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotevents:CreateAlarmModel",
        "iotevents:UpdateAlarmModel"
      ],
      "Resource": "arn:aws:iotevents:us-east-1:123456789012:alarmModel/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribeAssetModel"
      ],
      "Resource": "arn:aws:iotsitewise:us-east-1:123456789012:asset-model/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:UpdateAssetModelPropertyRouting"
      ],
      "Resource": [
        "arn:aws:iotsitewise:us-east-1:123456789012:asset-model/12345678-90ab-
cdef-1234-567890abcdef"
      ],
      "Condition": {
        "StringLike": {
          "iotsitewise:propertyId": "abcdef12-3456-7890-abcd-ef1234567890",
          "iotevents:alarmModelArn": "arn:aws:iotevents:us-
east-1:123456789012:alarmModel/MyAlarmModel"
        }
      }
    }
  ]
}
```

## ( 可选 ) ListInputRoutings 权限

更新或删除资产模型时，AWS IoT SiteWise 可以检查中的警报模型是否 AWS IoT Events 正在监视与该资产模型关联的资产属性。这可以防止您删除 AWS IoT Events 警报当前正在使用的资产属性。要在中启用此功能 AWS IoT SiteWise，您必须拥有 `iotevents:ListInputRoutings` 权限。此权限 AWS IoT SiteWise 允许调用支持的 [ListInputRouting](#) s API 操作。AWS IoT Events

### Note

强烈建议您添加 `ListInputRoutings` 权限。

### Example 权限策略示例

以下政策允许您更新和删除资产模型，并使用中的 `ListInputRoutings` API AWS IoT SiteWise。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:UpdateAssetModel",
        "iotsitewise>DeleteAssetModel",
        "iotevents:ListInputRoutings"
      ],
      "Resource": "arn:aws:iotsitewise:us-east-1:123456789012:asset-model/*"
    }
  ]
}
```

### SiteWise 监控器所需的权限

如果要在 SiteWise 监控门户中使用警报功能，则必须使用以下策略更新 [SiteWise 监控服务角色](#)：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribePortal",

```

```

        "iotsitewise:CreateProject",
        "iotsitewise:DescribeProject",
        "iotsitewise:UpdateProject",
        "iotsitewise>DeleteProject",
        "iotsitewise:ListProjects",
        "iotsitewise:BatchAssociateProjectAssets",
        "iotsitewise:BatchDisassociateProjectAssets",
        "iotsitewise:ListProjectAssets",
        "iotsitewise:CreateDashboard",
        "iotsitewise:DescribeDashboard",
        "iotsitewise:UpdateDashboard",
        "iotsitewise>DeleteDashboard",
        "iotsitewise:ListDashboards",
        "iotsitewise:CreateAccessPolicy",
        "iotsitewise:DescribeAccessPolicy",
        "iotsitewise:UpdateAccessPolicy",
        "iotsitewise>DeleteAccessPolicy",
        "iotsitewise:ListAccessPolicies",
        "iotsitewise:DescribeAsset",
        "iotsitewise:ListAssets",
        "iotsitewise:ListAssociatedAssets",
        "iotsitewise:DescribeAssetProperty",
        "iotsitewise:GetAssetPropertyValue",
        "iotsitewise:GetAssetPropertyValueHistory",
        "iotsitewise:GetAssetPropertyAggregates",
        "iotsitewise:BatchPutAssetPropertyValue",
        "iotsitewise:ListAssetRelationships",
        "iotsitewise:DescribeAssetModel",
        "iotsitewise:ListAssetModels",
        "iotsitewise:UpdateAssetModel",
        "iotsitewise:UpdateAssetModelPropertyRouting",
        "sso-directory:DescribeUsers",
        "sso-directory:DescribeUser",
        "iotevents:DescribeAlarmModel",
        "iotevents:ListTagsForResource"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "iotevents:BatchAcknowledgeAlarm",
        "iotevents:BatchSnoozeAlarm",
        "iotevents:BatchEnableAlarm",

```

```

        "iotevents:BatchDisableAlarm"
    ],
    "Resource": "*",
    "Condition": {
        "Null": {
            "iotevents:keyValue": "false"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "iotevents:CreateAlarmModel",
        "iotevents:TagResource"
    ],
    "Resource": "*",
    "Condition": {
        "Null": {
            "aws:RequestTag/iotsitewisemonitor": "false"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "iotevents:UpdateAlarmModel",
        "iotevents>DeleteAlarmModel"
    ],
    "Resource": "*",
    "Condition": {
        "Null": {
            "aws:ResourceTag/iotsitewisemonitor": "false"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "iam:PassRole"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "iam:PassedToService": [

```

```

    "iotevents.amazonaws.com"
  ]
}
}
}
]
}

```

## 防止跨服务混淆代理

混淆代理问题是一个安全性问题，即不具有操作执行权限的实体可能会迫使具有更高权限的实体执行该操作。在中 AWS，跨服务模仿可能会导致混乱的副手问题。一个服务（呼叫服务）调用另一项服务（所谓的被调服务）时，可能会发生跨服务模拟。可以操纵调用服务以使用其权限对另一个客户的资源进行操作，否则该服务不应有访问权限。为了防止这种情况，AWS 提供可帮助您保护所有服务的委托服务委托人的工具，这些服务委托人有权访问账户中的资源。

我们建议在资源策略中使用 [aws:SourceArn](#) 和 [aws:SourceAccount](#) 全局条件上下文密钥来限制为资源 AWS IoT SiteWise 提供其他服务的权限。如果 `aws:SourceArn` 值不包含账户 ID，例如 Amazon S3 桶 Amazon 资源名称（ARN），您必须使用两个全局条件上下文键来限制权限。如果同时使用全局条件上下文密钥和包含账户 ID 的 `aws:SourceArn` 值，则 `aws:SourceAccount` 值和 `aws:SourceArn` 值中的账户在同一策略语句中使用，必须使用相同的账户 ID。

- 如果您只希望将一个资源与跨服务访问相关联，请使用 `aws:SourceArn`。
- 如果您想允许该账户中的任何资源与跨服务使用操作相关联，请使用 `aws:SourceAccount`。

的值 `aws:SourceArn` 必须是与 `sts:AssumeRole` 请求关联的 AWS IoT SiteWise 客户资源。

防范混淆代理问题最有效的方法是使用 `aws:SourceArn` 全局条件上下文键和资源的完整 ARN。如果您不知道资源的完整 ARN，或正在指定多个资源，请针对 ARN 未知部分使用带有通配符 (\*) 的 `aws:SourceArn` 全局上下文条件键。例如，`arn:aws:service:*:123456789012:*`。

### Example — 混乱的副手预防

以下示例显示了如何在中使用 `aws:SourceArn` 和 `aws:SourceAccount` 全局条件上下文键 AWS IoT SiteWise 来防止出现混淆的副手问题。

```

{
  "Version": "2012-10-17",
  "Statement": {

```

```
"Sid": "ConfusedDeputyPreventionExamplePolicy",
"Effect": "Allow",
"Principal": {
  "Service": "iotsitewise.amazonaws.com"
},
"Action": "sts:AssumeRole",
"Resource": [
  "arn:aws:iotsitewise::ResourceName/*"
],
"Condition": {
  "ArnLike": {
    "aws:SourceArn": "arn:aws:iotsitewise:*:123456789012:*"
  },
  "StringEquals": {
    "aws:SourceAccount": "123456789012"
  }
}
}
```

## 对 AWS IoT SiteWise 身份和访问进行故障排除

使用以下信息来帮助您诊断和修复使用 AWS IoT SiteWise 和 AWS Identity and Access Management (IAM) 时可能遇到的常见问题。

### 主题

- [我无权在以下位置执行操作 AWS IoT SiteWise](#)
- [我无权执行 iam:PassRole](#)
- [我想允许我以外的人 AWS 账户 访问我的 AWS IoT SiteWise 资源](#)

### 我无权在以下位置执行操作 AWS IoT SiteWise

如果 AWS Management Console 告诉您您无权执行某项操作，则必须联系管理员寻求帮助。管理员是指提供用户名和密码的人员。

当 mateojackson IAM 用户尝试使用控制台查看有关资产的详细信息，但不具有 iotsitewise:DescribeAsset 权限时，会发生以下示例错误。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
iotsitewise:DescribeAsset on resource: a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

在这种情况下，Mateo 请求管理员更新其策略，以允许他使用 `iotsitewise:DescribeAsset` 操作访问 ID 为 `a1b2c3d4-5678-90ab-cdef-2222EXAMPLE` 的资产资源。

## 我无权执行 `iam:PassRole`

如果您收到一个错误，表明您无权执行 `iam:PassRole` 操作，则必须更新策略以允许您将角色传递给 AWS IoT SiteWise。

有些 AWS 服务 允许您将现有角色传递给该服务，而不是创建新的服务角色或服务相关角色。为此，您必须具有将角色传递到服务的权限。

当名为 `marymajor` 的 IAM 用户尝试使用控制台在 AWS IoT SiteWise 中执行操作时，会发生以下示例错误。但是，服务必须具有服务角色所授予的权限才可执行此操作。Mary 不具有将角色传递到服务的权限。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

在这种情况下，必须更新 Mary 的策略以允许她执行 `iam:PassRole` 操作。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

## 我想允许我以外的人 AWS 账户 访问我的 AWS IoT SiteWise 资源

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以担任角色。对于支持基于资源的策略或访问控制列表 (ACL) 的服务，您可以使用这些策略向人员授予对您的资源的访问权。

要了解更多信息，请参阅以下内容：

- 要了解是否 AWS IoT SiteWise 支持这些功能，请参阅 [如何 AWS IoT SiteWise 与 IAM 配合使用](#)。
- 要了解如何提供对您拥有的资源的访问权限 AWS 账户，请参阅 [IAM 用户指南中的向您拥有 AWS 账户的另一个 IAM 用户提供访问权限](#)。
- 要了解如何向第三方提供对您的资源的访问 [权限 AWS 账户](#)，请参阅 [IAM 用户指南中的向第三方提供访问权限](#)。AWS 账户
- 要了解如何通过身份联合验证提供访问权限，请参阅《IAM 用户指南》中的 [为经过外部身份验证的用户 \(身份联合验证\) 提供访问权限](#)。
- 要了解使用角色和基于资源的策略进行跨账户存取之间的差别，请参阅《IAM 用户指南》中的 [IAM 角色与基于资源的策略有何不同](#)。



## 合规性验证 AWS IoT SiteWise

AWS IoT SiteWise 不在任何 AWS 合规计划的范围内。

有关特定合规计划范围内的 AWS 服务列表，请参阅按合规计划划分的[范围内的AWS服务按合规计划](#)。有关一般信息，请参阅[AWS 合规计划AWS](#)。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅中的“[下载报告](#)”中的“[AWS Artifact](#)”。

您在使用 AWS IoT SiteWise 时的合规责任取决于您的数据的敏感性、贵公司的合规目标以及适用的法律和法规。AWS 提供了以下资源来帮助实现合规性：

- [安全与合规性快速入门指南](#) — 这些部署指南讨论了架构注意事项，并提供了在上部署以安全性和合规性为重点的基准环境的步骤。AWS
- [HIPAA 安全与合规架构白皮书 — 本白皮书](#)描述了公司如何使用来 AWS 创建符合 HIPAA 标准的应用程序。
- [AWS 合规资源AWS](#) — 此工作簿和指南集可能适用于您所在的行业和所在地区。
- [使用AWS Config 开发人员指南中的规则评估资源](#) — 该 AWS Config 服务评估您的资源配置在多大程度上符合内部实践、行业准则和法规。
- [AWS Security Hub](#)— 此 AWS 服务可全面了解您的安全状态 AWS ，帮助您检查是否符合安全行业标准和最佳实践。
- [工业物联网解决方案的十大安全黄金法则](#) — 这篇博客文章介绍了有助于保护工业控制系统 (ICS)、工业物联网 (IIoT) 和云环境的十条黄金法则。
- [制造 OT 的安全最佳实践 — 本白皮书描述了为](#) AWS 云设计、部署和架构这些本地混合制造工作负载的安全最佳实践。

## 韧性在 AWS IoT SiteWise

AWS 全球基础设施是围绕 AWS 区域和可用区构建的。AWS 区域提供多个物理隔离和隔离的可用区，这些可用区通过低延迟、高吞吐量和高度冗余的网络相连。利用可用区，您可以设计和操作在可用区之间无中断地自动实现失效转移的应用程序和数据库。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错性和可扩展性。

AWS IoT SiteWise 完全托管，使用高度可用和耐用的 AWS 服务，例如 Amazon S3 和 Amazon EC2。为了确保可用区中断时的可用性，需要跨多个可用区 AWS IoT SiteWise 运行。

有关 AWS 区域和可用区的更多信息，请参阅[AWS 全球基础设施](#)。

除了 AWS 全球基础架构外，还 AWS IoT SiteWise 提供多项功能来帮助支持您的数据弹性和备份需求：

- 您可以 AWS IoT Core 通过 MQTT 消息将属性值更新发布到，然后配置规则以对这些数据进行操作。使用此功能，您可以备份其他 AWS 服务中的数据，例如 Amazon S3 和 Amazon DynamoDB。有关更多信息，请参阅 [与其他 AWS 服务互动](#) 和 [使用资产属性通知将数据导出到 Amazon S3](#)。
- 您可以使用 AWS IoT SiteWise Get\* API 来检索和备份历史资产属性数据。有关更多信息，请参阅 [查询历史资产属性值](#)。
- 您可以使用 AWS IoT SiteWise Describe\* API 来检索资源的定义，例如资产和模型。可以对这些定义进行备份，并稍后使用它们重新创建资源。有关更多信息，请参阅 [AWS IoT SiteWise API 参考](#)。

## 基础设施安全 AWS IoT SiteWise

作为一项托管服务 AWS IoT SiteWise，受 AWS 全球网络安全的保护。有关 AWS 安全服务以及如何 AWS 保护基础设施的信息，请参阅[AWS 云安全](#)。要使用基础设施安全的最佳实践来设计您的 AWS 环境，请参阅 S AWS security Pillar Well-Architected Framework 中的[基础设施保护](#)。

您可以使用 AWS 已发布的 API 调用 AWS IoT SiteWise 通过网络进行访问。客户端必须支持以下内容：

- 传输层安全性协议 (TLS)。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 具有完全向前保密 (PFS) 的密码套件，例如 DHE (临时 Diffie-Hellman) 或 ECDHE (临时椭圆曲线 Diffie-Hellman)。大多数现代系统 (如 Java 7 及更高版本) 都支持这些模式。

此外，必须使用访问密钥 ID 和与 IAM 委托人关联的秘密访问密钥来对请求进行签名。或者，您可以使用 [AWS Security Token Service](#) (AWS STS) 生成临时安全凭证来对请求进行签名。

SiteWise 在上 AWS IoT Greengrass 运行的 Edge 网关使用 X.509 证书和加密密钥来连接云并进行身份验证。AWS 有关更多信息，请参阅 AWS IoT Greengrass Version 1 开发人员指南中的[AWS IoT Greengrass 设备身份验证和授权](#)。

## 配置和漏洞分析

IoT 实例集可能由大量具有不同功能、长期存在且地理位置分散的设备组成。这些特性导致实例集设置复杂且容易出错。由于设备的处理能力、内存和存储空间通常有限，因此它们并不总是支持加密和其

他安全措施。此外，设备经常使用具有已知漏洞的软件。这些因素不仅令 IoT 实例集成为吸引黑客的目标，而且导致难以持续保护设备实例集安全。

AWS IoT Device Defender 通过提供工具来识别安全问题和与最佳实践的偏差，从而应对这些挑战。AWS IoT Device Defender 用于分析、审计和监控连接的设备，以检测异常行为并降低安全风险。AWS IoT Device Defender 可以对设备队列进行审计，以确保它们遵守安全最佳实践并检测设备上的异常行为。这使得在您的 AWS IoT 设备群中实施一致的安全策略成为可能，并在设备遭到入侵时快速做出响应。有关更多信息，请参阅 AWS IoT 开发人员指南中的 [AWS IoT Device Defender](#)。

如果您使用 SiteWise Edge 网关向服务采集数据，则需要您负责配置和维护 SiteWise Edge 网关的环境。此职责包括升级到最新版本的 SiteWise Edge 网关的系统 AWS IoT Greengrass 软件、软件和 AWS IoT SiteWise 连接器。有关更多信息，请参阅 [《AWS IoT Greengrass Version 1 开发人员指南》](#) 中的 [配置内 AWS IoT Greengrass 核和升级连接器](#)。

## VPC 端点

接口 VPC 终端节点在您的虚拟私有云 (VPC) 和之间建立私有连接 AWS IoT SiteWise。 [AWS PrivateLink](#) 为接口端点提供支持，允许对 AWS IoT SiteWise API 操作进行私有访问。您可以无需互联网网关、NAT 设备、VPN 连接或 AWS Direct Connect。VPC 中的实例即使没有公有 IP 地址也可与 AWS IoT SiteWise API 操作进行通信。您的 VPC 和 VPC 之间的流量 AWS IoT SiteWise 不会离开 AWS 网络。

每个接口端点均由子网中的一个或多个 [弹性网络接口](#) 表示。

在为设置接口 VPC 终端节点之前 AWS IoT SiteWise，请查看 Amazon VPC 用户指南中的 [接口终端节点属性和限制](#)。

有关更多信息，请参阅《Amazon VPC 用户指南》中的 [接口 VPC 端点 \(AWS PrivateLink\)](#)。

## VPC 终端节点支持的 API 操作

AWS IoT SiteWise 支持从您的 VPC 调用以下 AWS IoT SiteWise API 操作：

- 对于所有数据层面 API 操作，请使用以下端点：*region* 替换为你的 AWS 区域

```
data.iotsitewise.region.amazonaws.com
```

数据平面 API 操作包括以下内容：

- [BatchGetAssetProperty](#) 价值

- [BatchGetAssetPropertyValueHistory](#)
  - [BatchPutAssetProperty价值](#)
  - [GetAssetPropertyAggregates](#)
  - [GetAssetPropertyValue](#)
  - [GetAssetPropertyValue历史](#)
  - [GetInterpolatedAssetProperty价值观](#)
- 对于用于管理资产模型、资产、SiteWise 边缘网关、标签和账户配置的控制平面 API 操作，请使用以下端点。将 *region* 替换为您的 AWS 区域

```
api.iotsitewise.region.amazonaws.com
```


支持的控制平面 API 操作包含以下内容：

- [AssociateAssets](#)
- [CreateAsset](#)
- [CreateAsset模型](#)
- [DeleteAsset](#)
- [DeleteAsset模型](#)
- [DeleteDashboard](#)
- [DescribeAsset](#)
- [DescribeAsset模型](#)
- [DescribeAsset财产](#)
- [DescribeDashboard](#)
- [DescribeLogging选项](#)
- [DisassociateAssets](#)
- [ListAsset模特](#)
- [ListAsset人际关系](#)
- [ListAssets](#)
- [ListAssociated资产](#)
- [PutLogging选项](#)
- [UpdateAsset](#)

支持的 API 操作

- [UpdateAsset模型](#)

- [UpdateAsset财产](#)
- [CreateGateway](#)
- [DeleteGateway](#)
- [DescribeDefaultEncryptionConfiguration](#)
- [DescribeGateway](#)
- [DescribeGatewayCapabilityConfiguration](#)
- [DescribeStorage配置](#)
- [ListGateways](#)
- [ListTagsForResource](#)
- [UpdateGateway](#)
- [UpdateGatewayCapabilityConfiguration](#)
- [PutDefaultEncryptionConfiguration](#)
- [PutStorage配置](#)
- [TagResource](#)
- [UntagResource](#)

 Note

控制平面 API 操作的接口 VPC 终端节点目前不支持调用以下 SiteWise 监控 API 操作：

- [BatchAssociateProjectAssets](#)
- [BatchDisassociateProjectAssets](#)
- [CreateAccess政策](#)
- [CreateDashboard](#)
- [CreatePortal](#)
- [CreateProject](#)
- [DeleteAccess政策](#)
- [DeletePortal](#)
- [DeleteProject](#)
- [DescribeAccess政策](#)
- [DescribePortal](#)
- [DescribeProject](#)

- [ListAccess政策](#)
- [ListDashboards](#)
- [ListPortals](#)
- [ListProjects](#)
- [ListProject资产](#)
- [UpdateAccess政策](#)
- [UpdateDashboard](#)
- [UpdatePortal](#)
- [UpdateProject](#)

## 为 AWS IoT SiteWise 创建接口 VPC 端点

要为 AWS IoT SiteWise 服务创建 VPC 终端节点，请使用 Amazon VPC 控制台或 AWS Command Line Interface (AWS CLI)。有关更多信息，请参阅《Amazon VPC 用户指南》中的[创建接口端点](#)。

使用以下服务名称之一 AWS IoT SiteWise 为创建 VPC 终端节点：

- 对于数据平面 API 操作，请使用以下服务名称：

```
com.amazonaws.region.iotsitewise.data
```

- 对于控制平面 API 操作，请使用以下服务名称：

```
com.amazonaws.region.iotsitewise.api
```

## AWS IoT SiteWise 通过接口 VPC 终端节点进行访问

当您创建接口终端节点时，我们会生成特定于终端节点的 DNS 主机名，您可以用来与之通信。AWS IoT SiteWise 默认情况下启用私有 DNS。有关更多信息，请参阅 Amazon VPC 用户指南中的[使用私有托管区域](#)。

如果您为终端节点启用私有 DNS，则可以通过以下 VPC 终端节点之一向 AWS IoT SiteWise 发出 API 请求。

- 对于数据层面 API 操作，请使用以下终端节点：将##替换为您的 AWS 区域。

```
data.iotsitewise.region.amazonaws.com
```

- 对于控制平面 API 操作，请使用以下终端节点：将##替换为您的 AWS 区域。

```
api.iotsitewise.region.amazonaws.com
```

如果您为终端节点禁用私有 DNS，则必须执行以下操作才能 AWS IoT SiteWise 通过终端节点进行访问：

#### 1. 在 API 请求中指定 VPC 端点 url。

- 对于数据层面 API 操作，请使用以下端点 URL。将 *vpc-endpoint-id* 和 ## 替换为您的 VPC 端点 ID 和区域。

```
vpc-endpoint-id.data.iotsitewise.region.vpce.amazonaws.com
```

- 对于控制平面 API 操作，请使用以下端点 URL。将 *vpc-endpoint-id* 和 ## 替换为您的 VPC 端点 ID 和区域。

```
vpc-endpoint-id.api.iotsitewise.region.vpce.amazonaws.com
```

#### 2. 禁用主机前缀注入功能。当您调用每个 API 操作时，AWS CLI 和 AWS SDK 会在服务端点前面加上各种主机前缀。此功能会导致 AWS CLI 和 AWS 软件开发工具包生成在您指定 VPC 终端节点 AWS IoT SiteWise 时无效的 URL。

#### Important

您无法在 AWS CLI 或中禁用主机前缀注入 AWS Tools for PowerShell。这意味着，如果您禁用私有 DNS，则无法使用这些工具 AWS IoT SiteWise 通过 VPC 终端节点进行访问。允许私有 DNS 使用 AWS CLI 或 AWS IoT SiteWise 通过终端节点 AWS Tools for PowerShell 进行访问。

有关如何在 SDK 中禁用主机前缀注入的更多信息，请参阅每个 AWS SDK 的以下文档部分：

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Go v2](#)

- [AWS SDK for Java](#)
- [AWS SDK for Java 2.x](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for .NET](#)
- [AWS SDK for PHP](#)
- [AWS SDK for Python \(Boto3\)](#)
- [AWS SDK for Ruby](#)

有关更多信息，请参阅《Amazon VPC 用户指南》中的[通过接口端点访问服务](#)。

## 为创建 VPC 终端节点策略 AWS IoT SiteWise

您可以为 VPC 端点附加控制对 AWS IoT SiteWise 的访问的端点策略。该策略指定以下信息：

- 可执行操作的主体。
- 可执行的操作。
- 可对其执行操作的资源。

有关更多信息，请参阅《Amazon VPC 用户指南》中的[使用 VPC 端点控制对服务的访问权限](#)。

示例：用于 AWS IoT SiteWise 操作的 VPC 终端节点策略

以下是的终端节点策略示例 AWS IoT SiteWise。当连接到终端节点时，此策略授予用户访问 *iotsitewiseadmin* 在 AWS 账户 *123456789012* 中列出的对指定资产的 AWS IoT SiteWise 操作的访问权限。

```
{
  "Statement": [
    {
      "Action": [
        "iotsitewise:CreateAsset",
        "iotsitewise:ListGateways",
        "iotsitewise:ListTagsForResource"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:iotsitewise:us-west-2:123456789012:asset/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
      "Principal": {
```



```
        "AWS": [
            "123456789012:user/iotsitewiseadmin"
        ]
    }
}
]
```

## 以下方面的安全最佳实践 AWS IoT SiteWise

本主题包含的安全最佳实践 AWS IoT SiteWise。

### 在 OPC-UA 服务器上使用身份验证凭证

连接到您的 OPC-UA 服务器需要身份验证凭证。要执行此操作，请参考适用于您的服务器的文档。然后，要允许您的 SiteWise Edge 网关连接到 OPC-UA 服务器，请向 SiteWise Edge 网关添加服务器身份验证密钥。有关更多信息，请参阅 [配置源身份验证](#)。

### 为 OPC-UA 服务器使用加密通信模式

在为 Edge 网关配置 OPC-UA 源时，请选择未弃用的加密消息安全模式。SiteWise 这有助于在工业数据从 OPC-UA 服务器转移到 SiteWise Edge 网关时对其进行保护。有关更多信息，请参阅 [通过本地网络传输的数据](#) 和 [配置数据源](#)。

### 保持组件更新

如果您使用 SiteWise Edge 网关向服务采集数据，则您有责任配置和维护 SiteWise Edge 网关的环境。此责任包括升级到网关的系统软件、AWS IoT Greengrass 软件和连接器的最新版本。

#### Note

AWS IoT SiteWise Edge 连接器在您的文件系统上存储机密。这些密钥控制谁可以查看您的 SiteWise Edge 网关中缓存的数据。强烈建议您为运行 SiteWise Edge 网关的系统开启磁盘或文件系统加密。

## 加密您的 SiteWise Edge 网关的文件系统

加密和保护您的 SiteWise Edge 网关，因此您的工业数据在通过 SiteWise Edge 网关时是安全的。如果您的 SiteWise Edge 网关具有硬件安全模块，则可以进行配置 AWS IoT Greengrass 以保护您的

SiteWise Edge 网关。有关更多信息，请参阅 AWS IoT Greengrass Version 1 开发人员指南中的[硬件安全性集成](#)。否则，请参阅适用于您的操作系统的文档，了解如何加密和保护文件系统。

## 安全访问您的边缘配置

不要共享您的 Edge 控制台应用程序密码或 Mon SiteWise itor 应用程序密码。请勿将此密码放在任何人都能看到的地方。为密码配置适当的过期时间，实施健康的密码轮换策略。

## 向 SiteWise 监控器用户授予可能的最低权限

通过对门户用户使用最小访问策略权限集，遵循最低权限原则。

- 创建门户时，请定义一个角色，允许该门户所需的最小资源集。有关更多信息，请参阅[将服务角色用于 AWS IoT SiteWise Monitor](#)。
- 当您和门户管理员创建并共享项目时，请使用该项目所需的最小资源集。
- 当身份不再需要访问门户或项目时，请将其从该资源中删除。如果该身份不再适用于您的组织，请从您的身份存储中删除该身份。

最低权限原则最佳实践也适用于 IAM 角色。有关更多信息，请参阅[策略最佳实践](#)。

## 不要暴露敏感信息

您应该禁止记录凭证和其他敏感信息，例如个人身份信息 (PII)。尽管访问 SiteWise Edge 网关上的本地日志需要根权限，而访问 CloudWatch 日志需要 IAM 权限，但我们仍建议您实施以下保护措施。

- 请勿在资产或模型的名称、描述或属性中使用敏感信息。
- 请勿在 SiteWise Edge 网关或源名称中使用敏感信息。
- 请勿在门户、项目或控制面板的名称或描述中使用敏感信息。

## 遵循 AWS IoT Greengrass 安全最佳实践

遵循 SiteWise Edge 网关 AWS IoT Greengrass 的安全最佳实践。有关更多信息，请参阅 AWS IoT Greengrass Version 1 开发人员指南中的[安全最佳实践](#)。

## 另请参阅

- AWS IoT 开发人员指南中的[安全最佳实践](#)

- [工业物联网解决方案的十大安全黄金法则](#)

# 登录和监控 AWS IoT SiteWise

监控是维护和其他 AWS 解决方案的可靠性、可用性和性能的重要组成部分。AWS IoT SiteWise AWS IoT SiteWise 支持以下监控工具来监视服务，在出现问题时进行报告，并在适当时自动采取措施：

- Amazon 会实时 CloudWatch 监控您的 AWS 资源和您运行 AWS 的应用程序。收集和跟踪指标，创建自定义仪表板，并设置警报，以便在指定指标达到特定阈值时通知您或采取行动。例如，您可以 CloudWatch 跟踪您的 Amazon EC2 实例的 CPU 使用率或其他指标，并在需要时自动启动新实例。有关更多信息，请参阅 [Amazon CloudWatch 用户指南](#)。
- Amazon CloudWatch Logs 监控、存储和访问来自 SiteWise 边缘网关和其他来源的日志文件。CloudTrail CloudWatch 日志可以监视日志文件中的信息，并在达到特定阈值时通知您。您还可以在高持久性存储中检索您的日志数据。有关更多信息，请参阅 [Amazon CloudWatch 日志用户指南](#)。
- AWS CloudTrail 捕获由您的账户或代表您的 AWS 账户进行的 API 调用和相关事件。然后，将日志文件 CloudTrail 传送到您指定的 Amazon S3 存储桶。您可以识别哪些用户和帐户拨打了电话 AWS、发出呼叫的源 IP 地址以及呼叫发生的时间。有关更多信息，请参阅 [AWS CloudTrail 用户指南](#)。

## 主题

- [使用 Amazon CloudWatch 日志进行监控](#)
- [监控 SiteWise 边缘网关日志](#)
- [AWS IoT SiteWise 使用 Amazon CloudWatch 指标进行监控](#)
- [使用记录 AWS IoT SiteWise API 调用 AWS CloudTrail](#)

## 使用 Amazon CloudWatch 日志进行监控

配置 AWS IoT SiteWise 为将信息记录到 CloudWatch 日志，以监控服务并对其进行故障排除。

使用 AWS IoT SiteWise 控制台时，AWS IoT SiteWise 会创建一个服务相关角色，允许服务代表您记录信息。如果您不使用 AWS IoT SiteWise 控制台，则必须手动创建服务相关角色才能接收日志。有关更多信息，请参阅 [为 AWS IoT SiteWise 创建服务相关角色](#)。

您必须有一个允许将日志事件放入 AWS IoT SiteWise 入 CloudWatch 流中的资源策略。要创建和更新 CloudWatch 日志的资源策略，请运行以下命令。*logging-policy-name* 替换为要创建的策略的名称。

```
aws logs put-resource-policy --policy-name logging-policy-name --policy-
document "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Sid\":
  \"IoTSiteWiseToCloudWatchLogs\", \"Effect\": \"Allow\", \"Principal\": { \"Service\":
  [ \"iotsitewise.amazonaws.com\" ] }, \"Action\": \"logs:PutLogEvents\", \"Resource\":
  \"*\"} ] }"
```

CloudWatch 日志还支持 a [ws: SourceArn](#) 和 a [ws: SourceAccount](#) 条件上下文密钥。这些条件上下文键是可选的。

要创建或更新仅允许 AWS IoT SiteWise 将与指定资源关联的日志放入 CloudWatch 流的 AWS IoT SiteWise 资源策略，请运行命令并执行以下操作：

- *logging-policy-name* 替换为要创建的策略的名称。
- 将 *Source-ARN* 替换为 AWS IoT SiteWise 资源的 ARN，例如资产模型或资产。要查找每种 AWS IoT SiteWise 资源类型的 ARN，请参阅《服务授权参考》AWS IoT SiteWise 中[定义的资源类型](#)。
- 将 *## ID #* 换为与指定 AWS IoT SiteWise 资源关联的 AWS 账户 ID。

```
aws logs put-resource-policy --policy-name logging-policy-name --policy-
document "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Sid\":
  \"IoTSiteWiseToCloudWatchLogs\", \"Effect\": \"Allow\", \"Principal\": { \"Service
  \": [ \"iotsitewise.amazonaws.com\" ] }, \"Action\": \"logs:PutLogEvents\", \"Resource
  \": \"*\", \"Condition\": { \"StringLike\": { \"aws:SourceArn\": [ \"source-ARN\" ],
  \"aws:SourceAccount\": [ \"account-ID\" ] } } } ] }"
```

默认情况下，AWS IoT SiteWise 不将信息记录到 CloudWatch 日志。要激活日志记录，请选择“禁用” (OFF) 以外的日志级别。AWS IoT SiteWise 支持以下日志级别：

- OFF – 日志记录处于关闭状态。
- ERROR – 记录错误。
- INFO – 记录错误和信息性消息。

您可以将 SiteWise Edge 网关配置为通过 Lo CloudWatch gs 记录信息 AWS IoT Greengrass。有关更多信息，请参阅 [监控 SiteWise 边缘网关日志](#)。

如果 AWS IoT Core 要对 AWS IoT SiteWise 规则操作进行故障排除，也可以配置为将信息 CloudWatch 记录到日志。有关更多信息，请参阅 [对 AWS IoT SiteWise 规则操作进行故障排除](#)。

## 目录

- [管理登录 AWS IoT SiteWise](#)
  - [查找您的日志级别](#)
  - [更改您的日志级别](#)
- [示例：AWS IoT SiteWise 日志文件条目](#)

## 管理登录 AWS IoT SiteWise

使用 AWS IoT SiteWise 控制台或 AWS CLI 执行以下日志配置任务。

### 查找您的日志级别

#### Console

使用以下过程可在 AWS IoT SiteWise 控制台中查找当前的日志记录级别。

要查找您当前的 AWS IoT SiteWise 日志级别

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在左侧导航窗格中，选择日志记录选项。

当前日志记录状态显示在日志记录状态下。如果已激活日志记录，则当前日志记录级别将显示在详细程度级别下。

#### AWS CLI

运行以下命令以查找您当前的 AWS IoT SiteWise 日志记录级别 AWS CLI。

```
aws iotsitewise describe-logging-options
```

此操作将采用以下格式返回包含日志记录级别的响应。

```
{
  "loggingOptions": {
    "level": "String"
  }
}
```

## 更改您的日志级别

使用以下步骤在 AWS IoT SiteWise 控制台或使用中更改日志记录级别 AWS CLI。

### Console

更改您的 AWS IoT SiteWise 日志级别

1. 导航到 [AWS IoT SiteWise 控制台](#)。
2. 在左侧导航窗格中，选择日志记录选项。
3. 选择编辑。
4. 选择要激活的详细程度级别。
5. 选择保存。

### AWS CLI

运行以下 AWS CLI 命令来更改您的 AWS IoT SiteWise 日志级别。将 *logging-level* 替换为所需的日志记录级别。

```
aws iotsitewise put-logging-options --logging-options level=logging-level
```

## 示例：AWS IoT SiteWise 日志文件条目

每个 AWS IoT SiteWise 日志条目都包含事件信息和该事件的相关资源，因此您可以了解和分析日志数据。

以下示例显示了成功创建资产模型时 AWS IoT SiteWise 记录的 CloudWatch 日志条目。

```
{
  "eventTime": "2020-05-05T00:10:22.902Z",
  "logLevel": "INFO",
  "eventType": "AssetModelCreationSuccess",
  "message": "Successfully created asset model.",
  "resources": {
    "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
  }
}
```

## 监控 SiteWise 边缘网关日志

您可以将 AWS IoT SiteWise Edge 网关配置为将信息记录到 Amazon CloudWatch Logs 或本地文件系统。

### 主题

- [使用 Amazon CloudWatch 日志](#)
- [使用服务日志](#)
- [使用事件日志](#)

## 使用 Amazon CloudWatch 日志

您可以将 SiteWise Edge 网关配置为向日志发送 CloudWatch 日志。有关更多信息，请参阅《AWS IoT Greengrass Version 2 开发人员指南》中的[启用 CloudWatch 日志记录](#)。

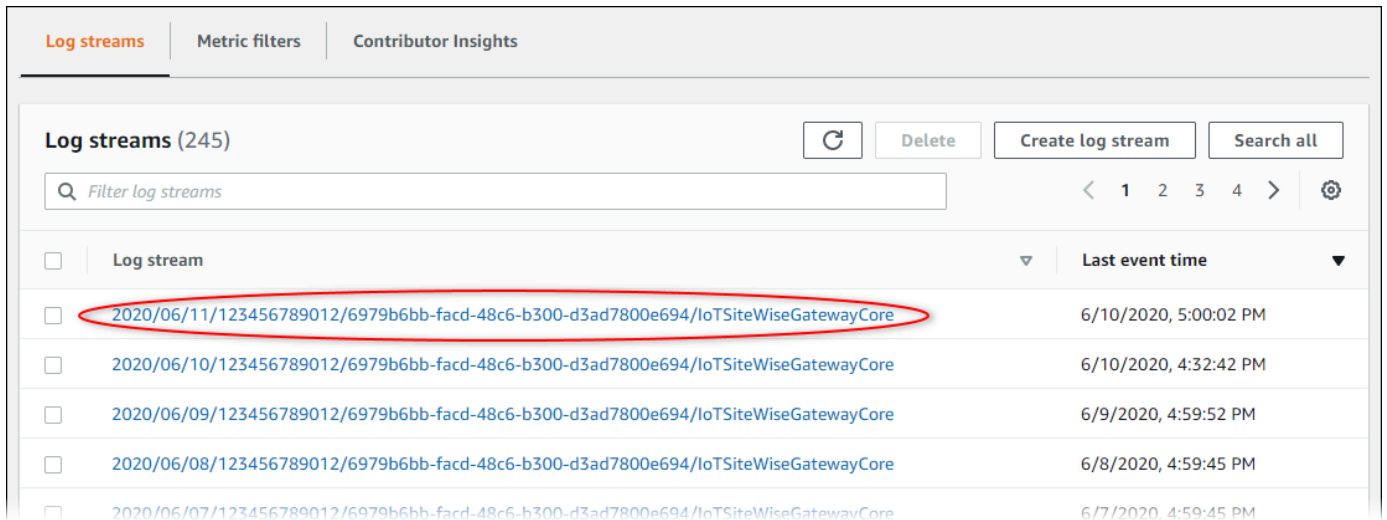
### 配置和访问 CloudWatch 日志 (控制台)

1. 导航到 [CloudWatch 控制台](#)。
2. 在导航窗格中，选择 日志组。
3. 您可以在以下日志 AWS IoT SiteWise 组中找到组件日志：
  - `/aws/greengrass/UserComponent/region/aws.iot.SiteWiseEdgeCollectorOpcua`— 边 SiteWise 边缘网关组件的日志，该组件从边 SiteWise 边缘网关的 OPC-UA 源收集数据。
  - `/aws/greengrass/UserComponent/region/aws.iot.SiteWiseEdgePublisher`— 将 OPC-UA 数据流发布到的 SiteWise 边缘网关组件的日志。AWS IoT SiteWise

选择要调试的函数的日志组。

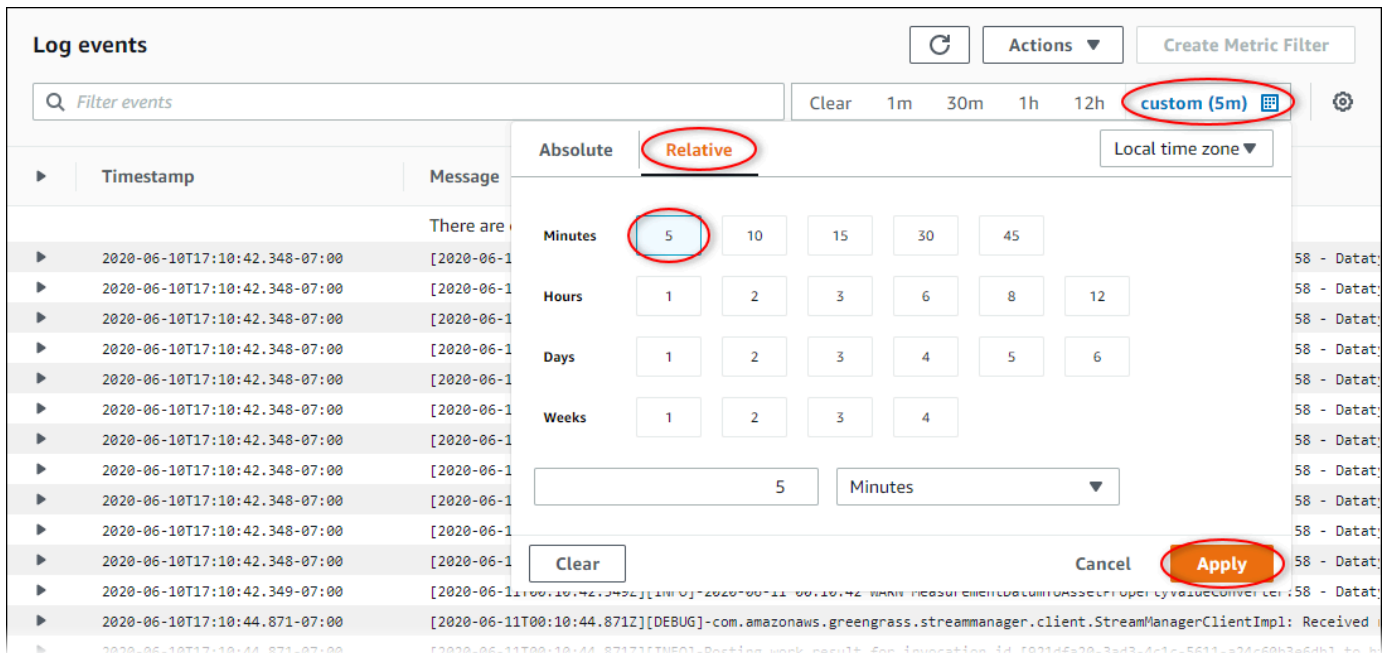
4. 选择名称以您的 AWS IoT Greengrass 群组名称结尾的日志流。默认情况下，首先 CloudWatch 显示最新的日志流。





5. 要显示最近 5 分钟的日志，请执行以下操作：

- a. 选择右上角的 自定义。
- b. 选择 相对。
- c. 选择 5 分钟。
- d. 选择 应用。



6. (可选) 要查看更少的日志，可以从右上角选择 1 分钟。

7. 滚动到日志条目底部可以显示最近的日志。

## 使用服务日志

SiteWise Edge 网关设备包含服务日志文件以帮助调试问题。以下各节将帮助您查找和使用 AWS IoT SiteWise OPC-UA Collector 和 P AWS IoT SiteWise Publisher 组件的服务日志文件。

### AWS IoT SiteWise OPC-UA 收集器服务日志文件

AWS IoT SiteWise OPC-UA 收集器组件使用以下日志文件。

#### Linux

```
/greengrass/v2/logs/aws.iot.SiteWiseEdgeCollectorOpcua.log
```

#### Windows

```
C:\greengrass\v2\logs\aws.iot.SiteWiseEdgeCollectorOpcua.log
```

#### 查看此组件的日志

- 在核心设备上运行以下命令以实时查看此组件的日志文件。将 */greengrass/v2* 或 *C:\greengrass\v2* 替换为 AWS IoT Greengrass 根文件夹的路径。

#### Linux

```
sudo tail -f /greengrass/v2/logs/aws.iot.SiteWiseEdgeCollectorOpcua.log
```

#### Windows (PowerShell)

```
Get-Content C:\greengrass\v2\logs\aws.iot.SiteWiseEdgeCollectorOpcua.log -Tail 10 -Wait
```

### AWS IoT SiteWise 发布者服务日志文件

P AWS IoT SiteWise Publisher 组件使用以下日志文件。

#### Linux

```
/greengrass/v2/logs/aws.iot.SiteWiseEdgePublisher.log
```

## Windows

```
C:\greengrass\v2\logs\aws.iot.SiteWiseEdgePublisher.log
```

### 查看此组件的日志

- 在核心设备上运行以下命令以实时查看此组件的日志文件。将 `/greengrass/v2` 或 `C:\greengrass\v2` 替换为 AWS IoT Greengrass 根文件夹的路径。

## Linux

```
sudo tail -f /greengrass/v2/logs/aws.iot.SiteWiseEdgePublisher.log
```

## Windows (PowerShell)

```
Get-Content C:\greengrass\v2\logs\aws.iot.SiteWiseEdgePublisher.log -Tail 10 -Wait
```

## 使用事件日志

SiteWise Edge 网关设备包含事件日志文件，以帮助调试问题。以下各节将帮助您查找和利用 AWS IoT SiteWise OPC-UA 收集器和 AWS IoT SiteWise 发布者组件的事件日志文件。

### AWS IoT SiteWise OPC-UA 收集器事件日志

AWS IoT SiteWise OPC-UA Collector 组件包括事件日志，可帮助客户识别和修复问题。日志文件与本地日志文件分开，位于以下位置。将 `/greengrass/v2` 或 `C:\greengrass\v2` 替换为 AWS IoT Greengrass 根文件夹的路径。

## Linux

```
/greengrass/v2/work/aws.iot.SiteWiseEdgeCollector0pcua/logs/  
IotSiteWiseOpcUaCollectorEvents.log
```

## Windows

```
C:\greengrass\v2\work\aws.iot.SiteWiseEdgeCollector0pcua\logs  
\IotSiteWiseOpcUaCollectorEvents.log
```

此日志包含详细信息和故障排除说明。故障排除信息与诊断一起提供，并附有如何解决问题的说明，有时还会提供指向更多信息的链接。诊断信息包括以下内容：

- 严重性级别
- Timestamp
- 其他活动特定信息

### Example 示例日志

```
dataSourceConnectionSuccess:
  Summary: Successfully connected to OpcUa server
  Level: INFO
  Timestamp: '2023-06-15T21:04:16.303Z'
  Description: Successfully connected to the data source.
  AssociatedMetrics:
  - Name: FetchedDataStreams
    Description: The number of fetched data streams for this data source
    Value: 1.0
    Namespace: IoTSiteWise
    Dimensions:
    - Name: SourceName
      Value: SourceName{value=OPC-UA Server}
    - Name: ThingName
      Value: test-core
  AssociatedData:
  - Name: DataSourceTrace
    Description: Name of the data source
    Data:
    - OPC-UA Server
  - Name: EndpointUri
    Description: The endpoint to which the connection was attempted.
    Data:
    - '"opc.tcp://10.0.0.1:1234"'
```

### AWS IoT SiteWise 发布者事件日志

AWS IoT SiteWise Publisher 组件包括事件日志，可帮助客户识别和修复问题。日志文件与本地日志文件分开，位于以下位置。将 `/greengrass/v2` 或 `C:\greengrass\v2` 替换为 AWS IoT Greengrass 根文件夹的路径。

## Linux

```
/greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/logs/  
IotSiteWisePublisherEvents.log
```

## Windows

```
C:\greengrass\v2\work\aws.iot.SiteWiseEdgePublisher\logs  
\IotSiteWisePublisherEvents.log
```

此日志包含详细信息和故障排除说明。故障排除信息与诊断一起提供，并附有如何解决问题的说明，有时还会提供指向更多信息的链接。诊断信息包括以下内容：

- 严重性级别
- Timestamp
- 其他活动特定信息

## Example 示例日志

```
accountBeingThrottled:  
  Summary: Data upload speed slowed due to quota limits  
  Level: WARN  
  Timestamp: '2023-06-09T21:30:24.654Z'  
  Description: The IoT SiteWise Publisher is limited to the "Rate of data points  
  ingested"  
  quota for a customers account. See the associated documentation and associated  
  metric for the number of requests that were limited for more information. Note  
  that this may be temporary and not require any change, although if the issue  
  continues  
  you may need to request an increase for the mentioned quota.  
  FurtherInformation:  
  - https://docs.aws.amazon.com/iot-sitewise/latest/userguide/quotas.html  
  - https://docs.aws.amazon.com/iot-sitewise/latest/userguide/troubleshooting-gateway.html#gateway-issue-data-streams  
  AssociatedMetrics:  
  - Name: TotalErrorCount  
    Description: The total number of errors of this type that occurred.  
    Value: 327724.0  
  AssociatedData:  
  - Name: AggregatePropertyAliases
```

Description: The aggregated property aliases of the throttled data.

FileLocation: /greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/./logs/data/  
AggregatePropertyAliases\_1686346224654.log

## AWS IoT SiteWise 使用 Amazon CloudWatch 指标进行监控

您可以使用 AWS IoT SiteWise 进行监控 CloudWatch，它收集原始数据并将其处理为可读的近乎实时的指标。这些统计数据会保存 15 个月，从而使您能够访问历史信息，并能够更好地了解您的 Web 应用程序或服务的执行情况。此外，可以设置用于监测特定阈值的警报，并在达到相应阈值时发送通知或执行操作。有关更多信息，请参阅 [Amazon CloudWatch 用户指南](#)。

AWS IoT SiteWise 将以下各节中列出的指标和维度发布到 AWS/IoTSiteWise 命名空间。

### Tip

AWS IoT SiteWise 每隔一分钟发布指标。当您在 CloudWatch 控制台的图表中查看这些指标时，我们建议您选择 1 分钟的时间段。这样，您便能查看指标数据的最高可用解析。

### 主题

- [AWS IoT Greengrass Version 2 网关指标](#)
- [AWS IoT Greengrass Version 1 网关指标](#)

## AWS IoT Greengrass Version 2 网关指标

AWS IoT SiteWise 发布以下 SiteWise Edge 网关指标。所有 SiteWise Edge 网关指标每隔一分钟发布一次。

### SiteWise 边缘网关指标

指标	描述
Gateway.CpuUsage	SiteWise 边缘网关的 CPU 使用率。  单位：百分比  维度：无
Gateway.TotalDiskSpace	SiteWise Edge 网关的总磁盘空间。

指标	描述
	单位：字节 维度：无
Gateway.UsedDiskSpace	SiteWise Edge 网关的已用磁盘空间。 单位：字节 维度：无
Gateway.AvailableDiskSpace	SiteWise Edge 网关的可用磁盘空间。 单位：字节 维度：无
Gateway.UsedPercentageDiskSpace	SiteWise Edge 网关占磁盘空间的已用百分比。 单位：字节 维度：无
Gateway.TotalMemory	SiteWise 边缘网关的总内存。 单位：字节 维度：无
Gateway.UsedMemory	SiteWise 边缘网关的已用内存。 单位：字节 维度：无
Gateway.AvailableMemory	SiteWise 边缘网关的可用内存。 单位：字节 维度：无

指标	描述
<code>Gateway.UsedPercentageMemory</code>	SiteWise Edge 网关的已用内存百分比。  单位：字节  维度：无
<code>Gateway.CloudConnectivity</code>	SiteWise 边缘网关的云连接状态。  单位：无  尺寸：GatewayId
<code>Gateway.SWE.Component.RunningStatus</code>	SiteWise Edge 网关上组件的运行状态。  单位：无  尺寸：GatewayId

### OPC-UA 收集器指标

指标	描述
<code>OpcUaCollector.Heartbeat</code>	每分钟为连接到 SiteWise Edge 网关 (sourceName ) 的每个 OPC-UA 源 (gatewayId ) 生成一次。  单位：计数 ( 1 表示源已连接 , 0 表示源已断开连接。 )  尺寸：GatewayId , SourceName
<code>OpcUaCollector.ActiveDataStreamCount</code>	SiteWise 边缘网关 (gatewayId ) 为 OPC-UA 源 () 订阅的数据流数量。sourceName  单位：计数  尺寸: GatewayId, SourceName, PropertyGroup



指标	描述
<code>OpcUaCollector.IncomingValuesCount</code>	<p>SiteWise 边缘网关 (gatewayId ) 每分钟为 OPC-UA 源 (sourceName ) 接收的数据点数量。</p> <p>单位 : 计数</p> <p>尺寸: GatewayId, SourceName, PropertyGroup</p>
<code>OpcUaCollector.IncomingValuesError</code>	<p>SiteWise 边缘网关 (gatewayId ) 从 OPC-UA 源 (sourceName ) 接收的无效值的数据点数量。这些数据点不是由 OpcUa 收集器摄取的 , 而是每分钟生成一次。</p> <p>单位 : 计数</p> <p>尺寸: GatewayId, SourceName, PropertyGroup</p>
<code>OpcUaCollector.ConversionErrors</code>	<p>SiteWise 边缘网关 (gatewayId ) 接收到的 OPC-UA 源 (sourceName ) 的数据点数量 , 这些数据点在向其发送数据时导致转换错误。AWS IoT SiteWise 这些数据点不会被 C OpcUa collector 摄取。</p> <p>单位 : 计数</p> <p>尺寸 : GatewayId , SourceName</p>

### AWS IoT SiteWise 处理器指标

指标	描述
<code>Gateway.DataProcessor.IngestionSuccess</code>	<p>成功摄取的数据点数 , 每分钟生成。</p> <p>单位 : 计数</p> <p>维度 : None</p>

指标	描述
<code>Gateway.DataProcessor.IngestionThrottled</code>	<p>每分钟生成的被限制的数据点数量。</p> <p>单位：计数</p> <p>尺寸：ThrottledAt</p>
<code>Gateway.DataProcessor.MeasurementRejected</code>	<p>每分钟生成的被拒绝的测量次数。</p> <p>单位：计数</p> <p>尺寸：原因</p>
<code>Gateway.DataProcessor.MeasurementUnmodeled</code>	<p>每分钟生成的未建模测量值的数量。</p> <p>单位：计数</p> <p>尺寸：原因</p>
<code>Gateway.DataProcessor.MessagesRemaining</code>	<p>每分钟生成的流中剩余的消息数。</p> <p>单位：计数</p> <p>尺寸：StreamName</p>
<code>Gateway.DataProcessor.ProcessingError</code>	<p>每分钟生成的处理错误数。</p> <p>单位：计数</p> <p>尺寸：原因</p>
<code>IoTSiteWiseProcessor.IsConnectedToMqttBroker</code>	<p>由 SiteWise Edge 网关中的处理器每分钟生成一次。</p> <p>单元：1 ( 1 表示处理器已连接到 MQTT 代理。 )</p> <p>尺寸：GatewayId</p>

指标	描述
<code>IoTSiteWiseProcessor.NumberOfSubscriptionsToMqttBroker</code>	<p>处理器每分钟生成的订阅 MQTT 代理的主题数量。多级通配符主题计为 1。</p> <p>单位：计数</p> <p>尺寸：GatewayId</p>
<code>IoTSiteWiseProcessor.NumberOfUniqueMqttTopicsReceived</code>	<p>处理器每分钟从 MQTT 代理收到的唯一主题的数量。</p> <p>单位：计数</p> <p>尺寸：GatewayId</p>
<code>IoTSiteWiseProcessor.MqttMessageReceivedSuccessCount</code>	<p>处理器成功接收来自 MQTT 代理的消息数，每分钟生成。</p> <p>单位：计数</p> <p>尺寸：GatewayId</p>
<code>IoTSiteWiseProcessor.MqttReceivedSuccessBytes</code>	<p>每分钟生成处理器从 MQTT 代理成功接收的消息数据的字节数。</p> <p>单位：计数</p> <p>尺寸：GatewayId</p>

### AWS IoT SiteWise 发布者指标

指标	描述
<code>IoTSiteWisePublisher.Heartbeat</code>	<p>由发布者在 SiteWise Edge 网关中每分钟生成一次。</p> <p>单位：1 ( 1 表示发布服务器正在运行，但缺少表示发布服务器未运行的数据点。 )</p>

指标	描述
	尺寸 : GatewayId
<code>IoTSiteWisePublisher.PublisherSuccessCount</code>	SiteWise 边缘网关 (GatewayId ) 每分钟成功发布到云端的数据点数量。  单位 : 计数  尺寸 : GatewayId
<code>IoTSiteWisePublisher.PublisherFailureCount</code>	SiteWise 边缘网关 (GatewayId ) 发布失败的数据点数量 , 每分钟生成。  单位 : 计数  尺寸 : GatewayId
<code>IoTSiteWisePublisher.PublisherRejectedCount</code>	SiteWise Edge 网关 (GatewayId ) 每分钟从云端拒绝的数据点数量。  单位 : 计数  尺寸 : GatewayId
<code>IoTSiteWisePublisher.DroppedCount</code>	由 SiteWise Edge 网关 (GatewayId ) 丢弃但未发布到云端的数据点数量 , 每分钟生成一次。  单位 : 计数  尺寸 : GatewayId
<code>IoTSiteWisePublisher.IsConnectedToMqttBroker</code>	由发布者在 SiteWise Edge 网关中每分钟生成一次。  单位 : 1 ( 1 表示发布者已连接到 MQTT 代理。 )  尺寸 : GatewayId

指标	描述
<code>IoTSiteWisePublisher.NumberOfSubscriptionsToMqttBroker</code>	发布者每分钟生成的订阅 MQTT 代理的主题数量。多级通配符主题计为 1。  单位：计数  尺寸：GatewayId
<code>IoTSiteWisePublisher.NumberOfUniqueMqttTopicsReceived</code>	发布者每分钟从 MQTT 代理收到的唯一主题的数量。  单位：计数  尺寸：GatewayId
<code>IoTSiteWisePublisher.MqttMessageReceivedSuccessCount</code>	发布者成功接收来自 MQTT 代理的消息数，每分钟生成。  单位：计数  尺寸：GatewayId
<code>IoTSiteWisePublisher.MqttReceivedSuccessBytes</code>	发布者每分钟从 MQTT 代理成功接收的消息数据的字节数。  单位：计数  尺寸：GatewayId

## AWS IoT Greengrass Version 1 网关指标

AWS IoT SiteWise 发布以下 SiteWise Edge 网关指标。所有 SiteWise Edge 网关指标每隔一分钟发布一次。

**⚠ Important**

要接收 SiteWise Edge 网关指标，您必须在 E SiteWise dge 网关上使用至少版本 6 的 AWS IoT SiteWise 连接器。有关更多信息，请参阅 [AWS IoT Greengrass Version 1 开发人员指南](#) 中的 [AWS IoT SiteWise OPC-UA 收集器](#)。

## SiteWise 边缘网关指标

指标	描述
Gateway.Heartbeat	<p>每分钟为连接的每个 SiteWise Edge 网关 (gatewayId ) 生成一次。</p> <p>单位：1 ( 1 表示 SiteWise Edge 网关已启动，缺少表示 SiteWise Edge 网关与云断开连接的数据点。 )</p> <p>尺寸：GatewayId</p>
Gateway.PublishSuccessCount	<p>SiteWise Edge 网关 (gatewayId ) 成功发布的数据点数量。</p> <p>单位：计数</p> <p>尺寸：GatewayId</p>
Gateway.PublishFailureCount	<p>SiteWise Edge 网关 (gatewayId ) 未能发布的数据点数量。</p> <p>该指标统计由 SiteWise Edge 网关对 <a href="#">BatchPutAssetPropertyValue</a> 操作的调用所产生的错误。有关 SiteWise Edge 网关故障排除的更多信息，请参阅 <a href="#">对 SiteWise 边缘网关进行故障排除</a>。</p> <p>单位：计数</p> <p>尺寸：GatewayId</p>
Gateway.ProcessFailureCount	<p>SiteWise Edge 网关 (gatewayId ) 未能处理的数据点数量。</p>

指标	描述
	<p>该指标统计了 SiteWise Edge 网关和 Ed SiteWise ge 网关源之间发生的错误，包括来源报告的错误。有关 SiteWise Edge 网关故障排除的更多信息，请参阅<a href="#">对 SiteWise 边缘网关进行故障排除</a>。</p> <p>单位：计数</p> <p>尺寸：GatewayId</p>
Gateway.PublishRejectedCount	<p>来自 SiteWise Edge 网关 (gatewayId ) 的被拒绝的数据点的数量。</p> <p>单位：计数</p> <p>尺寸：GatewayId</p>

## OPC-UA 相关指标

指标	描述
OPCUACollector.Heartbeat	<p>每分钟为连接到 SiteWise Edge 网关 (sourceName ) 的每个 OPC-UA 源 (gatewayId ) 生成一次。</p> <p>单位：计数 ( 1 表示源已连接，0 表示源已断开连接。 )</p> <p>尺寸：GatewayId , SourceName</p>
OPCUACollector.ActiveDataStreamCount	<p>SiteWise 边缘网关 (gatewayId ) 为 OPC-UA 源 () 订阅的数据流数量。sourceName</p> <p>单位：计数</p> <p>尺寸: GatewayId, SourceName, PropertyGroup</p>

指标	描述
<code>OpcUaCollector.IncomingValuesCount</code>	<p>SiteWise 边缘网关 (gatewayId ) 每分钟为 OPC-UA 源 (sourceName ) 接收的数据点数量。</p> <p>单位：计数</p> <p>尺寸: GatewayId, SourceName, PropertyGroup</p>
<code>OpcUaCollector.IncomingValuesError</code>	<p>SiteWise 边缘网关 (gatewayId ) 从 OPC-UA 源 (sourceName ) 收到的无效值的数据点数量。这些数据点不会被 OpcUa 收集器摄取，而是每分钟生成一次。</p> <p>单位：计数</p> <p>尺寸: GatewayId, SourceName, PropertyGroup</p>
<code>OpcUaCollector.ConversionErrors</code>	<p>SiteWise 边缘网关 (gatewayId ) 接收到的 OPC-UA 源 (sourceName ) 的数据点数量，这些数据点在向其发送数据时导致转换错误。AWS IoT SiteWise 这些数据点不会被 C OpcUa collector 摄取。</p> <p>单位：计数</p> <p>尺寸： GatewayId , SourceName</p>

## EIP 相关指标

指标	描述
<code>EIPCollector.Heartbeat</code>	<p>每分钟为连接到 E SiteWise dge 网关 (sourceName ) 的每个 EIP 源 (gatewayId ) 生成一次。</p> <p>单位：1 ( 1 表示源已连接，缺少表示源的数据点已断开连接。 )</p>



指标	描述
	尺寸：GatewayId , SourceName
EIPCollector.IncomingValuesCount	SiteWise Edge 网关 (gatewayId ) 为 EIP 源 () 订阅的数据流数量 (sourceName )。  单位：计数  尺寸：GatewayId , SourceName
EIPCollector.ActiveDataStreamCount	SiteWise 边缘网关 (gatewayId ) 为 EIP 源 () 接收的数据点数量 (sourceName )。  单位：计数  尺寸：GatewayId , SourceName

### 与 Modbus 相关的指标

指标	描述
ModbusTCPCollector.Heartbeat	每分钟为连接到 SiteWise 边缘网关 (sourceName ) 的每个 Modbus 源 (gatewayId ) 生成一次。  单位：1 ( 1 表示 Modbus 源已连接，缺少表示源的数据点已断开连接。 )  尺寸：GatewayId , SourceName
ModbusTCPCollector.IncomingValuesCount	SiteWise 边缘网关 (gatewayId ) 为 Modbus 源 () 订阅的数据流数量 (sourceName )。  单位：计数  尺寸：GatewayId , SourceName
ModbusTCPCollector.ActiveDataStreamCount	SiteWise 边缘网关 (gatewayId ) 接收到的 Modbus 源 () 的数据点数量 (sourceName )。

指标	描述
	单位：计数
	尺寸：GatewayId, SourceName

## 使用记录 AWS IoT SiteWise API 调用 AWS CloudTrail

AWS IoT SiteWise 与 AWS CloudTrail 一项服务集成，该服务提供用户、角色或 AWS 服务在中执行的操作的记录 AWS IoT SiteWise。CloudTrail 将发出的 API 调用捕获 AWS IoT SiteWise 为事件。捕获的调用包括来自 AWS IoT SiteWise 控制台的调用和对 AWS IoT SiteWise API 操作的代码调用。如果您创建跟踪，则可以激活向 Amazon S3 存储桶持续传输事件，包括的事件 AWS IoT SiteWise。CloudTrail 如果您未配置跟踪，您仍然可以在 CloudTrail 控制台的“事件历史记录”中查看最新的事件。使用收集的信息 CloudTrail，您可以确定向哪个请求发出 AWS IoT SiteWise、发出请求的 IP 地址、谁发出了请求、何时发出请求以及其他详细信息。

有关的更多信息 CloudTrail，请参阅 [《AWS CloudTrail 用户指南》](#)。

## AWS IoT SiteWise 信息在 CloudTrail

CloudTrail 在您创建 AWS 账户时已在您的账户上激活。当支持的事件活动发生在中时 AWS IoT SiteWise，该活动会与其他 AWS 服务 CloudTrail 事件一起记录在事件历史记录中。您可以在自己的 AWS 账户中查看、搜索和下载最近发生的事件。有关更多信息，请参阅 [使用事件历史查看 CloudTrail 事件](#)。

要持续记录您 AWS 账户中的事件，包括的事件 AWS IoT SiteWise，请创建跟踪。跟踪允许 CloudTrail 将日志文件传输到 Amazon S3 存储桶。默认情况下，当您在控制台中创建跟踪时，该跟踪将应用于所有 AWS 区域。跟踪记录 AWS 分区中所有区域的事件，并将日志文件传送到您指定的 Amazon S3 存储桶。此外，您可以配置其他 AWS 服务，以进一步分析和处理 CloudTrail 日志中收集的事件数据。有关更多信息，请参阅下列内容：

- [创建跟踪记录概览](#)
- [CloudTrail 支持的服务和集成](#)
- [配置 Amazon SNS 通知 CloudTrail](#)
- [接收来自多个区域的 CloudTrail 日志文件和接收来自多个账户的 CloudTrail 日志文件](#)

每个事件或日记账条目都包含有关生成请求的人员信息。身份信息有助于您确定以下内容：

- 请求是使用根证书还是 AWS Identity and Access Management (IAM) 用户凭证发出。
- 请求是使用角色还是联合用户的临时安全凭证发出的。
- 请求是否由其他 AWS 服务发出。

有关更多信息，请参阅 [CloudTrail userIdentity 元素](#)。

## AWS IoT SiteWise 中的数据事件 CloudTrail

[数据事件](#) 可提供对资源或在资源中所执行资源操作（例如，读取或写入 Amazon S3 对象）的相关信息。这些也称为数据层面操作。数据事件通常是高容量活动。默认情况下，CloudTrail 不记录数据事件。CloudTrail 事件历史记录不记录数据事件。

记录数据事件将收取额外费用。有关 CloudTrail 定价的更多信息，请参阅 [AWS CloudTrail 定价](#)。

您可以使用 CloudTrail 控制台或 CloudTrail API 操作记录 AWS IoT SiteWise 资源类型的数据事件。AWS CLI 本节中的 [表格](#) 显示了可用的资源类型 AWS IoT SiteWise。

- 要使用 CloudTrail 控制台记录数据事件，请创建 [跟踪](#) 或 [事件数据存储](#) 以记录数据事件，或者 [更新现有的跟踪或事件数据存储](#) 以记录数据事件。
  1. 选择数据事件以记录数据事件。
  2. 从数据事件类型列表中，选择要为其记录数据事件的资源类型。
  3. 选择要使用的日志选择器模板。您可以记录资源类型的所有数据事件、记录所有 readOnly 事件、记录所有 writeOnly 事件，或者创建自定义日志选择器模板来筛选 readOnlyeventName、和 resources.ARN 字段。
- 要使用记录数据事件 AWS CLI，请将 --advanced-event-selectors 参数配置为将 eventCategory 字段设置为等于 Data 并将该 resources.type 字段设置为资源类型值（参见 [表](#)）。您可以添加条件来筛选 readOnlyeventName、和 resources.ARN 字段的值。
  - 要配置记录数据事件的跟踪，请运行 [AWS CloudTrail put-event-selectors](#) 命令。有关更多信息，请参阅 [使用记录跟踪的数据事件 AWS CLI](#)。
  - 要将事件数据存储配置为记录数据事件，请运行 [AWS CloudTrail create-event-data-store](#) 命令创建新的事件数据存储以记录数据事件，或者运行 [AWS CloudTrail update-event-data-store](#) 命令来更新现有的事件数据存储。有关更多信息，请参阅 [使用记录事件数据存储的数据事件 AWS CLI](#)。

下表列出了 AWS IoT SiteWise 资源类型。数据事件类型（控制台）列显示要从控制 CloudTrail 台上的数据事件类型列表中选择 的值。resources.type 值列显示该 resources.type 值，您将在使用或

API 配置高级事件选择器时指定该值。AWS CLI CloudTrail “记录到的数据 API CloudTrail” 列显示了 CloudTrail 针对该资源类型记录的 API 调用。

数据事件类型 (控制台)	resources.type 值	数据 API 已记录到 CloudTrail *
AWS IoT SiteWise 资产	AWS::IoTSiteWise::Asset	<ul style="list-style-type: none"> <li>• <a href="#">BatchPutAssetPropertyValue</a></li> <li>• <a href="#">GetAssetPropertyValue</a></li> <li>• <a href="#">GetAssetPropertyValueHistory</a></li> <li>• <a href="#">GetAssetPropertyAggregates</a></li> <li>• <a href="#">GetInterpolatedAssetPropertyValues</a></li> <li>• <a href="#">BatchGetAssetPropertyValue</a></li> <li>• <a href="#">BatchGetAssetPropertyValueHistory</a></li> <li>• <a href="#">BatchGetAssetPropertyAggregates</a></li> </ul>
AWS IoT SiteWise 时间序列	AWS::IoTSiteWise::TimeSeries	<ul style="list-style-type: none"> <li>• <a href="#">BatchPutAssetPropertyValue</a></li> <li>• <a href="#">GetAssetPropertyValue</a></li> <li>• <a href="#">GetAssetPropertyValueHistory</a></li> <li>• <a href="#">GetAssetPropertyAggregates</a></li> <li>• <a href="#">GetInterpolatedAssetPropertyValues</a></li> <li>• <a href="#">BatchGetAssetPropertyValue</a></li> <li>• <a href="#">BatchGetAssetPropertyValueHistory</a></li> </ul>

数据事件类型 (控制台)	resources.type 值	数据 API 已记录到 CloudTrail *
		<ul style="list-style-type: none"> <li><a href="#">BatchGetAssetPropertyAggregates</a></li> </ul>

### Note

Cloudtrail 事件中记录的 resources.type 取决于 API 请求中使用的标识符。如果在请求中指定了资产 ID，则会记录资产 resources.type，否则会记录 resource TimeSeries es.type。

\*您可以将高级事件选择器配置为在 eventName、和 resources.ARN 字段上进行筛选 readOnly，以仅记录那些对您很重要的事件。有关这些字段的更多信息，请参阅 [AdvancedFieldSelector](#)。

## AWS IoT SiteWise 中的管理事件 CloudTrail

[管理事件](#) 提供有关对您 AWS 账户中的资源执行的管理操作的信息。这些也称为控制层面操作。默认情况下，CloudTrail 记录管理事件。

AWS IoT SiteWise 将所有 AWS IoT SiteWise 控制平面操作记录为管理事件。有关 AWS IoT SiteWise 记录到的 AWS IoT SiteWise 控制平面操作的列表 CloudTrail，请参阅 [AWS IoT SiteWise API 参考](#)。

### 示例：AWS IoT SiteWise 日志文件条目

跟踪是一种配置，允许将事件作为日志文件传输到您指定的 Amazon S3 存储桶。CloudTrail 日志文件包含一个或多个日志条目。事件代表来自任何来源的单个请求，包括有关所请求的操作、操作的日期和时间、请求参数等的信息。CloudTrail 日志文件不是公共 API 调用的有序堆栈跟踪，因此它们不会按任何特定的顺序出现。

以下示例显示了演示该 CreateAsset 操作的 CloudTrail 日志条目。

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/Administrator",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
```

```
"userName": "Administrator",
"sessionContext": {
  "sessionIssuer": {},
  "webIdFederationData": {},
  "attributes": {
    "mfaAuthenticated": "false",
    "creationDate": "2020-03-11T17:26:40Z"
  }
},
"invokedBy": "signin.amazonaws.com"
},
"eventTime": "2020-03-11T18:01:22Z",
"eventSource": "iotsitewise.amazonaws.com",
"eventName": "CreateAsset",
"awsRegion": "us-east-1",
"sourceIPAddress": "203.0.113.0",
"userAgent": "signin.amazonaws.com",
"requestParameters": {
  "assetName": "Wind Turbine 1",
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
  "clientToken": "a1b2c3d4-5678-90ab-cdef-00000EXAMPLE"
},
"responseElements": {
  "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetArn": "arn:aws:iotsitewise:us-east-1:123456789012:asset/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetStatus": {
    "state": "CREATING"
  }
},
"requestID": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
"eventID": "a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}
```

## 为资源添加 AWS IoT SiteWise 标签

为 AWS IoT SiteWise 资源添加标签提供了一种高效地分类、管理和检索组织资产的强大方法。通过分配由键值对组成的标签，您可以将描述性元数据附加到您的资源中。标签中的元数据可用于简化操作。例如，在风电场场景中，标签允许您使用位置、容量和运行状态等特定属性标记涡轮机，从而在 AWS IoT SiteWise 中快速识别和管理。

将标签与 AWS Identity and Access Management (IAM) 策略集成可通过定义条件访问规则来增强安全性和操作控制。这意味着您可以仅指定具有特定标签的用户。例如，只有那些被标记为特定角色或部门的人才能访问或修改特定的资源。

## 在中使用标签 AWS IoT SiteWise

使用标签按用途、所有者、环境或用例的任何其他分类对 AWS IoT SiteWise 资源进行分类。在您具有同一类型的许多资源时，可以根据特定资源的标签快速识别该资源。

每个标签都由您指定的密钥和可选值组成。例如，您可以为资产模型建立一系列标签，以便根据它们支持的工业流程对其进行跟踪。建议您管理的每种资源开发一组量身定制的标签密钥。使用一组一致的标签密钥可以更轻松地管理资源。

## 用 tagging AWS Management Console

中的标签编辑器 AWS Management Console 提供了一种集中、统一的方式，供您为来自所有 AWS 服务的资源创建和管理标签。有关更多信息，请参阅 AWS Resource Groups 用户指南中的[标签编辑器](#)。

## 使用 API 进行标 AWS IoT SiteWise 记

AWS IoT SiteWise API 还使用标签。在创建标签之前，请注意标签限制。有关更多信息，请参阅 AWS 一般参考中的[标签命名和使用惯例](#)。

- 要在创建资源时添加标签，请在资源的 `tags` 属性中定义这些标签。
- 要向现有资源添加标签或更新标签值，请使用[TagResource](#)操作。
- 要从资源中移除标签，请使用[UntagResource](#)操作。
- 要检索与资源关联的标签，请使用[ListTagsForResource](#)操作，或者描述资源并检查其 `tags` 属性。

下表列出了您可以使用 AWS IoT SiteWise API 标记的资源及其对应的 `Create` 和 `Describe` 操作。

## 可标记的资源 AWS IoT SiteWise

资源	创建操作	描述操作
资产模型或组件模型	<a href="#">CreateAssetModel</a>	<a href="#">DescribeAssetModel</a>
资产	<a href="#">CreateAsset</a>	<a href="#">DescribeAsset</a>
SiteWise 边缘网关	<a href="#">CreateGateway</a>	<a href="#">DescribeGateway</a>
门户	<a href="#">CreatePortal</a>	<a href="#">DescribePortal</a>
项目	<a href="#">CreateProject</a>	<a href="#">DescribeProject</a>
控制面板	<a href="#">CreateDashboard</a>	<a href="#">DescribeDashboard</a>
访问策略	<a href="#">CreateAccessPolicy</a>	<a href="#">DescribeAccessPolicy</a>
时间序列	<a href="#">BatchPutAssetPropertyValue</a>	<a href="#">DescribeTimeSeries</a>

对于[BatchPutAssetPropertyValue](#)，在创建资产模型和资产 AWS IoT SiteWise 之前，您可以配置要将工业数据发送到的数据源。AWS IoT SiteWise 自动创建数据流以接收来自您的设备的原始数据流。有关更多信息，请参阅[管理数据摄取](#)。

使用以下操作可查看和管理支持标记的资源的标签：

- [TagResource](#)— 向资源添加标签，或更新现有标签的值。
- [ListTagsForResource](#)— 列出资源的标签。
- [UntagResource](#)— 从资源中移除标签。

随时为资源添加或移除标签。要更新现有标签密钥的值，请向资源添加一个具有相同密钥和所需新值的新标签。此操作将旧值替换为新值。虽然可以将空字符串指定为标签值，但不能分配空值。

删除资源也会移除与其关联的所有标签。

## 在 IAM 策略中使用标签

在 IAM 策略中使用资源标签来控制用户的访问权限和权限。例如，策略可以允许用户仅创建附加了特定标签的资源。策略还可以限制用户创建或修改具有特定标签的资源。



**Note**

如果您使用标签来允许或拒绝用户对资源的访问，则应拒绝用户对相同资源添加或删除这些标签的能力。否则，用户可以绕过您的限制，通过修改资源的标签来获得对资源的访问权限。

可以在策略语句的 Condition 元素（也称为 Condition 块）中使用以下条件上下文键和值。

```
aws:ResourceTag/tag-key: tag-value
```

允许或拒绝带特定标签的资源上的操作。

```
aws:RequestTag/tag-key: tag-value
```

要求在创建或修改可标记的资源时使用（或不使用）特定标签。

```
aws:TagKeys: [tag-key, ...]
```

要求在创建或修改可标记的资源时使用（或不使用）一组特定的标签键。

**Note**

IAM policy 中的条件上下文键和值仅适用于将可标记的资源作为必需参数的操作。例如，您可以为设置基于标签的条件访问权限。[ListAssets](#)您无法将基于标签的条件访问设置为开启，[PutLoggingOptions](#)因为请求中未引用任何可标记的资源。

有关更多信息，请参阅 IAM 用户指南中的[使用 AWS 资源标签控制资源访问权限](#)和[IAM JSON 策略参考](#)。

使用标签的 IAM 策略示例

- [根据标签查看 AWS IoT SiteWise 资产](#)

# 故障排除 AWS IoT SiteWise

使用这些部分中的信息来排除和解决问题 AWS IoT SiteWise。

## 主题

- [批量导入和导出操作疑难解答](#)
- [对 AWS IoT SiteWise 门户进行故障排除](#)
- [对 SiteWise 边缘网关进行故障排除](#)
- [对 AWS IoT SiteWise 规则操作进行故障排除](#)

## 批量导入和导出操作疑难解答

要处理和诊断传输任务期间产生的错误，请参阅 AWS IoT TwinMaker GetMetadataTransferJobAPI：

1. 创建并运行传输任务后，调用 GetMetadataTransferJobAPI：

```
aws iottwinmaker get-metadata-transfer-job \  
--metadata-transfer-job-id your_metadata_transfer_job_id \  
--region us-east-1
```

2. 作业状态更改为以下状态之一：

- COMPLETED
- CANCELLED
- ERROR

3. GetMetadataTransferJobAPI 会返回一个 [MetadataTransferJobProgress](#)对象。

4. 该MetadataTransferJobProgress对象包含以下参数：

- failedCount：表示在转移过程中失败的资产数量。
- skippedCount：表示在转移过程中跳过的资产数量。
- succeededCount：表示在转移过程中成功的资产数量。
- TotalCount：表示转移过程中涉及的资产总数。

5. 此外，API 调用还会返回一个 reportUrl 元素，其中包含一个预签名的网址。如果您的转移任务存在需要调查的错误，则可以通过此 URL 下载完整的错误报告。

## 对 AWS IoT SiteWise 门户进行故障排除

对 AWS IoT SiteWise 门户网站的常见问题进行故障排除。

### 用户和管理员无法访问 AWS IoT SiteWise 门户

如果用户或管理员无法访问您的 AWS IoT SiteWise 门户，则您的附加的 AWS Identity and Access Management (IAM) 策略可能会限制您的权限，从而阻止成功登录。

请参阅以下会导致登录失败的 IAM 策略示例：

#### Note

任何包含 "Condition" 元素的附加 IAM 策略都会导致登录失败。

示例 1：此处的条件是 IP 受限，这将导致登录失败。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribePortal"
      ],
      "Resource": "*",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": [
            "REPLACESAMPLEIP"
          ]
        }
      }
    }
  ]
}
```

示例 2：此处的条件是包含标签，这将导致登录失败。

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "iotsitewise:DescribePortal"
    ],
    "Resource": "*",
    "Condition": {
      "StringLike": {
        "aws:ResourceTag/project": "*"
      }
    }
  }
]
```

向门户添加用户或管理员时，请避免创建限制用户权限（例如受限 IP）的 IAM policy。任何具有受限权限的附加策略都将无法连接到 AWS IoT SiteWise 门户。

## 对 SiteWise 边缘网关进行故障排除

AWS IoT SiteWise 边缘网关运行一组 AWS IoT Greengrass 组件。您可以将 SiteWise Edge 网关配置为将事件记录到 Amazon CloudWatch 和 SiteWise Edge 网关的本地文件系统。然后，您可以查看日志文件以排除您的 SiteWise Edge 网关故障。

您还可以查看 SiteWise Edge 网关报告的 CloudWatch 指标，以解决连接或数据流问题。有关更多信息，请参阅 [AWS IoT SiteWise 使用 Amazon CloudWatch 指标进行监控](#)。

### 主题

- [配置和访问 SiteWise Edge 网关日志](#)
- [排除 SiteWise Edge 网关问题](#)
- [疑难解答 AWS IoT Greengrass 问题](#)

## 配置和访问 SiteWise Edge 网关日志

在查看 SiteWise Edge 网关日志之前，必须将 SiteWise Edge 网关配置为将日志发送到 Amazon CloudWatch Logs 或将日志存储在本地文件系统中。

- 如果您想使用 CloudWatch AWS Management Console 来查看 SiteWise Edge 网关的日志文件，请使用日志。有关更多信息，请参阅 [使用 Amazon CloudWatch 日志](#)。
- 如果您想使用命令行或本地软件查看 SiteWise Edge 网关的日志文件，请使用本地文件系统日志。有关更多信息，请参阅 [使用服务日志](#)。

## 排除 SiteWise Edge 网关问题

使用以下信息对 SiteWise Edge 网关问题进行故障排除。

### 问题

- [无法将包部署到 SiteWise Edge 网关](#)
- [AWS IoT SiteWise 无法从 OPC-UA 服务器接收数据](#)
- [控制面板中未显示任何数据](#)
- [aws.iot 中显示“找不到或加载主类”。SiteWiseEdgePublisher 在 /greengrass/v2/logs 处记录错误](#)

## 无法将包部署到 SiteWise Edge 网关

如果 n AWS IoT Greengrass ucleus 组件 (`aws.greengrass.Nucleus`) 已过期，则可能无法将包部署到您的 SiteWise Edge 网关。您可以使用 AWS IoT Greengrass V2 控制台升级 n AWS IoT Greengrass ucleus 组件。

### 升级 n AWS IoT Greengrass ucleus 组件（控制台）

1. 导航到 [AWS IoT Greengrass 控制台](#)。
2. 在导航窗格的 AWS IoT Greengrass 下，选择部署。
3. 在部署列表中，选择要修改的部署。
4. 选择修订。
5. 在指定目标页面，选择下一步。
6. 在选择组件页面上公共组件下的搜索框中，输入 `aws.greengrass.Nucleus`，然后选择 `aws.greengrass.Nucleus`。
7. 选择下一步。
8. 在配置组件页面上，选择下一步。
9. 在配置高级设置页面上，选择下一步。
10. 在检查页上，选择部署。

## AWS IoT SiteWise 无法从 OPC-UA 服务器接收数据

如果您的 AWS IoT SiteWise 资产未接收 OPC-UA 服务器发送的数据，则可以搜索 SiteWise Edge 网关的日志来解决问题。查找包含以下消息的信息级别 `swPublisher` 日志。

```
Emitting diagnostic name=PublishError.SomeException
```

根据日志 *SomeException* 中的类型，使用以下异常类型和相应的问题对您的 SiteWise Edge 网关进行故障排除：

- `ResourceNotFoundException`— 您的 OPC-UA 服务器发送的数据与任何资产的属性别名都不匹配。在以下两种情况下，可能会发生此异常：
  - 您的属性别名与 OPC-UA 变量不完全匹配，包括您定义的任何源前缀。检查属性别名和源前缀是否正确。
  - 您尚未将 OPC-UA 变量映射到资产属性。有关更多信息，请参阅 [将工业数据流映射到资产属性](#)。

如果您已经映射了想要输入的所有 OPC-UA 变量，则可以筛选 Ed AWS IoT SiteWise ge 网关发送了哪些 OPC-UA 变量。SiteWise 有关更多信息，请参阅 [使用 OPC-UA 节点筛选条件](#)。
- `InvalidRequestException`— 您的 OPC-UA 变量数据类型与您的资产属性数据类型不匹配。例如，如果 OPC-UA 变量具有整数数据类型，相应的资产属性则必须是整数数据类型。双精度类型的资产属性不能接收 OPC-UA 整数值。要解决此问题，请使用正确的数据类型定义新属性。
- `TimestampOutOfRangeException`— 您的 SiteWise Edge 网关正在发送超出可 AWS IoT SiteWise 接受范围的数据。AWS IoT SiteWise 拒绝过去时间戳早于 7 天或将来 5 分钟以上的任何数据点。如果您的 SiteWise Edge 网关断电或与 AWS 云的连接中断，则可能需要清除 SiteWise Edge 网关的缓存。
- `ThrottlingException` 或者 `LimitExceededException`— 您的请求超出了 AWS IoT SiteWise 服务配额，例如提取的数据点速率或资产属性数据 API 操作的请求速率。检查您的配置是否未超过 [AWS IoT SiteWise 配额](#)。

### 控制面板中未显示任何数据

如果您的控制面板中没有显示任何数据，则 SiteWise Edge 网关的发布者配置和 Data Source 可能不同步。如果它们不同步，更新数据源的名称可能会加快从云端到边缘的同步，从而修复不同步错误。

更新数据源的名称

1. 导航到 [AWS IoT SiteWise 控制台](#)。

2. 在导航窗格中，选择 Edge 网关。
3. 选择连接到仪表板的 SiteWise Edge 网关。
4. 在数据来源下，选择编辑。
5. 选择一个新的源名称，然后选择保存以确认您的更改。
6. 确认数据来源表中的数据来源名称已更新，以验证您的更改。

aws.iot 中显示“找不到或加载主类”。SiteWiseEdgePublisher 在 /greengrass/v2/logs 处记录错误

如果您看到此错误，则可能需要更新 SiteWise Edge 网关的 Java 版本。

- 从终端设备运行以下命令：

```
java -version
```

您的 SiteWise Edge 网关所运行的 java 版本将显示在下面 OpenJDK Runtime Environment。您会看到如下响应：

```
openjdk version "11.0.20" 2023-07-18 LTS
OpenJDK Runtime Environment Corretto011.0.20.8.1 (build 11.0.20+8-LTS)
OpenJDK 64-Bit Server VM Corretto-11.0.20.8.1 (build 11.0.20+8-LTS, mixed mode)
```

如果你运行的是 Java 版本 11.0.20.8.1，则必须将 IoT P SiteWise Publisher 包更新到 2.4.1 或更高版本。只有 java 版本 11.0.20.8.1 受到影响，具有其他 Java 版本的环境可以继续使用旧版本的 IoT SiteWise Publisher 组件。有关更新组件包的更多信息，请参阅[更改 SiteWise Edge 网关组件包的版本](#)。

## 疑难解答 AWS IoT Greengrass 问题

要查找在上配置或部署 SiteWise Edge 网关的许多问题的解决方案 AWS IoT Greengrass，请参阅 AWS IoT Greengrass 开发人员指南 AWS IoT Greengrass 中的[疑难解答](#)。

## 对 AWS IoT SiteWise 规则操作进行故障排除

要对中的 AWS IoT SiteWise 规则操作进行故障排除 AWS IoT Core，可以执行以下步骤之一：

- 配置 Amazon CloudWatch 日志
- 为规则配置重新发布错误操作

然后，将错误消息与本主题中的错误进行比较，以排查问题。

## 主题

- [配置 AWS IoT Core 日志](#)
- [配置重新发布错误操作](#)
- [排查问题](#)
- [排查规则问题](#)
- [排查规则问题](#)

## 配置 AWS IoT Core 日志

您可以配置为 AWS IoT 将不同级别的信息 CloudWatch 记录到 Logs 中。

### 配置和访问 CloudWatch 日志

1. 要为配置日志记录 AWS IoT Core，请参阅《AWS IoT 开发人员指南》中的[使用 CloudWatch 日志进行监控](#)。
2. 导航到 [CloudWatch 控制台](#)。
3. 在导航窗格中，选择 日志组。
4. 选择群AWSIoTLogs组。
5. 选择最近的日志流。默认情况下，首先 CloudWatch 显示最新的日志流。
6. 选择日志条目以展开日志消息。您的日志条目可能类似于以下屏幕截图。

CloudWatch > Log Groups > AWSIoTLogs > 9ca6614a-00fc-4f9e-8100-5c2a34918e90\_123456789012\_0

Expand all  Row  Text

Filter events  2020-02-10 (19:36:11) ▾

Time (UTC +00:00)	Message
2020-02-11	No older events found at the moment. <a href="#">Retry</a> .
19:36:11	2020-02-11 19:36:11.823 TRACEID:d4cd3bd0-ac41-cd4a-4f59-74a242ec70e6 PRINCIPALID:AIDAZ2YMUHYHIEDEL3VA3 [ERROR] EVENT:IotSiteWiseActionFailure TOPICNAME:/tutorial/device/SiteWiseTutorialDevice1/cpu CLIENTID:iotconsole-1581444173801-0 MESSAGE:Failed to send message data to IoT SiteWise asset properties. [Code: InvalidRequestException, Message: Property value does not match data type DOUBLE]. Message arrived on: /tutorial/device/SiteWiseTutorialDevice1/cpu, Action: iotSiteWise
	No newer events found at the moment. <a href="#">Retry</a> .



7. 将错误消息与本主题中的错误进行比较，以排查问题。

## 配置重新发布错误操作

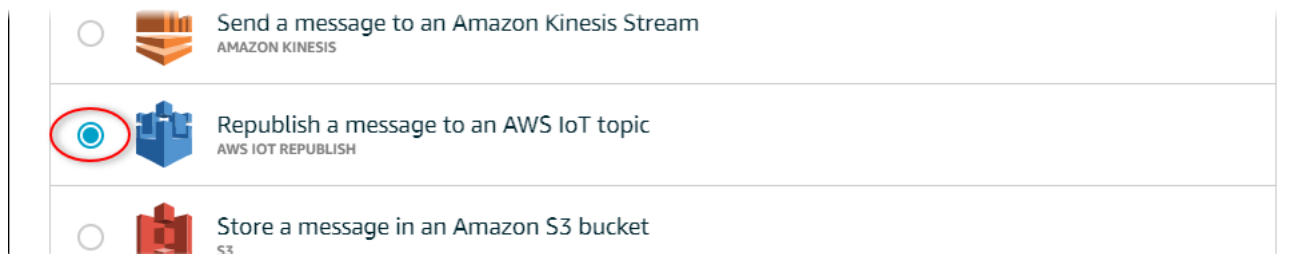
您可以对规则的错误操作进行配置以处理错误消息。在该过程中，您将重新发布规则操作配置为错误操作，以在 MQTT 测试客户端中查看错误消息。

### Note

重新发布错误操作仅输出 ERROR 级别日志的等效内容。如果需要更多详细日志，则必须[配置 CloudWatch](#) 日志。

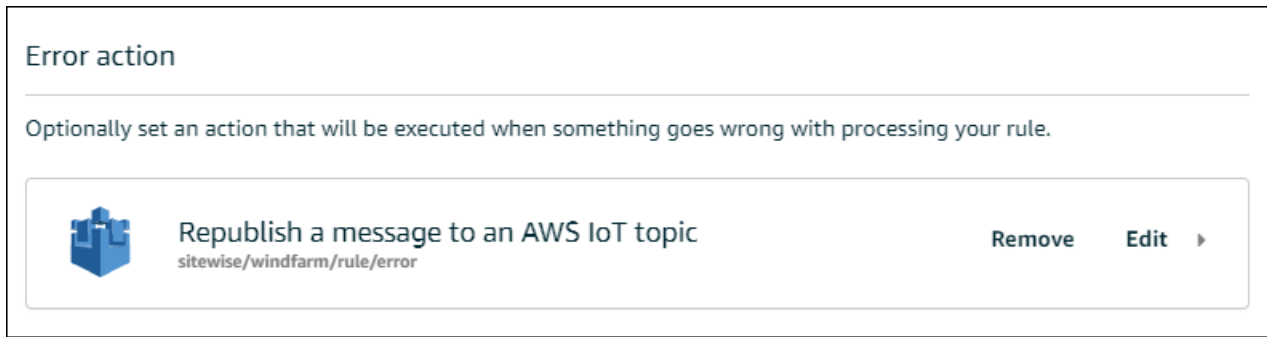
在规则中添加重新发布错误操作

1. 导航到 [AWS IoT 控制台](#)。
2. 在左侧导航窗格中，选择行动，然后选择规则。
3. 选择您的规则。
4. 在 错误操作 下面，选择 添加操作。
5. 选择“向 AWS IoT 主题重新发布消息”。



6. 选择页面底部的 配置操作。
7. 在主题中，输入唯一的主题（例如，**sitewise/windfarm/rule/error**）。AWS IoT Core 将重新发布此主题的错误消息。
8. 选择“选择”以授予执行错误操作的 AWS IoT Core 访问权限。
9. 选择为规则创建的角色旁边的 选择。
10. 选择 更新角色 以在角色中添加其他权限。
11. 选择添加操作。

您的规则的错误操作应类似于以下屏幕截图。



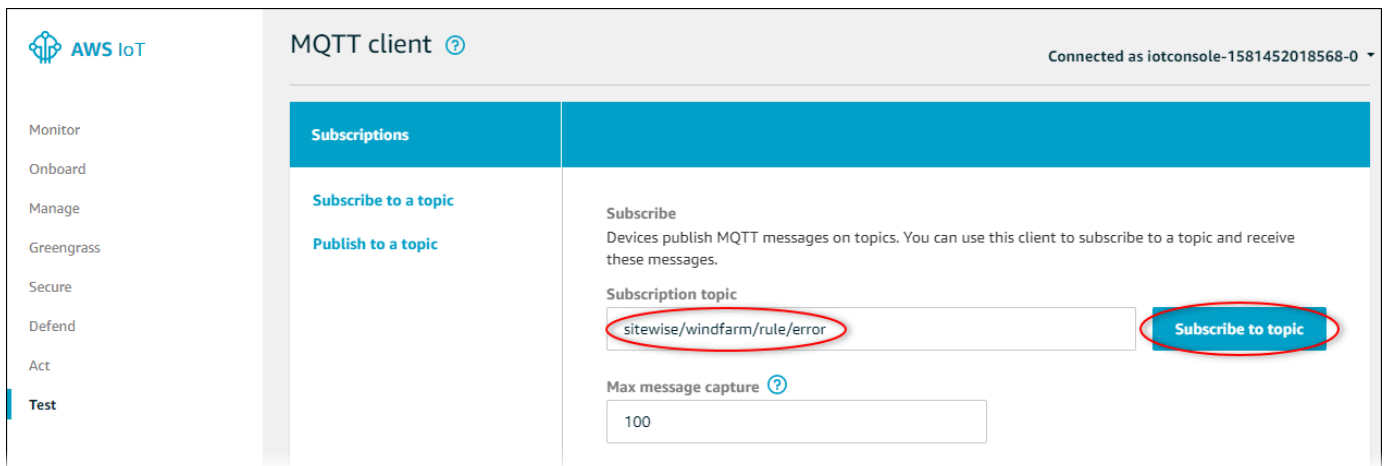
12. 选择主机左上角的返回箭头返回 AWS IoT 主机主页。

在设置重新发布错误操作后，您可以在 AWS IoT Core 上的 MQTT 测试客户端中查看错误消息。

在以下过程中，您在 MQTT 测试客户端中订阅错误主题。在 MQTT 测试客户端中，您可以接收规则的错误消息以排查问题。

#### 订阅错误操作主题

1. 导航到 [AWS IoT 控制台](#)。
2. 在左侧导航页面中，选择 **测试** 以打开 MQTT 测试客户端。
3. 在 **订阅主题** 字段中，输入您之前配置的错误主题（例如，**sitewise/windfarm/rule/error**），然后选择 **订阅主题**。



4. 等待错误消息出现，然后在任何错误消息中展开 **failures** 数组。

接下来，将错误消息与本主题中的错误进行比较，以排查问题。

## 排查问题

使用以下信息排查规则问题。

### 问题

- [错误：成员必须在当前时间戳之前 604800 秒和之后 300 秒范围内](#)
- [错误：Property value does not match data type <type> \(属性值与数据类型 <type> 不匹配\)](#)
- [错误：用户：<role-arn>无权在资源上执行：iotsitewise：BatchPutAssetPropertyValue](#)
- [错误：iot.amazonaws.com 无法在资源上执行：sts：AssumeRole <role-arn>](#)
- [信息：No requests were sent \(没有发送请求\)。PutAssetPropertyValueEntries 执行替换模板后为空。](#)

**错误：成员必须在当前时间戳之前 604800 秒和之后 300 秒范围内**

与当前 Unix 纪元时间相比，您的时间戳早了超过 7 天或晚了超过 5 分钟。尝试以下操作：

- 检查您的时间戳是否采用 Unix 纪元 (UTC) 时间。如果您提供具有不同时区的时间戳，则会收到此错误。
- 检查您的时间戳是否以秒为单位。AWS IoT SiteWise 期望时间戳分为以秒为单位的时间（以 Unix 纪元时间为单位）和以纳秒为单位的偏移量。
- 检查您正在上传的数据的时间戳是否是在过去 7 天之前。

**错误：Property value does not match data type <type> (属性值与数据类型 <type> 不匹配)**

规则操作中的条目与目标资产属性具有不同的数据类型。例如，您的目标资产属性为 DOUBLE，但您选择的数据类型为 整数，或者您用 integerValue 传递了值。尝试以下操作：

- 如果您从 AWS IoT 控制台配置规则，请检查是否为每个条目选择了正确的数据类型。
- 如果您通过 API 或 AWS Command Line Interface (AWS CLI) 配置规则，请检查您的 value 对象是否使用了正确的类型字段（doubleValue 例如，DOUBLE 属性字段）。

**错误：用户：<role-arn>无权在资源上执行：iotsitewise：BatchPutAssetPropertyValue**

未授权您的规则访问目标资产属性，或者目标资产属性不存在。尝试以下操作：

- 检查您的属性别名是否正确，以及您是否有具有给定属性别名的资产属性。有关更多信息，请参阅[将工业数据流映射到资产属性](#)。
- 检查您的规则是否具有角色以及该角色是否允许目标资产属性具有 `iotsitewise:BatchPutAssetPropertyValue` 权限，例如通过目标资产的层次结构。有关更多信息，请参阅[授 AWS IoT 予所需的访问权限](#)。

错误：iot.amazonaws.com 无法在资源上执行：`sts:AssumeRole <role-arn>`

您的用户无权代入您在 AWS Identity and Access Management (IAM) 中的规则中的角色。

检查是否对您的用户授予规则上角色的 `iam:PassRole` 权限。有关更多信息，请参阅 AWS IoT 开发人员指南中的[传递角色权限](#)。

信息：No requests were sent (没有发送请求)。PutAssetPropertyValueEntries 执行替换模板后为空。

#### Note

此消息是 INFO 级别日志。

您的请求必须至少具有一个包含所有必需参数的条目。

检查规则的参数（包括替换模板）是否会导致非空值。替换模板无法访问规则查询语句中的 AS 子句定义的值。有关更多信息，请参阅 AWS IoT 开发人员指南中的[替换模板](#)。

## 排查规则问题

如果 CPU 和内存使用率数据未按预期显示，请按照此过程中的 AWS IoT SiteWise 步骤对规则进行故障排除。在该过程中，您将重新发布规则操作配置为错误操作，以在 MQTT 测试客户端中查看错误消息。您也可以配置日志记录以 CloudWatch 进行故障排除。有关更多信息，请参阅[对 AWS IoT SiteWise 规则操作进行故障排除](#)。

在规则中添加重新发布错误操作

1. 导航到[AWS IoT 控制台](#)。
2. 在左侧导航窗格中，选择消息路由，然后选择规则。
3. 选择您以前创建的规则，然后选择编辑。

4. 在错误操作 - 可选下，选择添加错误操作。
5. 选择“向 AWS IoT 主题重新发布消息”。
6. 在主题中，输入错误路径（例如 `sitewise/rule/tutorial/error`）。AWS IoT Core 将重新发布此主题的错误消息。
7. 选择您之前创建的角色（例如，`SiteWiseTutorialDeviceRuleRole`）。
8. 选择更新。

在设置重新发布错误操作后，您可以在 AWS IoT Core 上的 MQTT 测试客户端中查看错误消息。

在以下过程中，您在 MQTT 测试客户端中订阅错误主题。

### 订阅错误操作主题

1. 导航到 [AWS IoT 控制台](#)。
2. 在左侧导航页面中，选择 MQTT 测试客户端以打开 MQTT 测试客户端。
3. 在主题筛选器字段中，输入 `sitewise/rule/tutorial/error` 并选择订阅。

出现错误消息时，查看任何错误消息中的 `failures` 阵列以诊断问题。有关可能的问题以及如何解决这些问题的更多信息，请参阅[对 AWS IoT SiteWise 规则操作进行故障排除](#)。

如果未显示错误，请检查是否启用了您的规则，以及您是否订阅了在重新发布错误操作中配置的不同主题。如果在执行此操作后仍未出现错误，请检查设备脚本是否正在运行并成功更新设备的影子。

#### Note

您还可以订阅设备的影子更新主题，以查看您的 AWS IoT SiteWise 操作解析的有效负载。为此，请订阅以下主题。

```
$aws/things/+/shadow/update/accepted
```

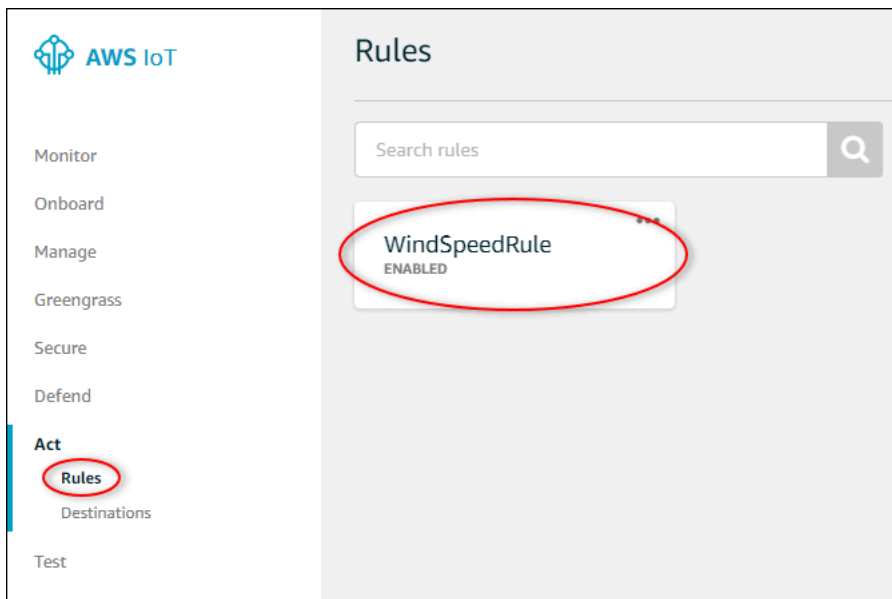
## 排查规则问题

如果演示资产数据未按预期方式显示在 DynamoDB 表中，请按照该过程中的步骤排除规则故障。在该过程中，您将重新发布规则操作配置为错误操作，以在 MQTT 测试客户端中查看错误消息。您也可

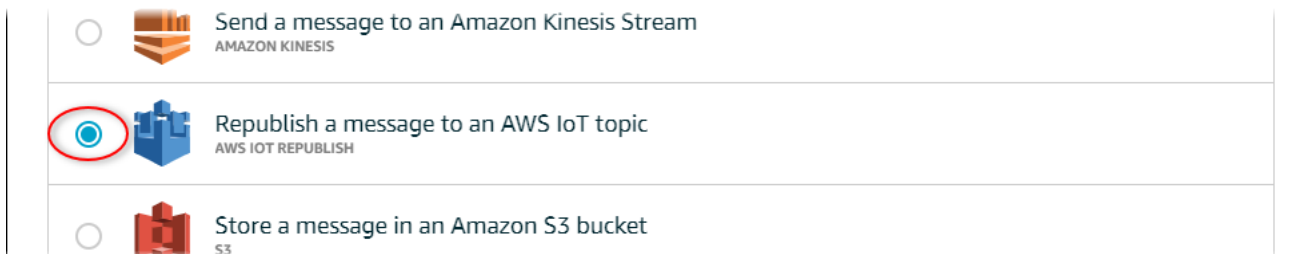
以配置日志记录以 CloudWatch 进行故障排除。有关更多信息，请参阅《AWS IoT 开发人员指南》中的[使用 CloudWatch 日志进行监控](#)。

在规则中添加重新发布错误操作

1. 导航到 [AWS IoT 控制台](#)。
2. 在左侧导航窗格中，选择行动，然后选择规则。
3. 选择您以前创建的规则。



4. 在 错误操作 下面，选择 添加操作。
5. 选择“向 AWS IoT 主题重新发布消息”。



6. 选择页面底部的 配置操作。
7. 在 Topic (主题) 中输入 **windspeed/error**。AWS IoT Core 将重新发布此主题的错误消息。

**Configure action**

**Republish a message to an AWS IoT topic**  
AWS IOT REPUBLISH

This action will republish the message to another AWS IoT topic.

\*Topic ?  
windspeed/error

Quality of Service ?  
 0 - The message is delivered zero or more times.  
 1 - The message is delivered one or more times.

Choose or create a role to grant AWS IoT access to perform this action.

No role selected Create Role **Select**

Cancel Add action

8. 选择 **Select** 以授予使用您之前创建的角色执行错误操作的 C AWS IoT ore 访问权限。
9. 选择您的角色旁边的 **选择**。

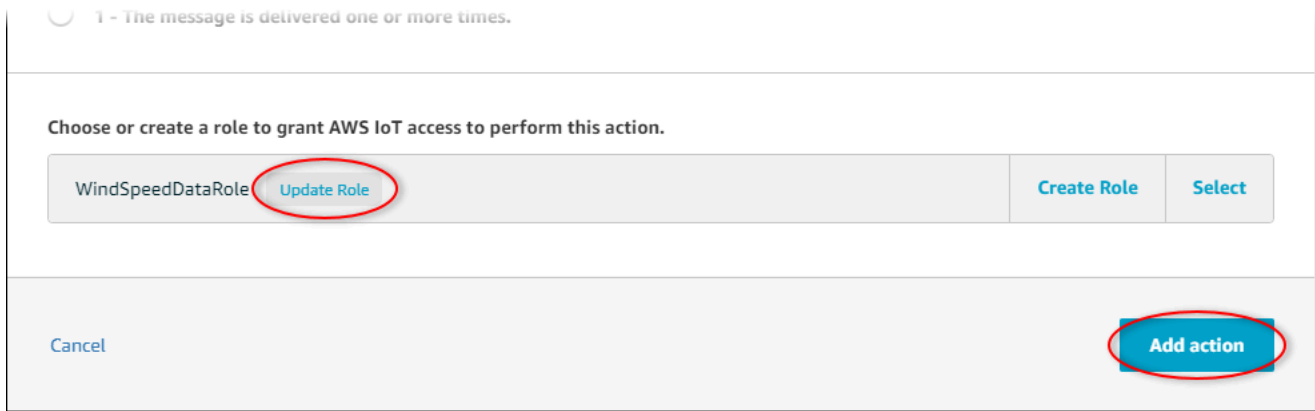
Choose or create a role to grant AWS IoT access to perform this action.

No role selected Refresh Create Role Close

Search for IAM roles

WindSpeedDataRole **Select**

10. 选择 **更新角色** 以在角色中添加其他权限。



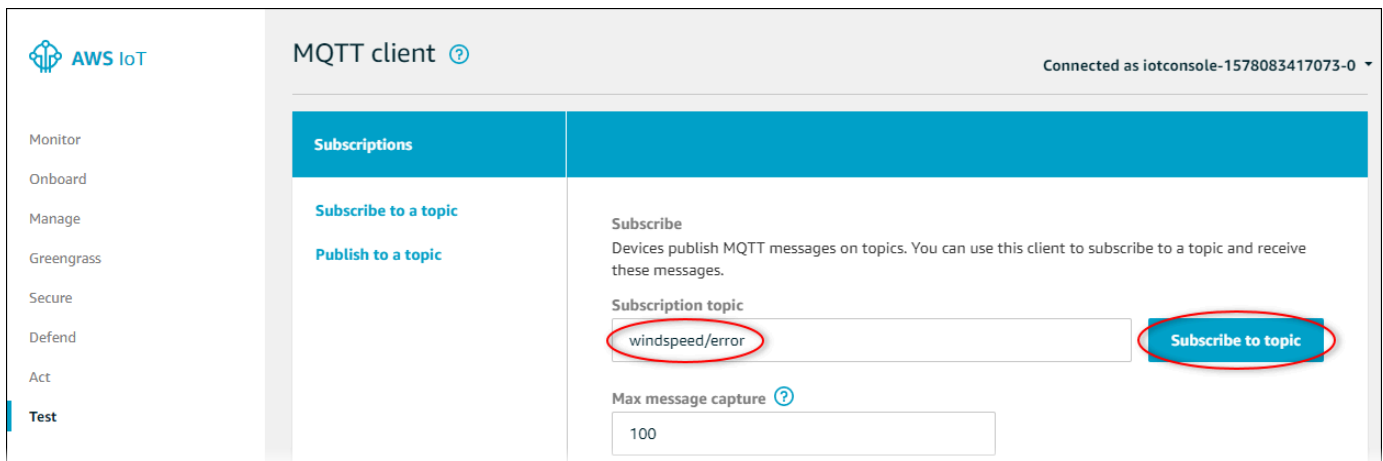
11. 选择 **添加操作** 以完成添加错误操作的过程。
12. 选择主机左上角的后退箭头返回 AWS IoT 酷睿主机主页。

设置重新发布错误操作后，您可以在 Core 的 MQTT 测试客户端中 AWS IoT 查看错误消息。

在以下过程中，您在 MQTT 测试客户端中订阅错误主题。

#### 订阅错误操作主题

1. 在 AWS IoT Core 控制台的左侧导航页中，选择“测试”。
2. 在 **订阅主题** 字段中，输入 **windspeed/error** 并选择 **订阅主题**。



3. 观察显示的错误消息，并在错误消息中浏览 **failures** 数组以诊断以下常见问题：
  - 规则查询语句中的错别字
  - 角色权限不足



如果未显示错误，请检查是否启用了您的规则，以及您是否订阅了在重新发布错误操作中配置的相同主题。如果仍未显示错误，请检查您的演示风电场资产是否仍然存在，以及是否在风速属性上启用了通知。如果您的演示资产已过期并从中消失 AWS IoT SiteWise，则可以创建一个新的演示并更新规则查询语句以反映更新的资产模型和属性 ID。

# AWS IoT SiteWise 终端节点和配额

以下各节描述了终端节点和配额 AWS IoT SiteWise。

内容

- [AWS IoT SiteWise 端点](#)
- [AWS IoT SiteWise 配额](#)

## AWS IoT SiteWise 端点

要以编程方式连接 AWS IoT SiteWise，请使用终端节点。AWS 软件开发工具包和 AWS Command Line Interface (AWS CLI) 会自动使用 AWS 区域中的默认终端节点。有关可用区域的更多信息，请参阅中的[AWS IoT SiteWise 终端节点和配额](#)[AWS 一般参考](#)。AWS IoT SiteWise

AWS IoT SiteWise 支持以下端点。

`data.iotsitewise.region.amazonaws.com`

使用此端点访问以下数据平面 API 操

作：[BatchPutAssetPropertyValue](#)、[GetAssetPropertyAggregates](#)、[GetAssetPropertyValue](#)、[GetAssetProperty](#)。  
`region` 替换为您所在 AWS 的地区。

`api.iotsitewise.region.amazonaws.com`

AWS IoT SiteWise 为控制平面 API 操作提供了此整合端点，用于管理资产模型、资产、SiteWise 边缘网关、标签和账户配置。将 `region` 替换为您的 AWS 区域。

### Note

- 默认情况下，当您调用 AWS IoT SiteWise 支持的控​​制平面 API 操作时，使用统一终端节点。
- 我们建议您将合并端点用于支持的控​​制面板 API 操作。
- 您不能使用整合终端节点访问 SiteWise 监控 API 操作。

支持的控制平面 API 操作包

包括[AssociateAssets](#)、[CreateAsset](#)、[CreateAssetModel](#)、[DeleteAsset](#)、[DeleteAssetModel](#)、[DeleteDashboard](#)和[UntagResource](#)。

控制面板 API 操作的接口 VPC 端点仅支持合并端点。有关更多信息，请参阅 [VPC 端点](#)。

[iotsitewise.region.amazonaws.com](https://iotsitewise.region.amazonaws.com)

使用此端点访问以下 API 操

作：[DescribeStorageConfigurationPutStorageConfiguration](#)、[DescribeDefaultEncryptionConfiguration](#)、[ListTagsForResource](#)和[UntagResource](#)。将 *region* 替换为您的 AWS 区域。

[model.iotsitewise.region.amazonaws.com](https://model.iotsitewise.region.amazonaws.com)

使用此端点访问以下 API 操

作：[AssociateAssets](#)、[CreateAsset](#)、[CreateAssetModel](#)、[DeleteAsset](#)、[DeleteAssetModel](#)、[DeleteDashboard](#)和[UpdateAssetProperty](#)。将 *region* 替换为您所在 AWS 的地区。

[edge.iotsitewise.region.amazonaws.com](https://edge.iotsitewise.region.amazonaws.com)

使用此端点访问以下 API 操

作：[CreateGatewayDeleteGateway](#)、[DescribeGateway](#)、[DescribeGatewayCapabilityConfiguration](#)、[ListGatewayCapabilities](#)和[UpdateGatewayCapabilityConfiguration](#)。将 *region* 替换为您所在 AWS 的地区。

[monitor.iotsitewise.region.amazonaws.com](https://monitor.iotsitewise.region.amazonaws.com)

使用此端点访问以下 API 操

作：[BatchAssociateProjectAssets](#)、[BatchDisassociateProjectAssets](#)、[CreateAccessPolicy](#)、[CreateDashboard](#)和[UpdateProject](#)。将 *region* 替换为您的 AWS 区域。

## AWS IoT SiteWise 配额

下表描述了中的配额 AWS IoT SiteWise。有关配额以及如何请求增加配额的更多信息，请参阅 AWS 一般参考 中的 [AWS 服务限额](#)。有关 AWS IoT SiteWise 配额的更多信息，请参阅中的 [AWS IoT SiteWise 服务配额](#) [AWS 一般参考](#)。

## 资产和资产模型的配额

资源	限额	可调整	注意事项
每个区域每个 AWS 账户的资产模型数量	1000	支持	
每个资产模型的资产数	10000	支持	
每个父资产的子资产数量	2000	支持	
资产模型层次结构树的深度	30	支持	
每个资产模型的层次结构定义数量	30	是	
每个资产模型在根层中的属性数量	500	支持	每种资产模型的最大数量 <code>assetModelProperties</code> 为。此计数不包括 <code>compositeModelProperties</code> 。此配额也适用于根据此资产模型创建的任何独特资产。
每个资产模型的属性数	5000	支持	类型为 <code>ASSET_MODEL</code> 或的资产模型的最大属性数 <code>COMPONENT_MODEL</code> 。这个数字是通过组合根资产模型和任何包含的 <code>component-model-based</code> 或内联的复合模型的属性来确定的。此

资源	限额	可调整	注意事项
			配额也适用于根据此资产模型创建的任何独特资产。
每个复合模型的属性数量	100	是	复合模型允许的最大属性数。此外，该类型的资产模型允许的最大属性数COMPONENT_MODEL。
每个资产模型的属性树深度	10	不支持	例如，一个模型具有转换属性 C，该属性使用转换属性 B，后者又使用测量属性 A，模型的深度为 3。
每个层次结构树的资产模型数	100	支持	
每个资产模型的直接相关的属性数	20	不支持	此配额限制了可直接依赖于单个属性的属性数量，如属性公式表达式中所定义。资产模型的“依赖属性数”必须大于“每个资产模型的直接依赖属性数”。如果每个资产模型的直接依赖属性数限制大于每个资产模型的依赖属性数限制，则必须请求增加两者的配额。

资源	限额	可调整	注意事项
每个资产模型的相关属性数	30	不支持	此配额限制了可以直接或间接依赖于单个属性的属性数量，如属性公式表达式中所定义。
每个资产模型的复合模型数量	50	可以	单个资产模型上允许的最大复合模型数量。
复合模型深度	2	可以	每个资产模型（包括内联模型和复合模型）的 component-model-based 复合模型树的最大深度。
使用相同组件模型的独特资产模型的数量	20	是	具有至少一个直接引用 COMPONENT_MODEL 类型的特定资产模型的 component-model-based 复合模型的唯一资产模型的最大数量。
每个属性公式表达式的属性变量数	10	不支持	例如，表达式 $\text{avg}(\text{power}) + \text{max}(\text{temp})$ 中有两个属性变量 <code>power</code> 和 <code>temp</code> 。这也适用于转换计算结果。
每个属性公式表达式的函数数量	10	不支持	例如，表达式 $\text{avg}(\text{power}) + \text{max}(\text{temp})$ 中有两个函数 <code>avg</code> 和 <code>max</code> 。

## 资产属性数据的配额

资源	限额	可调整	注意事项
资产属性数据 API 操作的请求速率	每个 AWS 账户每个区域每秒 1000 个请求	支持	此配额应用于 API 操作，例如 GetAssetPropertyValue 和 BatchPutAssetPropertyValue。
每个资产属性在每个数据质量中每秒的数据点数	10 数据点	不支持	此配额适用于每个资产属性的每个数据质量中具有相同时间戳的 timestamp-quality-value (TQV) 数据点的最大数量（以秒为单位）。对于每个资产属性，在任何给定的秒数内，最多可以存储此数量的高质量、不确定质量和低质量数据点。
每个 AWS 账户每个区域每个资产每秒摄取的 BatchPutAssetPropertyValue 条目数。	每个资产属性 10 个条目	不支持	此配额适用于来自所有来源的 BatchPutAssetPropertyValue 条目，包括 SiteWise Edge 网关、AWS IoT Core 规则和 API 调用。
提取的数据点的速率	每个 AWS 账户每个区域每秒 5000 个数据点	支持	Timestamp-quality-value (TQV) 数据点。
BatchGetAssetPropertyValue	200	支持	在当前的区域中，您每秒可以在此账户下执行的最

资源	限额	可调整	注意事项
BatchGetAssetPropertyAggregates 的请求速率			大 BatchGetAssetPropertyAggregates 请求数。
BatchGetAssetPropertyValues 的请求速率	500	支持	在当前的区域中，您每秒可以在此账户下执行的最大 BatchGetAssetPropertyValues 请求数。
BatchGetAssetPropertyHistory 的请求速率	200	支持	在当前的区域中，您每秒可以在此账户下执行的最大 BatchGetAssetPropertyHistory 请求数。
每个 AWS 账户每个区域每个资产每秒摄取的 BatchPutAssetPropertyValues 条目数。	每个资产属性 10 个条目	不支持	此配额适用于来自所有来源的 BatchPutAssetPropertyValues 条目，包括 SiteWise Edge 网关、AWS IoT Core 规则和 API 调用。



资源	限额	可调整	注意事项
每个资产属性的 GetAssetPropertyAggregates 请求和 BatchGetAssetPropertyAggregates 条目查询的速率	50	不可以	当前区域中该帐户中每秒每个资产属性的最大 GetAssetPropertyAggregates 请求和 BatchGetAssetPropertyAggregates 条目总数。
每个资产属性的 GetAssetPropertyValue 请求和 BatchGetAssetPropertyValue 条目查询的速率	500	不支持	当前区域中该帐户中每秒每个资产属性的最大 GetAssetPropertyValue 请求和 BatchGetAssetPropertyValue 条目总数。
每个资产属性的 GetAssetPropertyValueHistory 请求和 BatchGetAssetPropertyValueHistory 条目查询的速率	30	不支持	当前区域中该帐户中每秒每个资产属性的最大 GetAssetPropertyValueHistory 请求和 BatchGetAssetPropertyValueHistory 条目总数。

资源	限额	可调整	注意事项
GetInterpolatedAssetPropertyValues 请求速率	500	支持	在当前的区域中，您每秒可以在此账户下执行的最大 GetInterpolatedAssetPropertyValues 请求数。
每个 GetInterpolatedAssetPropertyValues 请求的结果数	10	是	每个分页 GetInterpolatedAssetPropertyValues 请求返回的最大结果数。

资源	限额	可调整	注意事项
从 GetAssetPropertyValueHistory 和 BatchGetAssetPropertyHistory 检索到的数据点的速率	每个 AWS 账户每个区域每秒 100MB 的读取响应。	支持	<p>跨和的每个账户每个区域每秒检索的数据点的最大字节速率 (MB/秒)。</p> <p>AWS GetAssetPropertyValueHistory BatchGetAssetPropertyValueHistory 针对此配额评估的响应负载使用每个数据点的时间戳-质量-值 (TQV) 字段，并将每个 API 请求的字节大小舍入到下一个 4KB 增量。</p> <p>每秒检索的 Timestamp-quality-value (TQV) 数据点因数据类型而异：</p> <ul style="list-style-type: none"> <li>• 整数 – 每秒高达 500 万 TQV</li> <li>• 双倍 – 每秒高达 400 万 TQV</li> <li>• 布尔值 – 每秒高达 600 万 TQV</li> <li>• 字符串 – 因每个字符串值大小而异。</li> </ul>

## SiteWise 边缘网关的配额

资源	限额	可调整
每个 AWS 账户每个区域的 SiteWise 边缘网关数量	100	是
每个边缘网关的 OPC-UA 源数量 SiteWise	100	不支持

## 的配额 AWS IoT SiteWise Monitor

资源	限额	可调整
每个 AWS 账户每个区域的门户数量	100	是
每个门户的项目数	100	是
每个项目的控制面板数	100	是
每个项目的根资产数	1	不支持
每个控制面板的可视化数量	10	是
每个控制面板可视化的指标数	5	支持
每个控制面板可视化的阈值数	12	不支持

## AWS IoT SiteWise 批量导入和导出元数据的配额

资源	描述	限额	可调整
队列中的元数据传输任务数量	队列中PENDING元数据传输任务的最大数量。	10	是
元数据传输任务导入文件的大小	导入文件的最大大小 (以 MB 为单位)。	100 MB	支持

资源	描述	限额	可调整
AWS IoT SiteWise 元数据传输任务的资源配额	在单个任务中导入或导出的资源的最大数量。资源包括资产和资产模型。	5000	不支持

### AWS IoT SiteWise 批量导入数据的配额

资源	限额	可调整
正在运行的批量导入作业数	100	不支持
CSV 文件的大小	10 GB	不支持
未压缩的 parquet 文件的大小	256 MB	不支持
用于缓冲CSV摄取的文件的大小	256 MB	不支持
未压缩的拼花地板排组的大小	64 MB	不支持
每个拼花地板行组的唯一测量次数	2000	支持
缓冲摄取的时间戳与今天的时间戳之间的天数	30	支持
每个 AWS 账户中CreateBulkImportJobs 每个区域的请求费率	10	是
每个 AWS 账户中ListBulkImportJobs 每个区域的请求费率	50	可以
每个 AWS 账户中DescribeBulkImportJobs 每个区域的请求费率	50	可以

## 异常检测配额

异常检测的配额由 AWS IoT SiteWise 和 Amazon Lookout for Equipment 共享。有关更多信息，请参阅 [Lookout for Equipment 的使用配额](#)。

## 《AWS IoT SiteWise 用户指南》的文档历史记录

下表描述了此版本的文档 AWS IoT SiteWise。

- API 版本：2019-12-02

变更	说明	日期
<a href="#">增加了对在西门子工业 SiteWise 边缘上运行 Edge 的支持</a>	AWS IoT SiteWise 现在支持在西门子工业 SiteWise 边缘设备上运行 Edge。	2023 年 11 月 26 日
<a href="#">增加了对温层存储的支持</a>	AWS IoT SiteWise 现在支持温层存储，这是一种完全托管的存储层，可让客户轻松安全地存储和访问工业数据。	2023 年 11 月 15 日
<a href="#">增加了对用户定义的唯一标识符的支持</a>	AWS IoT SiteWise 现在支持对资产、资产模型、属性和层次结构使用用户定义的唯一标识符。	2023 年 11 月 15 日
<a href="#">增加了对工业资产多变异常检测的支持</a>	AWS IoT SiteWise 现在，通过将历史和实时设备数据与 Amazon Lookout for Equipment 集成，支持对工业资产进行多变异常检测。	2023 年 11 月 15 日
<a href="#">增加了对经济高效且可扩展的时间序列数据摄取的支持 AWS IoT SiteWise</a>	AWS IoT SiteWise 现在支持以经济高效且可扩展的方式摄取分析用例所需的时间序列数据。	2023 年 11 月 15 日
<a href="#">增加了对批量导入、导出和更新的支持</a>	AWS IoT SiteWise 现在支持批量导入、导出和更新工业设备元数据。	2023 年 11 月 15 日

<a href="#">增加了对资产模型组件的支持</a>	AWS IoT SiteWise 现在支持资产模型组件，以帮助工业客户创建可重复使用的组件。	2023 年 11 月 15 日
<a href="#">增加了对 IoT 控制面板应用程序的支持</a>	AWS IoT SiteWise 现在支持开源仪表板应用程序，您可以在其中可视化操作数据并与之交互。	2023 年 11 月 15 日
<a href="#">更新了的服务相关角色 AWS IoT SiteWise</a>	AWS IoT SiteWise 具有新的服务相关角色，可以对 AWS IoT TwinMaker 数据库运行元数据搜索查询。	2023 年 11 月 6 日
<a href="#">更新了 AWS IoT SiteWise 数据流资源的标记</a>	添加了对标记数据流资源的支持。	2022 年 8 月 18 日
<a href="#">更新了 SiteWise 边缘网关</a>	现在，您可以配置发布者，以控制从边缘发送到云的数据以及发送到云的顺序。	2022 年 1 月 12 日
<a href="#">更新了 AWS IoT SiteWise 演示</a>	现在，您可以使用该演示创建 SiteWise 监控器门户。	2022 年 1 月 10 日
<a href="#">更新了存储管理</a>	现在可以定义保留期，以控制数据在热层中的保留时间。	2021 年 11 月 29 日
<a href="#">增加了对数据流管理的支持</a>	现在，您可以在创建资产模型和资产 AWS IoT SiteWise 之前提取数据。	2021 年 11 月 24 日
<a href="#">更新了资产模型层次结构</a>	现在，一个子资产模型可以与多个父资产模型相关联。	2021 年 10 月 28 日
<a href="#">区域启动</a>	AWS IoT SiteWise 在 AWS GovCloud (美国西部) 推出。	2021 年 9 月 29 日



<a href="#">更新了函数</a>	添加了以下功能 <ul style="list-style-type: none"><li>在指标中，您可以在<a href="#">聚合函数</a>和<a href="#">时态函数</a>中使用嵌套表达式。</li><li>在转换中，可以使用 <a href="#">pretrigger () 函数</a>在触发当前转换计算的属性更新之前检索变量的值。</li></ul>	2021 年 8 月 10 日
<a href="#">自定义指标时间间隔</a>	在指标中添加了对自定义时间间隔和偏移量的支持。	2021 年 8 月 3 日
<a href="#">AWS IoT SiteWise 在边缘使用</a>	边缘处理功能现已正式发布。	2021 年 7 月 29 日
<a href="#">将数据导出到 Amazon S3</a>	AWS IoT SiteWise 现在可以将数据导出到 Amazon S3。	2021 年 7 月 27 日
<a href="#">VPC 端点 (AWS PrivateLink)</a>	用于控制面板 API 操作的接口 VPC 端点现已正式发布。	2021 年 7 月 15 日
<a href="#">转换</a>	转换现在可以输入多个资产属性变量。	2021 年 7 月 8 日
<a href="#">更新了 timestamp() 函数</a>	在转换中，现在可以提供变量作为 timestamp() 函数的参数。	2021 年 6 月 16 日
<a href="#">警报正式发布</a>	警报功能现已正式发布。	2021 年 5 月 27 日
<a href="#">Modbus-TCP 协议适配器版本 2 发布</a>	<a href="#">Modbus-TCP 协议适配器连接器</a> 版本 2 现已推出。此版本增加了对 ASCII、UTF8 和 ISO8859 编码源字符串的支持。	2021 年 5 月 24 日

[更新了服务限额](#)

为 [GetInterpolatedAssetPropertyValues](#) API 添加了以下配额：GetInterpolatedAssetPropertyValues 请求率、每个GetInterpolatedAssetPropertyValues 请求的结果数以及从过去开始日期到今天之间的天数GetInterpolatedAssetPropertyValues。

2021 年 4 月 29 日

[更新了公式表达式](#)

添加了以下运算符和函数：

2021 年 4 月 22 日

- 添加了以下[运算符](#)：[<](#)、[>](#)、[<=](#)、[>=](#)、[==](#)、[!=](#)、[!](#)、[and](#)、[or](#) 和 [not](#)。
- 添加了以下[比较功能](#)：[neq\(x, y\)](#)。
- 添加了以下[字符串函数](#)：[join\(\)](#)、[format\(\)](#) 和 [f''](#)。

[VPC 端点 \(AWS PrivateLink\)](#)

添加了有关如何通过创建接口 VPC 终端节点在您的虚拟私有云 (VPC) 和 AWS IoT SiteWise 控制平面 API 之间建立私有连接的信息。

2021 年 3 月 16 日

[IAM 联合身份验证](#)

您的 M SiteWise onitor 门户管理员和用户现在可以使用 IAM 证书登录分配给他们的门户。

2021 年 3 月 16 日

[区域启动](#)

AWS IoT SiteWise 在中国（北京）推出。

2021 年 2 月 3 日

<a href="#">物联网 SiteWise 连接器版本 10 已发布</a>	物联网 SiteWise 连接器的版本 10 已推出。此版本对 StreamManager 进行了配置，以改进源连接丢失并重新建立时的处理。当没有 SourceTimestamp 可用时，此版本还接受带有 ServerTimestamp 的 OPC-UA 值。	2021 年 1 月 22 日
<a href="#">日期和时间函数</a>	AWS IoT SiteWise 现在支持日期和时间功能。	2021 年 1 月 21 日
<a href="#">函数语法</a>	您现在可以对函数使用统一函数调用语法 (UFCS)。AWS IoT SiteWise	2021 年 1 月 11 日
<a href="#">与 Grafana 集成</a>	添加了有关如何在 Grafana 仪表板中可视化 AWS IoT SiteWise 表板中可视化数据的信息。	2020 年 12 月 15 日

## [AWS IoT SiteWise 功能发布](#)

现在，您可以通过警报监控数据，在边缘处理工业数据，使用 Modbus TCP 和 Ethernet/IP 源连接到 SiteWise 边缘网关，使用死区过滤传入的数据等等。

2020 年 12 月 15 日

- 添加了 [通过警报监控数据](#) 部分，可用于定义、配置和响应 AWS IoT SiteWise 中的警报。
- 添加了 [边缘处理](#) 部分，可用于在边缘设备上配置工业数据的处理。
- 在 E SiteWise dge 网关源文档中添加了 [Modbus TCP 和 Ethernet/IP](#) 部分。
- 添加了 [源目标](#) 部分，您可以使用该部分自定义将传入的工业数据发送到何处。
- 添加了 [OPC-UA 过滤](#) 部分，您可以使用该部分来控制从工业本地服务器发送到 SiteWise Edge 网关的数据的频率和类型。

## [AWS IoT SiteWise 现在支持客户托管的 CMK。](#)

AWS IoT SiteWise 现在支持使用客户托管的 CMK 进行加密。

2020 年 11 月 24 日

## [物联网 SiteWise 连接器版本 8 发布](#)

物联网 SiteWise 连接器的版本 8 已推出。此版本改进了连接器遇到间歇性网络连接时的稳定性。

2020 年 11 月 19 日

<a href="#">在公式表达式中使用字符串和条件语句</a>	添加了有关如何在转换和指标的公式表达式中使用字符串和条件函数的信息。	2020 年 11 月 16 日
<a href="#">使用 AWS IoT Greengrass 流管理器摄取数据</a>	添加了有关如何使用 AWS IoT Greengrass 边缘设备从本地数据源提取大量 IoT 数据的信息。	2020 年 9 月 16 日
<a href="#">VPC 端点 (AWS PrivateLink)</a>	添加了有关如何通过创建接口 VPC 终端节点在您的虚拟私有云 (VPC) 和 AWS IoT SiteWise 数据 API 之间建立私有连接的信息。	2020 年 9 月 4 日
<a href="#">物联网 SiteWise 连接器版本 7 已发布</a>	物联网 SiteWise 连接器版本 7 已推出。此版本修复了 E SiteWise dge 网关指标的问题。	2020 年 8 月 14 日
<a href="#">通过 AWS IoT SiteWise 控制台创建 IAM 身份中心用户</a>	添加了有关如何在 AWS IoT SiteWise 控制台中创建 IAM Identity Center 用户的信息。现在，您可以在将用户分配到新门户或现有门户时创建 IAM Identity Center 用户。更新了 <a href="#">可视化和共享风电场数据教程</a> 以使用此功能。此更改减少了教程中的步骤数。	2020 年 8 月 4 日
<a href="#">改进了 SiteWise Edge 网关故障</a>	添加了有关如何对 SiteWise Edge 网关进行故障排除以及如何 <a href="#">导出源的 OPC-UA 客户端证书</a> 的其他信息。	2020 年 6 月 18 日

<a href="#">控制台任务文档</a>	添加了用于 <a href="#">工业资产建模</a> 、 <a href="#">查询资产属性数据</a> 和 <a href="#">与其他服务交互</a> 的控制台任务文档。您可以按照以下说明完成 AWS IoT SiteWise 控制台中的任务。	2020 年 6 月 11 日
<a href="#">“分析导出的数据”教程</a>	添加了一个教程，您可以按照该教程学习如何使用 Amazon Athena 分析使用导出功能模板导出到 S3 <a href="#">的</a> 资产数据。AWS CloudFormation	2020 年 5 月 27 日
<a href="#">改进了对公式表达式的使用</a>	添加了有关 AWS IoT SiteWise 公式属性行为的详细信息，并添加了如何计算筛选数据点的示例。	2020 年 5 月 18 日
<a href="#">物联网 SiteWise 连接器版本 6 已发布</a>	物联网 SiteWise 连接器的版本 6 已推出。此版本增加了对 CloudWatch 指标和自动发现新 OPC-UA 标签的支持。这意味着，当 OPC-UA 源的标签发生变化时，您无需重新启动 SiteWise Edge 网关。此版本的连接器需要流管理器和 AWS IoT Greengrass 核心软件 v1.10.0 或更高版本。	2020 年 4 月 29 日

## [AWS IoT SiteWise 功能发布](#)

AWS IoT SiteWise 功能发布。现在，您可以使用 API 管理 SiteWise Edge 网关、向门户添加徽标、查看 SiteWise Edge 网关指标等。

2020 年 4 月 29 日

- 添加了[将数据导出到 Amazon S3](#)部分，其中包含可用于将新数据值导出到 S3 存储桶的 AWS CloudFormation 模板。
- 添加了[配置数据源](#)部分，该部分改进了 SiteWise Edge 网关源文档，并包括新的 SiteWise Edge 网关 API。
- 添加了[SiteWise Edge 网关指标](#)部分，描述了 SiteWise Edge 网关发布的 CloudWatch 指标。
- 添加了“在 Amazon EC2 上配置 SiteWise 边缘网关”部分，其中包含可用于在 Amazon EC2 实例上快速配置 SiteWise 边缘网关依赖项的 AWS CloudFormation 模板。
- 添加了[门户服务角色](#)部分，该部分描述了 SiteWise 监控器门户的新权限功能。
- 针对门户服务角色和门户徽标更新了[门户文档](#)。
- 添加了为[AWS IoT SiteWise 资源添加标签](#)部分。

- 针对新的控制面板定义结构更新了[创建控制面板 \(CLI\)](#)部分。
- 增加了[安全性](#)部分。

### [正在从中提取数据 AWS IoT Events](#)

添加了有关如何从事件发生 AWS IoT Events 时提取数据的信息。

2020 年 4 月 20 日

### [在 Monitor 教程中可 SiteWise 可视化和共享风电场数据](#)

添加了一个教程，您可以按照该教程学习如何使用该教程 AWS IoT SiteWise Monitor 来可视化和共享资产数据。

2020 年 3 月 12 日

### [AWS IoT SiteWise 概念](#)

添加了 AWS IoT SiteWise 概念词汇表，可用于了解该服务及其常用术语。

2020 年 3 月 5 日

### [已删除 AWS IoT Greengrass 安装说明](#)

已从《AWS IoT SiteWise 用户指南》中删除 AWS IoT Greengrass 核心软件安装说明。《[AWS IoT Greengrass 开发人员指南](#)》提供了设备设置脚本以及 AWS IoT Greengrass 在 Amazon EC2 和 Docker 等其他平台上进行设置的说明。

2020 年 2 月 14 日

### [使用规则改进了数据采集功能 AWS IoT Core](#)

添加了有关[如何使用](#)规则操作以及[如何对 AWS IoT SiteWise 规则操作进行故障排除](#)的详细信息，您可以使用该操作从 MQTT 消息中提取数据。AWS IoT Core

2020 年 2 月 14 日



<a href="#">物联网 SiteWise 连接器版本 5 已发布</a>	物联网 SiteWise 连接器的版本 5 已推出。此版本修复了与 AWS IoT Greengrass 核心软件 v1.9.4 的兼容性问题。	2020 年 2 月 12 日
<a href="#">物联网 SiteWise 连接器第 4 版发布</a>	物联网 SiteWise 连接器的版本 4 现已推出。此版本修复了 OPC-UA 服务器重新连接的问题。	2020 年 2 月 7 日
<a href="#">新编了工业资产建模</a>	<p>将“更新资产和模型”部分重新组合为“工业资产建模”中的多个主题。</p> <ul style="list-style-type: none"><li>• <a href="#">资产和模型状态</a></li><li>• <a href="#">将工业数据流映射到资产属性</a></li><li>• <a href="#">更新属性值</a></li><li>• <a href="#">关联资产和取消资产关联</a></li><li>• <a href="#">更新资产和模型</a></li><li>• <a href="#">删除资产和模型</a></li></ul>	2020 年 2 月 4 日
<a href="#">从事物中提取数据教程 AWS IoT</a>	添加了一个教程，您可以按照该教程来学习如何配置 AWS IoT SiteWise 规则操作以从新的或现有的设备队列中提取数据 AWS IoT。	2020 年 2 月 4 日
<a href="#">重构了从中检索数据的结构 AWS IoT SiteWise</a>	将“检索数据”部分重组为两个顶级部分： <a href="#">查询资产属性值和聚合以及其他服务交互</a> 。 AWS	2020 年 1 月 21 日
<a href="#">将属性值更新发布到 Amazon DynamoDB 教程</a>	添加了一个教程，您可以遵循该教程了解如何使用属性值通知在 DynamoDB 中存储资产数据。	2020 年 1 月 8 日

<a href="#">使用公式表达式</a>	添加了公式表达式引用，以整理可用于转换属性和指标属性中的常量和函数。将 <a href="#">资产属性</a> 重新组合为每种属性类型的单独主题。	2020 年 1 月 7 日
<a href="#">使用 OPC-UA 节点筛选条件</a>	添加了有关在添加 Edge 网关源时如何使用 OPC-UA 节点过滤器来提高 SiteWise Edge 网关性能的信息。	2020 年 1 月 3 日
<a href="#">升级连接器</a>	添加了有关在新连接器版本发布时如何升级 SiteWise Edge 网关的信息。	2019 年 12 月 30 日
<a href="#">物联网 SiteWise 连接器第 3 版发布</a>	物联网 SiteWise 连接器的版本 3 现已推出。此版本消除了 <code>iot:*</code> 权限要求。	2019 年 12 月 17 日
<a href="#">物联网 SiteWise 连接器第 2 版发布</a>	物联网 SiteWise 连接器的版本 2 现已推出。此版本增加了对多个 OPC-UA 秘密资源的支持。	2019 年 12 月 10 日
<a href="#">创建控制面板 (AWS CLI)</a>	在 AWS IoT SiteWise Monitor 使用中添加了有关如何创建仪表板的信息 AWS CLI。	2019 年 12 月 6 日

## [AWS IoT SiteWise 版本 2 已发布](#)

已发布版本 2 的预览版 AWS IoT SiteWise。现在，您可以通过 OPC-UA、MQTT 和 HTTP 提取数据，在资产层次结构中对数据进行建模，并使用 Monitor 对数据进行可视化。SiteWise

2019 年 12 月 2 日

- 针对资产、资产模型和资产层次结构的更改重新编写了[资产建模](#)部分。
- 更新了[数据提取部分](#)，增加了[AWS IoT Greengrass 连接器步骤和非网关数据摄取](#)部分。
- 添加了该[AWS IoT SiteWise Monitor](#)部分和[单独的应用程序指南](#)，其中显示了如何使用 Monitor SiteWise 的 Web 应用程序。
- 添加了[从中查询数据 AWS IoT SiteWise](#)和[与其他 AWS 服务互动](#)部分。
- 重新编写了[入门](#)部分，以便与更新的演示体验相符。

## [AWS IoT SiteWise 版本 1 已发布](#)

已发布版本 1 的初始预览版 AWS IoT SiteWise。

2019 年 2 月 25 日

# AWS 术语表

有关最新的 AWS 术语，请参阅《AWS 词汇表参考》中的 [AWS 词汇表](#)。

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。